Technical Appendix For
Recovering The Propensity Score From Biased Positive Unlabeled Data

Implementation Details

Data Preparation and Experimental Details We used a random 70-30 train-test split on each dataset, each run. This was achieved using Scikit-Learn's `train_test_split` function, stratifying according to class. Results for each dataset were obtained over 10 runs.

Introducing Class Overlap As described in the Experiments section of the main paper, we introduce class overlap in the Scaled Propensity experiments. To achieve this, we generate new points along the border between the positive and negative class using Borderline SMOTE (Han, Wang, and Mao 2005). Specifically, we use `BorderlineSMOTE` function in imbalanced-learn (Lemaître, Nogueira, and Aridas 2017) Python package. We use the default settings of imbalanced-learn version 0.5.0. We flip the labes of the genrated points (so that negative instances are generated on the positive side, and vice versa). We repeat this process until a base logistic regression classifier gets roughly 70% accuracy on the resultant data, indicating a roughly 30% class overlap.

Generating Labels from Propensity Score We first generate a propensity function value for each datapoint according to whichever propensity score function we are using (where the choice of propensity score is described in the setup of each experiment). Then, for all true positive instances, we take a draw from a random uniform distribution between 0 and 1. If the draw is less than the propensity score for that instance, a label of 1 (positive label) is given. Otherwise, a 0 label is assigned for that instance (indicating that the instance is unlabeled). All true negative instances are unlabeled.

Alternative Posterior Estimation Models

As stated, we use a Gaussian Process method to determine the labeling posterior. As our methods are sensitive to a good posterior estimate, we show the performance of the Local Certainty method for three different posterior estimators: Gaussian Process, MLP, and Logistic Regression. Each estimator used Scikit-Learn's default hyperparameters. Results are shown in Figure 1.

Breaking the Probabilistic Gap Assumption

Our Probabilistic Gap method assumes that the propensity score is a linear function of the class posterior, $e = k \cdot p(y = 1|x)$. However, this is a somewhat strong assumption that may not hold in practice. We thus evaluate the ability of our Probabilistic Gap method to recover the propensity score when the propensity score follows the *order* of the class prior, but is not a scalar multiple of it. To achieve this, we generate labels using a propensity score that is a multiple of the posterior *squared* ($e = k \cdot p(y = 1|x)^2$) and a multiple of the sigmoid of the posterior ($e = k \cdot \sigma(p(y = 1|x))$, where σ is the sigmoid function). Results shown in Figures 2, 3, and 4 illustrate that while performance is degraded when its assumptions aren't met, our Probabilistic Gap method is usually still the best performing method.

Various Class Overlap

As described in the Experiments section of the main paper, we use a 30% class overlap for the Scaled Propensity experiments. We vary the overlap from 10% to 30% here on the Bank dataset, demonstrating that the Probabilistic Gap method is still the best performing and robust to the choice of overlap. Results shown in Figure 5.

Utility of Recovered Propensity Scores: Using Propensity Sores For Classification

As discussed in the Preliminaries section of the main paper, the propensity score can be used for classification. We thus illustrate the utility of our estimated propensity scores by comparing the class posteriors obtained from our estimated propensity scores to those obtained form the state-of-the-art propensity estimation methods. We achieve this by training a down-stream classifier for each dataset using the propensity-weighted risk (Equation 1 of the main paper). We utilize the propensity scores obtained in the the "Recovering the Propensity Score" experiments in the main paper (experiments for the results reported in Figure 2 and Figure 3 of the main paper); thus, the dataset preparation and details for those experiments hold true for this experiment as well.

The results, shown in Table 1 and Table 2, demonstrate that our Local Certainty method produces a downstream classifier with the lowest (best) error for the Arbitrary Propensity setting (Table 1), and our Probabilistic Gap

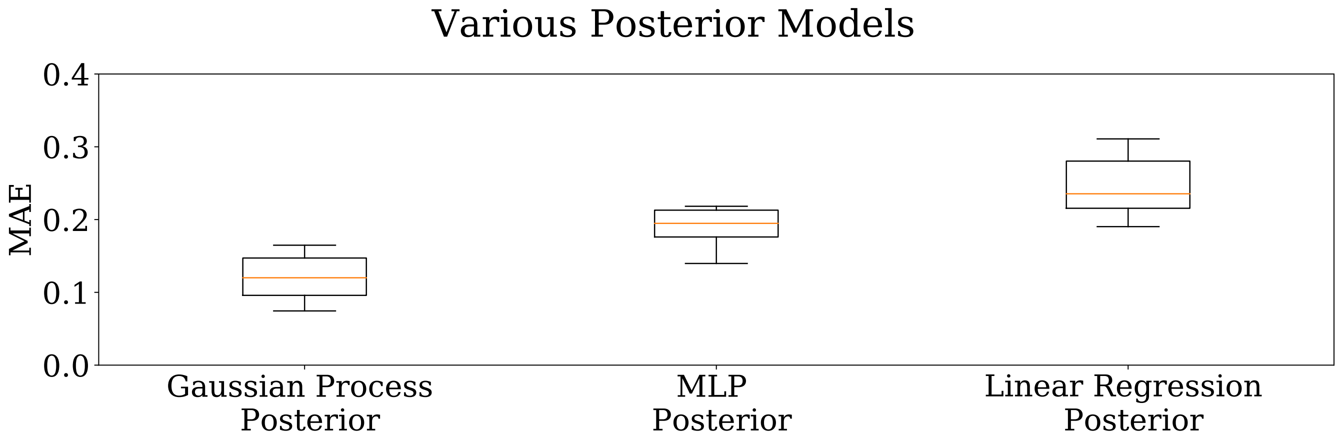

Figure 1: Results for Local Certainty when using various posteriors. Results reported on the Wine dataset.

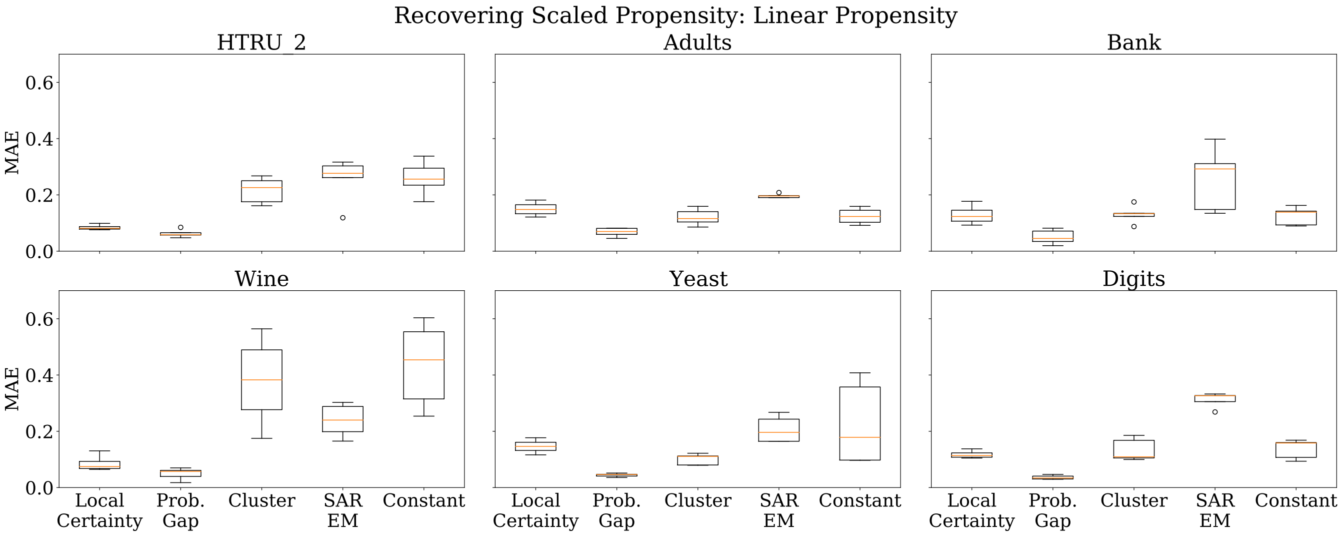

Figure 2: Scalar multiple.

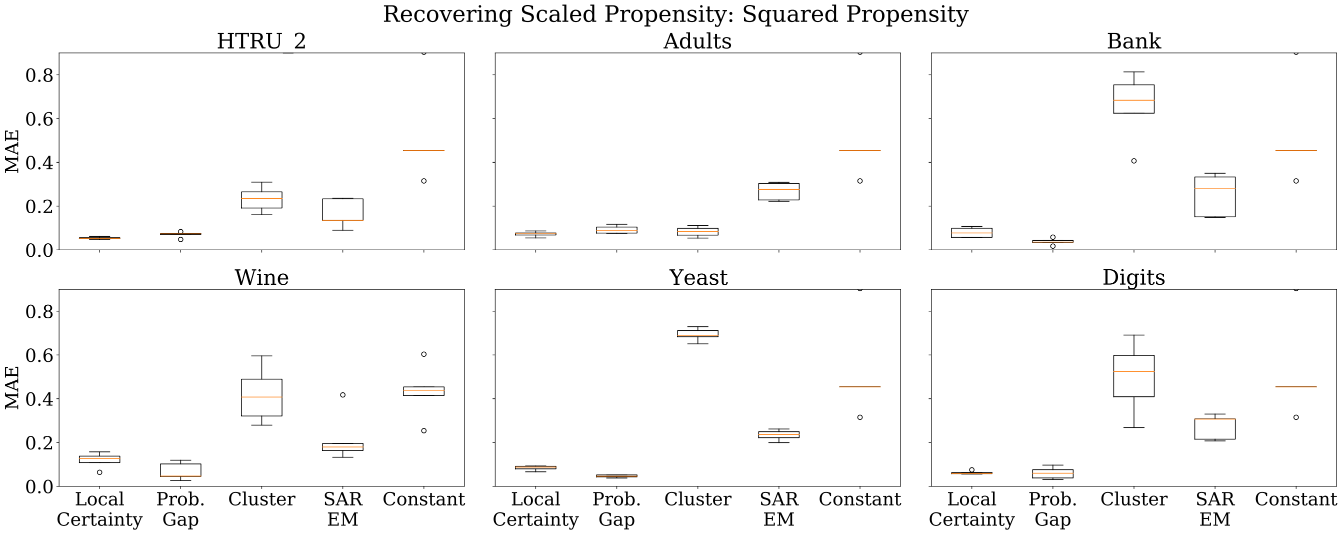

Figure 3: Squared posterior.

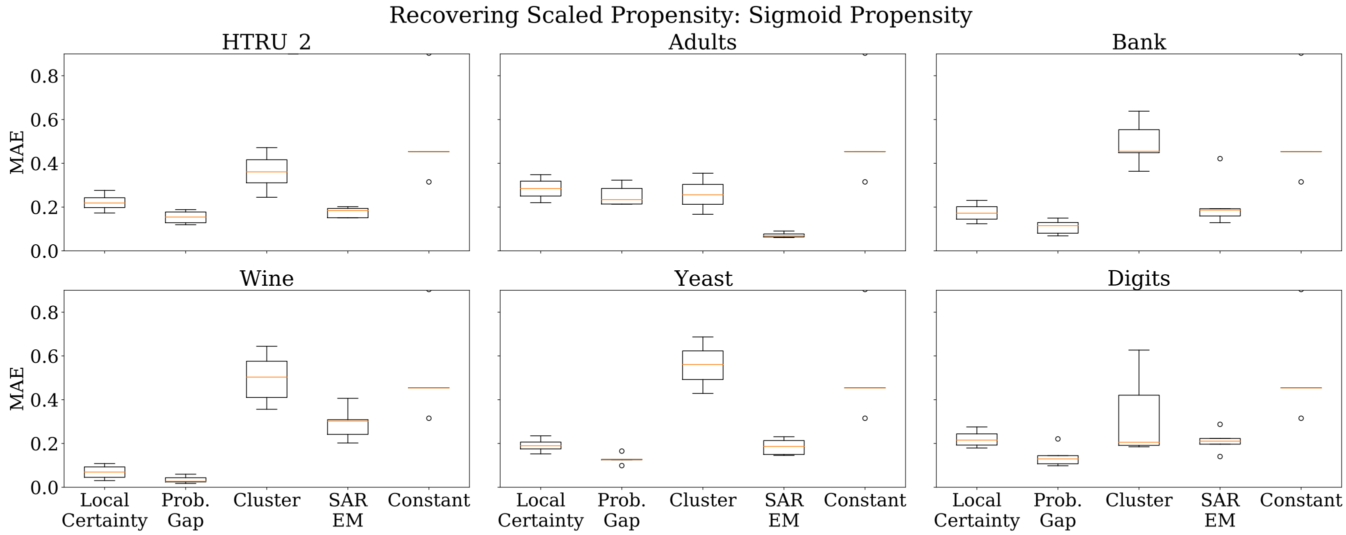

Figure 4: Sigmoid of posterior.

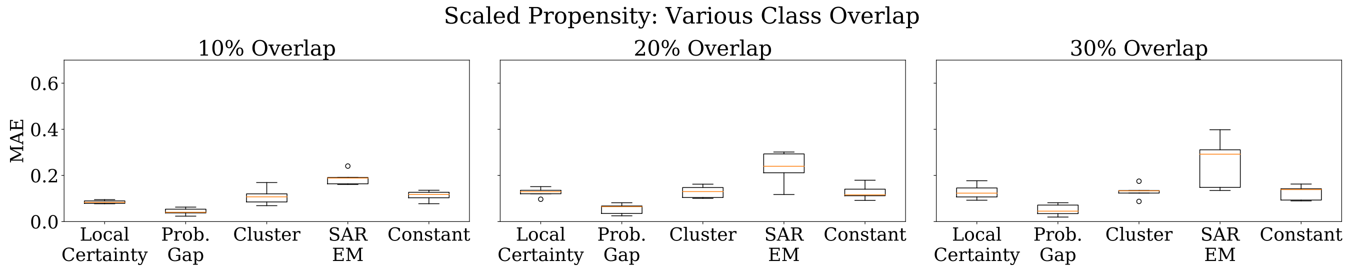

Figure 5: Performance for various amounts of class overlap on the Bank dataset.

Dataset	HTRU 2	Adult	Bank	Wine	Yeast	Digits
LC (Ours)	**0.04**+/-0.00	**0.35**+/-0.02	**0.06**+/-0.01	**0.24**+/-0.02	**0.44**+/-0.00	**0.21**+/-0.01
PG (Ours)	0.10+/-0.00	0.40+/-0.00	0.22+/-0.01	0.36+/-0.02	0.47+/-0.00	0.33+/-0.01
Cluster	0.05+/-0.00	0.37+/-0.00	0.09+/-0.00	0.48+/-0.01	0.46+/-0.00	0.23+/-0.00
SE	0.10+/-0.05	0.37+/-0.01	0.09+/-0.01	0.47+/-0.05	0.46+/-0.01	0.33+/-0.03
Constant	0.43+/-0.00	0.70+/-0.01	0.17+/-0.01	0.34+/-0.02	0.46+/-0.00	0.29+/-0.01

Table 1: Classification error for arbitrary propensity score scenario

Dataset	HTRU 2	Adult	Bank	Wine	Yeast	Digits
LC (Ours)	0.14+/-0.01	0.75+/-0.01	0.38+/-0.03	0.26+/-0.01	0.45+/-0.01	0.51+/-0.02
PG (Ours)	0.05+/-0.00	**0.25**+/-0.03	**0.30**+/-0.01	**0.25**+/-0.02	**0.41**+/-0.01	**0.27**+/-0.01
Cluster	**0.04**+/-0.00	0.27+/-0.00	0.33+/-0.00	0.78+/-0.01	0.45+/-0.00	0.51+/-0.01
SE	0.14+/-0.08	0.26+/-0.01	0.35+/-0.01	0.51+/-0.06	0.44+/-0.01	0.49+/-0.03
Constant	0.43+/-0.00	0.46+/-0.03	0.39+/-0.01	0.34+/-0.03	0.46+/-0.00	0.54+/-0.01

Table 2: Classification error for scaled propensity score scenario

Dataset	HTRU 2	Adult	Bank	Wine	Yeast	Digits
LC (Ours)	**0.20**+/-0.08	**0.19**+/-0.02	0.10+/-0.04	0.20+/-0.08	0.23+/-0.15	0.31+/-0.02
PG (Ours)	0.18+/-0.02	0.37+/-0.03	0.08+/-0.02	**0.13**+/-0.04	0.21+/-0.05	**0.27**+/-0.01
Cluster	0.47+/-0.14	0.48+/-0.16	0.31+/-0.17	0.26+/-0.17	0.30+/-0.20	0.51+/-0.01
SE	0.12+/-0.05	0.21+/-0.14	**0.05**+/-0.03	0.24+/-0.04	**0.19**+/-0.02	0.49+/-0.03
Constant	0.33+/-0.10	0.29+/-0.05	0.21+/-0.04	0.24+/-0.04	0.27+/-0.10	0.34+/-0.01

Table 3: MAE of propensity scores estimates when true propensity score is constant

method produces the best classifier in 5/6 of the datasets in the Scaled Propensity setting (Table 2). This shows that our methods nearly always result in the training of a more accurate classifier than the state-of-the-art propensity-recovering PU methods.

Constant Propensity Scenario

This experiment aims to answer the following question: Do the proposed methods still perform well when the propensity score is constant (i.e., in the unbiased setting)? To answer this question, we have performed another set of experiments for which the propensity score is constant (constant between 0.5 and 1.0, varied with a step size of 0.1, with 10 runs per value). Results are shown in Table 3. Our methods win 4/6 times and SAR-EM wins the remaining 2 times. Notably, the Constant method never wins. This is not surprising as the propensity score being constant does not break the assumption of our Local Certainty method, SAR EM, nor the Cluster method.

Proof of Theorem 2

We prove that identifiability does not hold for the propensity score in the Positive Subdomain, Positive Function, and Irreducubility assumptions in the case when e is allowed to be any arbitrary function of x by showing that under these assumptions there exists regions of the feature space $\mathcal{X}$ such that multiple values of e in this region will perfectly explain the observed data.

We begin with the Positive Subdomain assumption. Under this assumption, there is a region $\mathcal{A} \subset \mathcal{X}$ for which $p(y = 1|x^*) = 1 \; \forall x^* \in \mathcal{A}$. Notably, this region $\mathcal{A}$ is unknown, and $p(\ell = 1|x^*)$ can be less than 1 (i.e., labeling is not assumed to be perfect in $\mathcal{A}$. For some point $x^*in\mathcal{A}$, let $p(\ell = 1|x^*) = k$. Then, any value in $[k, 1]$ for the estimate of the corresponding propensity score (e^*) for this point will perfectly explain the data given a corresponding estimate of $y^* = \frac{k}{e^*}$. Thus, the propensity score is not identifiable under the Positive Subdomain assumption.

As the Positive Subdomain assumption implies the Positive Function and Irreducibility assumptions, the propensity score is likewise non-identifiable under these assumptions as well.

Proof of Theorem 3

This theorem is shown in (Bekker, Robberechts, and Davis 2019) and the proof is repeated here for the sake of completeness.

$$bias(R_{prop}(\hat{Y}|\hat{E},L)) = R(\hat{Y}) - \mathbb{E}[R_{prop}(\hat{Y}|\hat{E},L)]$$

$$\begin{aligned}\mathbb{E}[R_{prop}(\hat{Y}|\hat{E},L)] =& \frac{1}{n}\sum_{i=1}^{n} y_i e_i \left(\frac{1}{\hat{e}_i}\delta_1(\hat{y}_i) + (1 - \frac{1}{\hat{e}_i})\delta_0(\hat{y}_i)\right) \\ &+ (1 - y_i e_i)\delta_0(\hat{y}_i) \\ =& \frac{1}{n}\sum y_i \frac{e_i}{\hat{e}_i}\delta_1(\hat{y}_i) + (1 - y_i \frac{e_i}{\hat{e}_i})\delta_0(\hat{y}_i)\end{aligned}$$

$$\begin{aligned}bias(R_{prop}(\hat{Y}|\hat{E},L)) =& \frac{1}{n}\sum (y_i - y_i\frac{e_i}{\hat{e}_i})\delta_1(\hat{y}_i) \\ &+ (1 - y_i - 1 + y_i\frac{e_i}{\hat{e}_i})\delta_0(\hat{y}_i) \\ =& \frac{1}{n}\sum y_i(1 - \frac{e_i}{\hat{e}_i})\delta_1(\hat{y}_i) \\ &+ y_i(1 - \frac{e_i}{\hat{e}_i})\delta_0(\hat{y}_i) \\ =& \frac{1}{n}\sum y_i(1 - \frac{e_i}{\hat{e}_i})(\delta_1(\hat{y}_i) - \delta_0(\hat{y}_i)\end{aligned}$$

References

Bekker, J.; Robberechts, P.; and Davis, J. 2019. Beyond the Selected Completely at Random Assumption for Learning from Positive and Unlabeled Data. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2019, Würzburg, Germany, September 16-20, 2019, Proceedings, Part II*, volume 11907 of *Lecture Notes in Computer Science*, 71–85. Springer.

Han, H.; Wang, W.-Y.; and Mao, B.-H. 2005. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, 878–887. Springer.

Lemaître, G.; Nogueira, F.; and Aridas, C. K. 2017. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research*, 18(17): 1–5.