LEARNING UNDER TEMPORAL LABEL NOISE

Anonymous authors
Paper under double-blind review

ABSTRACT

Many time series classification tasks, where labels vary over time, are affected by *label noise* that also varies over time. Such noise can cause label quality to improve, worsen, or periodically change over time. We first propose and formalize *temporal label noise*, an unstudied problem for sequential classification of time series. In this setting, multiple labels are recorded over time while being corrupted by a time-dependent noise function. We first demonstrate the importance of modelling the temporal nature of the label noise function and how existing methods will consistently underperform. We then propose methods that can train noise-tolerant classifiers by estimating the temporal label noise function directly from data. We show that our methods lead to state-of-the-art performance under diverse types of temporal label noise on real-world datasets.

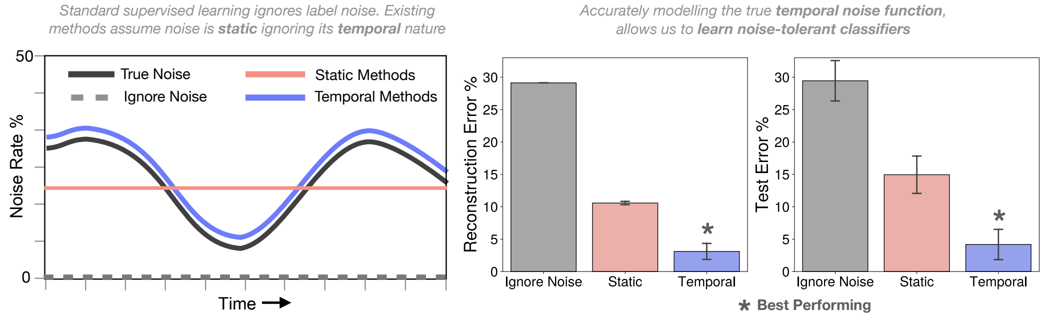
1 INTRODUCTION

Many supervised learning datasets contain *noisy* observations of ground truth labels. Such *label noise* can arise due to issues in human annotation or data collection [1, 25], including lack of expertise among annotators [28, 74], discrepancies in labelling difficulty [13, 27, 74], subjective labeling tasks [44, 52, 58], and systematic issues in automatic annotation like measurement error [29, 47]. Label noise is a key vulnerability of modern supervised learning [16, 23, 72]. Intuitively, models trained with noisy labels may learn to predict noise rather than the ground truth. Such models will then underperform at test time when they must predict the ground truth.

Label noise is a major problem in machine learning. Over the past decade, this has led to a stream of work studying label noise. However, the nearly all methods are designed for *static* prediction tasks where observations, labels, and label noise do not evolve over time [3, 37, 40, 63].

Many *non-static* prediction tasks suffer from noisy labels. For instance, time series classification is a non-static task which requires predicting a label for each observation over time. While the observations and labels in the time series change over time, we argue that the dynamics of the label *noise* can also evolve over time. Consider data collected from a patient in the emergency room of a hospital. The diagnostic label for this patient will be noisy at first, as they come from screening procedures. Over time, more testing and procedures are performed that influence the accuracy of the final diagnostic label. Existing approaches for static label noise *cannot* handle this phenomenon. We introduce *temporal label noise*, relaxing restrictive assumptions of static noise. Learning from temporal label noise is unstudied, yet clearly exists in a range of tasks. For example:

- *Human Activity Recognition*. Wearable device studies often ask participants to annotate their activities over time. But participants may mislabel due to recall bias, time of day, or labelling-at-random for monetized studies [24, 57].
- *Self-Reported Outcomes for Mental Health*: Mental health studies often collect self-reported survey data over long periods of time. Such self-reporting is known to be biased [5, 48, 50, 69], where participants are more or less likely to report certain features. For example, the accuracy of self-reported alcohol consumption is often seasonal [8].
- *Clinical Measurement Error*: The labels used to train clinical prediction models are often derived from clinician notes in electronic health records. These labels may capture noisier annotations during busier times, when a patient is deteriorating, or the bedside situation is more chaotic [73].



*Standard supervised learning ignores label noise. Existing methods assume noise is **static** ignoring its **temporal** nature*

*Accurately modelling the true **temporal noise function**, allows us to **learn noise-tolerant classifiers***

Figure 1: Label quality can vary over time due to *temporal* label noise. Existing methods assume noise is time-invariant (*static*) leading to loss in performance. Accurate modeling of temporal label noise, improves performance. We show performance improvements on a Human Activity Recognition task (`har` dataset) comparing reconstruction error and accuracy between static and temporal methods subject to 30% temporal label noise across 10 draws (see Appendix F for details).

Addressing the problem of temporal label noise is difficult from a technical perspective as adapting existing methods [43, 45] for this setting is not straightforward. Existing approaches have no mechanism to incorporate the notion of noise rates varying over time, therefore assuming they are constant over time. Additionally, the label noise function, temporal or otherwise, is often *unknown*; most datasets lack indication for which instances or time steps are more likely to be accurate.

We propose novel time series classification objectives which *learn* the temporal noise function directly from noisy data. These methods leverage a temporal loss function, which we prove is *robust to temporal label noise*. Our algorithms can be used out of the box and allow practitioners to learn noise-tolerant time series classifiers, even with unknown, temporal label noise.

Our main contributions include:

1. We formalize the problem of learning from noisy labels in temporal settings.
2. We propose a novel loss function for training models that are robust to temporal label noise. On its own, this can be used to improve prior methods.
3. We develop versatile methods to learn from temporal noise functions. Our methods can learn any temporal label noise function from noisy data alone and thereby lead to better time series classifiers. We pair these with extensions to address practical challenges of time series tasks – e.g., a discontinuous estimation procedure for irregular time intervals, and a plug-in approach for low sample regimes.
4. We present experiments that showcase how existing methods underperform in the presence of temporal noise, while our proposed methods lead to better classifiers. This highlights the necessity of accounting for temporal noise.

RELATED WORK

Time Series Machine learning for time series, especially in healthcare, often relies on combining autoregressive approaches to modelling sequences with deep neural network based methods [11, 35, 61]. Primarily in the context of time series modelling, Recurrent Neural Networks (RNNs) and state space models have gained attention for their ability to model observed data as emanations from underlying latent states that evolve over time. Attention-based mechanisms have further enabled models to prioritize relevant segments of data [68], thereby enhancing performance in tasks such as patient outcome prediction and treatment optimization [14, 30, 60, 64, 70, 79]. These approaches primarily rely on supervised learning to model changes in latent states over time. In healthcare for example, underlying latent states can be understood as clinical labels (i.e: sick vs healthy) that patients can transition in and out of. An understudied area in this domain is label noise - what happens when we embrace the idea that real-world time series datasets contain inaccurate labels?

Label Noise Our work identifies and addresses gaps in the literature on noisy labels [3, 37, 40, 63]. The vast majority of work on noisy labels studies label noise in static prediction tasks. We consider how label noise can arise in time series – a prediction task where the noise rates can change over time. There has been some recent work on label noise in this setting [see e.g., 2, 12]. This work has focused on the task of identifying noisy instances in time series, specifically by exploiting the notion that labels at neighboring time steps are unlikely to be corrupted together. In contrast, we focus on the task of developing algorithms to train models that are robust to temporal label noise. With respect to existing work, our approach can account for label noise in tasks where labels at neighboring time steps are *more* likely to be corrupted together (e.g., due to seasonal fluctuations in annotator error). We develop a way to learn via empirical risk minimization with *noise-robust loss functions* [19, 34, 37, 43, 71]. Our work establishes the potential to learn from noisy labels in time series when we have knowledge of the underlying noise process [c.f., 43, 45], and develops methods to fit the noise model directly from noisy data [see e.g., 33, 34, 45, 75, 77, 80].

2 FRAMEWORK

Preliminaries We consider a temporal classification task over C classes and T time steps. Each instance is characterized by a triplet of *sequences* over T time steps $(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}, \tilde{\mathbf{y}}_{1:T}) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}$ where $\mathcal{X} \subseteq \mathbb{R}^{d\times T}$ represents a multivariate time-series and $\mathcal{Y} = \{1, \ldots, C\}^T$ represents a sequence of labels. Here, $\mathbf{x}_{1:T}$, $\mathbf{y}_{1:T}$, and $\tilde{\mathbf{y}}_{1:T}$ are sequences of instances, *clean labels* and *noisy labels*, respectively. For example, we can capture settings where $\mathbf{x}_{1:T}$, $\mathbf{y}_{1:T}$ are recordings from an accelerometer with true activity labels, and $\tilde{\mathbf{y}}_{1:T}$ represent noisy annotations of activity.

Under temporal label noise, the *true* label sequence $\mathbf{y}_{1:T}$ is unobserved, and we only have access to a set of n noisy instances $D = \{(\boldsymbol{x}_{1:T}, \tilde{\boldsymbol{y}}_{1:T})_i\}_{i=1}^n$. We assume that each sequence in D is generated i.i.d. from a joint distribution $P_{\mathbf{x}_{1:T},\mathbf{y}_{1:T},\tilde{\mathbf{y}}_{1:T}}$, where the label noise process can vary over time. This distribution obeys two standard assumptions in temporal modeling and label noise [4, 10, 65]:

Assumption 1 (Future Independence)**.** *A label at time t depends only on the past sequence of feature vectors up to t:* $p(\mathbf{y}_{1:T} \mid \mathbf{x}_{1:T}) = \prod_{t=1}^{T} p(\mathrm{y}_t|\mathbf{x}_{1:t})$

Assumption 2 (Feature Independence)**.** *The sequence of noisy labels is conditionally independent of the features given the true labels:* $\tilde{\mathbf{y}}_{1:t} \perp\!\!\!\perp \mathbf{x}_{1:t} \mid \mathbf{y}_{1:t}$ *for* $t = 1, \ldots, T$

These assumptions are relatively straightforward, as they require that the current observation is independent of future observations and assume a feature-independent noise regime. Assumption 1 allows the joint sequence distribution to factorize as: $p(\tilde{\mathbf{y}}_{1:T}|\mathbf{x}_{1:T}) = \prod_{t=1}^{T} q_t(\tilde{\mathrm{y}}_t|\mathbf{x}_{1:t})$. Here, we introduce q_t, a slight abuse of notation, to denote a probability that is time-dependent. Assumption 2 allows for the noisy label distribution at t to be further decomposed as:

$$q_t(\tilde{\mathrm{y}}_t|\mathbf{x}_{1:t}) = \sum_{y\in\mathcal{Y}} q_t(\tilde{\mathrm{y}}_t \mid \mathrm{y}_t = y)p(\mathrm{y}_t = y \mid \mathbf{x}_{1:t}) \tag{1}$$

2.1 LEARNING FROM TEMPORAL LABEL NOISE

Our goal is to learn a temporal classification model $\boldsymbol{h}_\theta : \mathbb{R}^{d\times t} \to \mathbb{R}^C$ with model parameters $\theta \in \Theta$ and $t \leq T$. Here, $\boldsymbol{h}_\theta(\mathbf{x}_{1:t})$ returns an estimate of $p(\mathrm{y}_t|\mathbf{x}_{1:t})$. To infer the label at time step t, $\boldsymbol{h}_\theta$ takes as input a sequence of feature vectors up to t, and outputs a sequence of labels by taking the $\arg\max$ of the predicted distribution for each time step (see e.g., [4, 65]). We estimate parameters $\hat{\theta}$ for a model robust to noise, by maximizing the expected accuracy as measured in terms of the *clean* labels:

$$\hat{\theta} = \underset{\theta\in\Theta}{\operatorname{argmax}}\, \mathbb{E}_{\mathbf{y}_{1:T}|\mathbf{x}_{1:T}} \prod_{t=1}^{T} p(\mathrm{y}_t = \boldsymbol{h}_\theta(\mathbf{x}_{1:t}) \mid \mathbf{x}_{1:t})$$

However, during training time we only have access to sequences of *noisy* labels. To demonstrate how we can sidestep this limitation, we first need to introduce a flexible way to noise rates varying over time. Existing methods assume that noise is time-invariant (Fig. 1). To relax this assumption, we capture the temporal nature of noisy labels using a *temporal label noise function*, in Def. 1.

Definition 1. Given a temporal classification task with C classes and noisy labels, the *temporal label noise function* is a matrix-valued function $\boldsymbol{Q} : \mathbb{R}_+ \to [0, 1]^{C\times C}$ that specifies the label noise distribution at any time $t > 0$.

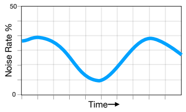
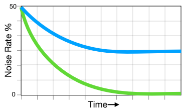
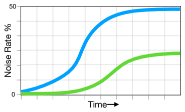
Pattern	**Depiction**	**Noise Model Q_{ij}**	**Parameters**	**Applications**
Periodic		$\frac{1}{2} + \frac{1}{2}\sin(\alpha_{ij}t + \phi_{ij})$	$\omega_{ij} = (\alpha_{ij}, \phi_{ij})$ α controls frequency ϕ controls shift	Annotation reliability **varies over time of day** – e.g., due to changes in annotator attentiveness over the day [15].
Decay		$\alpha_{ij}\exp(-\beta_{ij}t)$	$\omega_{ij} = (\alpha_{ij}, \lambda_{ij})$ α controls initial noise β controls decay rate	Annotation reliability **improves rapidly** (e.g., diagnostic accuracy of COVID-19 rapidly improved at the onset of the pandemic [46]) or **improves and plateaus** (e.g., irreducible uncertainty in labels [49])
Growth		$\frac{\alpha_{ij}}{1+\exp(-\beta_{ij}(t-\gamma_{ij}))}$	$\omega_{i,j} = (\alpha_{ij}, \beta_{ij}, \gamma_{ij})$ α controls limit β controls growth rate γ controls inflection point	Annotation reliability **decreases abruptly** – e.g., due to rapid adoption of improved clinical guidelines [62]. Or, it **decreases gradually** – e.g., due to underdiagnosis during the pandemic due to health-care disruptions [17]

Table 1: Overview of noise functions for time series classification tasks. We show the noise model $\boldsymbol{Q}(t)$ and parameters ω needed to model $q(\tilde{\mathrm{y}}_t = j \mid \mathrm{y}_t = i)$. Practitioners can model label noise by choosing a parameteric class to capture effects and fit parametric representations from data. The choice of noise model can differ across classes and subgroups (see the *Mixed* model in Section 4). We include other models and details in Appendix E.

We denote the output of the temporal noise function at time t as $\boldsymbol{Q}_t := \boldsymbol{Q}(t)$. This is a $C \times C$ matrix whose i, j^{th} entry encodes the flipping probability of observing a noisy label j given clean label i at time t: $q_t(\tilde{\mathrm{y}}_t = j \mid \mathrm{y}_t = i)$. We observe that $\boldsymbol{Q}_t$ is positive, row-stochastic, and diagonally dominant — ensuring that $\boldsymbol{Q}_t$ encodes a valid probability distribution [34, 45].

$\boldsymbol{Q}_\omega$ denotes a temporal noise function parameterized by a function with parameters ω. This parameterization can be constructed to encode essentially any temporal noise function. As shown in Table 1, we can capture a wide variety of temporal noise using this representation.

2.2 LOSS CORRECTION

Modeling temporal label noise is the first piece of the puzzle in training time series classifiers robust to label noise. However, we still need to consider how to leverage these noise models during empirical risk minimization. It remains unclear if and how existing loss correction techniques work for time series. Here we present theoretical results showing that learning is possible in our setting when we know the true temporal noise function $\boldsymbol{Q}(t)$. We include proofs in Appendix A.

We begin by treating the noisy posterior as the matrix-vector product of a noise transition matrix and a clean class posterior (Eq. (1)). To this effect, we define the *forward temporal loss*:

Definition 2. Given a temporal classification task over T time steps, a noise function $\boldsymbol{Q}(t)$, and a proper composite loss function ℓ_t [53], the *forward temporal loss* of a model $\boldsymbol{h}_\theta$ on an instance $(\tilde{\boldsymbol{y}}_{1:T}, \boldsymbol{x}_{1:T})$ is:

$$\overrightarrow{\ell}_{seq}(\tilde{\boldsymbol{y}}_{1:T}, \boldsymbol{x}_{1:T}, \boldsymbol{h}_\theta) := \sum_{t=1}^{T} \ell_t(\tilde{y}_t, \boldsymbol{Q}_t^\top \boldsymbol{h}_\theta(\boldsymbol{x}_{1:t}))$$

An intriguing property of the *forward temporal loss* is that the minimizer of the *forward temporal loss* over the noisy labels maximizes the likelihood of the data over the clean labels. This suggests that the *forward temporal loss* is robust to label noise:

Proposition 1. *A classifier that minimizes the empirical* forward temporal loss *over the noisy labels maximizes the empirical likelihood of the data over the clean labels.*

$$\underset{\theta \in \Theta}{\operatorname{argmin}}\, \mathbb{E}_{\tilde{\mathbf{y}}_{1:T}, \mathbf{x}_{1:T}} \overrightarrow{\ell}_{seq}(\tilde{\mathbf{y}}_{1:T}, \mathbf{x}_{1:T}, \boldsymbol{h}_\theta) = \underset{\theta \in \Theta}{\operatorname{argmin}} \sum_{t=1}^{T} \mathbb{E}_{\mathbf{y}_{1:t}, \mathbf{x}_{1:t}} \ell_t(\mathrm{y}_t, \boldsymbol{h}_\theta(\mathbf{x}_{1:t}))$$

Proposition 1 implies that we can *train on the noisy distribution* and learn a noise-tolerant classifier in expectation.

3 METHODOLOGY

Our results in the previous section show that we can train models that are robust to label noise. However, the temporal noise function $\boldsymbol{Q}$ is not available to the practitioner and must be learned from data. Our algorithm seeks to first learn this function from noisy data then account for noise using Def. 2. This strategy can be applied in multiple ways. In what follows, we propose a method where we jointly learn the temporal noise function and the model. We also present extensions for different use-cases in Section 3.3 and discuss how to choose between methods in Section 3.4

3.1 FORMULATION

We start with a method that simultaneously learns a time series classifier and temporal noise function. Given a noisy dataset, we learn these elements by solving the following optimization problem $\forall t \in [1, T]$:

$$\begin{aligned} \min_{\omega,\theta} \quad & \text{Vol}(\boldsymbol{Q}_\omega(t)) \\ \text{s.t.} \quad & \boldsymbol{Q}_\omega(t)^\top \boldsymbol{h}_\theta(\boldsymbol{x}_{1:t}) = p(\tilde{y}_t \mid \boldsymbol{x}_{1:t}) \end{aligned} \tag{2}$$

Eq. (2) is designed to return a faithful representation of the noise function $\hat{\omega}$ by imposing the *minimum-volume simplex* assumption [34], and a noise-tolerant temporal classifier $\hat{\theta}$ by minimizing the *forward temporal loss* in Def. 2.

Here, the objective minimizes the volume of the noise matrix, denoted as $\text{Vol}(\boldsymbol{Q}_t)$. This returns a matrix $\boldsymbol{Q}_t$, at each time step, that obeys the *minimum-volume simplex* assumption, which is a standard condition used to ensure identifiability in static classification tasks [see e.g., 34, 78]. In practice, the minimum-volume simplex assumption ensures that $\boldsymbol{Q}_t$ encloses the noisy conditional data distribution at time t: $p(\tilde{\mathrm{y}}_t \mid \mathbf{x}_{1:t})$. Here containment ensures that the estimated noise matrix could have generated each point in the noisy dataset – i.e., so that the corresponding noisy probabilities $p(\tilde{\mathrm{y}}_t \mid \mathbf{x}_{1:t})$ obey Eq. (1). Our use of this assumption guarantees the identifiability of $\boldsymbol{Q}_t$ when the posterior distribution is sufficiently-scattered over the unit simplex [see also 34, for details].

3.2 ESTIMATION PROCEDURE

The formulation above applies to any generic matrix-valued function according to Def. 1 with parameters ω. Because time series classification tasks can admit many types of temporal label noise functions (Table 1), we must ensure ω has sufficient representational capacity to handle many noise functions. Therefore in practice, we instantiate our solution as a fully connected neural network with parameters ω, $\boldsymbol{Q}_\omega(\cdot) : \mathbb{R} \to [0, 1]^{c \times c}$, adjusted to meet Def. 1 (see Appendix E.2 for implementation details).

We can now model any temporal label noise function owing to the universal approximation properties of neural networks. Provided the function space Θ defines autoregressive models of the form $p(y_t | \boldsymbol{x}_{1:t})$ (e.g., RNNs, Transformers, etc.), we can solve Eq. (2) using an augmented Lagrangian method for equality-constrained optimization problems [6]:

$$\mathcal{L}(\theta, \omega) = \frac{1}{T} \sum_{t=1}^{T} \left[\|\boldsymbol{Q}_\omega(t)\|_F + \lambda R_t(\theta, \omega) + \frac{c}{2} |R_t(\theta, \omega)|^2 \right] \tag{3}$$

Here: $\|\boldsymbol{Q}_\omega(t)\|_F$ denotes the Frobenius norm of $\boldsymbol{Q}_\omega(t)$, which acts as a convex surrogate for $\text{Vol}(\boldsymbol{Q}_\omega(t))$ [7]. Likewise, $R_t(\theta, \omega) = \frac{1}{n} \sum_{i=1}^{n} \ell_t(\tilde{y}_{t,i}, \boldsymbol{Q}_\omega(t)^\top \boldsymbol{h}_\theta(\boldsymbol{x}_{1:t,i}))$ denotes the violation of the equality constraint for each $t = 1 \dots T$. $\lambda \in \mathbb{R}_+$ is the Lagrange multiplier and $c > 0$ is a penalty parameter. Both are initially set to a default value of 1, we gradually increase the penalty parameter until the constraint holds and λ converges to the Lagrangian multiplier for the optimization problem Eq. (2) [6]. This approach recovers the best-fit parameters to the optimization problem in Eq. (2). We call this approach *Continuous Estimation*, and additional details on our implementation can be found in Appendix B.

3.3 EXTENSIONS AND ALTERNATIVE APPROACHES

We describe two alternative approaches that extend existing label noise methods to time series tasks. On there own, each can be used to fit different practitioner needs (see Section 3.4). However, these also serve as general purpose frameworks for how to modify *any* static label noise approach to the temporal setting.

Discontinuous Estimation Another approach is to assume that there is no temporal relationship between each $\boldsymbol{Q}_t$ across time and treat each time step independently. This approach is well suited for tasks where time steps are unevenly spaced (e.g., some clinical data involves labels collected over years or at different frequencies [51]). We can address such tasks through an estimation procedure where we learn a model that achieves the same objective without assuming continuity. In contrast to Eq. (3), this approach $\hat{\boldsymbol{Q}}_t$ is parameterized with a *separate* set of trainable real-valued weights, fitting the parameters at each time step using the data from that time step. This approach can be generalized to extend any state-of-the-art technique for noise transition matrix estimation in the static setting (e.g., Li et al. [34], Yong et al. [78], etc.).

Plug-In Estimation It is also advantageous to have a simple, plug-in estimator of the temporal noise function. Plug-in estimators are model-agnostic, can flexibly be deployed to other models, and can be efficient to estimate. We can construct a plug-in estimator of temporal label noise using *anchor points*, instances whose labels are known to be correct. Empirical estimates of the class probabilities of anchor points can be used to estimate the noise function. This estimate can then be plugged into Def. 2 to train a noise-tolerant time series classifier. Formally, in a temporal setting, anchor points [36, 45, 75] are instances that maximize the probability of belonging to class i at time step t:

$$\bar{\boldsymbol{x}}_t^i = \arg\max_{\boldsymbol{x}_t} p(\tilde{\mathrm{y}}_t = i \mid \boldsymbol{x}_{1:t}) \tag{4}$$

Since $p(\mathrm{y}_t = i \mid \bar{\boldsymbol{x}}_{1:t}^i) \approx 1$ for the clean label, we can express each entry of the label noise matrix as:

$$\hat{\boldsymbol{Q}}(t)_{i,j} = p(\tilde{\mathrm{y}}_t = j \mid \bar{\boldsymbol{x}}_{1:t}^i) \tag{5}$$

We construct a plug-in estimate of $\hat{\boldsymbol{Q}}(t)$ using a two-step approach: we identify anchor points for each class $y \in \mathcal{Y}$ and $t = 1, \ldots, T$ and set each entry of $\hat{\boldsymbol{Q}}(t)_{i,j}$ by Eq. (5) (see Appendix C for formalization). This is a generalization of the approach in static prediction tasks by Patrini et al. [45].

3.4 DISCUSSION

All three methods in Section 3.2 and 3.3 improve performance in temporal classification tasks by accounting for temporal label noise (see e.g., Fig. 1 and Section 4). However, they each have their own strengths and limitations. Here we provide practical guidance for users to discriminate between methods:

- Continuous Estimation (Continuous) imposes continuity across time steps – assuming that nearby points likely have similar noise levels. This assumption can improve reconstruction, and thus performance, in settings with multiple time steps as it reduces the effective number of model parameters. Conversely, it may also lead to misspecification in settings that exhibit discontinuity.
- Discontinuous Estimation (Discontinuous) can handle discontinuous temporal noise processes – assuming the noise levels of nearby points are independent of one another. However, it requires fitting more parameters, which scales according to T – this can lead to computational challenges and overfitting, especially for long sequences.
- Plug-In Estimation (Plug-In) has a simpler optimization problem, useful when separate datasets are used in noise estimation and classifier training, but verifying anchor points is difficult [75].

4 EXPERIMENTS

We benchmark our methods on a collection of temporal classification tasks from real-world applications. Our goal is to evaluate methods in terms of robustness to temporal label noise, and characterize when it is important to consider such noise. Appendix E has more details on setup and results.

4.1 SETUP

Datasets We use four real-world datasets from healthcare. Each dataset reflects binary classification tasks over a complex feature space where we have labeled examples across multiple time steps and where the labels are likely to exhibit label noise. The tasks include: activity recognition from temporal accelerometer data in adults (`moving`) [54]), activity recognition in seniors (`senior` [39]), and sleep detection (`sleeping` [20]), and blink detection (`blinking` [55]) both from continuous EEG data.

Noise Models We consider labels in the training sample as "ground truth" clean labels and corrupt them using one of five temporal label noise functions, as well as a time-independent function (i.e., static label noise) at various noise levels. Fig. 2 shows two of these functions. This setup reflects a standard approach used to evaluate algorithms for label noise in the literature [see e.g., 43, 37, 38, 78, 45]. In our setting, it allows us to control the noise model and explain the mechanism through ablation. Specifically, we can (1) evaluate the performance of each method across different noise models; (2) attribute the gains to our ability to capture for temporal noise and optimize jointly via ablation.

Methods We train time series classifiers using the three techniques described in Section 3: Continuous, Discontinuous, and Plug-In. As baselines, we compare these models to NLL loss that assumes no label noise exists (Ignore), and methods that assume a static noise model across time steps: Anchor [36, 45, 75] and VolMinNet [34].

Evaluation We split each dataset into a *noisy* training sample (80%, used to train the models and correct for label noise), and a *clean* test sample (20%, used to compute unbiased estimates of out-of-sample performance). We evaluate each model in terms of the *test accuracy* on test data and characterize how well each method learns the temporal noise function, using the *Approximation Error* $\text{Error}(\boldsymbol{Q}_t, \hat{\boldsymbol{Q}}_t) := \frac{1}{T}\sum_{t=1}^{T} \|\boldsymbol{Q}_t - \hat{\boldsymbol{Q}}_t\|$ between the true $\boldsymbol{Q}_t$ and estimated $\hat{\boldsymbol{Q}}_t$ for all t.

4.2 RESULTS

We summarize the results of our experiments for two temporal noise models in Table 2 and Table 3. We show the ability of our methods to accurately learn the underlying temporal noise function in Fig. 2. Additional results for other levels of noise and noise models as well as multi-class classification can be found in the Appendices and in our anonymized repository. In what follows we discuss these results.

On the Improved Performance from Modeling Temporal Noise First, we show clear value in accounting for temporal label noise. Table 2 shows the performance of each method on all five datasets. We find that the temporal methods are consistently more accurate than their non-temporal counterparts, highlighting the impact of modelling temporal noise. For example, Plug-In is a temporal extension of Anchor and we see that Plug-In outperforms Anchor in most settings. The results in Table 2 highlight the significant gains from accounting for temporal label noise. For example, on `moving`, the temporal method reduces the test error by over 10% compared to the nearest-performing static counterpart. This shows that our methods are robust to label noise despite being trained on data that is 30% corrupted with noisy labels, with no prior knowledge of the noise. This can have important consequences in e.g., health data in time series, where the noise is uknown and accurate models can make life-or-death decisions.

Among the temporal methods, Continuous achieves the best performance in comparison to Plug-In and Discontinuous. Continuous' superiority is even clearer when compared to the static methods. Comparing the results for Table 2 (which assumes a Mixed noise model) and Table 3 (which assumes a Sinusoidal noise model), we can see that these results hold across multiple types of temporal label noise. Fig. 2 demonstrates the ability of of our methods to accurately reconstruct the underlying label noise. More importantly, the benefit of Continuous becomes more evident as the amount of noise increases in the data. In all these cases, we observe that the temporal methods are consistently more robust to both temporal and static label noise. Overall, these findings suggest that we can improve performance by explicitly modeling how noise varies across time instead of assuming it is distributed uniformly in time.

			Static		*Temporal*		
Dataset	Metric	Ignore	Anchor	VolMinNet	Plug-In	Discontinuous	Continuous
`moving` [54]	Test Error	29.4 ± 1.7%	20.9 ± 2.6%	14.9 ± 2.7%	20.0 ± 1.8%	15.9 ± 2.7%	**4.2 ± 2.2%**
$n = 192, d = 14, T = 50$	Approx. Error	–	42.4 ± 3.4%	35.3 ± 0.8%	36.8 ± 1.7%	32.6 ± 0.5%	**10.3 ± 3.9%**
`senior` [39]	Test Error	22.7 ± 1.7%	20.7 ± .01%	19.0 ± 0.7%	18.8 ± 1.1%	13.6 ± 1.2%	**11.0 ± 0.3%**
$n = 444, d = 6, T = 100$	Approx. Error	–	35.9 ± 2.5%	36.3 ± 0.4%	26.9 ± 1.2%	21.7 ± 0.2%	**6.4 ± 0.8%**
`blinking` [55]	Test Error	34.1 ± 2.0%	34.1 ± 2.3%	29.6 ± 2.2%	29.6 ± 2.8%	29.9 ± 3.0%	**29.6 ± 2.3%**
$n = 299, d = 14, T = 50$	Approx. Error	–	35.3 ± 0.8%	35.2 ± 0.7%	19.6 ± 1.1%	26.6 ± 0.9%	**14.9 ± 2.3%**
`sleeping` [20]	Test Error	28.7 ± 0.8%	24.9 ± 1.1%	26.8 ± 1.4%	20.4 ± 1.8%	19.6 ± 0.8%	**16.3 ± 0.4%**
$n = 964, d = 7, T = 100$	Approx. Error	–	34.3 ± 1.8%	41.8 ± 0.1%	19.1 ± 3.5%	22.4 ± 0.2%	**4.9 ± 0.5%**

Table 2: Model performance and approximation error for all methods and datasets. We report the clean test error (%) and mean approximation error of $\hat{\boldsymbol{Q}}(t)$ ± st.dev over 10 runs. The best-performing methods are highlighted in Green. Continuous outperforms all baselines. Results are shown for the *Mixed* noise function (average 30% label noise across all time steps, one class with decreasing noise rate, one class with increasing noise rate). Additional noise model results are in Appendix F.

			Static		*Temporal*		
Dataset	Metric	Ignore	Anchor	VolMinNet	Plug-In	Discontinuous	Continuous
`moving` [54]	Test Error	24.0 ± 5.1%	18.0 ± 3.6%	13.5 ± 6.0%	18.5 ± 4.3%	14.0 ± 5.7%	**1.7 ± 0.6%**
$n = 192, d = 14, T = 50$	Approx. Error	–	49.9 ± 4.7%	43.5 ± 3.0%	48.1 ± 4.4	33.5 ± 5.0%	**9.1 ± 1.6%**
`senior` [39]	Test Error	22.5 ± 2.2%	20.1 ± 1.5%	16.4 ± 2.9%	19.9 ± 1.3%	14.1 ± 2.6%	**10.6 ± 0.7%**
$n = 444, d = 6, T = 100$	Approx. Error	–	47.3 ± 4.0%	42.4 ± 3.9%	46.9 ± 3.9%	20.5 ± 2.1%	**9.7 ± 2.7%**
`blinking` [55]	Test Error	37.4 ± 3.1%	35.8 ± 2.7%	33.2 ± 2.8%	35.8 ± 2.3%	32.0 ± 2.9%	**30.4 ± 3.1%**
$n = 299, d = 14, T = 50$	Approx. Error	–	49.3 ± 4.2%	42.4 ± 3.9%	46.4 ± 4.4%	24.7 ± 3.3%	**10.5 ± 3.3%**
`sleeping` [20]	Test Error	23.3 ± 3.1%	22.8 ± 2.7%	22.4 ± 3.0%	22.1 ± 2.0%	20.1 ± 3.5%	**15.2 ± 0.8%**
$n = 964, d = 7, T = 100$	Approx. Error	–	44.9 ± 4.0%	42.6 ± 4.1%	43.9 ± 4.3%	19.1 ± 1.8%	**8.8 ± 2.6%**

Table 3: Model performance and approximation error for all methods and datasets. We report the clean test error (%) and mean approximation error of $\hat{\boldsymbol{Q}}(t)$ ± st.dev over 10 runs. The best-performing methods are highlighted in Green. Continuous outperforms all baselines. Results are shown for the *Sinusoidal* noise function (average 30% label noise across all time steps, noise rate fluctuates over time). Additional noise model results are in Appendix F.

On Learning the Temporal Noise Function Our results show that the performance increases hand in hand with our ability to estimate the noise model. In particular, we observe that low values of reconstruction consistently lead to low values of error rates.

We note that Continuous has the best reconstruction error (Approx. Error) on all datasets (Table 2) We qualitatively compare estimated noise functions and the ground truth for Continuous, Discontinuous, and other baselines in Fig. 2. We can see that our method consistently estimates the noise function with lower mean absolute error across different families of noise functions. Appendix F has extended results on all other datasets (including multiclass).

On the Risk of Misspecification In many practical applications, we face the risk of misspecifying noise models due to a lack of access to true labels that would verify the underlying noise structure. Our results emphasize the limitations of static approaches, which will always fail to capture the correct noise function when it is temporal. As shown in Section 2.2, accurately specifying the temporal noise process leads to more noise-tolerant models. However, most existing $\boldsymbol{Q}$-estimators assume static, time-invariant label noise, thereby prone to misspecification.

We validate this experimentally by comparing our *forward temporal loss* method, which uses the true temporal noise function, to a static approximation (the average noise over time). As shown in Fig. 5 (Appendix F), static misspecification consistently results in poor performance, particularly in the *Mixed* noise setting, where class-conditional noise amplifies this weakness. Notably, when the temporal noise process is uniform, performance remains unchanged, indicating no downside to modeling temporal effects.rewrite this simpler two sentences

In summary, practitioners who assume that label noise is static in their time series datasets are at risk of poor model performance due to the inherent misspecification of the noise model. However, using

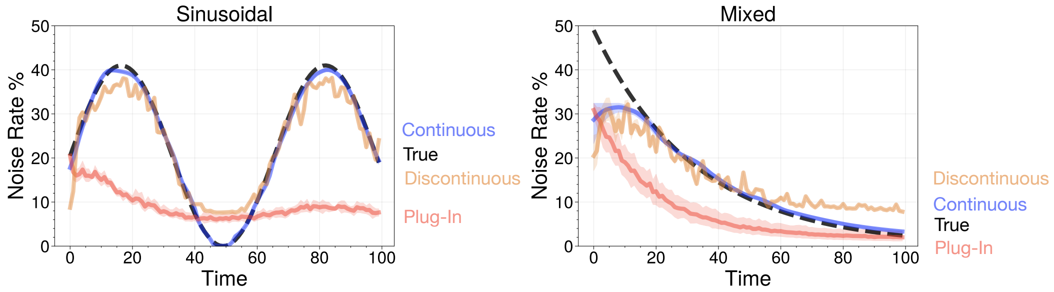


Figure 2: Our Temporal methods can learn the true label noise function across different noise patterns. The resulting noise models have variable lower reconstruction error but are superior to static approaches. We show results for *Sinusoidal* and *Mixed* noise (class-conditional noise each class has increasing or decreasing noise over time) on `adult`. We show the noise rate for the negative class only for clarity.

our temporal approaches they can relax this assumption to admit different types of temporal label noise. There is no cost to doing so, as our methods are competitive even if the noise model is truly static. Practitioners have everything to gain and nothing to lose by accounting for temporal label noise.

5 REAL-WORLD DEMONSTRATION

We now present experiments on a real-world stress detection task where we have both clean labels and noisy labels. This demonstrates a natural source of temporal label noise, without synthetic noise injection, and the clear effectiveness of our methods. We use a dataset from Goodday [22] monitoring stress in healthcare workers using both self-reported and physiological measures. Stress detection models are a common feature in smartwatches [9, 18] and can guide interventions to mitigate clinical burnout [21, 26].

Setup Temporal label noise arises naturally in this setting due to subjectivity, forgetfulness, and seasonal patterns in self-reported outcomes over time [48]. Given these effects, physiological measures are the standard to measure stress and guide interventions. However, self-reported stress labels are often the only information available for modeling. Given just the self-reported, *noisy* stress label, we want to train a time series classifier that generalizes well to the *clean* physiological indicator of stress.

The dataset contains $n = 289$ unique individuals over $T = 50$ time steps (days). Each sequence has $d = 9$ features. Here, the *noisy label* is: $\tilde{y}_{i,t} = 1$ if person i subjectively self-reports stress on day t. The *clean label* is $y_{i,t} = 1$ if the objective physiological measure of stress for person i indicates stress on day t. We train sequential classifiers to predict stress over time. Our training setup is identical to the one in Section 4, except that our training labels reflect real-world noisy labels. We report a subset of baselines to showcase the baseline via ablation.

Results In Fig. 3, the black line indicates the average disagreement rates between the noisy self-reported label of stress and the clean physiological label over time. We can see clear temporal patterns in label noise, where participants under-report stress in a seasonal manner. Temporal label noise methods (in this case, Continuous) perform the best at approximating this noise and therefore lead to superior predictive performance on the clean physiological labels. These results are consistent with the experiments in Section 4.2: improvements in estimation error across time lead to improvements in training and test accuracy (Table 4).

6 CONCLUSION

Many classification tasks, such as those found in healthcare, require classifying labels in a temporal fashion under unobserved label noise. It is well-known that noisy labels cause problems for static classification. In time series, however, labels that are observed temporally may admit differing noise rates over time. For example, label quality may improve or worsen over time. Existing methods work

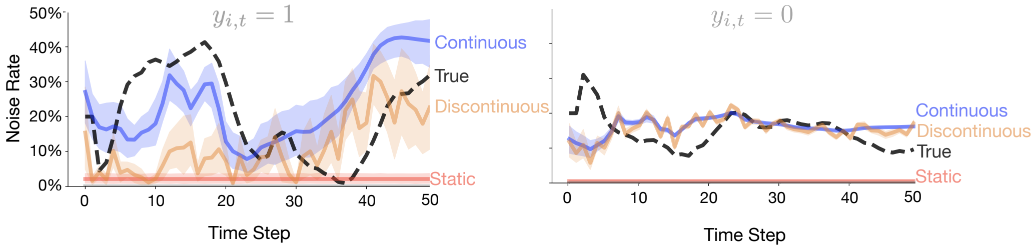


Figure 3: Temporal effects in label noise in a real-world stress detection task. We show noise rates when individuals are stressed (left) and not stressed (right). True noise rate is the average disagreement rates between clean and noisy labels over time. We can see clear, temporal label noise patterns, Our temporal label noise methods do a superior job of approximating it.

		Noise Estimation	**Model Performance**	
Method	**What It Represents**	**Approx. Error**	**Train Error**	**Test Error**
Ignore	ignoring label noise completely	17.6 ± 0.0%	31.0 ± 0.3%	31.1 ± 4.3%
Static	static label noise correction per time step	16.4 ± 1.1%	31.0 ± 0.3%	29.5 ± 4.0%
Discontinuous	temporal label noise correction with discontinuity	**11.4 ± 0.8%**	**27.1 ± 0.7%**	**26.0 ± 2.5%**
Continuous	temporal label noise correction imposing continuity	**9.7 ± 1.3%**	**21.1 ± 0.5%**	**25.5 ± 1.7%**

Table 4: Noise rate estimation error and model performance for stress detection. We train sequential classifiers to predict stress using the training setup in Section 4. We report results for four methods to highlight the mechanism driving performance improvements through an ablation study. We show the approximation error noise rates (left); and the clear label error rate on the training sample (middle) and test sample (right) All values correspond to the mean 10-CV estimates ± st.dev.

to construct noise-tolerant classifiers in the static setting and are ill-equipped to handle temporal label noise. Our work shows existing methods substantially underperform when subject to *temporal label noise*. To mitigate this, we show how to learn provably robust classifiers from time series data that have been corrupted with temporal label noise using knowledge of the underlying noise function. Given that the noise function is often unknown, we also propose methods to learn noise-tolerant time series classifiers without prior knowledge of the temporal noise function. Finally, we perform a demonstration on a real-world source of temporal label noise – self-reported labels in digital health studies. In this setting, we find our methods continue to outperform methods that ignore such noise. We hope that this can inspire the curation of more time-series datasets with noisy and clean annotations.

Limitations and Future Work Our methods rely on Assumption 1 and 2. Though these are standard assumptions in time series modeling, they may be difficult to verify in practice. Existing work in Empirical Risk Minimization (ERM) on highly dependent sequences may serve as a promising direction for relaxing these assumptions [41, 42, 56]. We assume that all time series in a dataset have the same underlying label noise function. This may not be true as different annotators can introduce differing noise patterns. We can potentially relax this assumption by considering multi-annotator noisy labels works [59, 32, 66] or generative models that admit clusters of temporal noise patterns (*i.e.*, mixture modeling).

REFERENCES

[1] Aroyo, Lora and Chris Welty. Truth is a lie: Crowd truth and the seven myths of human annotation. *AI Magazine*, 36(1):15–24, 2015.

[2] Atkinson, Gentry and Vangelis Metsis. Identifying label noise in time-series datasets. In *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, pages 238–243, 2020.

[3] Beigman, Eyal and Beata Beigman Klebanov. Learning with annotation noise. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 280–287, 2009.

[4] Bengio, Yoshua, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. *Advances in Neural Information Processing Systems*, 13, 2000.

[5] Bertrand, Marianne and Sendhil Mullainathan. Do people mean what they say? implications for subjective survey data. *American Economic Review*, 91(2):67–72, 2001.

[6] Bertsekas, Dimitri P. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

[7] Boyd, Stephen and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[8] Carpenter, Christopher. Seasonal variation in self-reports of recent alcohol consumption: racial and ethnic differences. *Journal of Studies on Alcohol*, 64(3):415–418, 2003.

[9] Chen, Jerry et al. Pain and Stress Detection Using Wearable Sensors and Devices—A Review. *Sensors*, 2021.

[10] Choe, Yo Joong, Jaehyeok Shin, and Neil Spencer. Probabilistic interpretations of recurrent neural networks. *Probabilistic Graphical Models*, 2017.

[11] Choi, Edward, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Using recurrent neural network models for early detection of heart failure onset. *Journal of the American Medical Informatics Association*, 24(2): 361–370, 2017.

[12] Cui, Beilei, Minqing Zhang, Mengya Xu, An Wang, Wu Yuan, and Hongliang Ren. Rectifying noisy labels with sequential prior: Multi-scale temporal feature affinity learning for robust video segmentation. *arXiv preprint arXiv:2307.05898*, 2023.

[13] Day, Thomas G, John M Simpson, Reza Razavi, and Bernhard Kainz. Improving image labelling quality. *Nature Machine Intelligence*, 5(4):335–336, 2023.

[14] Duan, Huilong, Zhoujian Sun, Wei Dong, Kunlun He, and Zhengxing Huang. On clinical event prediction in patient treatment trajectory using longitudinal electronic health records. *IEEE Journal of Biomedical and Health Informatics*, 24(7):2053–2063, 2019.

[15] Folkard, Simon. Diurnal variation in logical reasoning. *British Journal of Psychology*, 66(1):1–8, 1975.

[16] Frénay, Benoît and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869, 2013.

[17] Gandhi, Tejal K and Hardeep Singh. Reducing the risk of diagnostic error in the covid-19 era. *Journal of Hospital Medicine*, 15(6):363, 2020.

[18] Gedam, Shruti et al. A Review on Mental Stress Detection Using Wearable Sensors and Machine Learning Techniques. *IEEE Access*, 2021.

[19] Ghosh, Aritra, Himanshu Kumar, and P Shanti Sastry. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[20] Goldberger, Ary L, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101 (23):e215–e220, 2000.

[21] Goodday, Sarah M. et al. Unlocking stress and forecasting its consequences with digital technology. *npj Digital Medicine*, 2019.

[22] Goodday, Sarah M. et al. An Alternative to the Light Touch Digital Health Remote Study: The Stress and Recovery in Frontline COVID-19 Health Care Workers Study. *JMIR Formative Research*, 2021.

[23] Han, Bo, Quanming Yao, Tongliang Liu, Gang Niu, Ivor W Tsang, James T Kwok, and Masashi Sugiyama. A survey of label-noise representation learning: Past, present and future. *arXiv preprint arXiv:2011.04406*, 2020.

[24] Hsieh, Gary and Rafał Kocielnik. You get who you pay for: The impact of incentives on participation bias. In *Proceedings of the 19th ACM Conference on Computer-supported Cooperative work & Social Computing*, pages 823–835, 2016.

[25] Inel, Oana, Khalid Khamkham, Tatiana Cristea, Anca Dumitrache, Arne Rutjes, Jelle Ploegvan der , Lukasz Romaszko, Lora Aroyo, and Robert-Jan Sips. Crowdtruth: Machine-human computation framework for harnessing disagreement in gathering annotated data. In *The Semantic Web–ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part II 13*, pages 486–504. Springer, 2014.

[26] Iqbal, Talha et al. A Review of Biophysiological and Biochemical Indicators of Stress for Connected and Preventive Healthcare. *Diagnostics*, 2021.

[27] Jambigi, Neetha, Tirtha Chanda, Vishnu Unnikrishnan, and Myra Spiliopoulou. Assessing the difficulty of labelling an instance in crowdworking. In *Workshops of the European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 363–373. Springer, 2020.

[28] Jung, Hyun, Yubin Park, and Matthew Lease. Predicting next label quality: A time-series model of crowdwork. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 2, pages 87–95, 2014.

[29] Kiyokawa, Takuya, Keita Tomochika, Jun Takamatsu, and Tsukasa Ogasawara. Fully automated annotation with noise-masked visual markers for deep-learning-based object detection. *IEEE Robotics and Automation Letters*, 4(2):1972–1977, 2019.

[30] Lee, Jeong Min and Milos Hauskrecht. Modeling multivariate clinical event time-series with recurrent temporal mechanisms. *Artificial intelligence in medicine*, 112:102021, 2021.

[31] Li, Junnan, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020.

[32] Li, Shikun, Shiming Ge, Yingying Hua, Chunhui Zhang, Hao Wen, Tengfei Liu, and Weiqiang Wang. Coupled-view deep classifier learning from multiple noisy annotators. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4667–4674, 2020.

[33] Li, Shikun, Xiaobo Xia, Hansong Zhang, Yibing Zhan, Shiming Ge, and Tongliang Liu. Estimating noise transition matrix with label correlations for noisy multi-label learning. *Advances in Neural Information Processing Systems*, 35:24184–24198, 2022.

[34] Li, Xuefeng, Tongliang Liu, Bo Han, Gang Niu, and Masashi Sugiyama. Provably end-to-end label-noise learning without anchor points. In *International Conference on Machine Learning*, pages 6403–6413. PMLR, 2021.

[35] Lipton, Zachary C, David C Kale, Charles Elkan, and Randall Wetzel. Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*, 2015.

[36] Liu, Tongliang and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):447–461, 2015.

[37] Liu, Yang and Hongyi Guo. Peer loss functions: Learning from noisy labels without knowing noise rates. In *International Conference on Machine Learning*, pages 6226–6236. PMLR, 2020.

[38] Liu, Yang, Hao Cheng, and Kun Zhang. Identifiability of label noise transition matrix. In *International Conference on Machine Learning*, pages 21475–21496. PMLR, 2023.

[39] Logacjov, Aleksej and Astrid Ustad. HAR70+. UCI Machine Learning Repository, 2023. DOI: https://doi.org/10.24432/C5CW3D.

[40] Long, Philip M and Rocco A Servedio. Random classification noise defeats all convex potential boosters. In *International Conference on Machine Learning*, pages 608–615, 2008.

[41] Mariet, Zelda and Vitaly Kuznetsov. Foundations of sequence-to-sequence modeling for time series. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 408–417. PMLR, 2019.

[42] McDonald, Daniel J, Cosma Rohilla Shalizi, and Mark Schervish. Nonparametric risk bounds for time-series forecasting. *The Journal of Machine Learning Research*, 18(1):1044–1083, 2017.

[43] Natarajan, Nagarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. *Advances in Neural Information Processing Systems*, 26, 2013.

[44] Norden, Matthias, Oliver T Wolf, Lennart Lehmann, Katja Langer, Christoph Lippert, and Hanna Drimalla. Automatic detection of subjective, annotated and physiological stress responses from video data. In *10th International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 1–8. IEEE, 2022.

[45] Patrini, Giorgio, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1944–1952, 2017.

[46] Peacock, Hanna M, Cindy De Gendt, Geert Silversmit, Sandra Nuyts, Jan Casselman, Jean-Pascal Machiels, Francesco Giusti, Bart Van Gool, Vincent Vander Poorten, and Liesbet Van Eycken. Stage shift and relative survival for head and neck cancer during the 2020 covid-19 pandemic: a population-based study of temporal trends. *Frontiers in Oncology*, 13, 2023.

[47] Philbrick, Kenneth A, Alexander D Weston, Zeynettin Akkus, Timothy L Kline, Panagiotis Korfiatis, Tomas Sakinis, Petro Kostandy, Arunnit Boonrod, Atefeh Zeinoddini, Naoki Takahashi, et al. Ril-contour: a medical imaging dataset annotation tool for and with deep learning. *Journal of Digital Imaging*, 32: 571–581, 2019.

[48] Pierson, Emma et al. Daily, weekly, seasonal and menstrual cycles in women's mood, behaviour and vital signs. *Nature Human Behaviour*, 2021.

[49] Pusic, Martin V, Amy Rapkiewicz, Tenko Raykov, and Jonathan Melamed. Estimating the irreducible uncertainty in visual diagnosis: Statistical modeling of skill using response models. *Medical Decision Making*, 43(6):680–691, 2023.

[50] Quisel, Tom, Luca Foschini, Alessio Signorini, and David C Kale. Collecting and analyzing millions of mhealth data streams. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1971–1980, 2017.

[51] Ramirez, Andrea H, Lina Sulieman, David J Schlueter, Alese Halvorson, Jun Qian, Francis Ratsimbazafy, Roxana Loperena, Kelsey Mayo, Melissa Basford, Nicole Deflaux, et al. The all of us research program: data quality, utility, and diversity. *Patterns*, 3(8), 2022.

[52] Raykar, Vikas C, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(4), 2010.

[53] Reid, Mark D. and Robert C. Williamson. Composite binary losses. *Journal of Machine Learning Research*, 11(83):2387–2422, 2010. URL http://jmlr.org/papers/v11/reid10a.html.

[54] Reyes-Ortiz, Jorge, Davide Anguita, Alessandro Ghio, Luca Oneto, and Xavier Parra. Human Activity Recognition Using Smartphones. UCI Machine Learning Repository, 2012. DOI: https://doi.org/10.24432/C54S4K.

[55] Roesler, Oliver. EEG Eye State. UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C57G7J.

[56] Roy, Abhishek, Krishnakumar Balasubramanian, and Murat A Erdogdu. On empirical risk minimization with dependent and heavy-tailed data. *Advances in Neural Information Processing Systems*, 34:8913–8926, 2021.

[57] Sano, Akane and Rosalind W Picard. Stress recognition using wearable sensors and mobile phones. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, pages 671–676. IEEE, 2013.

[58] Schaekermann, Mike, Edith Law, Kate Larson, and Andrew Lim. Expert disagreement in sequential labeling: A case study on adjudication in medical time series analysis. In *SAD/CrowdBias@ HCOMP*, pages 55–66, 2018.

[59] Schmarje, Lars, Vasco Grossmann, Claudius Zelenka, Sabine Dippel, Rainer Kiko, Mariusz Oszust, Matti Pastell, Jenny Stracke, Anna Valros, Nina Volkmann, et al. Is one annotation enough?-a data-centric image classification benchmark for noisy and ambiguous label estimation. *Advances in Neural Information Processing Systems*, 35:33215–33232, 2022.

[60] Sha, Ying and May D Wang. Interpretable predictions of clinical outcomes with an attention-based recurrent neural network. In *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 233–240, 2017.

[61] Shamshirband, Shahab, Mahdis Fathi, Abdollah Dehzangi, Anthony Theodore Chronopoulos, and Hamid Alinejad-Rokny. A review on deep learning approaches in healthcare systems: Taxonomies, challenges, and open issues. *Journal of Biomedical Informatics*, 113:103627, 2021.

[62] Shekelle, Paul, Martin P Eccles, Jeremy M Grimshaw, and Steven H Woolf. When should clinical guidelines be updated? *Bmj*, 323(7305):155–157, 2001.

[63] Sugiyama, Masashi, Tongliang Liu, Bo Han, Yang Liu, and Gang Niu. Learning and mining with noisy labels. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 5152–5155, 2022.

[64] Suo, Qiuling, Fenglong Ma, Giovanni Canino, Jing Gao, Aidong Zhang, Pierangelo Veltri, and Gnasso Agostino. A multi-task framework for monitoring health conditions via attention-based recurrent neural networks. In *AMIA annual symposium proceedings*, volume 2017, page 1665. American Medical Informatics Association, 2017.

[65] Sutskever, Ilya, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[66] Tanno, Ryutaro, Ardavan Saeedi, Swami Sankaranarayanan, Daniel C Alexander, and Nathan Silberman. Learning from noisy labels by regularized estimation of annotator confusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11244–11253, 2019.

[67] Vaizman, Yonatan, Katherine Ellis, and Gert Lanckriet. Recognizing detailed human context in the wild from smartphones and smartwatches. *IEEE pervasive computing*, 16(4):62–74, 2017.

[68] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[69] Wallace, Shaun, Tianyuan Cai, Brendan Le, and Luis A Leiva. Debiased label aggregation for subjective crowdsourcing tasks. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, pages 1–8, 2022.

[70] Wang, Lu, Wei Zhang, Xiaofeng He, and Hongyuan Zha. Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2447–2456, 2018.

[71] Wang, Yisen, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330, 2019.

[72] Wei, Jiaheng, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. *arXiv preprint arXiv:2110.12088*, 2021.

[73] Westbrook, Johanna I, Enrico Coiera, William TM Dunsmuir, Bruce M Brown, Norm Kelk, Richard Paoloni, and Cuong Tran. The impact of interruptions on clinical task completion. *BMJ Quality & Safety*, 19(4):284–289, 2010.

[74] Whitehill, Jacob, Ting-fan Wu, Jacob Bergsma, Javier Movellan, and Paul Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in Neural Information Processing Systems*, 22, 2009.

[75] Xia, Xiaobo, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? *Advances in Neural Information Processing Systems*, 32, 2019.

[76] Xiao, Tong, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2691–2699, 2015.

[77] Yao, Yu, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. Dual t: Reducing estimation error for transition matrix in label-noise learning. *Advances in Neural Information Processing Systems*, 33:7260–7271, 2020.

[78] Yong, LIN, Renjie Pi, Weizhong Zhang, Xiaobo Xia, Jiahui Gao, Xiao Zhou, Tongliang Liu, and Bo Han. A holistic view of label noise transition matrix in deep learning and beyond. In *International Conference on Learning Representations*, 2023.

[79] Zhang, Shao, Jianing Yu, Xuhai Xu, Changchang Yin, Yuxuan Lu, Bingsheng Yao, Melanie Tory, Lace M Padilla, Jeffrey Caterino, Ping Zhang, et al. Rethinking human-ai collaboration in complex medical decision making: A case study in sepsis diagnosis. *arXiv preprint arXiv:2309.12368*, 2023.

[80] Zhang, Yivan, Gang Niu, and Masashi Sugiyama. Learning noise transition matrix from only noisy labels via total variation regularization. In *International Conference on Machine Learning*, pages 12501–12512. PMLR, 2021.

A PROOFS

In what follows, we use vector notation for completeness and clarity of exposition. We use Assumption 1 and Assumption 2 to allow us to factor the noisy label distribution as follows:

$$\boldsymbol{p}(\tilde{\mathbf{y}}_{1:T} \mid \mathbf{x}_{1:T}) = \prod_{t=1}^{\top} \boldsymbol{q}_t(\tilde{\mathrm{y}}_t \mid \mathbf{x}_{1:t}). \tag{6}$$

Definitions We start by defining some of the quantities that will be important for the proof:

Quantity	**Definition**
$\boldsymbol{p}(\mathrm{y}_t \mid \mathbf{x}_{1:t}) := \big[p(\mathrm{y}_t = c \mid \mathbf{x}_{1:t})\big]^{\top}_{c=1:C}$	Vector of probabilities for each label value, for the clean label distribution $\boldsymbol{p}(\mathrm{y}_t \mid \mathbf{x}_{1:t}) \in \mathbb{R}^{C\times 1}$
$\boldsymbol{p}(\tilde{\mathrm{y}}_t \mid \mathbf{x}_{1:t}) := \big[p(\mathrm{y}_t = c \mid \mathbf{x}_{1:t})\big]^{\top}_{c=1:C}$	Vector of probabilities for each possible label value, for the noisy label distribution $\boldsymbol{p}(\tilde{\mathrm{y}}_t \mid \mathbf{x}_{1:t}) \in \mathbb{R}^{C\times 1}$
$\boldsymbol{h}_\theta(\boldsymbol{x}_{1:t}) = \boldsymbol{p}_\theta(\mathrm{y}_t \mid \mathbf{x}_{1:t} = \boldsymbol{x}_{1:t})$	Classifier that predicts label distribution at t given preceding observations $\mathbb{R}^{d\times t} \to \mathbb{R}^C$
$\boldsymbol{h}_\theta(\boldsymbol{x}_{1:t}) = \boldsymbol{\psi}^{-1}(\boldsymbol{g}_\theta(\boldsymbol{x}_{1:t}))$	When h_θ is a deep network, g_θ is the final logits, and $\boldsymbol{\psi} : \Delta^{C-1} \to \mathbb{R}^C$ represents an invertible link function (e.g., softmax)
$\boldsymbol{Q}_t := \big[q_t(\tilde{\mathrm{y}}_t = k \mid \mathrm{y}_t = j)\big]_{j,k}$	The temporal noise matrix at time t: $\boldsymbol{Q}_t \in \mathbb{R}^{C\times C}$
$\ell_t(y_t, \boldsymbol{h}_\theta(\boldsymbol{x}_{1:t})) = -\log p_\theta(\mathrm{y}_t = y_t \mid \mathbf{x}_{1:t} = \boldsymbol{x}_{1:t})$	Loss at t: $\mathcal{Y} \times \mathbb{R}^C \to \mathbb{R}$
$\ell_{\psi,t}(y_t, \boldsymbol{h}_\theta(\boldsymbol{x}_{1:t})) = \ell_t(y_t, \boldsymbol{\psi}^{-1}\boldsymbol{h}_\theta(\boldsymbol{x}_{1:t}))$	A composite loss function using a link function $\boldsymbol{\psi}$
$\boldsymbol{\ell}_t(\boldsymbol{h}_\theta(\boldsymbol{x}_{1:t})) = \big[\ell_t(c, \boldsymbol{h}_\theta(\boldsymbol{x}_{1:t}))\big]^{\top}_{c=1:C}$	Vector of NLL losses for each possible value of the ground truth $\mathbb{R}^C \to \mathbb{R}^C$
$\overrightarrow{\ell}_{t,\psi}(c, \boldsymbol{h}_\theta(\boldsymbol{x}_{1:t})) = \ell_t(c, \boldsymbol{Q}_t^{\top} \cdot \boldsymbol{\psi}^{-1}(\boldsymbol{g}_\theta))$	Forward loss for class c
$\overrightarrow{\ell}_{seq,\psi}(\boldsymbol{y}_{1:T}, \boldsymbol{h}_\theta(\boldsymbol{x}_{1:t})) = \sum_{t=1}^{T} \overrightarrow{\ell}_{t,\psi}(c, \boldsymbol{h}_\theta(\boldsymbol{x}_{1:t}))$	Sequence forward loss

Table 5: Quantities and definitions used for the following proof

Proof of Proposition 1

Proof. Our goal is to show:

$$\operatorname*{argmin}_{\theta} \mathbb{E}_{\tilde{\mathbf{y}}_{1:T},\mathbf{x}_{1:T}} \overrightarrow{\ell}_{seq,\psi}(\mathbf{y}_{1:T}, \boldsymbol{g}_\theta(\mathbf{x}_{1:T})) = \operatorname*{argmin}_{\theta} \sum_{t=1}^{\top} \mathbb{E}_{\mathbf{y}_{1:t},\mathbf{x}_{1:t}} \ell_{t,\phi}(\mathbf{y}_{1:T}, \boldsymbol{g}_\theta(\mathbf{x}_{1:T})).$$

First, note that:

$$\begin{aligned} \overrightarrow{\ell}_{t,\psi}(\mathrm{y}_t, \boldsymbol{h}_\theta(\mathbf{x}_{1:t})) &= \ell_t(\mathrm{y}_t, \boldsymbol{Q}_t^{\top}\boldsymbol{\psi}^{-1}(\boldsymbol{g}_\theta(\mathbf{x}_{1:t}))) && (7) \\ &= \ell_{\boldsymbol{\phi}_t,t}(\mathrm{y}_t, \boldsymbol{g}_\theta(\mathbf{x}_{1:t})), && (8) \end{aligned}$$

where $\boldsymbol{\phi}_t^{-1} = \boldsymbol{\psi}^{-1} \circ \boldsymbol{Q}_t^{\top}$. Thus, $\boldsymbol{\phi}_t : \Delta^{C-1} \to \mathbb{R}^C$ is invertible, and is thus a proper composite loss [53].

Thus, as shown in Patrini et al. [45]:

$$\operatorname*{argmin}_{\theta} \mathbb{E}_{\tilde{\mathrm{y}}_t, \mathbf{x}_{1:t}} \ell_{\boldsymbol{\phi},t}(\mathrm{y}_t, \boldsymbol{g}_\theta(\mathbf{x}_{1:t})) = \operatorname*{argmin}_{\theta} \mathbb{E}_{\tilde{\mathrm{y}}_t | \mathbf{x}_{1:t}} \ell_{\boldsymbol{\phi}_t,t}(\mathrm{y}_t, \boldsymbol{g}_\theta(\mathbf{x}_{1:t})) \tag{9}$$

$$= \boldsymbol{\phi}_t(\boldsymbol{p}(\tilde{\mathrm{y}}_t \mid \mathbf{x}_{1:t})) \qquad \text{(property of proper composite losses)}$$

$$= \boldsymbol{\psi}((\boldsymbol{Q}_t^{-1})^\top \boldsymbol{p}(\tilde{\mathrm{y}}_t \mid \mathbf{x}_{1:t}))) \tag{10}$$

$$= \boldsymbol{\psi}(\boldsymbol{p}(\mathrm{y}_t \mid \mathbf{x}_{1:t})) \tag{11}$$

The above holds for the minimizer at a single time step, not the sequence as a whole. To find the minimizer of the loss over the entire sequence:

$$\operatorname*{argmin}_{\theta} \mathbb{E}_{\mathbf{x}_{1:T}, \tilde{\mathbf{y}}_{1:T}} \overrightarrow{\ell}_{seq,\psi}(\tilde{\mathbf{y}}_{1:T}, \boldsymbol{g}_\theta(\mathbf{x}_{1:T})) = \operatorname*{argmin}_{\theta} \mathbb{E}_{\tilde{\mathbf{y}}_{1:T} | \mathbf{x}_{1:T}} \overrightarrow{\ell}_{seq,\psi}(\tilde{\mathbf{y}}_{1:T}, \boldsymbol{g}_\theta(\mathbf{x}_{1:T})) \tag{12}$$

$$= \operatorname*{argmin}_{\theta} \mathbb{E}_{\tilde{\mathbf{y}}_{1:T} | \mathbf{x}_{1:T}} \sum_{t=1}^{\top} \overrightarrow{\ell}_{t,\psi}(\tilde{\mathrm{y}}_t, \boldsymbol{g}_\theta(\mathbf{x}_{1:t})) \tag{13}$$

$$= \operatorname*{argmin}_{\theta} \sum_{t=1}^{\top} \mathbb{E}_{\tilde{\mathbf{y}}_{1:T} | \mathbf{x}_{1:T}} \overrightarrow{\ell}_{t,\psi}(\tilde{\mathrm{y}}_t, \boldsymbol{g}_\theta(\mathbf{x}_{1:t})) \tag{14}$$

$$= \operatorname*{argmin}_{\theta} \sum_{t=1}^{\top} \mathbb{E}_{\tilde{\mathrm{y}}_t | \mathbf{x}_{1:t}} \overrightarrow{\ell}_{t,\psi}(\tilde{\mathrm{y}}_t, \boldsymbol{g}_\theta(\mathbf{x}_{1:t})) \tag{15}$$

$$= \operatorname*{argmin}_{\theta} \sum_{t=1}^{\top} \mathbb{E}_{\tilde{\mathrm{y}}_t | \mathbf{x}_{1:t}} \ell_{t,\phi}(\tilde{\mathrm{y}}_t, \boldsymbol{g}_\theta(\mathbf{x}_{1:t})) \tag{16}$$

As the minimizer of the sum will be the function that minimizes each element of the sum, then $\operatorname{argmin}_\theta \mathbb{E}_{\tilde{\mathbf{y}}_{1:T}, \mathbf{x}_{1:T}} \overrightarrow{\ell}_{seq,\psi}(\mathbf{y}_{1:T}, \boldsymbol{g}_\theta(\mathbf{x}_{1:T})) = \boldsymbol{\psi}(\boldsymbol{p}(\mathbf{y}_{1:T} \mid \mathbf{x}_{1:T}))$. Note that the $\operatorname{argmin}_\theta \sum_{t=1}^{\top} \mathbb{E}_{\mathbf{y}_{1:t}, \mathbf{x}_{1:t}} \ell_{t,\phi}(\mathbf{y}_{1:T}, \boldsymbol{g}_\theta(\mathbf{x}_{1:T})) = \boldsymbol{\psi}(\boldsymbol{p}(\mathbf{y}_{1:T} \mid \mathbf{x}_{1:T}))$, because the minimizer of the NLL is the data distribution. Thus, $\operatorname{argmin}_\theta \mathbb{E}_{\tilde{\mathbf{y}}_{1:T}, \mathbf{x}_{1:T}} \overrightarrow{\ell}_{seq,\psi}(\mathbf{y}_{1:T}, \boldsymbol{g}_\theta(\mathbf{x}_{1:T})) = \operatorname{argmin}_\theta \sum_{t=1}^{\top} \mathbb{E}_{\mathbf{y}_{1:t}, \mathbf{x}_{1:t}} \ell_{t,\phi}(\mathbf{y}_{1:T}, \boldsymbol{g}_\theta(\mathbf{x}_{1:T}))$.

$\square$

B CONTINUOUS LEARNING ALGORITHM

We summarize the augmented Lagrangian approach to solving the Continuous objective in Algorithm 1

Algorithm 1 Continuous Learning Algorithm

```
Input: Noisy Training Dataset $D$, hyperparameters $\gamma$ and $\eta$
Output: Model $\theta$, Temporal Noise Function $\omega$
  $c \leftarrow 1$ and $\lambda \leftarrow 1$
  for $k = 1, 2, 3, \ldots,$ do
    $\theta^k, \omega^k = \arg\min_{\theta,\omega} \mathcal{L}(\theta, \omega)$        ▷ Computed with SGD using the Adam optimizer
    $\lambda \leftarrow \lambda + c * R_t(\theta^k, \omega^k)$        ▷ Update Lagrange multiplier
    if $k > 0$ and $R_t(\theta^k, \omega^k) > \gamma R_t(\theta^{k-1}, \omega^{k-1})$ then
      $c \leftarrow \eta c$
    else
      $c \leftarrow c$
    end if
    if $R_t(\theta^k, \omega^k) == 0$ then
      break
    end if
  end for
```

For all experiments we set $\lambda = 1$,$c = 1$, $\gamma = 2$, and $\eta = 2$. k and the maximum number of SGD iterations are set to 15 and 10, respectively. This is to ensure that the total number of epochs is 150, which is the max number of epochs used for all experiments.

C PLUG-IN PROCEDURE

1. Fit a probabilistic classifier to predict noisy labels from the observed data.
2. For each class $y \in \mathcal{Y}$ and time $t \in [1 \ldots T]$:
    i Identify anchor points for class y: $\bar{\boldsymbol{x}}_t^j = \arg\max_{\boldsymbol{x}_t} p(\tilde{y}_t = y \mid \boldsymbol{x}_{1:t})$.
    ii Set $\hat{\boldsymbol{Q}}(t)_{y,y'}$ as the probability of classifier predicting class y' at time t given $\bar{\boldsymbol{x}}_t^j$.

D NOISE FUNCTION INDEPENDENT APPROACHES

We also include static methods that do *not* rely on the noise transition matrix (e.g., DivideMix [31]) in Appendix E, but find these methods are difficult to get working in the time-series setting and underperform (i.e., hyper-parameter tuning, complexity issues).

E EXPERIMENTAL DETAILS

Our code is available in an anonymized repository.

Dataset	**Classification Task**	n	d	T
`eeg_eye` [55]	Eye Open vs Eye Closed	299	14	50
`eeg_sleep` [20]	Sleep vs Awake	964	7	100
`har` [54]	Walking vs Not Walking	192	9	50
`har70` [39]	Walking vs Not Walking	444	6	100
`synth`	$\mathcal{N}$(0,1.5) vs $\mathcal{N}$(1,1.5)	1,000	50	100

Table 6: Datasets used in the experiments. Classification tasks, number of samples (n), dimensionality at each time step (d), and sequence length (T) are shown.

E.1 DATASET DETAILS

Synthetic We generate data for binary and multiclass classification with $n = 1\,000$ samples and $d = 50$ features over $T = 100$ time steps. We generate the class labels and obvservations for each time step using a Hidden Markov Model (HMM). The transition matrix generating the markov chain is uniform ensuring an equal likelihood of any state at any given time. We corrupted them using multidimensional (50) Gaussian emissions. The mean of the gaussian for state/class c is set to c with variance 1.5 (i.e. class 1 has mean 1 and variance 1.5). The high-dimensionality and overlap in feature-space between classes makes this a sufficiently difficult task, especially under label noise. We use a batchsize of 256

HAR from UC Irvine [54] consists of inertial sensor readings of 30 adult subjects performing activities of daily living. The sensor signals are already preprocessed and a vector of features at each time step are provided. We apply z-score normalization at the participant-level, then split the dataset into subsequences of a fixed size 50. We use a batchsize of 64.

HAR70 from UC Irvine [39] consists of inertial sensor readings of 18 elderly subjects performing activities of daily living. The sensor signals are already preprocessed and a vector of features at each time step are provided. We apply z-score normalization at the participant-level, then split the dataset into subsequences of a fixed size 100. We use a batchsize of 256.

EEG SLEEP from Physionet [20] consists of EEG data measured from 197 different whole nights of sleep observation, including awake periods at the start, end, and intermittently. We apply z-score normalization at the whole night-level. Then downsample the data to have features and labels each minute, as EEG data is sampled at 100Hz and labels are sampled at 1Hz. We then split the data into subsequences of a fixed size 100. We use a batchsize of 512.

EEG EYE from UC Irvine [55] consists of data measured from one continuous participant tasked with opening and closing their eyes while wearing a headset to measure their EEG data . We apply z-score normalization for the entire sequence, remove outliers (>5 SD away from mean), and split into subsequences of a fixed size 50. We use a batchsize of 128.

Train-Test Splitting Splitting strategies depended on the dataset. For example, in the real-world stress detection demonstration, the training and testing splits did not share individuals, as there is one time series per individual. For all other datasets, such as ‘moving’ and ‘senior’, we used the given train-test splits in the dataset.

E.2 SPECIFIC IMPLEMENTATION DETAILS

GRU the GRU $\boldsymbol{r} : \mathbb{R}^d \times \mathbb{Z} \to \mathbb{R}^C \times \mathbb{Z}$ produces an *output vector* such that the output of $\boldsymbol{r}(\boldsymbol{x}_t, \boldsymbol{z}_{t-1})$ is our model for $\boldsymbol{h}_\theta(\boldsymbol{x}_{1:t})$, and a *hidden state* $\boldsymbol{z}_t \in \mathbb{Z}$ that summarizes $\boldsymbol{x}_{1:t}$. We use a softmax activation on the output vector of the GRU to make it a valid parameterization of $\boldsymbol{p}_\theta(y_t \mid \boldsymbol{x}_{1:t})$. The GRU has a single hidden layer with a 32 dimension hidden state.

Continuous Continuous uses an additional fully-connected neural network with 10 hidden layers that outputs a $C * C$-dimensional vector to represent each entry of a flattened $\hat{\boldsymbol{Q}}_t$. To ensure the output of this network is valid for Def. 1, we reshape the prediction to be $C \times C$, apply a row-wise softmax function, add this to the identity matrix to ensure diagonal dominance, then rescale the rows to be row-stochastic. These operations are all differentiable, ensuring we can optimize this network with standard backpropagation.

VolMinNet and Independent We do a similar parameterization for VolMinNet and Independent, using a set of differentiable weights to represent the entries of $\boldsymbol{Q}_t$ rather than a neural network.

Anchor and Plug-In Patrini et al. [45] show that in practice taking the 97th percentile anchor points rather than the maximum yield better results, so we use that same approach in our experiments. They also describe a two-stage approach: 1) estimate the anchor points after a warmup period 2) use the anchor points to train the classifier with forward corrected loss. We set the warmup period to 25 epochs.

E.3 EXPERIMENTAL PARAMETERS

Given that the learning algorithm only has access to a noisy training dataset and performance is evaluated on a clean test set, a validation set must be drawn from clean test data or by manually cleaning the noisy training dataset which may be impractical. This makes hyperparameter tuning difficult in noisy label learning. As the optimal set of hyperparameters within each could vary for each method, noise type, amount of noise, and dataset, this represents a difficult task. To be fair for our experimental evaluations, we use the same set of hyperparameters for experiment, and only manually set batch size for each dataset.

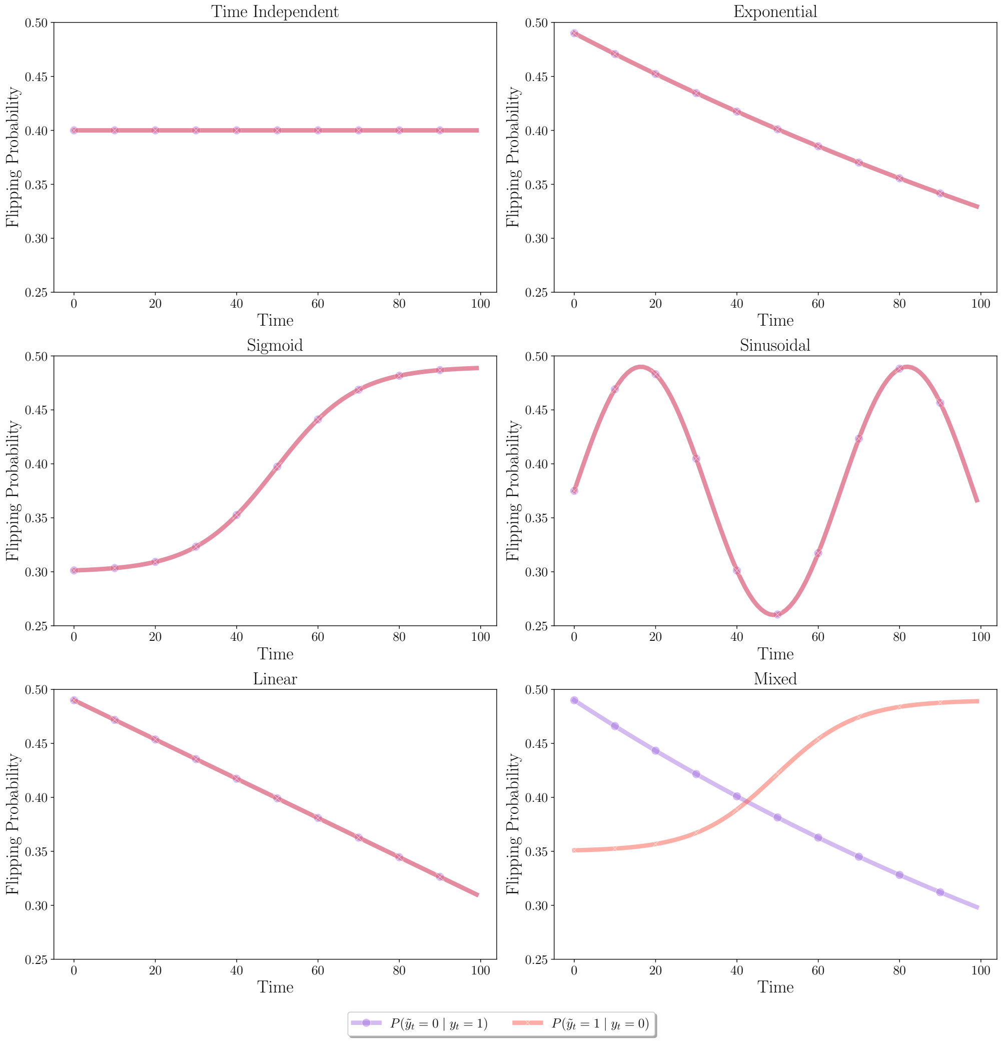
Each model was trained for 150 epochs using the adam optimizer with default parameters and a learning rate of 0.01.

For VolMinNet, Independent, and Continuous we use adam optimizer with default parameters and a learning rate of 0.01 to optimize each respective $\hat{\boldsymbol{Q}}_t$-estimation technique. λ was set to $1e-4$ for VolMinNet and Independent for all experiments, based on what was published previously [34] .

E.4 NOISE INJECTION

To the best of our knowledge there are no noisy label time series datasets (i.e.: standardized datasets with both clean and noisy labels) to evaluate our methods. In line with prior experimental approaches, we propose a noise injection strategy which assumes some temporal noise function that can give us a noisy distribution to evaluate from. We deliberately pick a wide variety of noise types, varying the amount and functional form of time-dependent noise, including static noise setting (uniform noise at every time, akin to what baseline methods assume), and class-dependent noise structure Fig. 4.

Figure 4: Temporal functions that can be specified using a temporal label noise function $\boldsymbol{Q}(t)$. We present six examples for binary classification task (from top-left clockwise): time independent, exponential decay sinusoidal noise, mixed class-dependent noise, linear decay noise, sigmoid increasing noise. Each plot shows the off-diagonal entries of various parameterized forms of $\boldsymbol{Q}(t)$.

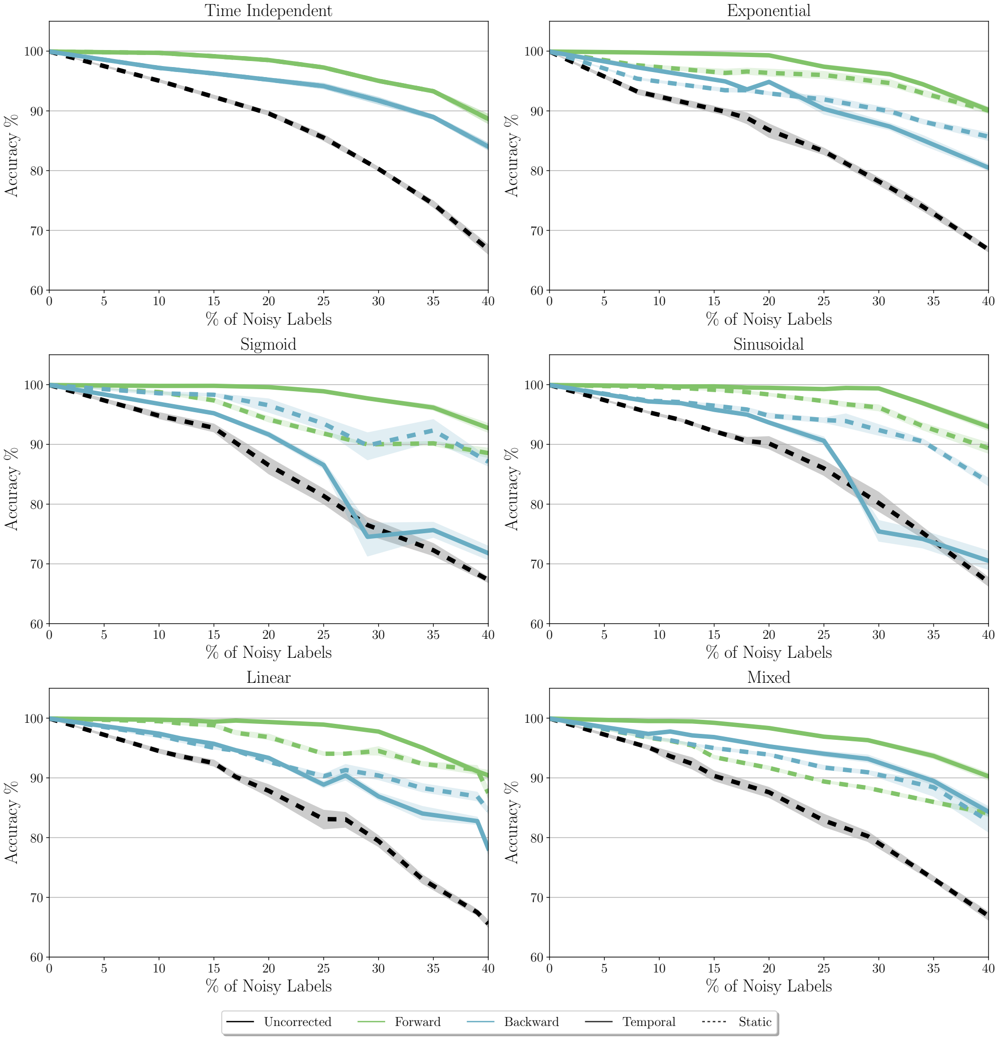


F COMPLETE RESULTS

F.1 STATIC APPROXIMATION

Here we study the performance of *forward temporal loss* where we know the noise function $\boldsymbol{Q}(t)$ – that is, even if we could *perfectly* estimate the noise process – and where we have a static estimate of $\boldsymbol{Q}(t)$ (the average over time). We find that even if the noise process is perfectly estimated, accounting for temporal noise outperforms a static estimate.

Figure 5: Comparing performance of models trained with *backward temporal loss* and *forward temporal loss* on `synth` with varying degrees of temporal label noise using either the true temporal noise function (Temporal) or the average temporal noise function (Static). Error bars are st. dev. over 10 runs.



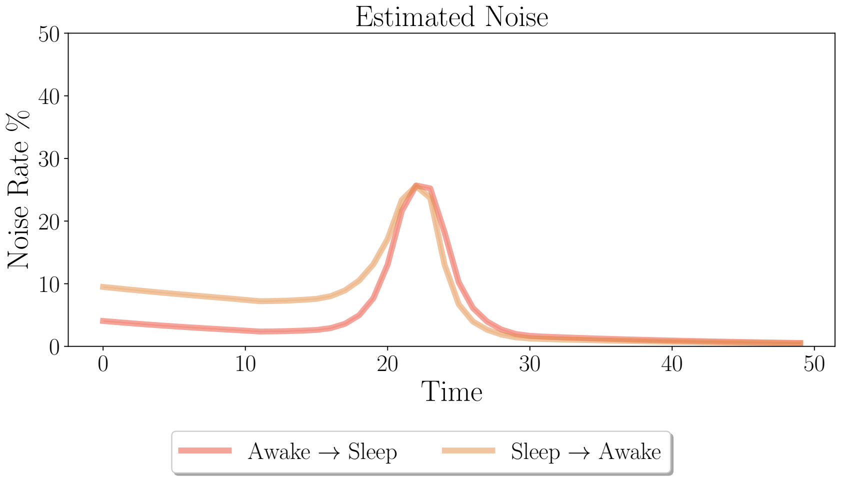
F.2 REAL TEMPORAL LABEL NOISE

Prior work in the static noisy label literature typically aim to demonstrate the effectiveness of their methods on a real world noisy dataset, where the noise function is not imposed by the researcher. The primary dataset used is the `Clothing1M` dataset [76]. Despite containing real label noise, Clothing1M is inapplicable in our setting: it is not temporal data and each instance has only one label. In the spirit of evaluating Continuous on real-world noisy labels, we discovered and experimented with `extrasensory`, a noisy-labelled time series dataset [67]. `extrasensory` includes human activity data from smartphones and smartwatches collected from 60 users spread across 300,000 minutes of measurements. In contrast to `har` and `har70` (datasets originally used in our paper), `extrasensory` has no expert-labelled annotations, all the labels are user-provided and therefore are highly noisy. Users often misreport falling asleep and waking up, so we expect particularly high label noise during sleep/awake transitions

In order to identify the label noise in this temporal data, we partition and center the dataset from all users around sleep/awake transition periods. That is, for a fixed length window of 50, sleep/awake transitions occur around the $t = 25$ point. We then train our Continuous objective with the same model architecture and hyperparameters as above to classify sleep and awake over time. Since there are no 'clean' labels, we demonstrate that Continuous successfully identifies an interpretable temporal noise function.

In Fig. 6, we see Continuous predicts there exists higher label noise near sleep/awake transitions (around $t = 25$). We hope our work also encourages the community to seek further sources of real temporal noise.

Figure 6: Continuous-estimated $\hat{\boldsymbol{Q}}_t$ for `extrasensory`. Error bars are st. dev. over 10 runs.



F.3 CLASS DEPENDENT AND CLASS INDEPENDENT

Table 7: Comparison of clean test-set Accuracy (%) and MAE of all methods on Class Indepenent and Class Dependent sinusoidal label noise for a fixed degree of label noise (30%) on `har`. Dashed line separates *Static* and *Temporal* methods.

		Class Independent		Class Dependent	
		Accuracy ↑	MAE ↓	Accuracy ↑	MAE ↓
Static	Uncorrected	76.0±5.1	–	76.4±3.1	–
	Anchor	82.0±3.6	0.15±0.014	84.2±2.2	0.13±0.012
	VolMinNet	86.5±6.0	0.13±0.009	92.6±1.9	0.12±0.012
Temporal	Plug-In	81.5±4.3	0.14±0.013	84.1±2.3	0.13±0.010
	Discontinuous	86.0±5.7	0.10±0.015	91.5±2.1	0.08±0.004
	Continuous	98.3±0.6	0.03±0.005	98.4±0.7	0.02±0.005

F.4 MULTICLASS CLASSIFICATION

Figure 7: Comparison of clean test set Accuracy (%) for `synth` across varying degrees of temporal label noise comparing all methods for 3-class classification. Error bars are st. dev. over 10 runs.

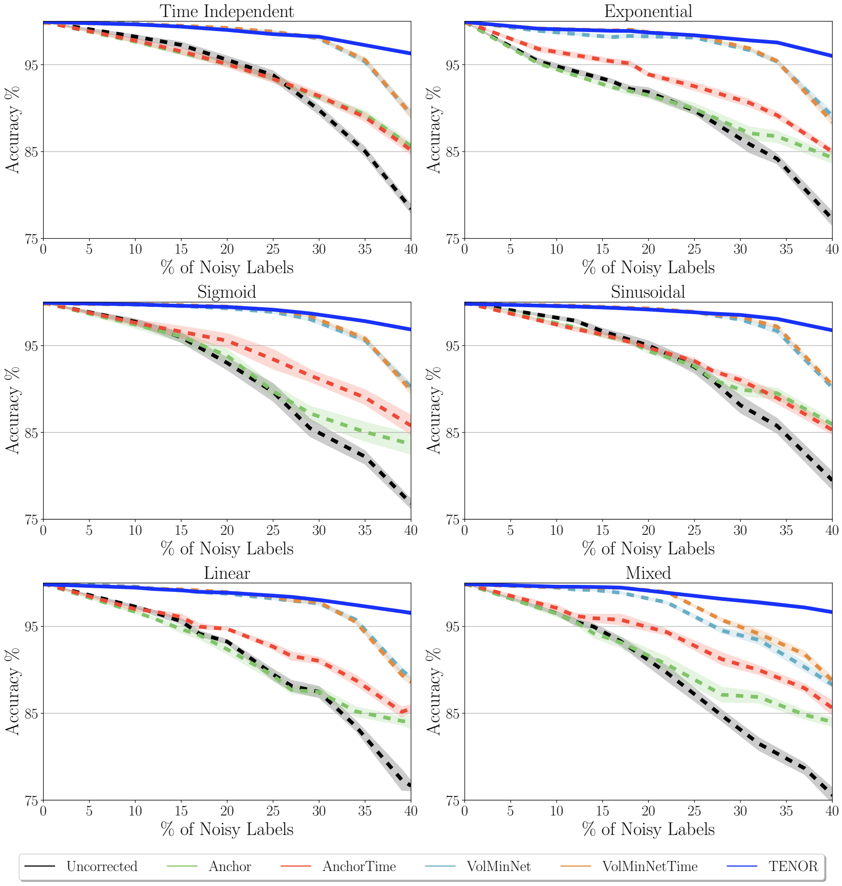


Figure 8: Comparison of noisy function reconstruction Mean Absolute Error (MAE) for `synth` across varying degrees of temporal label noise comparing all methods for 3-class classification. Error bars are st. dev. over 10 runs.

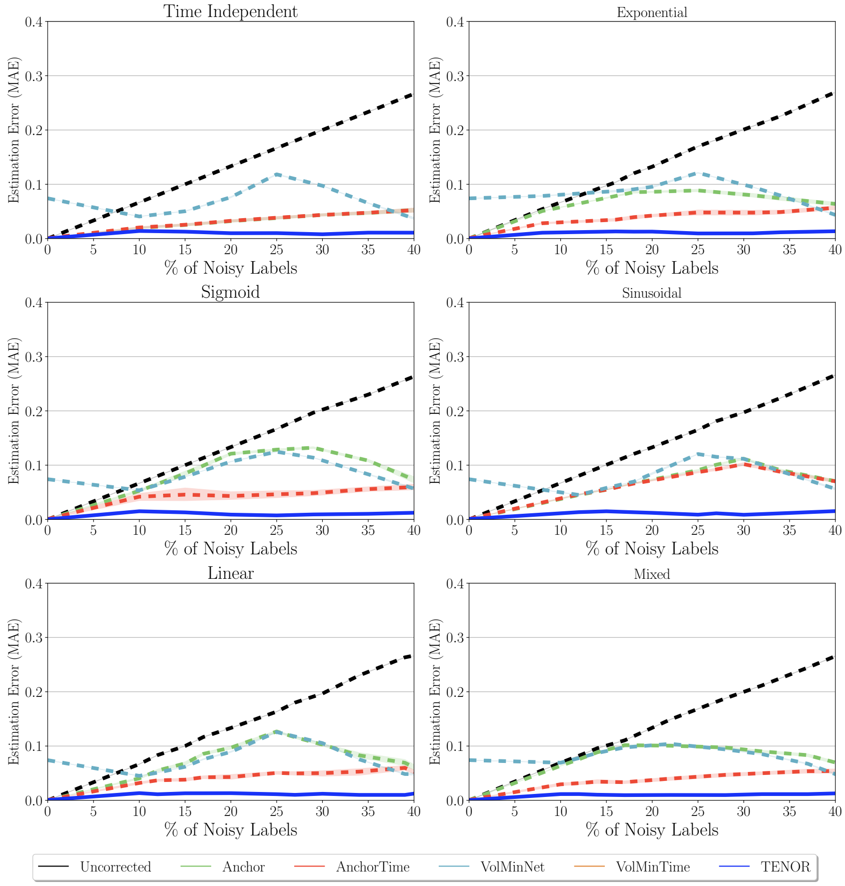


Figure 9: Comparison of clean test set Accuracy (%) for `eeg_sleep` across varying degrees of temporal label noise comparing all methods for 3-class classification. Error bars are st. dev. over 10 runs.

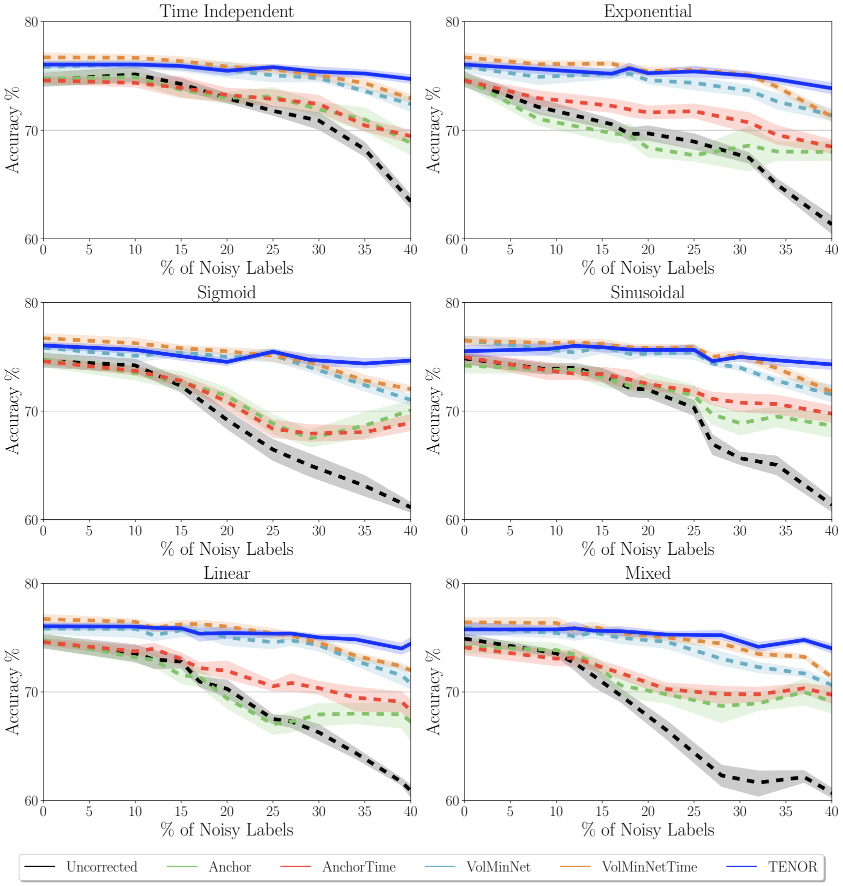


Figure 10: Comparison of noisy function reconstruction Mean Absolute Error (MAE) for `eeg_sleep` across varying degrees of temporal label noise comparing all methods for 3-class classification. Error bars are st. dev. over 10 runs.

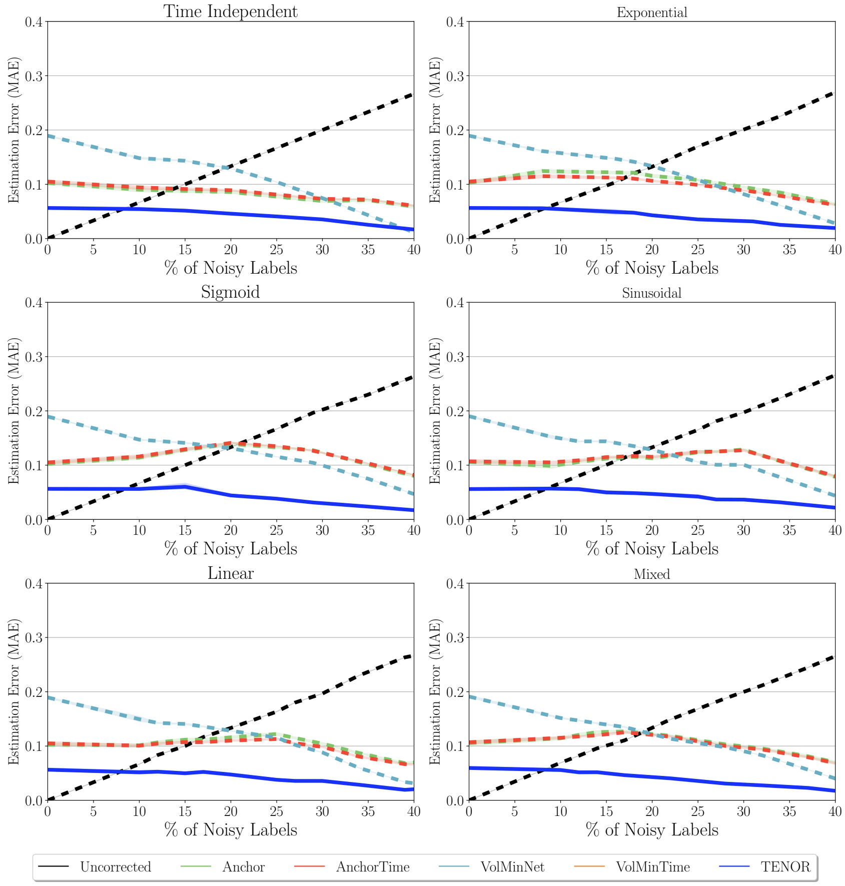


Figure 11: Comparison of clean test set Accuracy (%) for `har` across varying degrees of temporal label noise comparing all methods for 4-class classification. Error bars are st. dev. over 10 runs.

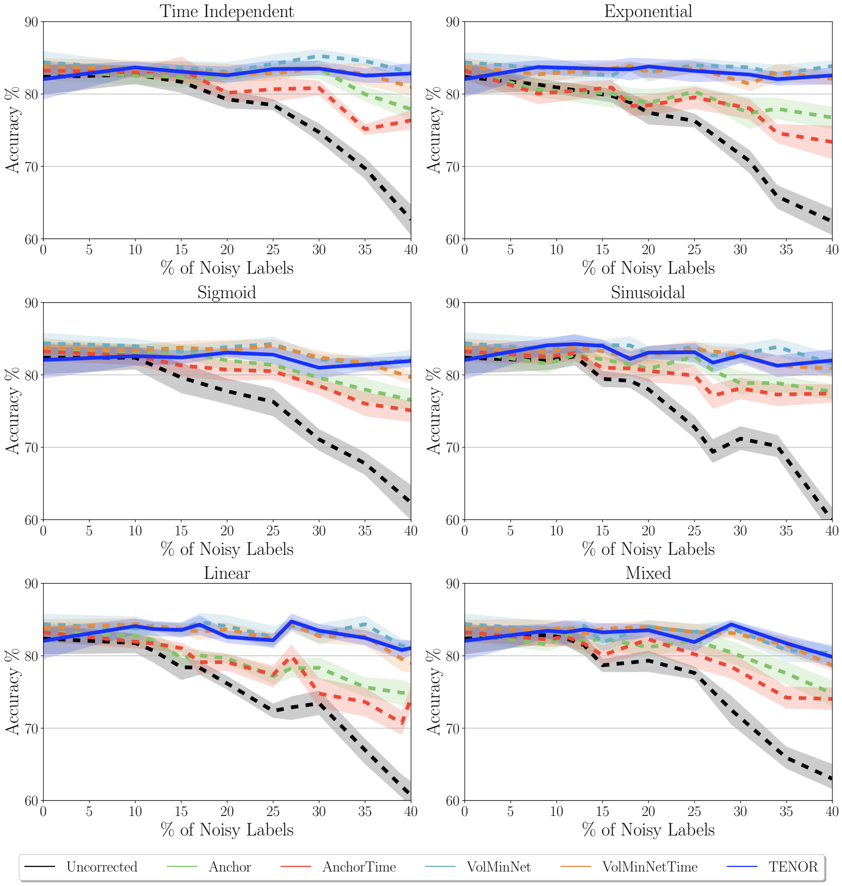
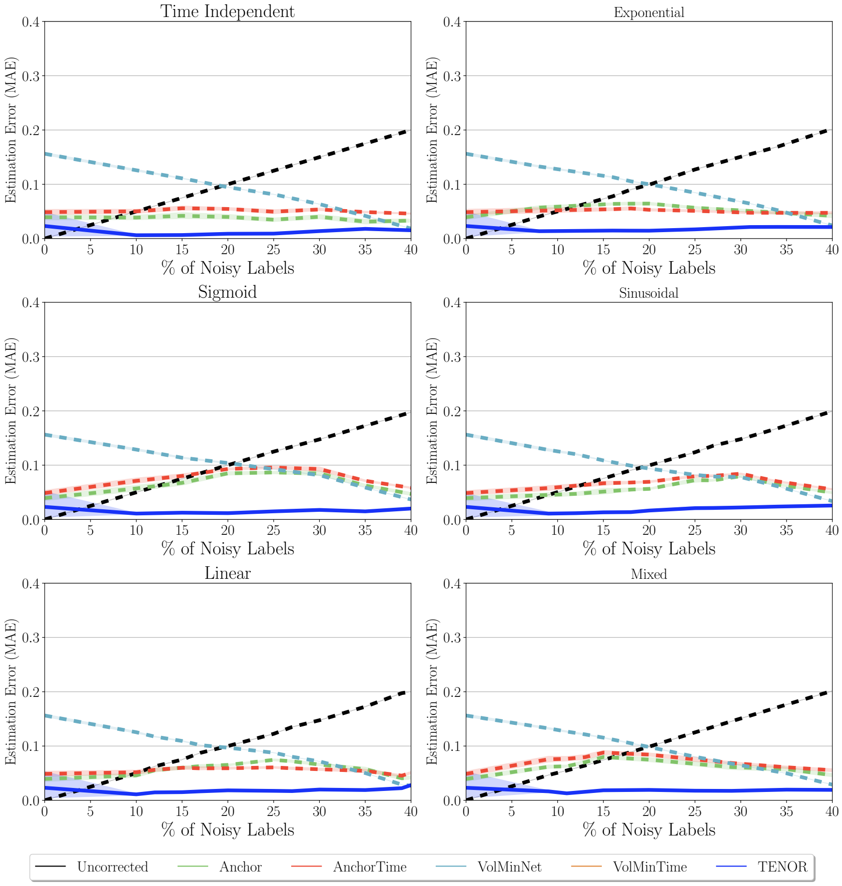


Figure 12: Comparison of noisy function reconstruction Mean Absolute Error (MAE) for `har` across varying degrees of temporal label noise comparing all methods for 4-class classification. Error bars are st. dev. over 10 runs.


Time Independent
Exponential
Sigmoid
Sinusoidal
Linear
Mixed
Estimation Error (MAE)
% of Noisy Labels
Uncorrected
Anchor
AnchorTime
VolMinNet
VolMinTime
TENOR