Appendices A. Pseudo-Code of FASTMIPL

Algorithm A1 is the complete procedure of FASTMIPL. First, the algorithm uniformly initializes the model parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ (Step 1). In each epoch, the training set is divided into multiple mini-batches (Step 3). The subsequent step involves calculating the unbiased estimator of the ELBO on batch of training examples, and updating parameters $\boldsymbol{\theta}$, $\boldsymbol{\phi}$ via mini-batch gradient descent (Step 4-6). For an unseen multi-instance bag, the predicted label is the category corresponding to the highest prediction probability (Step 7-8).

Algorithm A1: The FASTMIPL Algorithm

Input:
$\mathcal{D}$: the MIPL training set $\{(\boldsymbol{X}_i, \mathcal{S}_i) \mid 1 \leq i \leq m\}$
T : the number of training epochs
$\boldsymbol{X}_\star$: the unseen multi-instance bag with $n_\star$ instances
Parameter: model parameters $\boldsymbol{\theta}$, variational parameters $\boldsymbol{\phi}$
Output:
$y_\star$: the predicted label for $\boldsymbol{X}_\star$
Process:

1: Initialize model parameters $\boldsymbol{\theta}$, variational parameters $\boldsymbol{\phi}$;
2: **for** $t = 1$ to T **do**
3: Sample B mini-batches from training set $\mathcal{D}$;
4: Compute unbiased estimator of the ELBO on B by Equation (6) and Equation (7);
5: Update parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ via mini-batch gradient descent.
6: **end for**
7: Calculatethe the predefined bag embeddings $\boldsymbol{f}_{\boldsymbol{\gamma}}(\boldsymbol{X}_\star)$ according to Equation (1) and Equation (2);
8: **return** $y_\star$ according to Equation (9)

Appendices B. Effectiveness Comparison with on Benchmark Datasets

Given that MIL and PLL algorithms cannot natively address MIPL problems, we employ the Mean-based approach and MaxMin strategy as transformation techniques to evaluate the effectiveness of PLL algorithms within MIPL tasks. Additionally, we utilize the One-vs-Rest (OvR) decomposition strategy to adapt MIL algorithms for handling partial-label data.

Table B1 represents the effectiveness comparison between FASTMIPL and comparative PLL algorithms. We involve two types of PLL algorithms: the deep-learning-based approach with liniear classifiers (i.e., PRODEN (Lv et al. 2020), RC (Feng et al. 2020), LWS (Wen et al. 2021) and CAVL (Zhang et al. 2022a)) and the feature-aware disambiguation algorithm (PL-AGGD (Wang, Li, and Zhang 2019)). FASTMIPL consistently surpasses PLL algorithms in all cases across four benchmark datasets. There are two applied data transformation techniques to address the applicability of PLL algorithms on multi-instance structured data, named the Mean-based approach and the MaxMin strategy. The Mean-based approach calculates the feature representation at the bag-level by obtaining the average of the feature values of all instances in a data bag. The MaxMin strategy extracts the maximum and minimum instance-level feature values in a data bag and concatenates them together to represent data bag-level features.

Algorithm	r	MNIST	FMNIST	Birdsong	SIVAL
FASTMIPL	1	**.999±.002**	**.911±.022**	**.797±.024**	**.779±.030**
	2	**.998±.004**	**.901±.027**	**.792±.021**	**.708±.026**
	3	**.975±.074**	**.816±.071**	**.772±.022**	**.615±.031**
Mean					
PRODEN	1	.605±.023	.696±.051	.296±.014	.219±.014
	2	.481±.036	.573±.026	.272±.019	.185±.013
	3	.283±.028	.345±.027	.210±.023	.166±.017
RC	1	.658±.031	.753±.049	.362±.015	.279±.011
	2	.598±.033	.648±.030	.335±.011	.258±.017
	3	.391±.037	.408±.044	.298±.016	.237±.020
LWS	1	.462±.051	.726±.031	.265±.010	.240±.014
	2	.209±.028	.720±.025	.254±.012	.223±.008
	3	.204±.013	.577±.098	.205±.016	.194±.026
CAVL	1	.597±.078	.727±.057	.370±.013	.260±.014
	2	.412±.039	.586±.035	.334±.010	.216±.011
	3	.315±.020	.352±.036	.313±.017	.175±.020
PL-AGGD	1	.670±.026	.743±.026	.354±.019	.355±.014
	2	.597±.037	.678±.020	.313±.012	.314±.018
	3	.381±.032	.474±.057	.296±.015	.286±.019
MaxMin					
PRODEN	1	.508±.025	.426±.045	.389±.013	.316±.019
	2	.401±.037	.375±.043	.356±.014	.285±.023
	3	.345±.049	.309±.056	.336±.012	.248±.020
RC	1	.518±.033	.731±.027	.390±.015	.306±.023
	2	.469±.035	.665±.027	.371±.013	.289±.021
	3	.380±.048	.390±.058	.363±.010	.267±.019
LWS	1	.241±.042	.535±.049	.225±.038	.289±.017
	2	.238±.048	.404±.040	.205±.035	.271±.015
	3	.218±.017	.318±.064	.216±.029	.245±.021
CAVL	1	.480±.030	.543±.015	.354±.015	.251±.023
	2	.387±.027	.263±.038	.235±.003	.216±.011
	3	.289±.032	.283±.024	.195±.015	.175±.020
PL-AGGD	1	.529±.035	.397±.011	.384±.013	.399±.030
	2	.440±.020	.373±.036	.373±.025	.363±.031
	3	.321±.039	.327±.028	.345±.012	.329±.024

Table B1: The classification accuracy (mean±std) of FASTMIPL and comparative PLL algorithms on benchmark datasets with the varying numbers of false positive labels ($r \in \{1, 2, 3\}$).

Algorithm	MNIST	FMNIST	Birdsong	SIVAL
FASTMIPL	**.999±.002**	**.911±.022**	**.797±.024**	**.779±.030**
MIVAE	.694±.241	.601±.193	.140±.201	.101±.153
ATTEN	.502±.018	.409±.079	.133±.019	.099±.025
ATTEN-GATE	.499±.098	.354±.109	.160±.028	.114±.032
LOSS-ATTEN	.843±.059	.774±.061	.537±.021	.319±.037

Table B2: The classification accuracy (mean±std) of FASTMIPL and comparative MIL algorithms on benchmark datasets with one false positive labels ($r = 1$).

Table B2 summarizes the effectiveness comparison between FASTMIPL and comparative MIL algorithms on benchmark datasets with one false positive label ($r = 1$).

We consider two types of MIL algorithms, containing a variational autoencoder (VAE)-based algorithm (MIVAE (Zhang 2021)) and three attention-based algorithms, containing ATTEN (Ilse, Tomczak, and Welling 2018), ATTEN-GATE (Ilse, Tomczak, and Welling 2018) and LOSS-ATTEN (Shi et al. 2020)). FASTMIPL presents statistically effectiveness advantages compared to four MIL algorithms. The One vs. Rest (OvR) decomposition strategy has been applied to comparative MIL algorithms, excepted the multi-class MIL algorithm LOSS-ATTEN, to address the adaptation problem of several candidate labels in complex MIPL scenarios. The strategy aims to assign each label from the candidate set to the bag, and matches each multi-instance bags to a singular bag-level label. There are c ($c \in \{1, 2, \ldots, k\}$) binary-class classifiers in the MIL algorithm adopted OvR decomposition strategy, meaning that the c-th classifier focuses on the c-th label, and treats c-th label and other labels as 1 (positive) and 0 (negative) respectively.

The effectiveness of PLL and MIL algorithms diminishes as the complexity of the benchmark datasets increases, as indicated by the decline in prediction effectiveness from the MNIST-MIPL to SIVAL-MIPL dataset. This effectiveness variation highlights the applicability limitations of these algorithms in complex data.

Appendices C. Effectiveness Comparison on the Real-world Dataset

Table C1 presents the effectiveness comparison between FASTMIPL and PLL algorithms on the real-world dataset.

Algorithm	C-Row	C-SBN	C-KMeans	C-SIFT
FASTMIPL	**.487±.038**	**.573±.031**	**.573±.013**	**.526±.029**
Mean				
PRODEN	.365±.009	.392±.008	.233±.018	.334±.029
RC	.214±.011	.242±.012	.226±.009	.209±.007
LWS	.291±.010	.310±.006	.237±.008	.270±.007
CAVL	.312±.043	.364±.066	.286±.062	.329±.033
PL-AGGD	.412±.008	.480±.005	.358±.008	.363±.012
MaxMin				
PRODEN	.401±.007	.447±.011	.265±.027	.291±.011
RC	.227±.012	.338±.010	.208±.007	.246±.008
LWS	.299±.008	.382±.009	.247±.005	.230±.007
CAVL	.368±.054	.503±.025	.311±.038	.274±.018
PL-AGGD	.460±.008	.524±.008	.434±.009	.285±.009

Table C1: The classification accuracy (mean±std) of FASTMIPL and comparative PLL algorithms on the real-world dataset.

Obviously, FASTMIPL surpasses comparative algorithms in all cases. Additionaly, the MaxMin strategy outperforms the Mean-based approach among the PLL algorithms. This discrepancy likely stems from the Mean-based approach's failure to consider the significant differences between the tissue cells and the background on the CRC-MIPL dataset. Notably, in the CRC-MIPL dataset, which presents complex scenarios, FASTMIPL introduces random effects component to capture the intricate heterogeneity for instance bags. This approach significantly enhances the effectiveness of the predictive task of identifying true label from candidate labels within bags.

Appendices D. Time Consumption Comparison

Tables D1 and Table D2 present the time consumption values for both benchmark and real-world datasets. FASTMIPL outperforms the comparative algorithms by an order of magnitude in terms of time efficiency. Specifically, FASTMIPL reduces time consumption by up to 35 times compared to ELIMIPL and by up to 38 times compared to DEMIPL. This significant improvement in efficiency is due to the fact that FASTMIPL optimizes Equation (6) only once per epoch, whereas attention-based MIPL algorithms optimize the loss function multiple times, corresponding to the number of bags in each epoch.

Algorithm	r	MNIST-MIPL	FMNIST-MIPL	Birdsong-MIPL	SIVAL-MIPL
FASTMIPL	1	**16.909**	**16.783**	**70.804**	**110.180**
	2	**16.769**	**17.096**	**70.966**	**101.345**
	3	**16.894**	**16.519**	**83.552**	**109.472**
ELIMIPL	1	579.413	553.961	709.847	823.999
	2	569.412	571.749	702.397	792.799
	3	565.581	580.469	714.688	816.927
DEMIPL	1	561.336	580.774	679.021	1686.970
	2	570.101	576.258	693.691	1640.633
	3	561.268	572.451	681.662	1656.847

Table D1: The time consumption (seconds) of FASTMIPL and comparative MIPL algorithms on the benchmark datasets with the varying numbers of false positive labels ($r \in \{1, 2, 3\}$).

Algorithm	C-Row	C-SBN	C-KMeans	C-SIFT
FASTMIPL	**394.607**	**481.659**	**225.845**	**204.472**
ELIMIPL	4060.509	3996.979	4028.986	4271.141
DEMIPL	7889.763	7737.639	8102.833	7964.447

Table D2: The time consumption (seconds) of FASTMIPL and comparative MIPL algorithms on the real-world datasets.