Leveraging an Alignment Set in Tackling Instance-Dependent Label Noise

Donna Tjandra  
Jenna Wiens  
Computer Science & Engineering, University of Michigan, Ann Arbor, MI, USA

Abstract
Noisy training labels can hurt model performance. Most approaches that aim to address label noise assume label noise is independent from the input features. In practice, however, label noise is often feature or instance-dependent, and therefore biased (i.e., some instances are more likely to be mislabeled than others). E.g., in clinical care, female patients are more likely to be under-diagnosed for cardiovascular disease compared to male patients. Approaches that ignore this dependence can produce models with poor discriminative performance, and in many healthcare settings, can exacerbate issues around health disparities. In light of these limitations, we propose a two-stage approach to learn in the presence instance-dependent label noise. Our approach utilizes alignment points, a small subset of data for which we know the observed and ground truth labels. On several tasks, our approach leads to consistent improvements over the state-of-the-art in discriminative performance (AUROC) while mitigating bias (area under the equalized odds curve, AUEOC). For example, when predicting acute respiratory failure onset on the MIMIC-III dataset, our approach achieves a harmonic mean (AUROC and AUEOC) of 0.84 (SD [standard deviation] 0.01) while that of the next best baseline is 0.81 (SD 0.01). Overall, our approach improves accuracy while mitigating potential bias compared to existing approaches in the presence of instance-dependent label noise.

Data and Code Availability  This paper uses the MIMIC-III dataset (Johnson et al., 2016b), which is available on the PhysioNet repository (Johnson et al., 2016a). We also use two public datasets outside of the healthcare domain: 1) the Adult dataset¹, and 2) the COMPAS dataset². A link to the source code is provided in the footnote³.

Institutional Review Board (IRB)  This work is not regulated as human subjects research since data are de-identified.

1. Introduction
Motivation and Problem Setting  Datasets used to train machine learning models can contain incorrect labels (i.e., label noise), which can lead to overfitting. While label noise is widely studied, the majority of past work focuses on instance-independent label noise (i.e., when the noise is independent from an instance’s features) (Song et al., 2022). However, label noise can depend on instance features (Wei et al., 2022b; Chang et al., 2022), leading to different noise rates within subsets of the data. Furthermore, in settings where the noise rates differ with respect to a sensitive attribute, this can lead to harmful disparities in model performance (Liu, 2021). For example, consider the task of predicting cardiovascular disease among patients admitted to a hospital. Compared to male patients, female patients may be more likely to be under-diagnosed (Maserejian et al., 2009) and thus mislabeled, potentially leading to worse predictions for female patients. Although instance-dependent label noise has recently received more attention (Cheng et al., 2020b; Xia et al., 2020; Wang et al., 2021a), the effect of these approaches on model bias has been relatively understudied (Liu, 2021). Here, we address current limitations and propose a novel method for learning with instance-dependent label noise in a setting inspired by healthcare, specifically examining how modeling assumptions affect existing issues around potential model bias.

¹. https://github.com/AissatouPaye/Fairness-in-Classification-and-Representation-Learning
². https://www.kaggle.com/danofer/compass
Gaps in Existing Work  Broadly, current work addressing instance-dependent label noise takes one of two approaches: 1) learn to identify mislabeled instances (Cheng et al., 2020a; Xia et al., 2022; Zhu et al., 2022a), or 2) learn to optimize a noise-robust objective function (Feng et al., 2020; Wei et al., 2022a). In the first category, instances identified as mislabeled are either filtered out (Kim et al., 2021) or relabeled (Berthon et al., 2021). In some settings, this approach can have a negative effect on model bias. Revisiting our example on cardiovascular disease, approaches that filter out mislabeled individuals could ignore more female patients, since they have a potentially higher noise rate. While relabeling approaches use all available data, they can be sensitive to assumptions around the noise distribution (Ladouceur et al., 2007). In the second category, current approaches rely on objective functions that are less prone to overfitting to the noise and use all of the data and observed labels (Chen et al., 2021). However, past work has empirically shown that these improve discriminative performance the most when used to augment filtering approaches, and thus, the limitations and scenarios described above still potentially hold.

Our Idea  In light of these limitations, we propose an approach that addresses instance-dependent label noise, makes no assumptions about the noise distribution, and uses all data during training. We focus on a setting that frequently arises in healthcare, where we are given observed labels for a condition of interest (e.g., cardiovascular disease) and have a clinical expert who can evaluate whether the observed labels are correct for a small subset of the data (e.g., by manual chart review). Using this subset, which we refer to as the ‘alignment’ set, we learn the underlying pattern of label noise in a pre-training step. We then minimize a weighted cross-entropy over all the data. Note that our alignment set is a special case of anchor points (Liu and Tao, 2015), with the added requirement that the set contains instances for which the ground truth and observed labels do and do not match.

On synthetic and real data, we demonstrate that our approach improves on state-of-the-art baselines from the noisy labels and fairness literature, such as stochastic label noise (Chen et al., 2021) and group-based peer loss (Wang et al., 2021b). Overall, our contributions include:

- A novel approach to learn from datasets with instance-dependent noise that highlights a setting frequently found in healthcare
- A systematic examination of different settings of label noise, evaluating discriminative performance and bias mitigation
- Empirical results showing that the proposed approach is robust to both to the noise rate and amount of noise disparity between subgroups, reporting the model’s ability to maintain discriminative performance and mitigate potential bias
- A demonstration of how performance of the proposed approach changes when assumptions about the alignment set are violated

2. Methods
We introduce a two-stage approach for learning with instance-dependent label noise that leverages a small set of alignment points for which we have both observed and ground truth labels.

Table 1: Notation. A summary of notation used throughout. Superscripts in parentheses specify instances (e.g., \(x^{(i)}\)). Subscripts specify indexes into a vector (e.g., \(x_i\))

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(d)</td>
<td>number of features</td>
</tr>
<tr>
<td>(g)</td>
<td>number of groups</td>
</tr>
<tr>
<td>(x \in \mathbb{R}^d)</td>
<td>feature vector</td>
</tr>
<tr>
<td>(y \in [0,1])</td>
<td>predicted class probabilities</td>
</tr>
<tr>
<td>(\hat{y} \in {-1,1})</td>
<td>observed label</td>
</tr>
<tr>
<td>(y \in {-1,1})</td>
<td>ground truth label</td>
</tr>
<tr>
<td>(\beta = P(\hat{y} = y</td>
<td>\hat{y}, x; \phi))</td>
</tr>
<tr>
<td>(A)</td>
<td>alignment set, has (a) instances</td>
</tr>
<tr>
<td>(\overline{A})</td>
<td>non-alignment set, (\pi) instances</td>
</tr>
<tr>
<td>(\theta)</td>
<td>main model parameters</td>
</tr>
<tr>
<td>(\phi)</td>
<td>auxiliary model parameters</td>
</tr>
</tbody>
</table>

Notation and Setting  Our notation is summarized in Table 1, with additional notation defined throughout as needed. Our dataset, \(D = A \cup \overline{A}\) consists of instances in \(A = \{x^{(j)}, \hat{y}^{(j)}, y^{(j)}\}_{j=1}^a\) and \(\overline{A} = \{x^{(i)}, \hat{y}^{(i)}\}_{i=1}^\pi\). \(A\) is the set of alignment points (i.e., the alignment set), where both \(\hat{y}^{(j)}\) and \(y^{(j)}\) are known, and we assume that it includes instances
where $\hat{y}^{(i)} \neq y^{(i)}$. Alignment points are a special case of anchor points (Liu and Tao, 2015), where points that do and do not have matching observed and ground truth labels are both required. $\mathcal{A}$ is the non-alignment set and contains instances for which we do not know the ground truth labels. In the presence of noisy labels, we assume that whether $\hat{y} = y$ is dependent on $x$ (i.e., $P(\hat{y} = y) \neq P(\hat{y} = y|x$)). Given this dataset, we aim to train a model to learn $f : \mathbb{R}^d \rightarrow [0,1]$ (i.e. the function used to predict the ground truth labels), so that we can map unseen instances into one of two classes based on their feature vectors. Our learned model parameters, $\theta$, are such that the output of the corresponding model represents the predicted class probabilities, (i.e., $\hat{y}$). Although we focus on binary classification, our setup can be applied to multiclass classification.

**Justification and Desired Properties** Our setting is inspired by the use of pragmatic labeling tools in healthcare. Such tools are often based on various components of the electronic health record (EHR), and they are applied to identify cohorts or outcomes under noise rates between subgroups can also vary in practice (Kirby et al., 2016). Similar to how different noise rates can arise in practice, differences in noise rates between subgroups can also vary in practice (Kostopoulos et al., 2008). As a result, achieving these properties can potentially make our approach generalize to a wide variety of settings.

**Proposed Network** Our proposed network (Figure 1(a)) consists of two components. The first, parameterized by $\theta$, is a feed-forward network that uses feature vector $x$ to predict the class probability, $\hat{y} = P(y = 1|x; \theta)$. The second component, parameterized by $\phi$, is an auxiliary feed-forward network that uses observed label $\hat{y}$ and features $x$ to compute $\hat{\beta} = P(y = \hat{y}|\hat{y}, x; \phi)$, an instance-dependent prediction for whether the observed label is correct based on $x$ and $\hat{y}$. $\hat{\beta}$ can be considered as a confidence score for the observed label, with higher values indicating higher confidence. Learning $\hat{\beta}$ models the underlying pattern of label noise by forcing the model to learn which instances are correctly labeled. We use $\hat{\beta}$ to reweight the objective function during the second step of training, as described below. By including the observed label as input to $\phi$, our approach also applies to instance-independent label noise because it accounts for the case when the underlying pattern of label noise does not depend on the features. In order to learn $\hat{\beta}$, we assume that the label noise pattern can be represented as some function, though the specific form of this function (e.g., linear) does not need to be known. During training, we compute the loss using the outputs from both networks. At inference time (i.e., in practical use after training), we compute the class predictions from the network parameterized by $\theta$ only since $\hat{y}$ is unavailable.

**Training Procedure.** Our training procedure is summarized in Figure 1(b) and Appendix A. In Step 1, we pre-train both networks using the alignment points, $A$, minimizing an objective function based on cross entropy: $\theta', \phi' = \arg\min_{\theta, \phi} \mathcal{L}_\theta + \alpha_1 \mathcal{L}_\phi$. $\alpha_1 \in \mathbb{R}^+$ is a scalar hyperparameter; $\theta'$ and $\phi'$ are parameters that represent the initial values of $\theta$ and $\phi$. $\mathcal{L}_\theta$ is the cross-entropy loss between the class predictions and ground truth labels. It aids in learning the parameter values for $\theta$, and thus, the model’s decision boundary. $\mathbb{I}$ is an indicator function.

$$\mathcal{L}_\theta = \frac{-1}{|A|} \sum_{j \in A} \mathbb{I}(y^{(j)} = 1) \log(\hat{y}^{(j)}) + \mathbb{I}(\hat{y}^{(j)} = 1) \log(1 - \hat{y}^{(j)})$$

$\mathcal{L}_\phi$ is the cross-entropy loss between the predicted confidence score $\hat{\beta}^{(j)}$ and the actual agreement between $\hat{y}^{(j)}$ and $y^{(j)}$. It aids in learning the weights...
Our approach aims to mitigate bias by up-weighting advantage of the model’s learned noise pattern. Inference rate, $\hat{r}_k$ is associated with estimated noise rate $k$ is as follows. We introduce sets $G_k$ for $k = 1, 2, ..., g$ to represent disjoint subgroups of interest in the data, which are assumed to be known in advance. $G_a \cap G_b = \emptyset$ for all $a = 1, 2, ..., g$, $b = 1, 2, ..., g$ with $a \neq b$ and $\cup_{k=1}^g G_k = D$. Each group $G_k$ is then associated with estimated noise rate $\hat{r}_k = \frac{1}{|G_k|} \sum_{i \in G_k} 1 - \hat{\beta}_i(j)$. Although weighting each instance by $\hat{\beta}$ is a form of soft filtering, weighting each group by the inverse of its overall ‘clean’ rate avoids the effect of de-emphasizing groups with higher predicted noise rates. As a result, the expected value of $L'_\theta$ with respect to $\hat{\beta}$ is equal to the cross-entropy loss between the model’s predictions and ground truth labels (see Appendix A for proof). However, this assumes accurate estimates of $\hat{\beta}$. Thus, we expect that the proposed approach will perform best when the alignment set is representative of the target population. In scenarios where the alignment set is biased (e.g., some groups are underrepresented), if the learned noise function does not transfer to the underrepresented group, then the proposed approach may not be beneficial. In Section 4, we test this.

Figure 1: Our Approach. a) The model predicts, $\hat{y}$, at training and inference time using $\theta$. At training time, it also predicts whether the observed label is correct using $\phi$, $\theta$ and $\phi$ are pre-trained using $A$ and then fine tuned with the complete dataset. b) We pretrain the model using the alignment points, then train on the noisy data. $L_\theta$, $L_\phi$, and $L'_\theta$ are the objectives for the alignment points, label confidence score ($\hat{\beta}_\phi$), and noisy data, respectively. $\alpha_1$, $\alpha_2$, $\gamma$ are scalar hyperparameters.

for $\phi$, and thus, the underlying label noise pattern.

$$L_\phi = -\frac{1}{|A|} \sum_{j \in A} \mathbb{I}(\hat{y}^{(j)} = y^{(j)}) \log(\hat{\beta}^{(j)}) + \mathbb{I}(\hat{y}^{(j)} \neq y^{(j)}) \log(1 - \hat{\beta}^{(j)})$$

In Step 2, we initialize $\theta$ and $\phi$ as $\theta'$ and $\phi'$ and fine tune using the complete dataset. Step 2 consists of two parts, Step 2a and Step 2b. Each part aims to improve a specific component of the network (e.g., $\theta$) using another component of the network (e.g., $\phi$). We begin with Step 2a, move to Step 2b, and continue to alternate between Step 2a and Step 2b in a manner similar to expectation maximization so that we continually improve both $\theta$ and $\phi$. In Step 2a, we freeze $\phi$ and find that $\theta$ that minimizes the objective $L'_\theta + \gamma L_\theta$. $\gamma \in \mathbb{R}^+$ is a scalar hyperparameter. In Step 2b, we freeze $\theta$ and find $\phi$ that minimizes the objective $L'_\phi + \alpha_2 L_\phi$. $\alpha_2 \in \mathbb{R}^+$ is a scalar hyperparameter. $L'_\phi$ computes the cross-entropy loss over the potentially noisy, non-alignment points. Each instance is weighted by the model’s confidence in whether the observed label is correct via $\hat{\beta}(i)$, taking advantage of the model’s learned noise pattern. Our approach aims to mitigate bias by up-weighting groups, $k = 1, 2, ..., g$ with a higher estimated noise rate, $\hat{r}_k$, so that they are not dominated by/ignored compared to groups with a lower estimated noise rate.

$$L'_\theta = \frac{-1}{|A|} \sum_{i=1}^g \frac{1}{1 - \hat{r}_k} \sum_{i \in X \cap G_k} \sum_{j=1}^c \hat{\beta}_\theta(i) \mathbb{I}(\hat{y}(i) = j) \log(\hat{y}_j^{(i)})$$

We calculate $1 - \hat{r}_k$ as follows. For each instance $i$ in group $G_k$, we calculate the probability of the correct label being predicted. Then, we take the average over all instances in $G_k$. This is $\hat{r}_k = \frac{1}{|G_k|} \sum_{i \in G_k} 1 - \hat{\beta}_i(i)$.
During Step 2a, \( L'_\phi \) is used to train \( \theta \) by learning to predict \( \hat{y} \) such that it matches observed label \( \hat{y} \) on instances that are predicted to be correctly labeled. During Step 2b, \( L'_\phi \) is used to train \( \phi \). Here, since \( \theta \) is frozen and \( \phi \) is not, the network learns to predict the optimal \( \hat{\beta} \). Based on \( L'_\phi \) alone, there are two possible options to learn \( \hat{\beta} \): 1) consistently make \( \hat{\beta} \) close to 0, and 2) predict \( \hat{\beta} \) such that it is close to 1 when \( \hat{y} \) matches \( \hat{y} \) and close to 0 when \( \hat{y} \) does not match \( \hat{y} \). Since \( \hat{y} \) is used as a proxy for \( y \) in this step, the second option aligns with what we want \( \hat{\beta} \) to represent. To encourage this over the first option (i.e., consistently predicting 0 for \( \hat{\beta} \)), we include \( L_\phi \) in Step 2b, which is not minimized by consistently predicting 0 for \( \hat{\beta} \). Note that, in Step 2b, we rely on the cluster assumption (Singh et al., 2008) from semi-supervised learning, which broadly states that labeled data fall into clusters and that unlabeled data aid in defining these clusters. In the context of Step 2b, ‘labeled’ and ‘unlabeled’ are analogous to whether we know if the ground truth and observed labels match (i.e., alignment point versus non-alignment point), rather than the actual class labels themselves. As a result, we also rely on the alignment set being representative of the target population here to avoid dataset shift.

In contrast to previous filtering approaches, our approach utilizes all data during training. Moreover, it does not require a specialized architecture beyond the auxiliary network to compute \( \hat{\beta} \). Thus, it can be used to augment existing architectures.

3. Experimental Setup

We empirically explore the performance of our proposed approach relative to state-of-the-art baselines on five benchmark prediction tasks with two different label noise settings. For reproducibility, full implementation details are provided in Appendices B and C. We aim to test 1) the extent to which our desired properties hold, 2) the extent to which the proposed approach is robust to changes in the composition of the alignment set, and 3) which components of the proposed approach contribute the most.

Datasets We consider five different binary prediction tasks on four datasets from several domains with synthetic and real datasets. Though inspired by healthcare, we also consider domains outside of healthcare to show the broader applicability of our approach in areas where harmful biases can arise (e.g., predicting recidivism and income). Through our experiments, we start by assuming the labels in the dataset are noise free, and we inject varying amounts of synthetic label noise. In this subsection, we describe the tasks, features, and ‘ground truth’ labels we use. The next subsection will describe how we introduce synthetic label noise.

Synthetic: We generate a dataset containing 5,000 instances according to the generative process in Appendix B. The positive rates for the majority and minority groups are 37.5% and 32.3%, respectively.

MIMIC-III: Within the healthcare domain, we leverage a publicly available dataset of electronic health record data (Johnson et al., 2016b). We consider two separate prediction tasks: onset of 1) acute respiratory failure (ARF) and 2) shock in the ICU (intensive care unit) (Oh et al., 2019). MIMIC-III includes data pertaining to vital signs, medications, diagnostic and procedure codes, and laboratory measurements. We consider the four hour prediction setup for both tasks as described by Tang et al. (2020), resulting in 15,873 and 19,342 ICU encounters, respectively. After preprocessing (see Appendix B), each encounter had 16,278 and 18,186 features for each task respectively. We use race as a sensitive attribute, with about 70% of patients being white (positive rate 4.5% [ARF], 4.1% [shock]) and 30% being non-white (positive rate 4.4% [ARF], 3.7% [shock]).

Beyond healthcare, we use two benchmark datasets frequently considered in the fairness domain.

Adult: a publicly available dataset of census data (Dua and Graf, 2017). We consider the task of predicting whether an individual’s income is over $50,000. This dataset includes data pertaining to age, education, work type, work sector, race, sex, marital status, and country. Its training and test sets contain 32,561 and 16,281 individuals, respectively. We use a pre-processed version of this dataset and randomly select 1,000 individuals out of 32,561 for training. We also only include features pertaining to age, education, work type, marital status, work sector, and sex to make the task more difficult (see Appendix B). After preprocessing, each individual was associated with 56 features, and all features had a range of 0-1. We use sex as a sensitive attribute, with 67.5% of individuals being male (positive rate 30.9%) and 32.5% being female (positive rate 11.3%).

COMPAS: a publicly available dataset collected by ProPublica from Broward County, Florida, USA (Angwin et al., 2016). We consider the task of pre-
dicting recidivism within two years, i.e., whether a criminal defendant is likely to re-offend. COMPAS includes data pertaining to age, race, sex, and criminal history. We use a pre-processed version of this dataset and also normalize each feature to have a range of 0-1 (see Appendix B). After preprocessing, the dataset included 6,172 individuals with 11 features per individual. We use race as a sensitive attribute, with 65.8% of individuals being white (positive rate 39.1%) and 34.2% being non-white (positive rate 44.5%).

Label Noise To test the robustness of our approach in different settings of label noise, we introduce synthetic instance-dependent label noise to our datasets. Like past work (Song et al., 2022), our setup is limited for the real datasets because our added noise is synthetic and we use the labels provided in the dataset as ground truth, since we do not have access to actual ground truth labels on these public datasets.

To introduce instance-dependent noise, mislabeling was a function of the features. Let \( w_m \sim N(0, 0.33)^D \) and \( z_m = \sigma(x \cdot w_m) \), where \( \sigma \) is the sigmoid function, denote the coefficients describing the contribution of each feature to mislabeling and the risk of mislabeling, respectively. Whether an instance was mislabeled was based on \( z_m \) and the desired noise rate. For example, for a noise rate of 30%, instances whose value for \( z_m \) was above the 70th percentile had their labels flipped. This allowed us to vary the noise rate within subgroups in a straightforward manner. Across datasets, we focused on cases where the noise rate in the ‘minority’ population was always greater than or equal to that of the ‘majority’ group since this is more likely to occur (Suit et al., 2007).

Evaluation Metrics We evaluate our proposed approach in terms of discriminative performance and model bias. For discriminative performance, we evaluate using the area under the receiver operating characteristic curve (AUROC) (higher is better).

With respect to model bias, while there exist many different measures, we focus on equalized odds (Hardt et al., 2016), since it is commonly used in the context of healthcare (Pfohl et al., 2019; Xu et al., 2022; Yogarajan et al., 2023), when similar performance across groups is desired (Rajkomar et al., 2018; Pfohl et al., 2021). Because equalized odds focuses on the difference between the true and false positive rates among groups, it is applicable to many settings in healthcare since the consequences of failing to treat a patient in need (Pingleton, 1988; Bone, 1994), or giving an inappropriate treatment (Bogun et al., 2004; Nasrallah, 2015) can be serious. More specifically, we measure the area under the equalized odds curve (AUROC) (de Freitas Pereira and Marcel, 2020) (higher is better). For classification threshold \( \tau \), we calculate the equalized odds (EO) between two groups, called 1 and 2, as shown below. \( TP_a(\tau) \) and \( FP_a(\tau) \) denote true and false positive rates for group \( a \) at threshold \( \tau \), respectively. The AUROC is obtained by plotting the EO against all possible values of \( \tau \) and calculating the area under the curve.

We compute the harmonic mean (HM) between the AUROC and AUROC to highlight how the different approaches simultaneously maintain discriminative performance and mitigate bias. In the harmonic mean the worse performing metric dominates. For example, if a classifier has AUROC=0.5 and AUROC=1.0, the harmonic mean will emphasize the poor discriminative performance.

\[
EO(\tau) = \frac{2 - |TP_1(\tau) - TP_2(\tau)| - |FP_1(\tau) - FP_2(\tau)|}{2}
\]

Baselines We evaluate our proposed approach with several baselines to test different hypotheses.

Standard does not account for label noise and assumes that \( \hat{y} = y \) is always true.

SLN + Filter (Chen et al., 2021) combines filtering (Arpit et al., 2017) and SLN (Chen et al., 2021) and was shown to outperform state-of-the-art approaches like Co-Teaching (Han et al., 2018) and DivideMix (Li et al., 2020). It relies on filtering heuristics, which indirectly rely on uniform random label noise to maintain discriminative performance and mitigate bias.

JS (Jensen-Shannon) Loss (Englesson and Azizpour, 2021) builds on semi-supervised learning and encourages model consistency when predicting on perturbations of the input features. It was shown to be competitive with other state-of-the-art noise-robust loss functions (Ma et al., 2020). It was proposed for instance-independent label noise.

Transition (Xia et al., 2020) learns to correct for noisy labels by learning a transition function and was shown to outperform state-of-the-art approaches such as MentorNet (Jiang et al., 2018). It applies to instance-dependent label noise, but it assumes that the contributions of each feature to mislabeling and input reconstruction are identical.

CSIDN (confidence-scored instance-dependent noise) (Berthon et al., 2021) also learns a transition
function and was shown to outperform state-of-the-art approaches such as forward correction (Patrini et al., 2017). Like our approach, CSIDN uses the concept of ‘confidence’ in the observed label to help with training. Unlike our approach, CSIDN uses the model’s class predictions directly as confidence scores (instead predicting them via an auxiliary network) and uses them to learn the transition function (as opposed to re-weighting the loss).

Fair GPL (Wang et al., 2021b) builds on work addressing uniform random label noise (Jiang and Nachum, 2020) and uses peer loss (i.e., data augmentation that reduces the correlation between the observed label and model’s predictions) within subgroups (Wang et al., 2021b). It assumes that label noise only depends on group membership.

We also train a model using the ground truth labels (called Clean Labels) as an empirical upper bound for discriminative performance.

Implementation Details For each dataset, we randomly split the data into 80/20% training/test, ensuring that data from the same individual did not appear across splits. For the Adult dataset, we used the test set provided and randomly selected 1,000 individuals from the training set. We then randomly selected 10% of the training data for all datasets except MIMIC-III from each subgroup to be alignment points, thereby ensuring that they were representative of the overall population. For the MIMIC-III dataset, 2% from each subgroup were selected as alignment points due to the larger size of the dataset. Alignment points were selected randomly to simulate our setting of focus, where we have a proxy labeling function and then randomly select a subset of the data to chart review in order to validate the proxy function. Then, for all datasets, half of the alignment points were then set aside as a validation set to use during training for early stopping and hyperparameter selection, while the other half remained in the training set. Later, in our experiments, we evaluated when the alignment set size varied and when the alignment set was biased. All approaches (i.e., baselines and proposed) were given the ground truth labels for data in the alignment set (i.e., no noise added to alignment points) during training so that some approaches did not have an unfair advantage.

All models were trained in Python3.7 and PyTorch1.7.1 (Paszke et al., 2017), using Adam (Kingma and Ba, 2014). Hyperparameters, including the learning rate, L2 regularization constant, and objective function scalars (e.g., $\alpha$), were tuned using random search, with a budget of 20. We used early stopping (patience=10) based on validation set performance, which we measured with the HM. We report results on the held-out test set, showing the mean and standard deviation over 10 replications.

4. Results and Discussion

We describe the results from experiments with instance-dependent noise. For each plot, we combined discriminative performance and bias mitigation and plotted the HM of the AUROC and AUEOC to assess general performance with respect to both metrics. We show the AUROC and AUEOC separately in Appendix D. Additional experiments are provided in Appendix D. Their results are summarized here.

Robustness to Noise Rate Here, we investigated how robust the proposed approach and baselines were to varying amounts of instance-dependent label noise (Figure 2). Since noise was synthetically introduced and not dataset specific, we conducted two experiments on the synthetic dataset. In the first, we varied the overall noise rate from 10-60% in the majority group. For the minority group, we considered noise rates that were consistently 20% higher than that of the majority group, to keep the noise disparity level (i.e., the difference in noise rates between subgroups) constant. In the second, we varied the minority noise rate from 20-90% with a majority noise rate fixed at 20% throughout (i.e., from 0-70% disparity) on the synthetic dataset.

Part 1: Overall Noise Rate Overall, our proposed approach demonstrated robustness to a variety of noise rates within a realistic range (Figure 2(a)). At low minority noise rates (i.e., below 40%), the proposed approach and baselines, with the exception of JS Loss, were competitive. As the noise rate increased, many of the baselines experienced noticeable degradation in performance. The proposed approach and Transition showed more robustness, with the proposed approach being the most robust until a minority noise rate of 80%, which represents an extreme case of label noise.

Part 2: Noise Disparity Like the previous experiment, the proposed approach was robust over a variety of noise disparities (Figure 2(b)). This is likely because the objective function $L^p_f$ from Step 2 of training accounts for disparities by scaling each instance-specific loss term with the reciprocal of its
Leveraging an Alignment Set in Tackling Instance-Dependent Label Noise

estimated group clean rate (i.e., 1 - the estimated group noise rate). Similar to the previous experiment, at a minority noise rate of 80% and above, the proposed approach was no longer the most robust, though this setting is unlikely to occur in practice.

Sensitivity to Alignment Set Composition

Our next set of experiments tested the proposed approach in settings where we relax key settings about the alignment set. We considered all datasets with instance-dependent noise. The majority/minority noise rates were 20%/40%, respectively. Here we show performance with respect to the proposed approach, Standard, and Clean Labels. Results for the other baselines are included in Appendix D.

Part 1: Alignment set size. We varied the size of the alignment set, from 1% and 15% of the training set, with the alignment set being representative of the test set (Figure 3(a)). The proposed approach was robust to a wide range of alignment set sizes, only showing noticeable degradation at alignment set sizes of 3% or lower. As the size of the alignment set grew, performance improved, likely since having a larger alignment set provided access to a larger set of ground truth labels at training time. Although the minimum number of points required in the alignment set is likely to vary depending on the task, our results are promising in that they show that our approach is effective on a variety of real life tasks, even when the alignment set is small (i.e., as little as 3% of the data).

Part 2: Biased alignment set. Here, we test how the proposed approach performs when the alignment set is not representative of the population. We varied the amount of bias in the alignment set by changing the proportion at which the subgroups were present. We kept the size of the alignment set constant at 10% of the training data (2% for MIMIC-III on both tasks). We observed that the proposed approach was robust over a wide range of conditions, i.e., when the minority proportion is 20%-80% (Figure 3(b)). We hypothesize that this is because the learned relationship between the features and noise can generalize across groups to an extent. In scenarios where performance of the proposed approach degraded, one subgroup heavily dominated the alignment set. This is shown in Figure 3(b) on the extremes of the x-axis of some datasets, which correspond to an alignment set that is heavily over-represented for one subgroup and heavily under-represented for the other. Our approach relies, in part, on having a relatively unbiased alignment set for estimating $\beta$ in order to avoid introducing dataset shift between the two steps of our training pipeline. Thus, these results are in line with our expectations and highlight a limitation of our approach. However, despite this reliance, we observe that our approach is still robust in some scenarios where the alignment set is biased.

Which Parts of Our Approach Matter? Our last set of results examines the individual components...
Leveraging an Alignment Set in Tackling Instance-Dependent Label Noise

(a) As we decrease the alignment set size (proportion of training data) performance decreases. Still, at an alignment set size of 3%, the proposed approach is robust.

(b) We varied the alignment set bias (proportion of minority instances). The proposed approach is generally robust to variations in bias. The dashed vertical black line shows an unbiased alignment set.

Figure 3: Robustness to varying alignment sets. Mean and standard deviation for 10 random seeds.

5. Related Work

We build from previous work in label noise and address key limitations. Generally, many state-of-the-art approaches (Song et al., 2022) are limited in that they do not consider instance-dependent noise, do not consider the potential consequences of bias in label noise, or do not leverage the information our setting provides. We tackle these limitations by accounting for differences in noise rates among subsets of the data and taking advantage of additional information that can be found in our setting. In this section, we summarize past work and highlight our contributions.

Which Parts of Our Approach Matter? Our last set of results aims to more closely examine the individual components of the approach itself. We include results for an ablation study and a hyperparameter sensitivity analysis in Appendix D. In summary, while each component improved performance, we find that the most improvement came from adding $L_\theta$ and $L_\phi$ during Steps 2a and 2b, respectively, as opposed to using only $L'_\theta$ during those steps. The approach was most sensitive to the $\alpha_2$ hyperparameter and more robust to $\alpha_1$ and $\gamma$.
Leveraging an Alignment Set in Tackling Instance-Dependent Label Noise

Net (Jiang et al., 2018), Co-teaching (Han et al., 2018), FINE (Kim et al., 2021)). Many are based on the idea that correctly labeled instances are easier to classify than mislabeled instances (i.e., the memorization effect) (Arpit et al., 2017). For example, mislabeled instances could be those that the model incorrectly classifies (Verbaeten, 2002; Khoshgoftaar and Rebours, 2004; Thongkam et al., 2008; Chen et al., 2019), have a high loss value (Yao et al., 2020a), or significantly increase the complexity of the model (Gamberger et al., 1996). Given the identified mislabeled instances, these approaches either ignore them during training (Zhang et al., 2020) or treat them as ‘unlabeled’ and apply techniques from semisupervised learning (e.g., DivideMix (Li et al., 2020), SELF (Nguyen et al., 2020)). Overall, these heuristics have been shown to improve discriminative performance. However, depending on the setting, they can disproportionately discard subsets of data, which could exacerbate biases in model performance.

For binary classification, some approaches ‘correct’ (i.e., switch) the observed label for instances that are predicted to be incorrect (Han et al., 2020; Zheng et al., 2020). Building on this idea, others make use of a transition function that estimates the probability of the observed label being correct. Model predictions can then be adjusted by applying the transition function to the classifier’s predictions for each class. Some works manually construct the transition function from expert knowledge (Patrini et al., 2017), while others learn it (Xiao et al., 2015; Xu et al., 2019; Yao et al., 2020b; Zheng et al., 2021; Jiang et al., 2022; Bae et al., 2022; Cheng et al., 2022; Li et al., 2022). However, such approaches often make assumptions on the form of the noise distribution, and past work has shown that results are sensitive to the choice of distribution (Ladouceur et al., 2007).

To date, much of the work described above assumes instance-independent label noise (i.e., mislabeling is independent of the features). However, when this assumption is violated, the model may overfit to label noise (Lukasik et al., 2020). From an emerging body of work in instance-dependent label noise (Cheng et al., 2020b; Xia et al., 2020; Wang et al., 2021c; Zhu et al., 2022b), current approaches remain limited in that they still rely on filtering heuristics. Although we use soft filtering, we filter based on the learned relationship between the features and noise rather than existing heuristics and upweight groups with a higher estimated noise rate. While similar to a transition function in some aspects, our approach requires fewer probability estimates on label correctness (two estimates compared to the number of classes squared for a transition function) while achieving state-of-the-art performance.

**Noise-Robust Loss Functions** Prior work examines how regularization techniques can be adapted to the noisy labels setting, addressing issues related to overfitting on noisy data (Menon et al., 2019; Lukasik et al., 2020; Englesson and Azizpour, 2021). Label smoothing, and in some cases negative label smoothing, were found to improve the accuracy on both correctly labeled and mislabeled data (Lukasik et al., 2020; Wei et al., 2022a). With this approach, the observed labels are perturbed by a small, predetermined value, with all labels receiving the same perturbation at each training epoch. Follow-up work found that, instead of applying the same perturbation at each epoch, adding a small amount of Gaussian stochastic label noise (SLN) at each epoch resulted in further improvements, as it helped to escape from local optima (Chen et al., 2021). However, these approaches were most beneficial in the context of augmenting existing methods that identify mislabeled instances (e.g., stochastic label noise is applied to instances that are identified as correctly labeled by filtering approaches), and thus, potentially suffer from the same limitations. Alternatively, recent work has also proposed perturbing the features to encourage consistency in the model’s predictions (Englesson and Azizpour, 2021), though mainly in the context of instance-independent label noise. Others have proposed noise-robust variations of cross entropy loss (Feng et al., 2020; Wang et al., 2021a) but generally relied on assumptions like the memorization effect.

**Label Noise in Fairness** Label noise has also been addressed within the fairness literature recently. When the frequencies at which subgroups (defined by a sensitive attribute) appear are different within a dataset, past work has shown that common approaches addressing label noise can increase the prediction error for minority groups (i.e., rarer subgroups) (Liu, 2021). Past work proposed to re-weight instances from subgroups during training where model performance is poorer (Jiang and Nachum, 2020) in the instance-independent noise setting. Others use peer loss (Liu and Guo, 2020) within subgroups (Wang et al., 2021b) but assume that noise depends only on the sensitive attribute. We also train with a weighted loss, but weights are based on predicted label correctness rather than performance on
the observed labels. Recently, Wu et al. (2022) addressed some of the gaps of past work by examining the instance-dependent case. Our proposed approach differs from theirs in that we do not require our features to be grouped into distinct categories, such as root and low level attributes.

Anchor Points for Addressing Label Noise

Another related setting in past work uses anchor points. Anchor points are subsets of the data where the ground truth labels are known (Liu and Tao, 2015). To date, anchor points are generally used to learn a transition function (Xia et al., 2019, 2020; Berthon et al., 2021) or for label correction directly (Wu et al., 2021). We use a similar concept, alignment points, to 1) pre-train the model, and 2) predict label correctness. The first part builds from work in semi-supervised learning (Cascante-Bonilla et al., 2021), which has shown improvements from pre-training on labeled data. The second part is similar to a transition function, but differs in that we use the correctness predictions to re-weight the loss rather than adjust the predictions. We also assume that, for some alignment points, the ground truth and observed labels do not match. Generally, anchor-based approaches mitigate model bias by implicitly assuming that the anchor points are representative of the target population. Our approach also uses this assumption, but we empirically explore how model performance changes when the anchor points are biased (i.e., not representative), since it may be easier to obtain correct labels for specific subgroups Spector-Bagdady et al. (2021).

6. Conclusion

We introduce a novel approach for learning with instance-dependent label noise. Our two-stage approach uses the complete dataset and learns the relationship between the features and label noise using a small set of alignment points. On several datasets, we show that the proposed approach leads to improvements over state-of-the-art baselines in maintaining discriminative performance and mitigating bias. Our approach is not without limitations. We demonstrated that the success of the approach depends, in part, on the representativeness in the alignment set. Our experiments were also on pseudo-synthetic data in which we injected noise; this assumes we start from a noise free dataset. Finally, we only examined one form of bias in a specific case of instance-dependent label noise. Nonetheless, our case frequently arises in healthcare, especially when pragmatic (e.g., automated) labeling tools are used on large datasets, and chart review on the entire dataset is infeasible.

Acknowledgments

This work was supported by Cisco Systems Inc. and the National Science Foundation (NSF award no. IIS 2124127). 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 Cisco Systems Inc. or the National Science Foundation. We also thank the anonymous reviewers for their valuable feedback.

References


Lei Feng, Senlin Shu, Zhuozi Lin, Fengmao Lv, Li Li, and Bo An. Can cross entropy loss be robust to label noise? In International Joint Conferences on Artificial Intelligence, pages 2206–2212, 2020.


Leveraging an Alignment Set in Tackling Instance-Dependent Label Noise


Leveraging an Alignment Set in Tackling Instance-Dependent Label Noise


Leveraging an Alignment Set in Tackling Instance-Dependent Label Noise


Appendix A. Proposed Approach: Additional Details

We provide additional details on our approach, including a general overview in the form of pseudocode as well as a justification for the proposed objective function and its relation to the clean label loss.

A.1. General Overview

We summarize our approach with pseudocode below in Algorithm 1. We begin with the dataset and initial model parameters, and we aim to use the dataset to learn the final model parameters. A is the set of anchor points. $\theta^{'}$ and $\phi^{'}$ are the initial model parameters for the $\theta$ and $\phi$ networks. Here, 'stopping criteria' may refer to any stopping criteria, such as early stopping. The Freeze() function takes as input model parameters and freezes them, and the Unfreeze() function takes as input model parameters and unfreezes them.

A.2. Proposed and Clean Label Loss

We show that minimizing the proposed loss $L_{\theta}$ from Step 2 of the proposed method is equal to minimizing cross entropy on the clean labels in expectation.

$$L_{\theta} = \frac{1}{|A|} \sum_{k=1}^{g} \sum_{i \in A \cap G_k} \frac{1}{1 - \hat{r}_k} \sum_{j=1}^{c} \hat{\beta}_\phi \mathbb{I}(\hat{y}(i) == j) \log(\hat{y}_j(i))$$

Therefore,

$$\mathbb{E}\left[ \sum_{k=1}^{g} \sum_{i \in A \cap G_k} \frac{1}{1 - \hat{r}_k} \sum_{j=1}^{c} \hat{\beta}_\phi \mathbb{I}(\hat{y}(i) == j) \log(\hat{y}_j(i)) \right]$$

$$= \sum_{k=1}^{g} \sum_{i \in A \cap G_k} \frac{1}{1 - \hat{r}_k} \sum_{j=1}^{c} \mathbb{E}\left[ \hat{\beta}_\phi \mathbb{I}(\hat{y}(i) == j) \log(\hat{y}_j(i)) \right]$$

$$= \sum_{k=1}^{g} \sum_{i \in A \cap G_k} \frac{1}{1 - \hat{r}_k} \sum_{j=1}^{c} \mathbb{I}(\hat{y}(i) == j) \log(\hat{y}_j(i))$$

As a reminder, each group $G_k$ is then associated with estimated noise rate $\hat{r}_k = \frac{1}{|G_k|} \sum_{i \in G_k} 1 - \hat{\beta}_\phi^{(i)}$ and estimated clean (i.e., correct) rate $1 - \hat{r}_k = \hat{\beta}_\phi^{(i)}$.

Algorithm 1: Proposed approach.

| Input: | $\{x(i), \hat{y}(i), y(i)\}_{i \in A \cap G}$, $\{x(i), \hat{y}(i)\}_{i \in A}$, $\theta, \phi$ |
| Output: | $\theta, \phi$ (final model parameters) |
| Hyperparameters: | Scalars $\alpha_1, \alpha_2, \gamma$ |
| Train | $(\{x(i), \hat{y}(i), y(i)\}_{i \in A \cap G}$, $\{x(i), \hat{y}(i)\}_{i \in A}$, $\theta, \phi$) |

1. While ¬(stopping criteria) (Step 1)
   (a) $\hat{y} = \theta^{'}(x)$ (Predict label)
   (b) $\hat{\beta}_\phi = \phi^{'}(x, \hat{y})$ (Predict label confidence)
   (c) $L_\theta = \frac{1}{|A|} \sum_{i \in A \cap G_k} \sum_{j=1}^{c} \mathbb{I}(\hat{y}(i) == j) \log(\hat{y}_j(i))$
   (d) $L_\phi = \frac{1}{|A|} \sum_{i \in A} \sum_{j=1}^{c} \mathbb{I}(\hat{y}(i) == y(i)) \log(\hat{\beta}_\phi^{(i)}) + \mathbb{I}(\hat{y}(i) \neq y(i)) \log(1 - \hat{\beta}_\phi^{(i)})$
   (e) Loss $= L_\theta + \alpha_1 L_\phi$
   (f) Update model parameters
   (g) Compute stopping criteria

2. $\theta, \phi \leftarrow \theta^{'}$, $\phi^{'}$

3. Freeze($\phi$)

4. While ¬(stopping criteria) (Step 2)
   (a) $\hat{y} = \theta^{'}(x)$
   (b) $\hat{\beta}_\phi = \phi^{'}(x, \hat{y})$
   (c) $L_\theta = \frac{1}{|A|} \sum_{i \in A \cap G_k} \sum_{j=1}^{c} \mathbb{I}(\hat{y}(i) == j) \log(\hat{y}_j(i))$
   (d) $L_\phi = \frac{1}{|A|} \sum_{i \in A} \sum_{j=1}^{c} \mathbb{I}(\hat{y}(i) == y(i)) \log(\hat{\beta}_\phi^{(i)}) + \mathbb{I}(\hat{y}(i) \neq y(i)) \log(1 - \hat{\beta}_\phi^{(i)})$
   (e) $L_{\theta}^{\beta} = \frac{1}{|A|} \sum_{k=1}^{G} \frac{1}{1 - \hat{r}_k} \sum_{i \in A \cap G_k} \sum_{j=1}^{c} \hat{\beta}_\phi \mathbb{I}(\hat{y}(i) == j) \log(\hat{y}_j(i))$ (Weighted loss)
   (f) If $\phi$ is frozen (Step 2a)
      i. Loss $= L_{\theta}^{\beta} + \gamma L_{\theta}$
      ii. Unfreeze($\phi$)
      iii. Freeze($\theta$)
   (g) Else (Step 2b)
      i. Loss $= L_{\theta}^{\beta} + \alpha_2 L_{\phi}$
      ii. Unfreeze($\theta$)
      iii. Freeze($\phi$)
   (h) Update model parameters
   (i) Compute stopping criteria

5. Return $\theta, \phi$ (Final model parameters)
\[ \frac{1}{|G_k|} \sum_{i \in G_k} \hat{\beta}^{(i)} \phi. \] We can express the noise and clean rates in terms of \( \hat{\beta}^{(i)} \) since
\[
1 - r_k = \frac{1}{|G_k|} \sum_{i \in G_k} \mathbb{1}(\hat{y}^{(i)} = y^{(i)})
= P(y = \hat{y}|\hat{x}, x) \text{ for a random instance in } G_k
= \frac{1}{|G_k|} \sum_{i \in G_k} P(y^{(i)} = \hat{y}^{(i)}|\hat{y}^{(i)}, x^{(i)})
\]
where \( r_k \) and \( 1 - r_k \) are the actual noise and clean rates within group \( k \), respectively. Therefore, since \( \hat{\beta}^{(i)} \) is trained to predict \( P(y = \hat{y}|\hat{x}, x) \), we estimate the noise and clean rates using \( \hat{\beta}^{(i)} \).

**Appendix B. Preprocessing Details**

Here, we provide more detail on our synthetic data generation process and real dataset pre-processing.

**B.1. Synthetic**

Our data generation process is as described below. Note that the \( \text{Percentile}(p, \{z\}) \) function outputs the \( p^\text{th} \) percentile over all values in \( \{z\} \). We defined the feature at index 0 to be a synthetic sensitive attribute. Instances with values below the \( 20^\text{th} \) percentile for this feature were considered as the ‘minority’, and the rest were considered as the ‘majority’. Features 10-19 for the majority instances and features 20-29 for the minority instances were set to 0 to provide more contrast between the two groups. For individual \( i \),
\[
\begin{align*}
    d &= 30, \mathbf{x}^{(i)} \sim N(0,1)^{30} \\
    \mathbf{w} &\sim N(0,1)^{30}, \mathbf{z}^{(i)} = \mathbf{x}^{(i)} \cdot \mathbf{w} \\
    y^{(i)} &= 1 \text{ if } z^{(i)} > \text{Percentile}(50, \{z^{(j)}\}) \text{ else } 0 \\
    x^{(i)}_j &= 0 \text{ for } j = 10, 11, ..., 19 \\
    &\text{ if } x^{(i)}_0 > \text{Percentile}(20, \{x^{(j)}_0\}) \text{ else } 0 \\
    x^{(i)}_j &= 0 \text{ for } j = 20, 21, ..., 29 \\
    &\text{ if } x^{(i)}_0 < \text{Percentile}(20, \{x^{(j)}_0\}) \text{ else } 0 \\
\end{align*}
\]

**B.2. MIMIC-III**

Data were processed using the FlexIble Data Driven pipeLinE (FIDDLE), ([Tang et al., 2020]), a publicly available pre-processing tool for electronic health record data. We used the same features as ([Tang et al., 2020]) for our tasks. More information can be found at https://physionet.org/content/mimic-ecu- fiddle-feature/1.0.0/.

**B.3. Adult**

Although, we used a pre-processed version of this dataset, we omitted features pertaining to education, work type, and work sector to make the task more difficult. More specifically, in the file ‘headers.txt’ at the repository mentioned in Footnote 1, we kept all features beginning with ‘age’, ‘workclass’, ‘education’, ‘marital status’, and ‘occupation’. We also kept the ‘Sex_Female’ feature. The remaining features were excluded to make the task more difficult. Values were normalized for each feature to have a range of 0-1 by subtracting the minimum value observed among all individuals and dividing by the range. During training, we only used 1,000 randomly selected individuals from the provided dataset to make the task more difficult, since there would be fewer samples from which to learn. We made the task more difficult for this dataset to further highlight the differences in performance between the approaches.

**B.4. COMPAS**

Although, we used a pre-processed version of this dataset, we omitted the feature ‘score_factor’ (i.e., the risk score for recidivism from the ProPublica model) to make the task more difficult. Values were normalized for each feature to have a range of 0-1 by subtracting the minimum value observed among all individuals and dividing by the range.

**Appendix C. Additional Network and Training Details**

Here, our ranges of hyperparameters and implementation choices for the proposed network. All networks were trained on Intel(R) Xeon(R) CPUs, E7-4850 v3 @ 2.20GHz and Nvidia GeForce GTX 1080 GPUs. All layers were initialized with He initialization from a uniform distribution. We divide our training data into five batches during training. All random seeds (for Pytorch, numpy, and Python’s random) were initialized with 123456789.

**C.1. Hyperparameter Values Considered**

Here, we show the range of values we considered for our random search. More details are provided in Ta-
Table 2: For each dataset, we list the range of hyperparameters considered for each dataset. For each hyperparameter, the lower bound is shown in the top row, and the upper bound is shown in the bottom row. For hyperparameters we did not tune, only one row is shown.

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Synthetic</th>
<th>MIMIC-ARF</th>
<th>MIMIC-Shock</th>
<th>Adult</th>
<th>COMPAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer Size</td>
<td>10</td>
<td>500</td>
<td>500</td>
<td>100</td>
<td>10</td>
</tr>
<tr>
<td>Learning Rate</td>
<td>0.00001</td>
<td>0.00001</td>
<td>0.000001</td>
<td>0.00001</td>
<td>0.0001</td>
</tr>
<tr>
<td></td>
<td>0.01</td>
<td>0.001</td>
<td>0.001</td>
<td>0.01</td>
<td>0.05</td>
</tr>
<tr>
<td>L2 Constant</td>
<td>0.0001</td>
<td>0.000001</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>0.01</td>
<td>0.1</td>
<td>0.1</td>
<td>0.01</td>
</tr>
<tr>
<td>Filter Threshold</td>
<td>0.40</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.90</td>
<td>0.90</td>
</tr>
<tr>
<td>Noise Added</td>
<td>0.00001</td>
<td>0.00001</td>
<td>0.00001</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td></td>
<td>0.01</td>
<td>0.001</td>
<td>0.001</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>Number of Parts</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>$\alpha_{GPL}$</td>
<td>0.01</td>
<td>0.1</td>
<td>0.001</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>$\alpha_{1Proposed}$</td>
<td>0.1</td>
<td>0.1</td>
<td>0.01</td>
<td>0.1</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
</tr>
<tr>
<td>$\gamma_{Proposed}$</td>
<td>0.1</td>
<td>0.1</td>
<td>0.01</td>
<td>0.1</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
</tr>
<tr>
<td>$\alpha_{2Proposed}$</td>
<td>0.1</td>
<td>0.1</td>
<td>0.01</td>
<td>0.1</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
</tr>
</tbody>
</table>

Table 2. For any hyperparameters associated with the Adam optimizer not mentioned above, we used the default values. Not all hyperparameters were used with each approach. ‘Filter Threshold’ and ‘Noise Added’ were only used with the baseline SLN + Filter. Here, Filter Threshold refers to the minimum value of the predicted probability of the observed label for an instance to be considered ‘correctly labeled’. For example, if Filter Threshold=0.5, then all examples whose predicted probability for the observed label is at least 0.5 are considered ‘correct’ and used during training. ‘Number of Parts’ was only used with the baseline Transition. ‘$\alpha_{GPL}$’ was only used with the baseline Fair GPL. ‘$\alpha_{1Proposed}$’, ‘$\alpha_{2Proposed}$’, and ‘$\gamma_{Proposed}$’ was only used with the proposed method. Here, ‘$\alpha_{1Proposed}$’ and ‘$\alpha_{2Proposed}$’ correspond to the terms $\alpha_1$ and $\alpha_2$ that were used in the objective functions. We refer to them with the added term ‘Proposed’ in the subscript in this section to distinguish it from the $\alpha$ value used by the baseline Fair GPL.

C.2. Network Details
For the overall architecture, we used a feed forward network with two hidden layers. The auxiliary $\beta$ prediction component was also implemented with two feed forward layers. All layer sizes are as described in Table 2. In addition, we used the ReLU activation function. The complete implementation can be found in the attached code.

Appendix D. Expanded Results
Here, we describe additional results that were not included in the main text. We begin with followup experiments on the synthetic data and then describes results from the real data.

D.1. Robustness to Noise Rate Expanded
Here we include the AUROC and AUEOC plotted separately for the experiments where we varied the overall noise rate and noise disparity.
Leveraging an Alignment Set in Tackling Instance-Dependent Label Noise

As we varied the overall noise rate (Figure 4), the proposed approach is able to consistently outperform the baselines with respect to discriminative performance until a minority noise rate of 80%. This observation is similar to what we observed with the HM. With respect to bias mitigation, the proposed approach is not more beneficial than the baselines up to a minority noise rate of 60%. At a minority noise rate above 60%, our approach experienced the least degradation compared to the baseline approaches. This is in line with our expectations since our approach explicitly accounts for differences in noise rates among groups during training.

As we varied the noise disparity (Figure 5), we have similar observations to the previous experiment in that the proposed approach is able to consistently outperform the baselines with respect to discriminative performance until a minority noise rate of 80%. With respect to bias mitigation, the proposed approach is not more beneficial than the baselines up to a minority noise rate of 40%. At a minority noise rate above 40%, our approach experienced the least degradation compared to most of the other baseline approaches and was comparable to the Transition baseline. Unlike the previous experiment, the degradation in AUEOC among many of the baseline approaches is
larger, which is in line with our expectations since we were directly changing the difference in noise rates between the groups while the previous experiment kept the difference constant.

D.2. Ablation Study

We also examined our approach more closely by conducting an ablation study and a hyperparameter sensitivity analysis on the synthetic data. We used the synthetic dataset since our noise was synthetically introduced and not dataset specific. In our ablation study (Figure 6), we began with training on only the alignment points (i.e., Step 1 only), which achieved the worst performance. We then introduced Step 2 and added the remaining training data (i.e., non-alignment points) but only trained using $L_\theta$. This led to an improvement in performance, but not to the level of the full approach. The next two ablations build on the previous one. In the first one, we added continued supervision on the alignment points with $L_\theta$, and observed an improvement in performance, likely due to the retention of high quality data in this step. In the second one, we added continued supervision on the alignment points using $L_{\phi}$, and observed an even larger improvement. This is likely because including $L_{\phi}$ prevented the model from learning a solution where $\hat{\beta}$ was small for all instances, as previously discussed. Finally, we end with our full proposed approach, which performed noticeably better than each of the ablations, showing the importance of each component.

D.3. Hyperparameter Sensitivity Analysis

In our sensitivity analysis on the synthetic data (Figure 7), we tested how performance of the (full) proposed approach varied to changes in the hyperparameters $\alpha_1$, $\alpha_2$, and $\gamma$. For each of these hyperparameters, we measured performance at values between 0.01 and 100 on a logarithmic scale while keeping the other two values constant at 1. We found that $\alpha_1$ and $\gamma$ were the most robust to changes in the value. We found that $\alpha_2$ was more sensitive, with values between 0.1 and 10 generally working best.

D.4. Sensitivity to Alignment Set Composition Expanded

In our analysis on sensitivity to alignment set composition, we include results for the other baselines in (Figure 8). At alignment set sizes of below 5% on the real datasets, the proposed approach was beneficial to the baselines. At larger alignment set sizes, the baseline Transition was able to match the proposed method due to the increased amount of clean data. When the alignment set was biased, the proposed approach outperformed the baselines in the unbiased settings and was competitive as bias in the alignment set increased.
(a) We varied $\alpha_1$.

(b) We varied $\alpha_2$.

(c) We varied $\gamma$.

Figure 7: Sensitivity analysis of proposed approach on objective function hyperparameters.

(a) As we decrease the alignment set size (proportion of training data) performance decreases. Still, at an alignment set size of 5%, the proposed approach generally outperforms the baselines.

(b) As we vary the alignment set bias (proportion of minority instances) performance varies. The proposed approach is generally robust to changes in the bias of the alignment set. The dashed vertical black line shows the proportion at which the minority group occurs in the dataset (i.e., an unbiased alignment set).

Figure 8: Robustness to varying alignment sets. Mean and standard deviation for 10 random seeds.