A Unified Approach to Count-Based Weakly-Supervised Learning

Vinay Shukla [1] **Zhe Zeng** [*1] **Kareem Ahmed** [*1] **Guy Van den Broeck** [1]

Abstract

High-quality labels are often very scarce, whereas unlabeled data with inferred weak labels occurs more naturally. In many cases, these weak labels dictate the frequency of each respective class over a set of instances. In this paper, we develop a unified approach to learning from such weakly-labeled data, which we call *count-based weakly-supervised learning*. At the heart of our approach is the ability to compute the probability of exactly k out of n outputs being set to true. This computation is differentiable, exact, and efficient. Building upon the previous computation, we derive a *count loss* penalizing the model for deviations in its distribution from an arithmetic constraint defined over label counts. We evaluate our approach on three common weakly-supervised learning paradigms and observe that our proposed approach achieves state-of-the-art or highly competitive results across all three of the paradigms.

1. Introduction

Weakly supervised learning (Zhou, 2018) enables a model to learn from data with restricted, partial or inaccurate labels, often known as *weakly-labeled data*. Weakly supervised learning fulfills a need arising in many real-world settings that are subject to privacy or budget constraints, such as privacy sensitive data (Wojtusiak et al., 2011), medical image analysis (Bortsova et al., 2018), clinical practice (Quellec et al.), personalized advertisement (Bekker & Davis, 2020) and knowledge base completion (Galárraga et al., 2015; Zupanc & Davis, 2018), to name a few. In all such settings, *instance-level labels* are unavailable. Instead, instances are grouped into *bags* with corresponding *bag-level labels* that are a function of the instance labels, e.g., the proportion of positive labels in a bag. A key insight that we bring forth is that such weak supervision can very often be construed as *enforcing constraints on label counts of data*.

More concretely, we consider three prominent weakly supervised learning paradigms. The first paradigm is known as *learning from label proportions* (Quadrianto et al., 2008). Here the weak supervision consists in the *proportion* of positive labels in a given bag, which can be interpreted as *the count of positive instances* in such a bag. The second paradigm, whose supervision is strictly weaker than the former, is *multiple instance learning* (Maron & Lozano-Pérez, 1997; Dietterich et al., 2001). Here the bag labels only indicate the *existence* of at least one positive instance in a bag, which can be recast as to whether *the count of positive instances* is greater than zero. The third paradigm, *learning from positive and unlabeled data* (De Comité et al., 1999; Letouzey et al., 2000), grants access to the ground truth labels for a subset of *only the positive instances*, providing only a class prior for what remains. We can recast the class prior as *a distribution of the count of positive labels*.

Leveraging the view of weak supervision as a constraint on label counts, we utilize a simple, efficient and probabilistically sound approach to weakly-supervised learning. More precisely, we train a neural network to make instance-level predictions that conform to the desired label counts. To this end, we propose a *differentiable count loss* that characterizes how close the network's distribution comes to the label counts; a loss which is surprisingly tractable. Compared to prior methods, this approach does not approximate probabilities but computes them *exactly*. Our empirical evaluation demonstrates that our proposed count loss significantly boosts the classification performance on all three aforementioned settings.

2. Problem Formulations

Notations. Let $\mathcal{X} \in \mathbb{R}^d$ be the input space over d features, $\mathcal{Y} = \{0, 1\}$ be a binary label space, and $\boldsymbol{x} \in \mathcal{X}$ and $y \in \mathcal{Y}$ be the input and output random variables respectively.

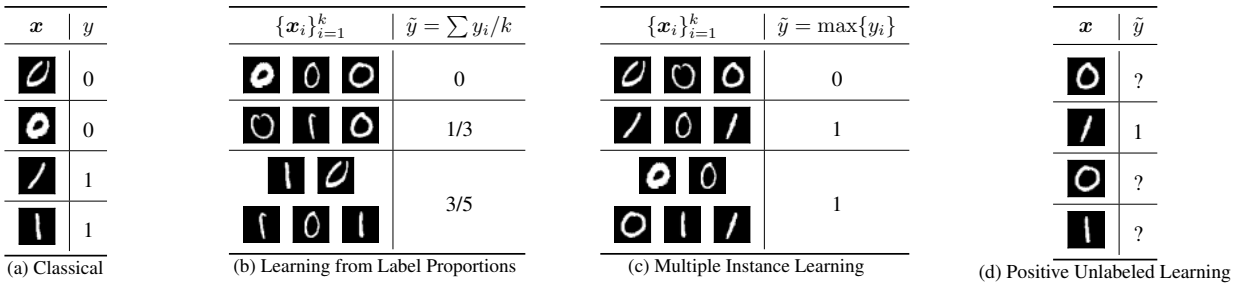
2.1. Classical Binary Classification

In fully-supervised binary classification, it is assumed that each feature and label pair $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$ is sampled inde-

[*]Equal contribution [1]Department of Computer Science, University of Los Angeles, Los Angeles, United States. Correspondence to: Vinay Shukla <vshukla@g.ucla.edu>, Zhe Zeng <zhezeng@cs.ucla.edu>, Kareem Ahmed <ahmedk@cs.ucla.edu>, Guy Van den Broeck <guyvdb@cs.ucla.edu>.

Published at the Differentiable Almost Everything Workshop of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. July 2023. Copyright 2023 by the author(s).

Table 1: Three weakly supervised settings, Learning from Label Proportions (Section 2.2), Multiple Instance Learning (Section 2.3) and Positive and Unlabeled learning (Section 2.4), against the classical fully supervised setting (Section 2.1) for binary classification, using digits from the MNIST dataset.



(a) Classical (b) Learning from Label Proportions (c) Multiple Instance Learning (d) Positive Unlabeled Learning

pendently from a joint distribution $p(\boldsymbol{x}, y)$. A classifier f is learned to minimize the risk $R(f) = \mathbb{E}_{(\boldsymbol{x},y)\sim p}[\ell(f(\boldsymbol{x}), y)]$, where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ is the loss function. We define a set of training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where the empirical loss is minimized as $\hat{R}(f) = \frac{1}{n}\sum_{i=1}^n \ell(f(\boldsymbol{x}_i), y_i)$.

2.2. Learning from Label Proportions

Learning from label proportions (LLP) (Quadrianto et al., 2008) assumes that each instance in the training set is assigned to bags and only the proportion of positive instances in each bag is known. One example is in light of the pandemic, where infection rates were typically reported based on geographical boundaries such as states and counties. Each boundary can be treated as a bag with the infection rate as the proportion annotation.

The goal of LLP is to learn an instance-level classifier $f : \mathcal{X} \to \mathcal{Y}$ even though it is trained on bag-level labeled data. Formally, the training dataset consists of m bags, denoted by $\mathcal{D} = \{(B_i, \tilde{y}_i)\}_{i=1}^m$ where each bag $B_i = \{\boldsymbol{x}_j\}_{j=1}^k$ consist of k instances and this k could vary among different bags. The bag proportions are defined as $\tilde{y}_i = \sum_{j=1}^k y_j/k$ with y_j being the instance label that cannot be accessed and only $\tilde{y}_i$ is available during training. An example is shown in Figure 1b. We do not assume that the bags are non-overlapping while some existing work suffer from this limitation including Scott & Zhang (2020).

2.3. Multiple Instance Learning

Multiple instance learning (MIL) (Maron & Lozano-Pérez, 1997; Dietterich et al., 2001) refers to the scenario where the training dataset consists of bags of instances and labels are provided at bag level. However, in MIL, the bag label is a single binary label indicating whether there is a positive instance in the bag or not as opposed to a bag proportion defined in LLP. A real-world application of MIL lies in the field of drug activity (Dietterich et al., 2001). We can observe the effects of a group of conformations but not for any specific molecule, motivating a MIL setting. Formally, in MIL, the training dataset consists of m bags, denoted by $\mathcal{D} = \{(B_i, \tilde{y}_i)\}_{i=1}^m$, with a bag consisting of k instances, i.e., $B_i = \{\boldsymbol{x}_j\}_{j=1}^k$. The size k can vary among different bags. For each instance $\boldsymbol{x}_j$, there exists an instance-level label y_j which is not accessible. The bag-level label is defined as $\tilde{y}_i = \max_j\{y_j\}$. An example is shown in Figure 1c.

The main goal of MIL is to learn a model that predicts a bag label while a more challenging goal is to learn an instance-level predictor that is able to discover positive instances in a bag. In this work, we aim to tackle both by training an instance-level classifier whose predictions can be combined into a bag level prediction as the last step.

2.4. Learning from Positive and Unlabeled Data

Learning from positive and unlabeled data or *PU learning* (De Comité et al., 1999; Letouzey et al., 2000) refers to the setting where the training dataset consists of only positive instances and unlabeled data, and the unlabeled data can contain both positive and negative instances. A motivation of PU learning is persistence in the case of shifts to the negative-class distribution (Plessis et al., 2015), for example, a spam filter. An attacker may alter the properties of a spam email, making a traditional classifier require a new negative dataset (Plessis et al., 2015). We note that taking a new unlabeled sample would be more efficient, motivating PU learning. Formally, in PU learning, the training dataset $\mathcal{D} = \mathcal{D}_p \cup \mathcal{D}_u$ where $\mathcal{D}_p = \{(\boldsymbol{x}_i, \tilde{y}_i = 1)\}_{i=1}^{n_p}$ is the set of positive instances with $\boldsymbol{x}_i$ from $p(\boldsymbol{x} \mid y = 1)$ and $\tilde{y}$ denoting whether the instance is labeled, and $\mathcal{D}_u = \{(\boldsymbol{x}_i, \tilde{y}_i = 0)\}_{i=1}^{n_u}$ the unlabeled set with $\boldsymbol{x}_i$ from

$$p_u(\boldsymbol{x}) = \beta\, p(\boldsymbol{x} \mid y = 1) + (1 - \beta)\, p(\boldsymbol{x} \mid y = 0), \quad (1)$$

where the mixture proportion $\beta := p(y = 1 \mid \tilde{y} = 0)$ is the fraction of positive instances among the unlabeled population. Although the instance label y is not accessible, its information can be inferred from the binary selection label $\tilde{y}$: if the selection label $\tilde{y} = 1$, it belongs to the positively labeled set, i.e., $p(y = 1 \mid \tilde{y} = 1) = 1$; otherwise, the instance $\boldsymbol{x}$ can be either positive or negative. An example

Table 2: A summary of the labels and objective functions for all the settings considered in the paper.

TASK	LABEL	LABEL LEVEL	OBJECTIVE
classical fully supervised	binary y	instance level	$-y \log p(y) - (1-y)\log(1-p(y))$
learning from label proportion	continuous $\tilde{y} = \sum_i y_i / k$	bag level	$-\log p(\sum \hat{y}_i = k\tilde{y})$
multiple instance learning	binary $\tilde{y} = \max\{y_i\}$	bag level	$-\tilde{y}\log p(\sum \hat{y}_i \geq 1) - (1-\tilde{y})\log p(\sum_i \hat{y}_i = 0)$
learning from PU data	binary $\tilde{y}$	instance level	1) $\mathbb{D}_{KL}(p(\sum_i \hat{y}_i) \parallel \texttt{Bin}(k,\beta))$; 2) $-\log p(\sum \hat{y}_i = k\beta)$

of such a dataset is shown in Figure 1d.

The goal of PU learning is to train an instance-level classifier. However, it is not straightforward to learn from PU data and it is necessary to make assumptions to enable learning with positive and unlabeled data (Bekker & Davis, 2020). In this work, we make a commonly-used assumption for PU learning, *selected completely at random (SCAR)*, which lies at the basis of many PU learning methods.

Definition 2.1 (**SCAR**)**.** Labeled instances are selected completely at random, independent from input $\boldsymbol{x}$ and the positive distribution, i.e., $p(\tilde{y} = 1 \mid \boldsymbol{x}, y = 1) = p(\tilde{y} = 1 \mid y = 1)$.

3. A Unified Approach: Count Loss

We aim to derive objectives for the three weakly supervised settings from first principles. We propose to bridge between neural outputs, which can be observed as counts, and arithmetic constraints derived from the weakly supervised labels. The idea is to capture how close the classifier is to satisfying the arithmetic constraints. They can be easily integrated with deep learning models and allow end-to-end training.

For the three objectives, we show that they share the same computational building block: given k instances $\{\boldsymbol{x}_i\}_{i=1}^k$ and an instance-level classifier f that predicts $p(\hat{y}_i \mid \boldsymbol{x}_i)$ with $\hat{y}$ being the prediction, inferring the probability of the count constraint $\sum_{i=1}^k \hat{y}_i = s$ is to compute

$$p(\sum_{i=1}^k \hat{y}_i = s \mid \{\boldsymbol{x}_i\}_{i=1}^k) := \sum_{\hat{\boldsymbol{y}} \in \mathcal{Y}^k} [\![\sum_{i=1}^k \hat{y}_i = s]\!] \prod_{i=1}^k p(\hat{y}_i \mid \boldsymbol{x}_i)$$

where $[\![\cdot]\!]$ denotes the indicator function and $\hat{\boldsymbol{y}}$ denotes the vector $(\hat{y}_1, \cdots, \hat{y}_k)$. For succinctness, we omit the dependency on the input and simply write the count probability as $p(\sum_{i=1}^k \hat{y}_i = s)$. Intractable as it seems, we show that it is indeed possible to derive a tractable computation for the count probability based on Ahmed et al. (2023b).

Proposition 3.1. *The count probability $p(\sum_{i=1}^k \hat{y}_i = s)$ of sampling k prediction variables with summation being s from an unconstrained distribution $p(\boldsymbol{y}) = \prod_{i=1}^k p(\hat{y}_i)$ can be computed exactly in time $\mathcal{O}(ks)$. Moreover, the set $\{p(\sum_{i=1}^k \hat{y}_i = s)\}_{s=0}^k$ can also be computed in time $\mathcal{O}(k^2)$.*

Next, we show how the objectives derived from first principles can be solved by using the count probability as an oracle, which we summarize in Table 2.

LLP setting. Given a bag $B = \{\boldsymbol{x}_i\}_{i=1}^k$and its bag-label $\tilde{y}$, by definition, it can be inferred that the number of positive instances (count) in the bag is $k\tilde{y}$. Our objective is to maximize the probability $p(\sum_i \hat{y}_i = k\tilde{y})$. In practice, the loss is defined in the log space for numerical stability. We also perform risk analysis with details in Appendix.

MIL setting. Given a bag $B = \{\boldsymbol{x}_i\}_{i=1}^k$ of size k and a single binary label $\tilde{y}$ as its weakly supervised label, we propose a cross-entropy loss as below

$$\ell(B, \tilde{y}) = -\tilde{y}\log p(\sum \hat{y}_i \geq 1) - (1-\tilde{y})\log p(\sum \hat{y}_i = 0).$$

In the above loss, the probability term $p(\sum \hat{y}_i = 0)$ can be obtained via the oracle for computing count probability and the other term $p(\sum \hat{y}_i \geq 1)$ can simply be obtained from $1 - p(\sum \hat{y}_i = 0)$, i.e., the same call to the oracle.

PU Learning setting. Recall that for the unlabeled data $\mathcal{D}_u$ in the training dataset, an unlabeled instance $\boldsymbol{x}_i$ is drawn from a mixture distribution as shown in Equation 1 parameterized by a mixture proportion $\beta = p(y = 1 \mid \tilde{y} = 0)$.

Proposition 3.2. *With SCAR assumption and a class prior, the mixture proportion β can be estimated from dataset $\mathcal{D}$.*

The probabilistic semantic of the mixture proportion is that, if we randomly draw an instance $\boldsymbol{x}_i$ from the unlabeled population, the probability that the true label y_i is positive would be β. Further, if we randomly draw k instances, the distribution of the summation of the true labels $\sum_{i=1}^k y_i$ conforms to a binomial distribution $\texttt{Bin}(k, \beta)$ parameterized by the mixture proportion β, i.e., $p(\sum_{i=1}^k y_i = s) = \binom{k}{s}\beta^s(1-\beta)^{k-s}$. Based on this observation, we propose an objective to minimize the KL divergence between the distribution of predicted label sum and the binomial distribution parameterized by the mixture proportion, that is,

$$\mathbb{D}_{KL}\left(p(\sum_{i=1}^k \hat{y}_i) \parallel \texttt{Bin}(k,\beta)\right) = \sum_{s=0}^k p(\sum_{i=1}^k \hat{y}_i = s) \log \frac{p(\sum_{i=1}^k \hat{y}_i = s)}{\texttt{Bin}(s; k, \beta)}$$

Table 3: LLP results showing test AUC with standard deviation aggregated over 5 trials for each experimental setting.

Dataset	Dist	Method	8	32	128	512
Adult	$[0, \frac{1}{2}]$	CL (Ours)	**0.8984 ± 0.0013**	**0.8848 ± 0.0041**	**0.8743 ± 0.0052**	**0.8703 ± 0.0070**
Adult	$[0, \frac{1}{2}]$	PL	0.8889 ± 0.0024	0.8782 ± 0.0036	**0.8743 ± 0.0039**	0.8678 ± 0.0085
Adult	$[0, \frac{1}{2}]$	LMMCM	0.8728 ± 0.0019	0.8693 ± 0.0047	0.8669 ± 0.0041	0.8674 ± 0.0040
Adult	$[\frac{1}{2}, 1]$	CL (Ours)	**0.8854 ± 0.0022**	**0.8738 ± 0.0039**	0.8675 ± 0.0043	**0.8607 ± 0.0056**
Adult	$[\frac{1}{2}, 1]$	PL	0.8781 ± 0.0038	0.8731 ± 0.0035	**0.8699 ± 0.0057**	0.8556 ± 0.0180
Adult	$[\frac{1}{2}, 1]$	LMMCM	0.8584 ± 0.0164	0.8644 ± 0.0052	0.8601 ± 0.0045	0.8500 ± 0.0186

Table 4: MIL experiment on MNIST dataset. Each block represents a different distribution from which we draw bag sizes—First Block: $\mathcal{N}(10, 2)$, Second Block: $\mathcal{N}(50, 10)$, Third Block: $\mathcal{N}(100, 20)$.

Training Bags	50	100	150	200	300	400	500
Gated Attention	0.775 ± 0.034	0.894 ± 0.012	0.935 ± 0.005	0.939 ± 0.006	**0.963 ± 0.002**	0.959 ± 0.002	**0.966 ± 0.003**
Attention	0.807 ± 0.026	**0.913 ± 0.006**	**0.940 ± 0.004**	0.942 ± 0.007	0.957 ± 0.002	0.961 ± 0.005	0.965 ± 0.004
CL (Ours)	**0.818 ± 0.024**	0.906 ± 0.009	0.929 ± 0.005	**0.946 ± 0.001**	0.952 ± 0.004	**0.962 ± 0.002**	0.963 ± 0.002
Gated Attention	**0.943 ± 0.005**	0.949 ± 0.009	**0.970 ± 0.005**	**0.977 ± 0.001**	0.983 ± 0.002	0.986 ± 0.004	**0.987 ± 0.002**
Attention	0.936 ± 0.010	**0.962 ± 0.006**	**0.970 ± 0.001**	**0.977 ± 0.002**	0.981 ± 0.002	**0.987 ± 0.001**	**0.987 ± 0.002**
CL (Ours)	0.939 ± 0.010	0.960 ± 0.002	0.964 ± 0.007	0.972 ± 0.002	**0.982 ± 0.003**	0.982 ± 0.001	**0.987 ± 0.002**
Gated Attention	0.975 ± 0.003	0.981 ± 0.004	0.992 ± 0.002	0.987 ± 0.004	**0.996 ± 0.001**	**0.998 ± 0.001**	0.990 ± 0.004
Attention	**0.984 ± 0.001**	0.982 ± 0.001	**0.996 ± 0.000**	0.987 ± 0.007	0.992 ± 0.004	0.994 ± 0.002	0.998 ± 0.000
CL (Ours)	0.981 ± 0.007	**0.989 ± 0.000**	**0.996 ± 0.002**	**0.995 ± 0.001**	**0.996 ± 0.002**	0.993 ± 0.003	**0.999 ± 0.001**

Table 5: PU Learning results on acccuracy.

Dataset	CL (Ours)	CVIR	nnPU	nPU
Binarized MNIST	**96.4 ± 0.01**	96.3 ± 0.07	96.1 ± 0.14	95.2 ± 0.19
MNIST17	**99.0 ± 0.19**	98.7 ± 0.09	98.4 ± 0.20	98.4 ± 0.09
Binarized CIFAR	80.1 ± 0.34	**82.3 ± 0.18**	77.2 ± 1.03	76.7 ± 0.74
Cat vs. Dog	**74.8 ± 1.64**	73.3 ± 0.94	71.8 ± 0.33	68.8 ± 0.53

where $\texttt{Bin}(s; k, \beta)$ denotes the probability mass function of the binomial distribution $\texttt{Bin}(k, \beta)$. Again, the KL divergence can be obtained by $k + 1$ calls to the oracle for computing count probability $p(\sum_{i=1}^{k} \hat{y}_i = s)$. The KL divergence is further combined with a cross entropy defined over labeled data $\mathcal{D}_p$ as in the classical binary classification as the overall objective. As an alternative, we propose an objective for the unlabeled data that requires fewer calls to the oracle: matching only the expectations of the two distributions, that is, to maximize $p(\sum_{i=1}^{k} \hat{y}_i = k\beta)$ where $k\beta$ is the expectation of the binomial distribution $\texttt{Bin}(k, \beta)$.

4. Experiments

We present a thorough empirical evaluation of our proposed count loss (CL) on the three tasks, *LLP*, *MIL* and *PU learning* respectively, with additional results in Appendix.

LLP. We experiment on datasets *Adult* with 8192 training samples where the task is to predict whether a person makes over $50k$ a year or not given personal information as input. We follow the experimental settings from Scott & Zhang (2020) where two settings are considered: one with label proportions uniformly on $[0, \frac{1}{2}]$ and the other uniformly on $[\frac{1}{2}, 1]$. We experiment on four bag sizes n with $n \in \{8, 32, 128, 512\}$. We compare our approach CL with LMMCM from (Scott & Zhang, 2020) and against Proportion Loss (PL)(Tsai & Lin, 2020) with results in Table 3, where CL showcases superior results against the baselines.

MIL. We experiment on the MNIST dataset (LeCun, 1998) and follow the setting in Ilse et al. (2018): the training and test set bags are randomly sampled from MNIST; each bag can have images of digits from 0 to 9, and bags with digit 9 are labeled positive. The task is to train a classifier that is able to predict bag labels; the more challenging task is to *discover key instances*, that is, to train a classifier that identifies images of digit 9. The number of bags in training set n is in $\{50, 100, 150, 200, 300, 400, 500\}$. We include the Attention and Gated Attention mechanism (Ilse et al., 2018) as baselines. Results are shown in Table 4, where CL is able to outperform all other baselines or exhibit highly comparable performance for bag-level predictions.

PU Learning. We experiment on MNIST (LeCun, 1998) and CIFAR-10 (Krizhevsky & Hinton, 2009), following the four simulated settings from Garg et al. (2021). The performance is evaluated using the accuracy on a test set of unlabeled data. We compare our proposed CL using the first objective in Table 2 with baseline Conditional Value Ignoring Risk approach (CVIR) (Garg et al., 2021), nnPU (Kiryo et al., 2017), and uPU (Plessis et al., 2015). Results are shown in Table 5, where our CL perform better than baselines on 3 out of the 4 simulated PU learning settings.

Acknowledgments

This work was funded in part by the DARPA Perceptually-enabled Task Guidance (PTG) Program under contract number HR00112220005, NSF grants #IIS-1943641, #IIS-1956441, #CCF-1837129, Samsung, CISCO, a Sloan Fellowship, and a gift from RelationalAI. GVdB discloses a financial interest in RelationalAI. ZZ is supported by an Amazon Doctoral Student Fellowship.

References

Ahmed, K., Wang, E., Chang, K.-W., and Van den Broeck, G. Leveraging unlabeled data for entity-relation extraction through probabilistic constraint satisfaction, mar 2021.

Ahmed, K., Li, T., Ton, T., Guo, Q., Chang, K.-W., Kordjamshidi, P., Srikumar, V., Van den Broeck, G., and Singh, S. Pylon: A pytorch framework for learning with constraints. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (Demo Track)*, feb 2022a.

Ahmed, K., Teso, S., Chang, K.-W., Van den Broeck, G., and Vergari, A. Semantic probabilistic layers for neuro-symbolic learning. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, dec 2022b.

Ahmed, K., Wang, E., Chang, K.-W., and den Broeck, G. V. Neuro-symbolic entropy regularization. In *The 38th Conference on Uncertainty in Artificial Intelligence*, 2022c.

Ahmed, K., Chang, K.-W., and Van den Broeck, G. Semantic strengthening of neuro-symbolic learning. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS)*, apr 2023a.

Ahmed, K., Zeng, Z., Niepert, M., and Van den Broeck, G. Simple: A gradient estimator for k-subset sampling. In *Proceedings of the International Conference on Learning Representations (ICLR)*, may 2023b.

Andrews, S., Tsochantaridis, I., and Hofmann, T. Support vector machines for multiple-instance learning. In *NIPS*, 2002.

Ardehaly, E. M. and Culotta, A. Co-training for demographic classification using deep learning from label proportions. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 1017–1024. IEEE, 2017.

Bekker, J. and Davis, J. Learning from positive and unlabeled data: A survey. *Machine Learning*, 109:719–760, 2020.

Belle, V., Passerini, A., and Van den Broeck, G. Probabilistic inference in hybrid domains by weighted model integration. In *Proceedings of IJCAI*, pp. 2770–2776, 2015a.

Belle, V., Van den Broeck, G., and Passerini, A. Hashing-based approximate probabilistic inference in hybrid domains. In *Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence (UAI)*, 2015b.

Bortsova, G., Dubost, F., Ørting, S., Katramados, I., Hogeweg, L., Thomsen, L., Wille, M., and de Bruijne, M. Deep learning from label proportions for emphysema quantification. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11*, pp. 768–776. Springer, 2018.

Carbonneau, M.-A., Cheplygina, V., Granger, E., and Gagnon, G. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 2018.

De Comité, F., Denis, F., Gilleron, R., and Letouzey, F. Positive and unlabeled examples help learning. pp. 219–230, 12 1999. ISBN 978-3-540-66748-3. doi: 10.1007/3-540-46769-6_18.

Dietterich, T., Lathrop, R., and Lozano-Pérez, T. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89:31–71, 03 2001. doi: 10.1016/S0004-3702(96)00034-3.

du Plessis, M. C., Niu, G., and Sugiyama, M. Analysis of learning from positive and unlabeled data. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Galárraga, L., Teflioudi, C., Hose, K., and Suchanek, F. M. Fast rule mining in ontological knowledge bases with amie++. *The VLDB Journal*, 24(6):707–730, 2015.

Garg, S., Wu, Y., Smola, A. J., Balakrishnan, S., and Lipton, Z. Mixture proportion estimation and pu learning: a modern approach. *Advances in Neural Information Processing Systems*, 34:8532–8544, 2021.

Hüllermeier, E. Learning from imprecise and fuzzy observations: Data disambiguation through generalized loss minimization. *International Journal of Approximate Reasoning*, 55(7):1519–1534, 2014.

Ilse, M., Tomczak, J., and Welling, M. Attention-based deep multiple instance learning. In *International conference on machine learning*, pp. 2127–2136. PMLR, 2018.

Kiryo, R., Niu, G., Du Plessis, M. C., and Sugiyama, M. Positive-unlabeled learning with non-negative risk estimator. *Advances in neural information processing systems*, 30, 2017.

Kobayashi, R., Mukuta, Y., and Harada, T. Risk consistent multi-class learning from label proportions. *arXiv preprint arXiv:2203.12836*, 2022.

Kolb, S., Morettin, P., Zuidberg Dos Martires, P., Sommavilla, F., Passerini, A., Sebastiani, R., and De Raedt, L. The pywmi framework and toolbox for probabilistic inference using weighted model integration. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI*, pp. 6530–6532, 7 2019. doi: 10.24963/ijcai.2019/946.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009.

LeCun, Y. The mnist database of handwritten digits. 1998.

Letouzey, F., Denis, F., and Gilleron, R. Learning From Positive and Unlabeled examples. In *Proceedings of the 11th International Conference on Algorithmic Learning Theory, ALT'00*, pp. 71–85, Sydney, Australia, 2000. Springer Verlag.

Manhaeve, R., Dumancic, S., Kimmig, A., Demeester, T., and De Raedt, L. Deepproblog: Neural probabilistic logic programming. In *NeurIPS*, 2018.

Maron, O. and Lozano-Pérez, T. A framework for multiple-instance learning. In Jordan, M., Kearns, M., and Solla, S. (eds.), *Advances in Neural Information Processing Systems*, volume 10. MIT Press, 1997.

Plessis, M. D., Niu, G., and Sugiyama, M. Convex formulation for learning from positive and unlabeled data. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, 2015.

Quadrianto, N., Smola, A., Caetano, T., and Le, Q. Estimating labels from label proportions. 2008.

Quellec, G., Cazuguel, G., Cochener, B., and Lamard, M. Multiple-instance learning for medical image and video analysis. *IEEE reviews in biomedical engineering*.

Scott, C. and Zhang, J. Learning from label proportions: A mutual contamination framework. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 22256–22267. Curran Associates, Inc., 2020.

Sirinukunwattana, K., Raza, S. e. A., Tsang, Y., Snead, D., Cree, I., and Rajpoot, N. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. *IEEE Transactions on Medical Imaging*, 35:1–1, 02 2016. doi: 10.1109/TMI.2016.2525803.

Tsai, K.-H. and Lin, H.-T. Learning from label proportions with consistency regularization. In *Asian Conference on Machine Learning*, pp. 513–528. PMLR, 2020.

Wang, X., Yan, Y., Tang, P., Bai, X., and Liu, W. Revisiting multiple instance neural networks. *Pattern Recognition*, 74:15–24, 2018.

Wojtusiak, J., Irvin, K., Birerdinc, A., and Baranova, A. V. Using published medical results and non-homogenous data in rule learning. In *2011 10th International Conference on Machine Learning and Applications and Workshops*, volume 2, pp. 84–89. IEEE, 2011.

Xu, J., Zhang, Z., Friedman, T., Liang, Y., and Van den Broeck, G. A semantic loss function for deep learning with symbolic knowledge. In *International conference on machine learning*, pp. 5502–5511. PMLR, 2018.

Yu, F. X., Choromanski, K., Kumar, S., Jebara, T., and Chang, S.-F. On learning from label proportions. *arXiv preprint arXiv:1402.5902*, 2014.

Zeng, Z. and Broeck, G. V. d. Collapsed inference for bayesian deep learning. *arXiv preprint arXiv:2306.09686*, 2023.

Zeng, Z. and Van den Broeck, G. Efficient search-based weighted model integration. *Proceedings of UAI*, 2019.

Zeng, Z., Morettin, P., Yan, F., Vergari, A., and Broeck, G. V. d. Scaling up hybrid probabilistic inference with logical and arithmetic constraints via message passing. In *Proceedings of the International Conference of Machine Learning (ICML)*, 2020a.

Zeng, Z., Morettin, P., Yan, F., Vergari, A., and Van den Broeck, G. Probabilistic inference with algebraic constraints: Theoretical limits and practical approximations. *Advances in Neural Information Processing Systems*, 33: 11564–11575, 2020b.

Zeng, Z., Morettin, P., Yan, F., Vergari, A., and Van den Broeck, G. Is parameter learning via weighted model integration tractable? In *Proceedings of the UAI Workshop on Tractable Probabilistic Modeling (TPM)*, jul 2021.

Zhou, Z.-H. A brief introduction to weakly supervised learning. *National science review*, 5(1):44–53, 2018.

Zupanc, K. and Davis, J. Estimating rule quality for knowledge base completion with the relationship between coverage assumption. In *Proceedings of the 2018 World Wide Web Conference*, pp. 1073–1081, 2018.

A. Related Work

Unified Approaches. Although not common, there exists some literature in regards to general approaches for weakly supervised learning. One example being the method proposed in Hüllermeier (2014), which provides a procedure that minimizes the empirical risk on "fuzzy" sets of data. The paper also establishes guarantees for model identification and instance level recognition.

Count Loss. The idea of computing the posterior of a bag is not a new concept for tackling weakly supervised learning—specifically for LLP. Many approaches have tried to approximate the bag posterior through averaging the instances in a bag Ardehaly & Culotta (2017); Tsai & Lin (2020). This computation is not exact and can be considered a heuristic estimate of our approach. However, it is certainly worth mentioning as they are motivated by the same principle.

LLP. Quadrianto et al. (2008) first introduced an exponential family based approach that used an estimation of mean for each class. Others seek to minimize "empirical proportion risk" or EPR as in Yu et al. (2014), which is centered around creating an instance level classifier that is able to reproduce the label proportions of each bag. As mentioned previously, more recent methods such as the work described in Ardehaly & Culotta (2017); Tsai & Lin (2020) use bag posterior approximation and neural based approaches. One such method is Proportion Loss (PL) (Tsai & Lin, 2020), which we contrast to our approach. This is computed by binary cross entropy between the averaged instance level probabilities and ground-truth bag proportion.

MIL. MIL finds some its earlier approaches with SVMs, which have been use quite prolifically and still remain one of the most common baselines. We start with MI-SVM/mi-SVM(Andrews et al., 2002) which are examples of transductive SVMs (Carbonneau et al., 2018) that seek a stable instance classification through repeated retraining iterations. MI-SVM is an example of an instance space method (Carbonneau et al., 2018), which identifies methods that classify instances as a preliminary step in the problem. This is in contrast to bag-space or embedded-space methods that omit the instance classification step. We then look toward Wang et al. (2018) which remains one of the hallmarks for the use of neural networks for Multi-Instance Learning. In Ilse et al. (2018), they utilized a similar approach but with Attention based mechanisms.

PU learning. Bekker & Davis (2020) groups PU Learning paradigms into three main classes: two step, biased, and class prior incorporation. Biased Learning techniques train a classifier on the entire dataset with the understanding that negative samples are subject to noise Bekker & Davis (2020). We will focus on a subset of biased learning techniques (Risk Estimators) as they are considered state of the art and relevant to us as baselines. The Unbiased Risk Estimator (uPU), which was originally proposed in du Plessis et al. (2014) and covered in Plessis et al. (2015), provides an alternative to the inefficiencies in manually biasing unlabeled data. And later, Non-negative Risk Estimator (nnPU) Kiryo et al. (2017) accounted for weaknesses in the unbiased risk estimator.

Neuro-Symbolic Losses. In this paper, we have dealt with a specific form of distributional constraint. Conversely, there has been a plethora of work exploring the integration of *hard* symbolic constraints into the learning of neural networks. This can take the form of enforcing a hard constraint (Ahmed et al., 2022b), whereby the network's predictions are guaranteed to satisfy the pre-specified constraints. Or it can take the form of a soft constraint (Xu et al., 2018; Manhaeve et al., 2018; Ahmed et al., 2021; 2022c;a; 2023a) whereby the network is trained with an additional loss term that penalizes the network for placing any probability mass on predictions that violate the constraint. While in this work we focus on discrete constraints defined over binary variables, there are existing work focusing on hybrid constraints defined over both discrete and continuous variables and their tractability (Belle et al., 2015b;a; Zeng et al., 2021; 2020b). The development of inference algorithms for such constraints and their applications such as Bayesian deep learning remain an active topic (Zeng & Van den Broeck, 2019; Kolb et al., 2019; Zeng et al., 2020a; Zeng & Broeck, 2023).

B. Tractable Computation of Count Probability

Algorithm 1 Count Probability $p(\sum_{i=1}^{k} \hat{y}_i = s)$

Input: A set of k log probabilities $\{t_i\}_{i=1}^{k}$ with $t_i := \log p(y_i = 1)$, the number of instances k, and a label sum s
Output: log probability $\log p(\sum_{i=1}^{k} \hat{y}_i = s)$ or a set of log probability $\{\log p(\sum_{i=1}^{k} \hat{y}_i = s)\}_{s=0}^{k}$

```
// A[i, m] = log p(Σ_{j=1}^{i} y_j = m) ∀i, m
Initialize an array A to be −Inf everywhere
A[0, 0] = 0    // p(Σ_{j=1}^{0} y_j = 0) = 1
Compute t'_i ← log1mexp(t_i) // log p(y_i = 0)
for i = 1 to k do
  for m = 0 to s do
    a_+ = A[i − 1, m − 1] + t_i
    a_− = A[i − 1, m] + t'_i
    A[i, m] = logsumexp(a_+, a_−)
  end for
end for
return A[k, s] or A[k, :]
```

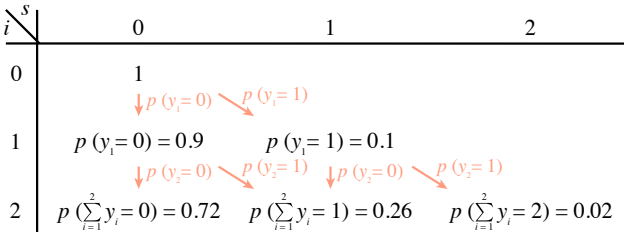


Figure 1: An example of how to compute the count probability in a dynamic programming manner. Assume that an instance-level classifier predicts three instances to have $p(y_1 = 1) = 0.1$, $p(y_2 = 1) = 0.2$, and $p(y_3 = 1) = 0.3$ respectively. The algorithm starts from the top-left cell and propagates the results down right. A cell has its probability $p(\sum_{j=0}^{i} y_j = s)$ computed by inputs from $p(\sum_{j=0}^{i-1} y_j = s)$ weighted by $p(y_i = 0)$, and $p(\sum_{j=0}^{i-1} y_j = s - 1)$ weighted by $p(y_i = 1)$ respectively, as indicated by the arrows.

We show that the count probability $p(\sum_{i=1}^{k} \hat{y}_i = s)$ can be computed in a dynamic programming manner. We provide an illustrative example of this computation in Figure 1. In practice, we implement this computation in log space for numeric stability which we summarized as Algorithm 1, where function `log1mexp` provides a numerically stable way to compute $\texttt{log1mexp}(x) = \log(1 - \exp(x))$ and function `logsumexp` a numerically stable way to compute $\texttt{logsumexp}(x, y) = \log(\exp(x) + \exp(y))$. Notice that since we show it is tractable to compute the set $\{p(\sum_{i=1}^{k} \hat{y}_i = s)\}_{s=0}^{k}$, for any two given label sum s_1 and s_2, a count probability $p(s_1 \leq \sum_i y_i) \leq s_2)$ where the count takes its form as an interval, can also be exactly and tractably computed. This implies that our tractable computation of count probabilities can potentially be leveraged by other count-based applications besides the three weakly supervised learning settings in the last section.

C. Additional Experiment Analysis

C.1. LLP

In addition to *Adult*, we experiment on *Magic Gamma Ray Telescope* with 6144 training samples where the task is to predict whether the electromagnetic shower is caused by primary gammas or not given information from the atmospheric Cherenkov gamma telescope (Dua & Graff, 2017).

We follow the experimental settings from Scott & Zhang (2020) where two settings are considered: one with label proportions uniformly on $[0, \frac{1}{2}]$ and the other uniformly on $[\frac{1}{2}, 1]$. Additionally, we experiment on a third setting with label proportions distributing uniformly on $[0, 1]$ which is not considered in Scott & Zhang (2020) but is the most natural setting since the label proportion is not biased toward either 0 or 1. We experiment on four bag sizes n with $n \in \{8, 32, 128, 512\}$.

Count loss (CL) denotes our proposed approach using the loss objective defined in Table 2 for LLP. We compare our approach with the method from Learning from (Scott & Zhang, 2020) or LMMCM and against Proportion Loss (PL)(Tsai & Lin, 2020).

Results and Discussion. We show our full results in Table 6. On almost every setting, our method showcases superior results against the baselines. This indicates CL is able to learn instance-level classification especially on settings where we have bags of small sizes, since bag-level information is closer to full supervision. We also empirically validate that techniques that approximate the bag posterior (PL) are less effective than optimizing the exact bag posterior with CL.

C.2. MIL

We first experiment on the MNIST dataset (LeCun, 1998) and follow the MIL experimental setting in Ilse et al. (2018): the training and test set bags are randomly sampled from the MNIST training and test set respectively; each bag can have images of digits from 0 to 9, and bags with digit 9 are labeled positive. Moreover, the dataset is constructed in a balanced way such that there is an equal amount of positively and negatively labeled bags as in Ilse et al. (2018). The task is to train a classifier that is able to predict bag labels; the more challenging task is to *discover key instances*, that is, to train a classifier that identifies images of digit 9. Following Ilse et al. (2018), we consider three settings that vary in bag generation process: in each setting, bags have their sizes generated from a normal distribution being $\mathcal{N}(10, 2), \mathcal{N}(50, 10), \mathcal{N}(100, 20)$ respectively. The number of bags in training set n is in $\{50, 100, 150, 200, 300, 400, 500\}$. Thus, we have $3 \times 7 = 21$ settings in total. Additionally, we introduce experimental analysis on *how performance of the learning methods would degrade as the number of bags and total samples in training set decreases*, by modulating the number of training bags n to be $\{10, 20, 30, 40\}$ and selecting bag sizes from $\mathcal{N}(5, 1)$ and $\mathcal{N}(10, 2)$.

We also experiment on the Colon Cancer dataset (Sirinukunwattana et al., 2016) to simulate a setting where bag instances are not independent. The dataset consists of 100 total H&E stained images, each of which contain images of cell nuclei that are classified as one of: epithelial, inflammatory, fibroblast, and miscellaneous. Each image represents a bag and instances are 27×27 patches extracted from the original image. A positively labeled bag or image is one that contains the epithelial nuclei.

For both datasets, we include the Attention and Gated Attention mechanism (Ilse et al., 2018) as baselines. We also use the MIL objective defined in Table 2.

Table 6: LLP results showing test AUC with standard deviation aggregated over 5 trials for each experimental setting. $*$ represents experiments that we ran with no early stopping. We highlight the highest test AUC. Full table for Table 3.

Dataset	**Dist**	**Method**	8	32	128	512
Adult	$[0, \frac{1}{2}]$	CL (Ours)	$\mathbf{0.8984 \pm 0.0013}$	$\mathbf{0.8848 \pm 0.0041}$	$\mathbf{0.8743 \pm 0.0052}$	$\mathbf{0.8703 \pm 0.0070}$
Adult	$[0, \frac{1}{2}]$	PL	0.8889 ± 0.0024	0.8782 ± 0.0036	$\mathbf{0.8743 \pm 0.0039}$	0.8678 ± 0.0085
Adult	$[0, \frac{1}{2}]$	LMMCM	0.8728 ± 0.0019	0.8693 ± 0.0047	0.8669 ± 0.0041	0.8674 ± 0.0040
Adult	$[\frac{1}{2}, 1]$	CL (Ours)	$\mathbf{0.8854 \pm 0.0022}$	$\mathbf{0.8738 \pm 0.0039}$	0.8675 ± 0.0043	$\mathbf{0.8607 \pm 0.0056}$
Adult	$[\frac{1}{2}, 1]$	PL	0.8781 ± 0.0038	0.8731 ± 0.0035	$\mathbf{0.8699 \pm 0.0057}$	0.8556 ± 0.0180
Adult	$[\frac{1}{2}, 1]$	LMMCM	0.8584 ± 0.0164	0.8644 ± 0.0052	0.8601 ± 0.0045	0.8500 ± 0.0186
Adult	$[0, 1]$	CL (Ours)	$\mathbf{0.8985 \pm 0.0010}$	$\mathbf{0.8891 \pm 0.0013}$	0.8871 ± 0.0021	$\mathbf{0.8790 \pm 0.0056}$
Adult	$[0, 1]$	PL	0.8884 ± 0.0030	0.8884 ± 0.0008	$\mathbf{0.8879 \pm 0.0025}$	0.8828 ± 0.0051
Adult	$[0, 1]$	LMMCM	0.8831 ± 0.0026	0.8819 ± 0.0006	0.8821 ± 0.0017	0.8786 ± 0.0052
Magic	$[0, \frac{1}{2}]$	CL (Ours)	$\mathbf{0.9088 \pm 0.0056}$	$\mathbf{0.8830 \pm 0.0097}$	$\mathbf{0.8926 \pm 0.0049}$	$*\mathbf{0.8864 \pm 0.0107}$
Magic	$[0, \frac{1}{2}]$	PL	0.8900 ± 0.0095	0.8510 ± 0.0032	0.8405 ± 0.0110	0.8332 ± 0.0149
Magic	$[0, \frac{1}{2}]$	LMMCM	0.8918 ± 0.0077	0.8799 ± 0.0113	0.8753 ± 0.0157	0.8734 ± 0.0092
Magic	$[\frac{1}{2}, 1]$	CL (Ours)	$\mathbf{0.9105 \pm 0.0020}$	$\mathbf{0.8980 \pm 0.0059}$	$\mathbf{0.8851 \pm 0.0255}$	$*\mathbf{0.8816 \pm 0.0083}$
Magic	$[\frac{1}{2}, 1]$	PL	0.9066 ± 0.0016	0.8818 ± 0.0108	0.8769 ± 0.0101	0.8429 ± 0.0443
Magic	$[\frac{1}{2}, 1]$	LMMCM	0.8911 ± 0.0083	0.8790 ± 0.0091	0.8684 ± 0.0046	0.8567 ± 0.0292
Magic	$[0, 1]$	CL (Ours)	$\mathbf{0.9173 \pm 0.0018}$	$\mathbf{0.9102 \pm 0.0057}$	$\mathbf{0.9146 \pm 0.0051}$	$\mathbf{0.9088 \pm 0.0039}$
Magic	$[0, 1]$	PL	0.9039 ± 0.0029	0.8870 ± 0.0037	0.9002 ± 0.0092	0.8807 ± 0.0200
Magic	$[0, 1]$	LMMCM	0.9070 ± 0.0026	0.9048 ± 0.0058	0.9113 ± 0.0058	0.8934 ± 0.0097

Table 7: MIL: We report mean test accuracy, AUC, F1, precision, and recall averaged over 5 runs with std. error on the Colon Cancer dataset. Highlighted are the highest mean values for each metric.

Method	**Accuracy**	**AUC**	**F1**	**Precision**	**Recall**
Gated Attention	0.909 ± 0.014	0.908 ± 0.013	0.886 ± 0.021	0.916 ± 0.020	0.879 ± 0.020
Attention	0.893 ± 0.015	0.890 ± 0.008	0.876 ± 0.017	0.908 ± 0.016	0.879 ± 0.018
CL (Ours)	$\mathbf{0.915 \pm 0.008}$	$\mathbf{0.912 \pm 0.010}$	$\mathbf{0.903 \pm 0.010}$	$\mathbf{0.936 \pm 0.014}$	$\mathbf{0.898 \pm 0.007}$

Results and Discussion. For the MNIST experiments, CL is able to outperform all other baselines or exhibit highly comparable performance for bag-level predictions as shown in Table 4. A more interesting setting is to compare how robust the learning methods are if the number of training bags decrease. Wang et al. (2018) claims that instance-level classifiers tend to lose against embedding based methods. However, we show in our experiment that this is actually not true in all cases as seen in Figure 2. While Attention and Gated Attention are based on embedding, they suffer from a more severe drop in predictive performance than CL when the number of training bags drops from 40 to 10; our method shows great robustness and consistently outperforms all baselines. The rationale we provide is that with lower number of training instances, we need more supervision over the limited samples we have. Our constraint provides this additional supervision, which accounts for the difference in performance.

We provide an exemplary investigation in Figure 3 to show that our approach learns effectively and delivers accurate instance-level predictions under bag-level supervision. We can see that even though the classifier is trained on feedback about whether a bag contains the digit 9 or not, it accurately discover all images of digit 9.

Our experimental results on the Colon Cancer dataset are shown in Table 7. We show that both our proposed objectives are able to consistently outperform baseline methods on all metrics. Interestingly, we do not expect CL perform well when instances in a bag are dependent; however, the results indicate that our count loss is robust to these settings.

C.3. PU

We experiment on dataset MNIST and CIFAR-10 (Krizhevsky & Hinton, 2009), following the four simulated settings from Garg et al. (2021): 1) Binarized MNIST: the training set consist of images of digits $0-9$ and images with digits in range $[0, 4]$ is defined as positive instances while others as negative; 2) MNIST17: the training set consist of images of digits 1 and 7 and images with digits 1 is defined as positive while 7 as negative; 3) Binarized CIFAR: the training set consist of images from ten classes and images from the first five classes is defined as positive instances

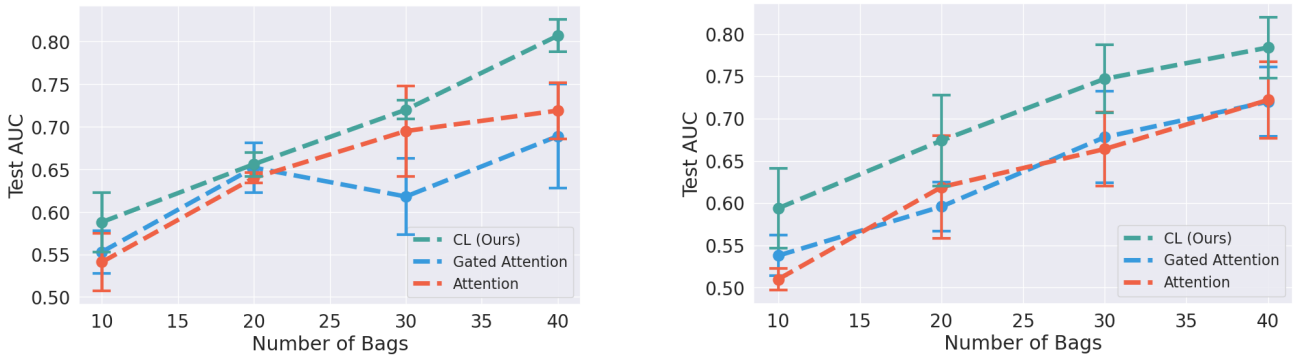


Figure 2: MIL MNIST dataset experiments with decreased training bags and lower bag size. Left: bag lengths samples from $\mathcal{N}(10, 2)$; Right: bag lengths sampled from $\mathcal{N}(5, 1)$. We plot the mean test AUC (aggregated over 3 trials) with standard error for 4 bag sizes. Best viewed in color.

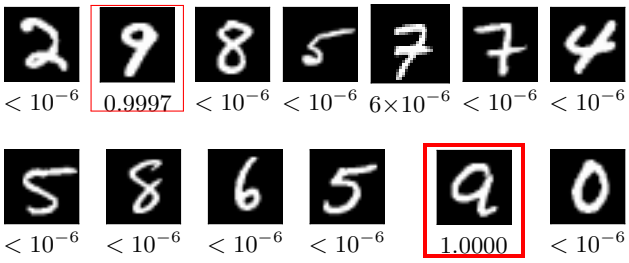


Figure 3: A test bag from our MIL experiments, where we set only the digit 9 as a positive instance. Bag sizes are distributed as $\mathcal{N}(10, 2)$, and we use 500 training bags (see Table 4 for details). Highlighted in red are digits identified to be positive with corresponding probability beneath.

while others as negative; 4) CIFAR Cat vs. Dog: the training set consist of images of cats and dogs and images of cats are defined as positive while dogs as negative. The mixture proportion is 0.5 in all experiments. The performance is evaluated using the accuracy on a test set of unlabeled data.

As shown in Table 2, we propose two objectives for PU learning. Our first objective is denoted by CL whereas the second approach is denoted by CL-expect. We compare against Conditional Value Ignoring Risk approach (CVIR) (Garg et al., 2021), nnPU (Kiryo et al., 2017), and uPU (Plessis et al., 2015).

Results and Discussion. Complete accuracy results are presented in Table 8 where we can see that our proposed methods perform better than baselines on 3 out of the 4 simulated PU learning settings. CL-expect builds off a similar "exactly-k" count approach, which we have shown to work well in the label proportion setting. The more interesting results are from CL where we fully leverage the information from a distribution as supervision instead of simply using the expectation. We think of this as applying a loss on each count weighted by their probabilities from the binomial distribution. We provide further evidence that our proposed count loss effectively guides the classifier towards predicting a binomial distribution as shown in Figure 4: we plot the count distributions predicted by CL and CVIR as well as the ground-truth binomial distribution. We can see that the CL is able to generate a count distribution close to the ground truth while the baseline approach does not.

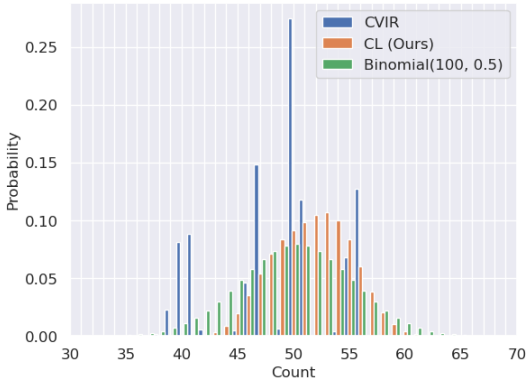


Figure 4: MNIST17 setting for PU Learning: We compute the average discrete distribution for CL and CVIR, a strong baseline over 5 test bags, each of which contains 100 instances. A ground truth binomial distribution of counts is also shown.

D. Proofs

Proposition 3.2 With SCAR assumption and a class prior, the mixture proportion β can be estimated from dataset $\mathcal{D}$.

Proof. Given a class prior $p(y = 1)$ denoted by α, the label frequency $p(\tilde{y} = 1 \mid y = 1)$ denoted by c can be obtained

Table 8: PU Learning: We report accuracy and standard deviation on a test set of unlabeled data, which is aggregated over 3 runs. We highlight the method with the highest mean accuracy. Results from CVIR, nnPU, and uPU are aggregated over 10 epochs, as defined in Garg et al. (2021), while we choose the single best epoch based on validation for our approaches. This is the full table for Table 5.

Dataset	Network	CL-expect (Ours)	CL (Ours)	CVIR	nnPU	nPU
Binarized MNIST	MLP	95.9 ± 0.15	$\mathbf{96.4 \pm 0.01}$	96.3 ± 0.07	96.1 ± 0.14	95.2 ± 0.19
MNIST17	MLP	98.7 ± 0.17	$\mathbf{99.0 \pm 0.19}$	98.7 ± 0.09	98.4 ± 0.20	98.4 ± 0.09
Binarized CIFAR	ResNet	79.2 ± 0.27	80.1 ± 0.34	$\mathbf{82.3 \pm 0.18}$	77.2 ± 1.03	76.7 ± 0.74
CIFAR Cat vs. Dog	ResNet	$\mathbf{76.5 \pm 1.86}$	74.8 ± 1.64	73.3 ± 0.94	71.8 ± 0.33	68.8 ± 0.53

by

$$\begin{aligned} c &= p(\tilde{y} = 1 \mid y = 1) \\ &= \frac{p(\tilde{y} = 1, y = 1)}{p(y = 1)} \\ &= \frac{p(\tilde{y} = 1)}{p(y = 1)}, \quad \textit{(by the definition of PU learning)} \end{aligned}$$

that is, $c = p(\tilde{y} = 1)/\alpha$. Notice that $p(\tilde{y} = 1)$ can be estimated from the dataset $\mathcal{D}$ by counting the proportion of the labeled instances. Further, we can obtain the mixture proportion as below,

$$\begin{aligned} \beta &= p(y = 1 \mid \tilde{y} = 0) \\ &= \frac{p(y = 1, \tilde{y} = 0)}{p(\tilde{y} = 0)} \\ &= \frac{p(y = 1)p(\tilde{y} = 0 \mid y = 1)}{1 - p(\tilde{y} = 1)} \\ &= \frac{p(y = 1)(1 - p(\tilde{y} = 1 \mid y = 1))}{1 - p(\tilde{y} = 1)} \\ &= \frac{\alpha(1 - c)}{1 - p(\tilde{y} = 1)}. \end{aligned}$$

□

Lemma D.1. *Let R_{llp} be our risk estimator defined over $p(\boldsymbol{x}, \tilde{y})$ as $R_{llp}(f) = \frac{1}{k(k+1)} \mathbb{E}_{p(\boldsymbol{x}^k, \tilde{y})}[\ell(f(\boldsymbol{x}), \boldsymbol{y})]$. Following the assumptions in Section 3.1 from Kobayashi et al. (2022), our proposed method is risk-consistent.*

Proof. In Kobayashi et al. (2022), it is shown that the risk R in classical multi-class classification as shown in Section 2.1 can be reduced to a risk R_{rc} over $p(\boldsymbol{x}^k, \tilde{y}^k)$ as shown in Equation 1 in Kobayashi et al. (2022) under certain assumptions.

Consider binary classification and follow our notations, we rewrite the Equation 1 in Kobayashi et al. (2022) as below,

$$R_{rc}(f) = \frac{1}{k(k+1)} \mathbb{E}_{p(\boldsymbol{x}^k, \tilde{y})} \sum_{\boldsymbol{y} \in \mathcal{Y}^k} \frac{\prod_{j=1}^k p(y_j \mid \boldsymbol{x}_j)}{\sum_{\boldsymbol{y}' \in \mathcal{Y}^k, \sum_j y'_j = \tilde{y}} \prod_{j=1}^k p(y'_j \mid \boldsymbol{x}_j)} \ell(f(\boldsymbol{x}^k), \boldsymbol{y})$$

We notice that the weight term attached to the loss can be further rewritten as a constrained probability as follows,

$$\frac{\prod_{j=1}^k p(y_j \mid \boldsymbol{x}_j)}{\sum_{\boldsymbol{y}' \in \mathcal{Y}^k, \sum_j y'_j = \tilde{y}} \prod_{j=1}^k p(y'_j \mid \boldsymbol{x}_j)} = p(\boldsymbol{y} \mid \sum_{j=1}^k y_j = \tilde{y}, \boldsymbol{x}^k)$$

This allows us to further rewrite the risk R_{rc} with likelihood loss being $\ell(f(\boldsymbol{x}^k), \boldsymbol{y}) = -p(\sum_{j=1}^k y_j = k\tilde{y} \mid \boldsymbol{x}^k)$:

$$\begin{aligned} R_{rc}(f) &= \frac{1}{k(k+1)} \mathbb{E}_{p(\boldsymbol{x}^k, \tilde{y})} \left[- \sum_{\boldsymbol{y} \in \mathcal{Y}^k} p(\boldsymbol{y} \mid \sum_{j=1}^k y_j = k\tilde{y}, \boldsymbol{x}^k) p(\sum_{j=1}^k y_j = k\tilde{y} \mid \boldsymbol{x}^k) \right] \\ &= \frac{1}{k(k+1)} \mathbb{E}_{p(\boldsymbol{x}^k, \tilde{y})} \left[- \sum_{\boldsymbol{y} \in \mathcal{Y}^k} p(\boldsymbol{y}, \sum_{j=1}^k y_j = k\tilde{y} \mid \boldsymbol{x}^k) \right] \\ &= \frac{1}{k(k+1)} \mathbb{E}_{p(\boldsymbol{x}^k, \tilde{y})} \left[-p(\sum_{j=1}^k y_j = k\tilde{y} \mid \boldsymbol{x}^k) \right] \\ &= \frac{1}{k(k+1)} \mathbb{E}_{p(\boldsymbol{x}^k, \tilde{y})} [\ell(f(\boldsymbol{x}^k), \boldsymbol{y})] = R_{llp}(f) \end{aligned}$$

The last few lines follow from the definition of conditional probabilities. This shows that the risk $R_{rc}(f) = R_{llp}(f)$, meaning that the reduction from risk $R_{rc}(f)$ to the classical risk $R(f)$ in Kobayashi et al. (2022) is applicable to our risk estimator R_{llp}, which proves that our learning method is risk-consistent. □

Proposition D.2. Assume that the loss function $\ell(f(\boldsymbol{x}), y)$ is ρ-Lipschitz with respect to $f(\boldsymbol{x})$ for any $y \in \mathcal{Y}$ bounded by some constant. Let f_{llp} be the hypothesis that minimizes the empirical risk, and f^*_{llp} is the hypothesis that minimizes the true risk, then f_{llp} converges to f^*_{llp} as $m \to \infty$.

Proof. This claim immediately follows Lemma D.1, where we shows that $R_{rc}(f) = R_{llp}(f)$. Therefore, it holds that $R_{llp}(\hat{f}) - R_{llp}(f^*) = R_{(sc)}(\hat{f}) - R_{(sc)}(f^*)$, where the latter term, an always positive term, is shown in Theorem 3.1 in Kobayashi et al. (2022) that it converges to 0 at rate $\sqrt{m}$. □

Proposition 3.1 *The count probability $p(\sum_{i=1}^{k} \hat{y}_i = s)$ of sampling k prediction variables with summation being s from an unconstrained distribution $p(\boldsymbol{y}) = \prod_{i=1}^{k} p(\hat{y}_i)$ can be computed exactly in time $\mathcal{O}(ks)$. Moreover, the set $\{p(\sum_{i=1}^{k} \hat{y}_i = s)\}_{s=0}^{k}$ can also be computed in time $\mathcal{O}(k^2)$.*

Proof. The claim that $p(\sum_{i=1}^{k} \hat{y}_i = s)$ can be computed exactly in time $\mathcal{O}(ks)$ follows immediately from Proposition 1 in Ahmed et al. (2023b): in Ahmed et al. (2023b), the unconstrained distribution is a factorized distribution obtained from k outputs from a single neural network model while in our case, the unconstrained distribution $p(\boldsymbol{y})$ is obtained from applying a classifier that gives a single output $p(y_i)$ on k inputs; the constructive proof of Proposition 1 in Ahmed et al. (2023b) still applies in our case. Moreover, the computation of $p(\sum_{i=1}^{k} \hat{y}_i = k)$ is done in a dynamic programming manner in the sense that for any $s < k$, $p(\sum_{i=1}^{k} \hat{y}_i = s)$ is an intermediate result for computing $p(\sum_{i=1}^{k} \hat{y}_i = k)$. By caching the intermediate result, the set $\{p(\sum_{i=1}^{k} \hat{y}_i = s)\}_{s=0}^{k}$ can be obtained by the time $p(\sum_{i=1}^{k} \hat{y}_i = k)$ is computed, which finishes our proof. □

E. Experimental Setup Details

In this section, we will provide relevant training details as it relates to each of our settings including hyperparameters and dataset details.

Table 9: Illustration of Adult and Magic datasets showing the number of training bags for each bag size. Note that we test on the same number of instances in all variations of bag size for both experiments: 16280 for Adult and 3804 for Magic. The breakdown of training bags is the same across all distributions of label proportion as well, i.e., $[0, \frac{1}{2}]$, $[\frac{1}{2}, 1]$, $[0, 1]$.

Bag Size	Training Bags Adult	Training Bags Magic
8	1024	768
32	256	192
128	64	48
512	16	12

E.1. Label Proportion

E.1.1. Adult Dataset

Hyperparameters. We use a learning rate of 0.00001 with the Adam Optimizer and $\beta_1 = 0.9, \beta_2 = 0.999$. The weight decay value is set to 0.001. We also notice that adding in $L1$ regularization of 0.001 improved the performance of our method. We train for 10000 epochs and use a set number of warm epochs for our experiments. All parameters were obtained by using a holdout of 12.5% of training data for validation on the $[0, 1]$ uniform setting. The network shown in Table 10 was also obtained grid search on this same validation set.

Table 10: Network used for Adult dataset in LLP Experiments.

Layer	Type
1	fc - 2048 + ReLU
2	fc - 64 + ReLU
3	fc - 1 + logsigmoid

Training Procedure. For CL, we use the parameters and network described in the previous paragraph and early stopping criterion based on validation loss from a held out validation set (12.5% of training data). For PL, we use the parameters and network except that we do not use $L1$ as we found this improves performance. We also use an early stopping criterion based on validation loss from a held out validation set (12.5% of training data).

Computing Resources. Trained on Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHzU and AMD EPYC 7313P 16-Core Processor CPU.

E.1.2. Magic Dataset

Hyperparameters. We use a learning rate of 0.0001 with the Adam Optimizer and $\beta_1 = 0.9, \beta_2 = 0.999$. The weight decay value is set to 0.001. We also notice that adding in $L1$ regularization of 0.001 improved the performance of our method. We train for 10000 epochs and use a set number of warm epochs for our experiments. All parameters were obtained by using a holdout of 12.5% of training data for validation on the $[0, 1]$ uniform setting. The network shown in Table 11 was also obtained grid search on this same validation set.

Table 11: Network used for Magic dataset in LLP Experiments.

Layer	Type
1	fc - 2048 + ReLU
2	fc - 1 + logsigmoid

Training Procedure. For CL, we use the parameters and network described in the previous paragraph and early stopping criterion based on validation loss from a held out validation set (12.5% of training data). For PL, we use the

parameters and network except that we do not use $L1$ regularization as we found this improves performance. We also use an early stopping criterion based on validation loss from a held out validation set (12.5% of training data). As shown in Table 6, there are two instances where we reran our results with no validation set. In these experiments, we only use 87.5% of training data and ran for a fixed number of epochs: 2000. This is because with only one validation bag, we can find ourselves with some instability in the training procedure. Note that PL did not benefit from rerunning with this method.

Computing Resources. Trained on Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHzU and AMD EPYC 7313P 16-Core Processor CPU.

E.2. Multi-Instance Learning

E.2.1. MNIST-BAGS

Hyperparameters. All of our hyperparameters derive from Ilse et al. (2018). This includes using the Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.999$, a learning rate of 0.0005, weight decay of 0.0001, and max epochs of 200. For the main experiment, we use a validation holdout of 20% to find a class weight for balancing the loss on positive bags versus negative bags. (We omit this step for our limited data experiments.)

Table 12: Network used for all MNIST experiments in MIL settings. Derived from the same network shown in Ilse et al. (2018).

Layer	Type
1	conv(5, 1, 0) - 20 + ReLU
2	maxpool(2, 2)
3	conv(5, 1, 0) - 50 + ReLU
4	maxpool(2, 2)
5	fc-500 + ReLU
6	fc-1 + logsigmoid

Training Procedure. For CL, we train on all the training data for the maximum number of iterations: 200. We also use all of the hyperparameters described in the last paragraph and Ilse et al. (2018). Because we were unable to reproduce the values in Ilse et al. (2018) for the Attention and Gated Attention mechanisms, we reran their experiments with our own implementation. To try and reproduce their results, we follow their optimization procedure. Specifically, we use a holdout of training data (20%) and validation loss + error for early stopping. We found that doing so provided the best values for Attention and Gated Attention.

Instance Pooling. To pool together instance level classification at the final stage, there are several operations that have been considered in the literature. Some include using the max and mean operator (Wang et al., 2018). We propose a new method based on our constraint. We compute the relevant probabilities defined in 3 for the MIL setting. More specifically, we compute the probability that a bag has at least one positive instance. We then round the probability of at least one positive instance to obtain our bag level classification.

Computing Resources. Trained on AMD EPYC 7313P 16-Core Processor CPU.

E.2.2. COLON CANCER DATASET

Hyperparameters. We derive our set of hyperparameters from Ilse et al. (2018). We use the Adam optimizer for all experiments with $\beta_1 = 0.9, \beta_2 = 0.999$. This includes weight decay of 0.0005, learning rate of 0.0001, and a maximum of 100 epochs.

Table 13: MIL: Network used for CL in colon cancer dataset. Derived from the same network shown in Ilse et al. (2018).

Layer	Type
1	conv(4, 1, 0) - 36 + ReLU
2	maxpool(2, 2)
3	conv(3, 1, 0) - 48 + ReLU
4	maxpool(2, 2)
5	fc-512 + ReLU
6	dropout
7	fc - 512 + ReLU
8	dropout
9	fc-2 + logsigmoid

Training Procedure. We perform 10-fold cross-validation and average the mean value of each metric over 5 seeds. For CL, we do not use early stopping and train on all data for the maximum number of epochs using the hyperparameters mentioned in the previous paragraph. For our baselines, Attention and Gated-Attention, we use the same hyperparameters as mentioned above. However, we follow the optimization procedure detailed in Ilse et al. (2018) to give try and reproduce the results given in the paper. This involves using a held out validation set for early stopping with validation loss + error as the stopping criteria. For this experiment, this validation set is assumed to be the size of 1 fold or one-ninth of the training data. (We find that including early stopping helps increase performance for both baselines.)

Computing Resources. Trained on NVIDIA RTX A6000 GPU.

E.3. PU Learning

E.3.1. MNIST Dataset

Table 14: Network used for MNIST data in PU Learning experiments. Resembles the network in Garg et al. (2021) except we replace the last layer with a single output and logsigmoid instead of softmax.

Layer	Type
1	fc - 5000 + ReLU
2	fc - 5000 + ReLU
3	fc - 50 + ReLU
4	fc-1 + logsigmoid

Hyperparameters. We fix weight decay to be 0.0005 and Adam optimizer for all experiments with $\beta_1 = 0.9, \beta_2 = 0.999$. We use a learning rate of 0.0001 and train for a maximum of 2000 epochs in all experiments for both CL and CL-expect. We use a validation set with size equal to 10% of training data in order to weigh the loss on positive data versus loss on unlabeled data.

Training Procedure. For MNIST dataset experiments, we use a fully connected multi-layer perceptron (MLP) defined in Table 14. We train CL and CL-expect with the hyperparameters defined in the previous paragraph. Furthermore, we use a held out validation set, equivalent to 10% of training data, for early stopping. While as results in Garg et al. (2021) are aggregated over 10 epochs, we choose to pick a single epoch based on our early stopping as this makes the most sense for our optimization technique.

Computing Resources. Trained on a singular NVIDIA RTX 2080-Ti GPU.

Table 15: Table taken almost directly from Garg et al. (2021). Table shows the break down of the various simulated PU datasets that we train on.

Dataset	Simulated PU Dataset	P vs N	Training		Test
			Positive	Unlabeled	Unlabeled
CIFAR	Binarized CIFAR	$[0-4]$ vs. $[5-9]$	12500	12500	2500
	CIFAR Cat vs. Dog	3 vs. 5	3000	3000	500
MNIST	Binarized MNIST	$[0-4]$ vs. $[5-9]$	15000	15000	2500
	MNIST-17	1 vs. 7	3000	3000	500

E.3.2. CIFAR Dataset.

Hyperparameters. We fix weight decay to be 0.0005 and Adam optimizer for all experiments with $\beta_1 = 0.9, \beta_2 = 0.999$. We use a learning rate of 0.0001 for all experiments except for CL-expect in the CIFAR Cat vs. Dog setting where we use 0.001. We use a validation set with size equal to 10% of training data in order to weigh the loss on positive data versus loss on unlabeled data.

Training Procedure. We use a ResNet-18 architecture for all CIFAR experiments. We train CL and CL-expect with the hyperparameters defined in the previous paragraph. Furthermore, we use a held out validation set, equivalent to 10% of training data, for early stopping. While as results in Garg et al. (2021) are aggregated over 10 epochs, we choose to pick a single epoch as this makes the most sense for our optimization technique.

Computing Resources. Trained on a singular NVIDIA 2080-Ti GPU.

E.3.3. Early Stopping

The early stopping procedure that we used in our experiments was a bit unique. Using our holdout of validation data, we do early stopping using the proximity to the class prior and validation loss to break ties. We can imagine that if we perfectly identify all positive and unlabeled samples and then calculate accuracy against the actually provided labels, we would get an accuracy equivalent to the class prior. This is because all the positive samples in the unlabeled set would be labeled incorrect.