Rethinking Learning from Label Proportions via Moment Matching

Tianhao Ma[1], **Wei Wang**[2], **Yivan Zhang**[1,2], **Dong-Dong Wu**[1,2], **Takashi Ishida**[2,1], **Gang Niu**[2], **Masashi Sugiyama**[2,1]
[1]The University of Tokyo, [2]RIKEN AIP

Abstract

Learning from label proportions (LLP) is a weakly supervised setting where training data are grouped into bags with only bag-level label proportions observed, and the goal is to learn a classifier that predicts labels for individual instances. Recently, a large body of methods has emerged; however, most of them rely on the assumption that instance–label pairs are sampled i.i.d. and randomly assembled into bags, an assumption often violated in real-world scenarios. Our experimental results indicate that these methods perform suboptimally under non-random settings. In this work, we adopt a more realistic assumption: bags are sampled i.i.d., and instance labels are conditionally independent given the instances. Under this setting, we show that the label counts follow a Poisson multinomial distribution. Motivated by this observation, we propose LLP via moment matching (LLP-MM), a simple yet effective approach which leverages multi-order factorial moments as training objectives, encouraging the classifier predictions to match these moments and thereby more fully exploit the underlying statistical structure of the data. Extensive experiments on benchmark datasets under various bag construction strategies demonstrate the effectiveness of our approach while maintaining high computational efficiency. Our code is available at https://github.com/TianhaoMa5/LLP-MM.

1 Introduction

Recently, the use of large-scale data has pushed machine learning and artificial intelligence to unprecedented levels of performance. However, in many real-world scenarios the training data are imperfect: labels may be ambiguous, weak, or even incorrect. These issues have motivated a popular machine learning paradigm, weakly supervised learning (WSL) [Sugiyama et al., 2022], which aims to train robust models from such imperfect supervision. Representative forms of WSL include semi-supervised learning [Li et al., 2021, Lai et al., 2022], noisy-label learning [Patrini et al., 2017], partial-label learning [Lv et al., 2020, Wang et al., 2025], and unlabeled-unlabeled learning [Lu et al., 2021, Wei et al., 2024].

Learning from label proportions (LLP) is an important WSL problem that has recently attracted a lot of attention [Quadrianto et al., 2009, Tsai and Lin, 2020, Zhang et al., 2022, Ma et al., 2025, 2026a]. It naturally arises in many real-world applications, such as privacy-sensitive scenarios [Diemert et al., 2022, Apple, 2025, Google, 2024], as well as in image analysis tasks where instance-level annotations are difficult or costly to obtain [Lai et al., 2014, Tokunaga et al., 2020]. In these scenarios, only label proportions are available for training instead of instance-level annotations, while the goal remains to learn a classifier that predicts labels for individual instances.

In recent years, many LLP methods with strong performance as well as theoretical guarantees have been proposed [Busa-Fekete et al., 2023, Li et al., 2024, Busa-Fekete et al., 2025]. These methods are

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

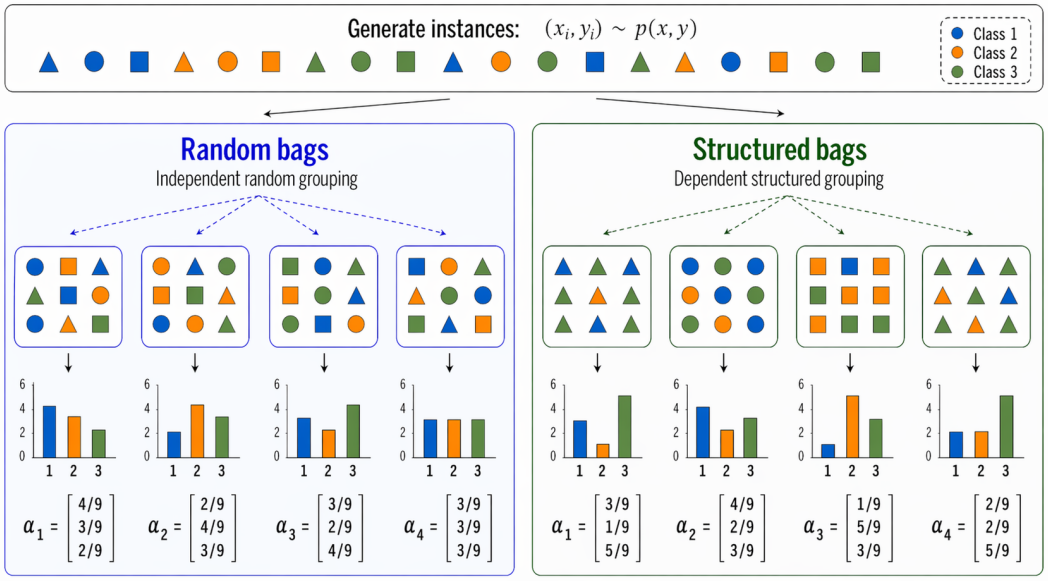

Figure 1: An illustration of structured bag construction based on instance features. A common example of a structured-bag assumption is to group instances with similar feature patterns into the same bag; for example, as shown in the figure, instances sharing the same shape feature are assigned to one bag.

often based on the *random-bag assumption* about the generation process of LLP data. First, instance–label pairs are sampled independently from the same joint density. Then, the bag is constructed by grouping instances uniformly at random. Finally, the label proportion is calculated as the fraction of instances from each class. However, in many real-world applications, bags are rarely formed uniformly at random. For example, election polling data is a practical application of LLP where instance-level labels are confidential while the results of regional support rates can be considered as label proportions [Sun et al., 2017, Ma et al., 2026a]. Here, voters within the same region often exhibit correlated voting behaviors, so the resulting groups are affected by geographic and demographic properties rather than random assignment. Similar structured-bag grouping mechanisms also arise in online advertising and privacy-preserving measurement, where instances are often grouped according to pre-defined categorical attributes or platform-defined grouping keys, rather than being grouped uniformly at random [Diemert et al., 2022, Brahmbhatt et al., 2024].

To address this problem, we adopt a more realistic data generation assumption, referred to as the *structured-bag assumption*, as illustrated in Fig. 1. In this setting, we allow instance correlations within the bag and assume that the grouping strategy is dependent on the individual examples. Besides, we assume that the instance labels are conditionally independent given the bag. Based on this assumption, we observe that the bag-level label counts correspond to the sum of independent but non-identically distributed categorical random variables, and therefore follow a Poisson multinomial distribution (PMD) [Daskalakis et al., 2015]. Consequently, the bag-level label statistics can be considered as following a normalized version of the PMD. This PMD perspective motivates the design of LLP algorithms that exploit the distributional information contained in the observed bag proportions.

Although it is natural to consider maximum likelihood estimation for LLP data, the likelihood of the PMD is difficult to compute, as its complexity grows exponentially with the number of classes [Lin et al., 2023]. Therefore, we propose *LLP via Moment Matching (LLP-MM)*, a simple yet effective approach that trains the classifier by matching classifier-induced multi-order factorial moments to the corresponding empirical factorial moments derived from the observed bag proportions. We also find that the classical proportion matching strategy [Yu et al., 2014, Ardehaly and Culotta, 2017] is a special case of our framework that only uses the first-order empirical moment as the target for the classifier-induced first-order moments. Theoretically, we show that considering higher orders of factorial moments introduces richer statistical information and yields a tighter solution space to facilitate

classifier training, which is in accord with our empirical results. Extensive comparative experiments show that moment matching remains effective under low computational cost, achieves state-of-the-art performance across various benchmark datasets, and is robust to different bag generation strategies.

Our contributions can be summarized as follows:

- We investigate the structured-bag assumption where the grouping strategy is dependent on examples. This assumption is more practical than the existing LLP data generation processes where bags are formed uniformly at random. Based on this assumption, we show that the bag-level label statistics follow the PMD, which offers a fresh perspective about LLP data.
- Motivated by the PMD structure, we propose moment matching, a simple and general framework that trains classifiers by matching classifier-induced moments with empirical multi-order factorial moments derived from label proportions. Our framework includes proportion matching, a widely used LLP strategy, as the special case of the first-order moments.
- Extensive experiments demonstrate that moment matching achieves strong performance on different benchmark datasets. Besides, moment matching is robust to different bag generation strategies and attains state-of-the-art results under practical computational budgets.

2 Preliminaries

In this section, we introduce the notation and background of multi-class classification and learning from label proportions.

2.1 Ordinary Multi-class Classification

Let $\mathcal{X}$ denote the input space and $\mathcal{Y} := \{1, \ldots, q\}$ denote the label space with q class labels. We assume that examples are sampled with probability density $p(x, y)$ over $\mathcal{X} \times \mathcal{Y}$. The goal of multi-class classification is to learn a *score-based predictor* $\boldsymbol{f} \in \mathcal{F}$ in a hypothesis class $\mathcal{F}$, where $\boldsymbol{f} : \mathcal{X} \to \Delta^{q-1}$ maps each input to the $(q-1)$-dimensional simplex

$$\Delta^{q-1} := \left\{ (p_1, \ldots, p_q) \in \mathbb{R}^q \;\middle|\; p_k \geq 0,\ \sum_{k=1}^{q} p_k = 1 \right\}.$$

Here, $\boldsymbol{f}_k(x)$ can be interpreted as the predicted probability of class k for input x.

Since direct minimization of the empirical 0-1 loss is computationally intractable for many natural hypothesis classes [Ben-David et al., 2003], we often instead use a *classification-calibrated surrogate loss* $\ell : \Delta^{q-1} \times \mathcal{Y} \to \mathbb{R}_+$ [Bartlett et al., 2006] and an empirical approximation of the following classification risk:

$$R(\boldsymbol{f}) := \mathbb{E}_{(x,y)\sim p(x,y)}[\ell(\boldsymbol{f}(x), y)]. \tag{1}$$

Under proper losses such as the cross-entropy loss or the mean squared error loss, the population-risk minimizer recovers the class-posterior distribution. We denote this Bayes-optimal predictor by $\boldsymbol{f}^*$, where

$$\boldsymbol{f}^*_k(x) = p(y = k \mid x), \quad \forall k \in \mathcal{Y},\ \forall x \in \mathcal{X}. \tag{2}$$

Lemma 2.1. *According to Yu et al. [2018], if ℓ is the cross-entropy loss or the mean squared error loss, then any optimal predictor $\boldsymbol{f}^\star \in \arg\min_{\boldsymbol{f}:\mathcal{X}\to\Delta^{q-1}} R(\boldsymbol{f})$ satisfies $\boldsymbol{f}^\star = \boldsymbol{f}^*$.*

2.2 Learning from Label Proportions

The learning objective in LLP is the same as that in ordinary multi-class classification: we still aim to learn an instance-level predictor $\boldsymbol{f} : \mathcal{X} \to \Delta^{q-1}$. The difference is that supervision is provided only at the bag level. Instead of observing a label for each individual example, the training data consist of groups of examples, referred to as *bags*, together with the class proportions of the labels in each bag [Yu et al., 2014]. Following Busa-Fekete et al. [2023, 2025], we assume for simplicity that every bag contains the same number m of examples. Formally, a *bag* is denoted by $\mathbf{B} = (x_1, \ldots, x_m) \in \mathcal{X}^m$, where each x_j is an unlabeled example. We assume that each instance-label pair (x_i, y_i) is drawn i.i.d. with probability density $p(x, y)$. These instances are then assembled

into bags by an unknown bag-construction mechanism, which may depend on the instance features or other latent grouping factors and is not restricted to uniform random assignment. We assume that, conditional on the bag $\mathbf{B} = (x_1, \ldots, x_m)$, the labels $y_1, \ldots, y_m$ are mutually independent, with $p(y_j \mid \mathbf{B}) = p(y_j \mid x_j)$ for every $j \in [m]$. Under this assumption, the conditional probability of the hidden label sequence given the bag can be factorized as

$$p(\boldsymbol{y} \mid \mathbf{B}) = \prod_{j=1}^{m} p(y_j \mid \mathbf{B}) = \prod_{j=1}^{m} p(y_j \mid x_j). \tag{3}$$

Instead of observing individual labels, we observe only the *class-proportion vector* $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_q)$. For each class $k \in \mathcal{Y}$, $\alpha_k := \frac{1}{m} \sum_{j=1}^{m} \mathbb{I}\{y_j = k\}$ is the *proportion* of examples in the bag that belong to class k. Here, $\mathbb{I}\{\cdot\}$ denotes the indicator function, which equals 1 if the condition is true and 0 otherwise. This induces an unknown density $p(\mathbf{B}, \alpha)$ over bags, and we model the observed bags as i.i.d. draws from this bag distribution. The posterior probability of $\boldsymbol{\alpha}$ is obtained by marginalizing over all label sequences consistent with $\boldsymbol{\alpha}$:

$$p(\boldsymbol{\alpha} \mid \mathbf{B}) = \sum_{\boldsymbol{y} \in \mathcal{Y}^{m/\boldsymbol{\alpha}}} p(\boldsymbol{y} \mid \mathbf{B}), \tag{4}$$

where $\mathcal{Y}^{m/\boldsymbol{\alpha}}$ denotes the set of all label sequences satisfying the proportion constraints:

$$\mathcal{Y}^{m/\boldsymbol{\alpha}} = \left\{ \boldsymbol{y} \in \mathcal{Y}^m \;\middle|\; \forall k \in \mathcal{Y},\ \frac{1}{m} \sum_{j=1}^{m} \mathbb{I}\{y_j = k\} = \alpha_k \right\}. \tag{5}$$

Accordingly, an LLP training dataset $\{(\mathbf{B}_i, \boldsymbol{\alpha}_i)\}_{i=1}^{n}$ consists of n i.i.d. examples of bag-proportion pairs drawn from the induced joint density $p(\mathbf{B}, \boldsymbol{\alpha})$. A common way to learn from such bag-proportion supervision is proportion matching [Yu et al., 2014, Ardehaly and Culotta, 2017]: given a predictor $\boldsymbol{f}$, one averages its instance-level predictions over each bag and encourages the resulting predicted class proportion to match the observed proportion α.

3 Methodology

In this section, we introduce the proposed moment matching approach. We first establish the connection between LLP and the PMD, and show that the popular proportion matching approach can be considered as the first-order moment matching. Then, we present the second-order and k-th-order moment matching. Finally, we present the theoretical analysis for the proposed approach.

3.1 Rethinking Label Proportions: A PMD Perspective

We show the connection between LLP and the PMD under certain assumptions about the data generation process. We denote by $\boldsymbol{e}^{y_j} \in \{0, 1\}^q$ the one-hot representation of label y_j. Conditioned on x_j, this one-hot vector can be equivalently viewed as a draw from a categorical distribution with parameter $p(\cdot \mid x_j)$, namely, $\boldsymbol{e}^{y_j} \sim \text{Categorical}(p(\cdot \mid x_j))$. We further define the bag-level label count vector by $\boldsymbol{z} := m\boldsymbol{\alpha} = \sum_{j=1}^{m} \boldsymbol{e}^{y_j}$. Since $\boldsymbol{e}^{y_j}$ are independent categorical random vectors conditioned on $\mathbf{B} = (x_1, \ldots, x_m)$, the count vector $\boldsymbol{z}$ follows a PMD [Daskalakis et al., 2015]. A straightforward strategy is to directly maximize the likelihood of the PMD in Equation (4). However, exact computation of the likelihood is computationally challenging. In particular, even the state-of-the-art method in computational statistics [Lin et al., 2023] has a time complexity of $O(m^{q-1} \log m)$. In typical LLP experimental settings, e.g., $m = 64$ and $q = 10$ [Ma et al., 2026b], this complexity is already astronomically large and therefore impractical for training. Instead of maximum likelihood, we leverage moment-based information and train the classifier by matching classifier-induced moments to empirical multi-order factorial moments.

3.2 Label Proportions Beyond First-Order Moments: A Second-Order Illustration

The classical algorithm Proportion Matching [Yu et al., 2014, Ardehaly and Culotta, 2017] can be viewed as a special case of our framework that performs first-order moment matching, and therefore

fails to exploit higher-order information; see Appendix B for details. Under the PMD, we provide an effective solution for exploiting richer information in label proportions through higher-order moments. As an illustration, we consider the following *second-order empirical moment matrix*:

$$\widehat{\mathbf{M}}_{\boldsymbol{\alpha}}^{(2)} = \frac{1}{m(m-1)} \sum_{i \neq j} \boldsymbol{e}^{y_i} (\boldsymbol{e}^{y_j})^\top, \tag{6}$$

where each entry $\left[\widehat{\mathbf{M}}_{\boldsymbol{\alpha}}^{(2)}\right]_{c,d}$ represents the normalized number of ordered instance index pairs (i, j) with $i \neq j$, $y_i = c$, and $y_j = d$. Since the number of instances in class c is $m\alpha_c$, this quantity can be expressed entirely in terms of $\boldsymbol{\alpha}$: when $c \neq d$, there are $m\alpha_c\, m\alpha_d$ such ordered pairs, while when $c = d$, there are $m\alpha_c(m\alpha_c - 1)$, since the same instance cannot be counted twice. This yields

$$\left[\widehat{\mathbf{M}}_{\boldsymbol{\alpha}}^{(2)}\right]_{c,d} = \frac{m\alpha_c(m\alpha_d - \mathbb{I}\{c = d\})}{m(m-1)}, \qquad c, d \in \{1, \ldots, q\}. \tag{7}$$

The detailed derivation is provided in Appendix C. For a bag $\mathbf{B}$, the classifier-induced second-order moment matrix is given by

$$\widehat{\mathbf{M}}_{\boldsymbol{f}}^{(2)}(\mathbf{B}) = \frac{1}{m(m-1)} \sum_{i \neq j} \boldsymbol{f}(x_i) \boldsymbol{f}(x_j)^\top. \tag{8}$$

Accordingly, $\widehat{\mathbf{M}}_{\boldsymbol{\alpha}}^{(2)}$ can serve as the target for the classifier-induced second-order moment. Following the objective of first-order moment matching, we minimize the *second-order moment matching risk*

$$\mathbb{E}_{p(\mathbf{B},\boldsymbol{\alpha})} \left[\mathcal{L}^{(2)} \Big(\widehat{\mathbf{M}}_{\boldsymbol{\alpha}}^{(2)}, \widehat{\mathbf{M}}_{\boldsymbol{f}}^{(2)}(\mathbf{B}) \Big) \right], \tag{9}$$

where $\mathcal{T}_r$ denotes the set of order-r tensors with nonnegative entries summing to one, and $\mathcal{L}^{(2)} : \mathcal{T}_2 \times \mathcal{T}_2 \to [0, +\infty]$ measures the discrepancy between the second-order empirical moment and the second-order classifier-induced moment.

3.3 Generalized Multi-Order Moment Matching

After the illustrative second-order example above, we now generalize the construction to arbitrary orders $r \leq m$, and describe how such multi-order information can be derived from label proportions and exploited for training. Following Johnson et al. [1997] and Ng et al. [2011], we introduce the normalized r-th order factorial moment tensor $\widehat{\mathbf{M}}_{\boldsymbol{\alpha}}^{(r)} \in (\mathbb{R}^q)^{\otimes r}$ as

$$\widehat{\mathbf{M}}_{\boldsymbol{\alpha}}^{(r)} := \frac{1}{(m)_r} \sum_{(j_1, \ldots, j_r) \in [m]^r_{\neq}} \boldsymbol{e}^{y_{j_1}} \otimes \boldsymbol{e}^{y_{j_2}} \otimes \cdots \otimes \boldsymbol{e}^{y_{j_r}}, \tag{10}$$

where $(m)_r := m(m-1)\cdots(m-r+1)$ is the falling factorial, $[m]^r_{\neq} := \{(j_1, \ldots, j_r) \in [m]^r \mid j_a \neq j_b\,, \forall a \neq b\}$ denotes the set of ordered r-tuples of distinct instance indices, and $\otimes$ denotes the tensor product, so that $\boldsymbol{e}^{y_{j_1}} \otimes \cdots \otimes \boldsymbol{e}^{y_{j_r}}$ is an order-r tensor in $(\mathbb{R}^q)^{\otimes r}$. Notably, the second-order moment matrix in Equation (6) is exactly the special case of Equation (10) with $r = 2$. Although the individual labels are not observed, the normalized r-th order factorial moment tensor can still be computed entirely from the label proportion vector $\boldsymbol{\alpha}$. To see this, consider any index tuple $(i_1, \ldots, i_r) \in [q]^r$. Define $\boldsymbol{k} = (k_1, \ldots, k_q) \in \mathbb{N}^q, k_c := \#\{\ell \in [r] \mid i_\ell = c\}, c = 1, \ldots, q$. Then $\boldsymbol{k}$ records the multiplicity of each class in $(i_1, \ldots, i_r)$, and satisfies $\sum_{c=1}^q k_c = r$. In other words, k_c records how many times class c appears in the tuple $(i_1, \ldots, i_r)$. Then the $(i_1, \ldots, i_r)$-th entry of $\widehat{\mathbf{M}}_{\boldsymbol{\alpha}}^{(r)}$ is given by

$$\left[\widehat{\mathbf{M}}_{\boldsymbol{\alpha}}^{(r)}\right]_{i_1, \ldots, i_r} = \frac{1}{(m)_r} \prod_{c=1}^{q} (z_c)_{k_c} = \frac{m^r}{(m)_r} \prod_{c=1}^{q} \prod_{t=0}^{k_c - 1} \left(\alpha_c - \frac{t}{m} \right). \tag{11}$$

Given a bag $\mathbf{B}$, for a classifier $\boldsymbol{f}$, we denote by $p_j := \boldsymbol{f}(x_j) \in \Delta^{q-1}$ the predicted class-probability vector of instance x_j. The classifier-induced normalized r-th order factorial moment tensor is then defined as

$$\widehat{\mathbf{M}}_{\boldsymbol{f}}^{(r)}(\mathbf{B}) := \frac{1}{(m)_r} \sum_{(j_1, \ldots, j_r) \in [m]^r_{\neq}} p_{j_1} \otimes p_{j_2} \otimes \cdots \otimes p_{j_r}. \tag{12}$$

Therefore, $\widehat{\mathbf{M}}_{\boldsymbol{\alpha}}^{(r)}$ computed from the observed label proportions can serve as the target tensor for $\widehat{\mathbf{M}}_{\boldsymbol{f}}^{(r)}(\mathbf{B})$, yielding a natural multi-order matching objective. In our work, we adopt a simple and direct strategy: we use the factorial moment tensor computed from the observed label proportions as the target, and encourage the classifier-induced moment tensor to match it at each order. Accordingly, the expected multi-order matching risk can be defined as

$$R_{\mathrm{MM}}^{(s)}(\boldsymbol{f}) := \mathbb{E}_{p(\mathbf{B},\boldsymbol{\alpha})}\left[\sum_{r=1}^{s} \lambda_r \, \mathcal{L}^{(r)}\Big(\widehat{\mathbf{M}}_{\boldsymbol{\alpha}}^{(r)}, \widehat{\mathbf{M}}_{\boldsymbol{f}}^{(r)}(\mathbf{B})\Big)\right], \tag{13}$$

where $s \in \{1, \ldots, m\}$ is the maximum order of moments used for training, $\lambda_r > 0$ is the weight associated with the r-th order moment, and $\sum_{r=1}^{s} \lambda_r = 1$. It is immediate that the PM risk in Equation (19) is recovered as the special case $s = 1$ of the above objective. Here, for each $r \in \{1, \ldots, s\}$, the loss function $\mathcal{L}^{(r)} : \mathcal{T}_r \times \mathcal{T}_r \to [0, +\infty]$ measures the discrepancy between the target r-th order factorial moment tensor $\widehat{\mathbf{M}}_{\boldsymbol{\alpha}}^{(r)}$ and the classifier-induced tensor $\widehat{\mathbf{M}}_{\boldsymbol{f}}^{(r)}(\mathbf{B})$. The empirical counterpart of $R_{\mathrm{MM}}^{(s)}(\boldsymbol{f})$ on a dataset $\{(\mathbf{B}_i, \boldsymbol{\alpha}_i)\}_{i=1}^{n}$ is given by

$$\widehat{R}_{\mathrm{MM}}^{(s)}(\boldsymbol{f}) := \frac{1}{n}\sum_{i=1}^{n}\left[\sum_{r=1}^{s} \lambda_r \, \mathcal{L}^{(r)}\Big(\widehat{\mathbf{M}}_{\boldsymbol{\alpha}_i}^{(r)}, \widehat{\mathbf{M}}_{\boldsymbol{f}}^{(r)}(\mathbf{B}_i)\Big)\right]. \tag{14}$$

Since the normalized factorial moment tensor has nonnegative entries that sum to one, it can be viewed, after vectorization, as a probability distribution over ordered class tuples $(c_1, \ldots, c_r) \in \{1, \ldots, q\}^r$. Therefore, matching two order-r factorial moment tensors is equivalent to matching two distributions over such class tuples, and $\mathcal{L}^{(r)}$ can be instantiated by standard distributional discrepancies, such as mean squared error or cross-entropy.

3.4 Theoretical Analysis

For each bag $\mathbf{B}$, let $\widehat{\mathbf{M}}_{\boldsymbol{f}^*}^{(r)}(\mathbf{B})$ denote the r-th order classifier-induced moment tensor associated with the Bayes-optimal classifier $\boldsymbol{f}^*$. Let $\mathcal{F}_{\mathrm{MM}}^{(s)}$ denote the set of minimizers of Equation (13). We then have the following proposition:

Proposition 3.1. *Assume that the hypothesis class $\mathcal{F}$ contains the Bayes-optimal classifier $\boldsymbol{f}^*$. For each $r \in \{1, \ldots, s_2\}$, assume that for any $\mathcal{T}_r$-valued random variable Z, its expectation $\mathbb{E}[Z]$ is the unique solution of $\min_{v \in \mathcal{T}_r} \mathbb{E}\Big[\mathcal{L}^{(r)}(Z, v)\Big]$. Assume further that $\lambda_r > 0$ for all $r \in \{1, \ldots, s_2\}$. Then, for any $1 \le s_1 < s_2 \le m$, we have $\mathcal{F}_{\mathrm{MM}}^{(s_2)} \subseteq \mathcal{F}_{\mathrm{MM}}^{(s_1)}$. That is, using more orders of factorial moments yields a tighter minimizer set.*

Proposition 3.1 partially explains the advantage of our method. By incorporating multi-order factorial moments as supervision, we impose progressively stronger constraints on the feasible minimizers. As a result, the solution set becomes tighter while still containing the Bayes-optimal classifier $\boldsymbol{f}^*$.

Theorem 3.2 (Generalization bound)**.** *Assume that the bag coordinates are exchangeable. Assume that, for each $r \in \{1, \ldots, s\}$, the loss $\mathcal{L}^{(r)}(A, \cdot)$ is L_r-Lipschitz with respect to the Frobenius norm, uniformly over all admissible target tensors A, and satisfies $0 \le \mathcal{L}^{(r)}(A, B) \le B_r$ for all admissible A, B. Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$,*

$$\sup_{\boldsymbol{f} \in \mathcal{F}} |R_{\mathrm{MM}}^{(s)}(\boldsymbol{f}) - \widehat{R}_{\mathrm{MM}}^{(s)}(\boldsymbol{f})| \le 2\sqrt{2m}\left(\sum_{r=1}^{s} \lambda_r L_r \, r\right)\left(\sum_{c=1}^{q} \mathfrak{R}_n(\mathcal{F}_c)\right) + \left(\sum_{r=1}^{s} \lambda_r B_r\right)\sqrt{\frac{\log(2/\delta)}{2n}}. \tag{15}$$

Under the current bag-i.i.d. assumption, our analysis establishes the optimality of matching the Bayes-optimal classifier-induced moments. Although this result does not by itself yield a full classifier-level consistency guarantee of the form $\boldsymbol{f} \to \boldsymbol{f}^*$, it provides a theoretical justification for the proposed moment-matching principle and, together with the empirical results in Section 4, suggests that the proposed approach is effective in practice. Establishing such consistency remains challenging for real-world structured bags [Bortsova et al., 2018, Sun et al., 2017, Pacheco et al., 2024], such as those in blood-cell detection, where shared, unobserved physiological factors may induce donor-specific

correlations among cells, making the within-bag dependence structure difficult to characterize. We leave this problem for future work.

For computing the loss, a naive approach would explicitly construct the full order-r moment tensor, whose number of entries scales as q^r. To avoid this dependence, we use a state-compressed dynamic programming method that reduces both time and memory to depend on the number of compressed histogram states. For example, in the second-order case, the moment tensor is a symmetric matrix, and therefore one only needs to compute its distinct entries rather than all q^2 ordered entries. The details are provided in Appendix F.2. The resulting complexity is summarized below.

Proposition 3.3 (Complexity of the state-compressed computation)**.** *Fix a bag of size m and a maximum order $s \leq m$. The compressed computation of all moments up to order s requires $O\big(mq\binom{q+s}{s}\big)$ time and $O\big(\binom{q+s}{s}\big)$ memory per bag.*

Although this complexity still grows combinatorially with the maximum order s, in practice only a small number of low-order moments is sufficient to achieve strong performance.

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets and Bag Construction

Following prior work [Busa-Fekete et al., 2023, 2025], we construct synthetic LLP datasets on three standard benchmarks: CIFAR-10, CIFAR-100 [Krizhevsky and Hinton, 2009], and miniImageNet [Vinyals et al., 2016]. Following Ma et al. [2026b], we evaluate three bag generation strategies, namely random bags, cluster bags, and $\boldsymbol{\alpha}$-first bags, with bag sizes $m \in \{16, 32, 64, 128\}$. Existing LLP benchmarks are mainly designed for binary classification [Brahmbhatt et al., 2024, Franco et al., 2023], and are therefore not well aligned with our multi-class classification setting. The detailed construction procedures are provided in Appendix G.1.

4.1.2 Baselines and Implementation Details

We compare LLP-MM with representative LLP baselines, including PM [Yu et al., 2014, Ardehaly and Culotta, 2017], DSQ [Li et al., 2024], LLP-PVC [Ma et al., 2026b], LLP-FC [Zhang et al., 2022], ROT [Dulac-Arnold et al., 2019], EasyLLP [Busa-Fekete et al., 2023], GeneralUPM [Applebaum et al., 2025], EasyLLP-ABS, GeneralUPM-ABS, and FlowLLP [Rakotomamonjy et al., 2026]. EasyLLP-ABS and GeneralUPM-ABS take the absolute value of the original loss to reduce the effect of negative loss values. For the main benchmark comparisons, LLP-MM uses $s = 8$ for CIFAR-10, and $s = 3$ for CIFAR-100 and miniImageNet. For each order r, we use the cross-entropy loss as $\mathcal{L}^{(r)}$, and set the weight λ_r to $1/s$. Further implementation details are provided in Appendix G.2. We also evaluate on patient-grouped blood cell data in Section 4.3.

4.2 Comparison with Baselines

LLP-MM ranks first in average accuracy across all nine dataset–bag-construction settings in Tables 1 and 2. DSQ and LLP-FC perform best in some larger-bag settings. EasyLLP and GeneralUPM can suffer from negative empirical losses during optimization. LLP-PVC is competitive on CIFAR-10 but has lower average accuracy on CIFAR-100 and miniImageNet. This gap may partly reflect its one-vs-rest decomposition, which creates increasingly imbalanced binary tasks as the number of classes grows and does not directly model the joint multiclass PMD. FFT-based evaluation of very small Poisson-binomial probabilities may also affect likelihood optimization through roundoff and cancellation errors [Peres et al., 2020].

4.3 Semi-Real-World Evaluation

Inspired by the blood-cell LLP setting of Pacheco et al. [2024], whose real data are not publicly available, we construct a semi-real-world LLP benchmark on the public 13-class KU-Optofil PBC dataset [Yarıkan et al., 2026]. We retain its official patient-separated split, merge training and validation cells, and form bags by patient ID, with synthetic subbags for missing IDs. Bag proportions

Table 1: Best test accuracy under Random Bag and Cluster Bag constructions (mean ± std over three runs, %). The best result for each bag size and average is bold. Ranks use average accuracy with standard competition ranking. † indicates severe overfitting caused by negative loss values, with predictions collapsing to a single class.

Dataset	Method	16	32	64	128	Average	Rank
			Random Bag				
CIFAR-10	PM	93.51 ± 0.11	90.05 ± 0.22	75.23 ± 0.70	60.02 ± 0.33	79.70	4
	DSQ	93.52 ± 0.15	91.39 ± 0.16	79.86 ± 0.77	61.90 ± 1.35	81.67	3
	LLP-PVC	91.95 ± 0.07	90.55 ± 0.08	83.79 ± 0.57	64.44 ± 1.45	82.68	2
	LLP-FC	75.29 ± 0.72	61.06 ± 0.87	53.89 ± 0.61	60.01 ± 0.24	62.56	7
	ROT	93.21 ± 0.09	**92.04 ± 0.10**	80.52 ± 0.49	35.03 ± 1.25	75.20	5
	EasyLLP	32.99 ± 0.80	31.98 ± 2.69	26.08 ± 0.21	26.40 ± 0.98	29.36	11
	EasyLLP-ABS	57.26 ± 2.36	50.54 ± 2.77	41.86 ± 3.53	33.80 ± 1.78	45.87	9
	GeneralUPM	45.43 ± 1.05	36.98 ± 3.42	29.80 ± 1.21	27.20 ± 0.68	34.85	10
	GeneralUPM-ABS	61.42 ± 1.33	52.29 ± 2.76	42.43 ± 3.18	33.08 ± 2.24	47.30	8
	FlowLLP	92.65 ± 0.10	88.82 ± 0.54	72.65 ± 0.69	40.12 ± 6.48	73.56	6
	LLP-MM	**93.97 ± 0.12**	91.99 ± 0.08	**84.13 ± 0.57**	**66.74 ± 0.20**	**84.21**	**1**
CIFAR-100	PM	71.12 ± 0.26	66.62 ± 0.05	54.34 ± 0.87	17.69 ± 0.57	52.44	3
	DSQ	71.01 ± 0.34	65.78 ± 0.26	45.95 ± 0.86	**25.70 ± 0.80**	52.11	4
	LLP-PVC	63.53 ± 0.21	61.73 ± 0.47	49.78 ± 2.78	12.13 ± 1.00	46.79	6
	LLP-FC	71.77 ± 0.08	67.10 ± 0.18	54.84 ± 0.84	17.39 ± 0.55	52.77	2
	ROT	72.02 ± 0.13	64.42 ± 0.56	14.81 ± 1.46	5.26 ± 0.15	39.13	7
	EasyLLP	2.43 ± 0.44	1.95 ± 0.27	1.34 ± 0.20	1.34 ± 0.05	1.77	11
	EasyLLP-ABS	17.66 ± 1.73	12.39 ± 1.44	9.62 ± 0.01	6.50 ± 0.41	11.55	9
	GeneralUPM	12.68 ± 0.33	8.36 ± 0.30	5.65 ± 1.05	4.07 ± 0.13	7.69	10
	GeneralUPM-ABS	24.00 ± 1.57	16.51 ± 1.61	11.33 ± 0.47	7.23 ± 0.40	14.77	8
	FlowLLP	71.49 ± 0.33	66.76 ± 0.06	51.68 ± 0.91	12.04 ± 2.05	50.49	5
	LLP-MM	**72.06 ± 0.10**	**67.14 ± 0.05**	**57.79 ± 0.64**	23.82 ± 0.54	**55.20**	**1**
miniImageNet	PM	64.76 ± 0.40	56.14 ± 0.35	38.31 ± 1.06	10.01 ± 2.04	42.30	3
	DSQ	63.38 ± 0.13	53.73 ± 0.40	31.45 ± 0.58	**14.81 ± 0.81**	40.84	4
	LLP-PVC	57.71 ± 0.14	54.63 ± 0.34	24.19 ± 1.57	5.21 ± 0.43	35.44	6
	LLP-FC	65.75 ± 0.05	58.13 ± 0.17	39.40 ± 1.75	11.05 ± 1.72	43.58	2
	ROT	**67.57 ± 0.82**	41.50 ± 6.10	7.21 ± 0.41	2.64 ± 0.57	29.73	7
	EasyLLP	1.92 ± 0.13	1.59 ± 0.15	1.51 ± 0.08	1.36 ± 0.09	1.59	11
	EasyLLP-ABS	11.88 ± 0.57	8.15 ± 0.25	5.35 ± 0.41	3.57 ± 0.28	7.24	9
	GeneralUPM	9.44 ± 0.11	6.08 ± 0.46	4.00 ± 0.25	2.91 ± 0.33	5.61	10
	GeneralUPM-ABS	19.61 ± 0.40	11.81 ± 0.77	7.79 ± 0.23	4.29 ± 0.52	10.87	8
	FlowLLP	62.83 ± 0.86	52.84 ± 1.46	28.13 ± 0.94	7.20 ± 0.77	37.75	5
	LLP-MM	65.94 ± 0.03	**59.79 ± 0.14**	**44.76 ± 0.29**	8.49 ± 1.08	**44.74**	**1**
			Cluster Bag				
CIFAR-10	PM	93.31 ± 0.19	91.29 ± 0.11	86.30 ± 0.45	72.94 ± 1.64	85.96	2
	DSQ	93.62 ± 0.19	91.97 ± 0.44	87.22 ± 0.53	64.65 ± 2.56	84.37	4
	LLP-PVC	92.30 ± 0.06	90.46 ± 0.24	85.29 ± 0.69	64.09 ± 2.06	83.04	6
	LLP-FC	56.22 ± 1.47	39.49 ± 0.69	34.29 ± 0.19	72.65 ± 1.04	50.66	7
	ROT	93.53 ± 0.07	92.05 ± 0.10	87.14 ± 0.15	59.96 ± 2.21	83.17	5
	EasyLLP	10.90 ± 1.25 †	10.90 ± 1.26 †	10.87 ± 1.20 †	10.78 ± 1.16 †	10.86 †	10
	EasyLLP-ABS	32.47 ± 1.31	26.93 ± 1.30	23.88 ± 0.35	19.73 ± 1.65	25.76	8
	GeneralUPM	10.90 ± 1.25 †	10.90 ± 1.26 †	10.87 ± 1.20 †	10.78 ± 1.16 †	10.86 †	10
	GeneralUPM-ABS	32.47 ± 1.31	26.93 ± 1.30	22.33 ± 1.84	20.25 ± 1.33	25.50	9
	FlowLLP	93.16 ± 0.16	91.20 ± 0.17	83.96 ± 0.28	69.56 ± 1.10	84.47	3
	LLP-MM	**94.04 ± 0.20**	**92.33 ± 0.10**	**89.12 ± 0.43**	**75.19 ± 2.14**	**87.67**	**1**
CIFAR-100	PM	71.35 ± 0.20	66.21 ± 0.68	57.78 ± 0.72	46.06 ± 1.02	60.35	3
	DSQ	**72.33 ± 0.15**	67.59 ± 1.04	54.33 ± 2.31	35.59 ± 1.86	57.46	5
	LLP-PVC	65.03 ± 0.55	61.13 ± 0.68	50.67 ± 0.69	29.77 ± 0.74	51.65	7
	LLP-FC	71.44 ± 0.51	66.59 ± 0.50	58.44 ± 1.02	**46.19 ± 0.66**	60.66	2
	ROT	72.06 ± 0.10	66.68 ± 0.28	48.45 ± 0.12	29.43 ± 1.12	54.16	6
	EasyLLP	0.90 ± 0.02 †	0.90 ± 0.02 †	0.90 ± 0.02 †	0.90 ± 0.02 †	0.90 †	10
	EasyLLP-ABS	7.07 ± 0.29	4.95 ± 0.41	3.22 ± 0.35	2.54 ± 0.27	4.44	8
	GeneralUPM	0.90 ± 0.02 †	0.90 ± 0.02 †	0.90 ± 0.02 †	0.90 ± 0.02 †	0.90 †	10
	GeneralUPM-ABS	7.14 ± 0.36	4.74 ± 0.06	3.26 ± 0.29	2.56 ± 0.25	4.42	9
	FlowLLP	70.41 ± 0.11	65.12 ± 0.07	55.15 ± 0.39	42.62 ± 0.52	58.32	4
	LLP-MM	72.24 ± 0.67	**67.89 ± 0.67**	**58.84 ± 0.65**	45.66 ± 0.50	**61.16**	**1**
miniImageNet	PM	65.21 ± 0.76	56.37 ± 0.10	44.06 ± 1.78	30.18 ± 0.90	48.95	3
	DSQ	66.03 ± 0.27	55.79 ± 1.36	37.02 ± 2.75	19.56 ± 0.15	44.60	4
	LLP-PVC	58.77 ± 0.39	52.50 ± 1.10	27.91 ± 8.11	15.31 ± 1.02	38.62	7
	LLP-FC	65.53 ± 0.62	56.41 ± 0.61	44.26 ± 1.01	**30.30 ± 1.07**	49.13	2
	ROT	**67.29 ± 0.40**	58.16 ± 0.87	33.94 ± 1.89	17.83 ± 1.32	44.30	5
	EasyLLP	1.04 ± 0.05 †	1.01 ± 0.01 †	1.01 ± 0.01 †	1.01 ± 0.01 †	1.01 †	10
	EasyLLP-ABS	5.31 ± 0.07	3.38 ± 0.33	2.86 ± 0.12	2.27 ± 0.05	3.45	8
	GeneralUPM	1.04 ± 0.05 †	1.01 ± 0.01 †	1.01 ± 0.01 †	1.01 ± 0.01 †	1.01 †	10
	GeneralUPM-ABS	5.31 ± 0.07	3.46 ± 0.26	2.78 ± 0.19	2.19 ± 0.07	3.44	9
	FlowLLP	61.86 ± 0.88	51.56 ± 0.43	37.44 ± 0.45	25.67 ± 0.53	44.13	6
	LLP-MM	66.99 ± 0.68	**59.65 ± 0.97**	**45.91 ± 1.63**	28.36 ± 0.71	**50.23**	**1**

are computed from cell labels. Table 3 reports three-run results, selecting each run's checkpoint by the highest test Macro-F1. Details are in Appendix G.4.

Table 2: Best test accuracy under $\boldsymbol{\alpha}$-First Bag construction (mean $\pm$ std over three runs, %). The best result for each bag size and average is bold. Ranks use average accuracy with standard competition ranking.

Dataset	Method	16	32	64	128	Average	Rank
CIFAR-10	PM	91.00 ± 0.40	70.37 ± 0.32	65.38 ± 1.14	61.35 ± 0.52	72.03	4
	DSQ	90.98 ± 0.16	74.24 ± 0.57	73.78 ± 1.11	68.83 ± 1.43	76.96	3
	LLP-PVC	91.85 ± 0.13	89.18 ± 0.11	**81.44 ± 0.61**	66.78 ± 0.54	82.31	2
	LLP-FC	58.53 ± 0.36	49.40 ± 0.64	49.00 ± 0.42	64.55 ± 1.15	55.37	7
	ROT	92.93 ± 0.07	**90.71 ± 0.11**	61.53 ± 3.70	38.24 ± 1.22	70.85	5
	EasyLLP	47.72 ± 1.40	37.40 ± 0.61	28.82 ± 2.44	24.09 ± 1.58	34.51	10
	EasyLLP-ABS	44.97 ± 0.77	36.84 ± 0.93	38.03 ± 0.35	39.59 ± 1.07	39.86	8
	GeneralUPM	41.79 ± 0.78	31.35 ± 2.18	29.76 ± 0.92	28.91 ± 1.56	32.95	11
	GeneralUPM-ABS	43.53 ± 1.03	36.56 ± 0.08	37.56 ± 0.59	37.93 ± 4.25	38.89	9
	FlowLLP	90.63 ± 0.08	72.68 ± 1.60	60.78 ± 0.28	55.39 ± 3.44	69.87	6
	LLP-MM	**93.54 ± 0.12**	88.25 ± 0.58	76.83 ± 0.94	**70.65 ± 0.69**	**82.32**	**1**
CIFAR-100	PM	70.81 ± 0.23	67.01 ± 0.14	54.53 ± 0.02	5.23 ± 0.34	49.40	3
	DSQ	70.96 ± 0.45	66.07 ± 0.47	46.56 ± 0.68	**13.60 ± 0.25**	49.30	4
	LLP-PVC	64.54 ± 0.34	61.43 ± 0.29	49.81 ± 1.90	4.41 ± 0.53	45.05	6
	LLP-FC	71.41 ± 0.13	66.93 ± 0.16	57.69 ± 0.43	4.75 ± 0.10	50.20	2
	ROT	**72.14 ± 0.23**	63.96 ± 0.40	16.58 ± 0.66	2.76 ± 0.04	38.86	7
	EasyLLP	1.95 ± 0.25	2.43 ± 0.63	2.13 ± 0.43	1.73 ± 0.16	2.06	11
	EasyLLP-ABS	17.51 ± 0.83	12.50 ± 0.36	9.39 ± 0.32	4.32 ± 0.66	10.93	9
	GeneralUPM	12.27 ± 0.43	8.87 ± 0.76	5.77 ± 0.18	2.63 ± 0.30	7.38	10
	GeneralUPM-ABS	23.98 ± 0.88	16.93 ± 0.54	11.20 ± 0.04	6.36 ± 0.37	14.61	8
	FlowLLP	70.90 ± 0.24	65.85 ± 0.46	50.37 ± 1.17	2.79 ± 0.53	47.48	5
	LLP-MM	72.01 ± 0.29	**67.36 ± 0.25**	**58.02 ± 0.35**	7.18 ± 0.20	**51.14**	**1**
miniImageNet	PM	63.76 ± 0.06	55.47 ± 0.29	36.44 ± 0.80	3.62 ± 0.09	39.82	4
	DSQ	63.14 ± 0.33	55.09 ± 0.60	33.03 ± 0.57	**8.72 ± 0.69**	40.00	3
	LLP-PVC	57.47 ± 0.37	54.74 ± 0.49	31.01 ± 1.47	2.89 ± 0.26	36.53	6
	LLP-FC	64.23 ± 0.32	54.71 ± 0.48	38.82 ± 0.38	3.80 ± 0.27	40.39	2
	ROT	65.87 ± 0.48	54.29 ± 1.00	11.19 ± 0.20	2.33 ± 0.28	33.42	7
	EasyLLP	1.62 ± 0.21	1.61 ± 0.08	1.71 ± 0.13	1.48 ± 0.08	1.60	11
	EasyLLP-ABS	11.47 ± 0.20	8.56 ± 0.18	5.49 ± 0.66	2.83 ± 1.04	7.09	9
	GeneralUPM	9.85 ± 0.13	6.89 ± 0.54	4.30 ± 0.21	2.41 ± 0.19	5.86	10
	GeneralUPM-ABS	20.73 ± 1.05	13.60 ± 0.78	7.70 ± 0.09	4.03 ± 0.30	11.51	8
	FlowLLP	62.76 ± 0.21	52.79 ± 1.01	30.89 ± 1.59	2.67 ± 0.33	37.28	5
	LLP-MM	**66.37 ± 0.16**	**59.96 ± 0.19**	**44.91 ± 0.45**	3.43 ± 0.09	**43.67**	**1**

Table 3: Test performance on KU-Optofil PBC (mean $\pm$ std over three runs, %). The best result in each column is bold.

Method	Accuracy ↑	Macro-F1 ↑	Balanced Acc. ↑	Weighted-F1 ↑
PM	77.35 ± 13.10	29.11 ± 10.74	32.70 ± 10.01	74.46 ± 15.73
DSQ	92.39 ± 1.33	56.89 ± 7.65	60.08 ± 8.30	92.70 ± 1.57
LLP-PVC	92.82 ± 1.94	58.42 ± 4.31	61.63 ± 4.27	91.58 ± 2.33
LLP-FC	75.11 ± 9.57	26.26 ± 4.28	27.12 ± 4.46	72.46 ± 11.60
ROT	**94.05 ± 0.65**	56.11 ± 7.56	56.82 ± 7.74	93.35 ± 1.17
EasyLLP	37.98 ± 8.32	11.51 ± 1.77	14.50 ± 0.85	39.28 ± 3.11
EasyLLP-ABS	38.33 ± 8.83	11.02 ± 1.59	13.85 ± 1.81	40.48 ± 5.11
GeneralUPM	66.33 ± 0.22	12.48 ± 1.28	14.16 ± 1.32	55.19 ± 2.51
GeneralUPM-ABS	66.33 ± 0.22	12.48 ± 1.28	14.16 ± 1.32	55.19 ± 2.51
FlowLLP	61.97 ± 4.34	25.67 ± 2.34	31.11 ± 1.90	55.75 ± 4.85
LLP-MM ($s = 3$)	91.88 ± 1.20	51.79 ± 5.80	53.23 ± 7.41	91.88 ± 0.48
LLP-MM ($s = 5$)	93.40 ± 0.84	66.40 ± 2.22	69.81 ± 0.91	94.20 ± 0.55
LLP-MM ($s = 8$)	93.93 ± 0.58	**69.75 ± 1.47**	**70.73 ± 2.74**	**94.57 ± 0.22**

LLP-MM ($s = 8$) leads in Macro-F1, Balanced Accuracy, and Weighted-F1, exceeding LLP-PVC by 11.33 Macro-F1 points; ROT has the highest Accuracy. Increasing s from 3 to 5 to 8 raises Macro-F1 from 51.79% to 66.40% to 69.75%.

4.4 Effect of the Order Parameter

Figure 2 shows the accuracy–runtime trade-off as s increases. Higher orders tighten the minimizer set toward $\boldsymbol{f}^*$ (Proposition 3.1) but increase computation. We report CIFAR-100 only up to $s = 3$, since fourth-order terms are prohibitively expensive with 100 classes. As the order increases from low values, prediction performance generally improves, suggesting that higher-order moments provide useful supervision beyond mean class proportions. At higher orders, performance tends to stabilize, and further increases bring only limited gains. Since computation continues to grow with the order, moderate orders offer a practical balance between predictive performance and computational cost.

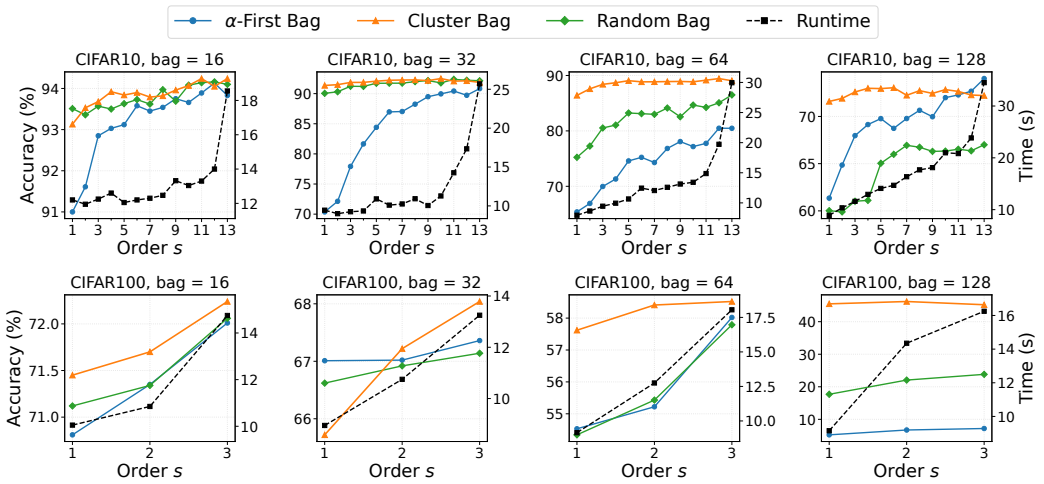

Figure 2: Effect of the order parameter on prediction performance and runtime per epoch across different bag construction strategies and bag sizes. Runtime was measured on a single NVIDIA A100-SXM4-40GB GPU with an AMD EPYC 7742 64-Core Processor.

5 Conclusion

We proposed LLP-MM, a moment-matching framework for learning from label proportions. By connecting proportions to the Poisson multinomial distribution, LLP-MM exploits multi-order factorial moments derived from observed label proportions and uses them as training supervision. Theoretical analysis shows that higher-order moments impose stronger constraints on the solution space, and experiments demonstrate that LLP-MM achieves robust performance across various settings while maintaining practical computational efficiency. A limitation is the scarcity of standardized real-world LLP datasets; beyond the semi-real-world study in Section 4.3, future work will develop broader realistic benchmarks. We will also design multiclass LLP benchmarks and evaluation metrics to compare different methods.

Acknowledgments and Disclosure of Funding

Part of the experiments was conducted using the Miyabi and Wisteria supercomputing systems at the Joint Center for Advanced High Performance Computing (JCAHPC) and the Information Technology Center, The University of Tokyo, respectively.

YZ was supported by JSPS KAKENHI Grant Number JP26K21299. DDW was supported by the RIKEN Junior Research Associate Program. TI was supported by KAKENHI Grant Number 22K17946. MS was supported by JST ASPIRE Grant Number JPMJAP25B1.

References

Apple. SKAdNetwork 4 release notes. https://developer.apple.com/documentation/storekit/skadnetwork/skadnetwork_release_notes/skadnetwork_4_release_notes, 2025. Apple Developer Documentation.

Lorne Applebaum, Travis Dick, Claudio Gentile, Haim Kaplan, and Tomer Koren. Optimal learning from label proportions with general loss functions. *arXiv preprint arXiv:2509.15145*, 2025.

Ehsan Mohammady Ardehaly and Aron Culotta. Co-training for demographic classification using deep learning from label proportions. In *IEEE International Conference on Data Mining Workshops*, pages 1017–1024, 2017.

Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006. URL https://doi.org/10.1198/016214505000000907.

Shai Ben-David, Nadav Eiron, and Philip M Long. On the difficulty of approximately maximizing agreements. *Journal of Computer and System Sciences*, 66(3):496–514, 2003. URL https://doi.org/10.1016/S0022-0000(03)00038-2.

Gerda Bortsova, Florian Dubost, Silas Ørting, Ioannis Katramados, Laurens Hogeweg, Laura Thomsen, Mathilde Wille, and Marleen de Bruijne. Deep learning from label proportions for emphysema quantification. In *Proceedings of the 21st International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 768–776. Springer, 2018.

Anand Brahmbhatt, Mohith Pokala, Rishi Saket, and Aravindan Raghuveer. LLP-Bench: A large scale tabular benchmark for learning from label proportions. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 4374–4381, 2024.

Róbert Busa-Fekete, Heejin Choi, Travis Dick, Claudio Gentile, and Andrés Muñoz Medina. Easy learning from label proportions. In *Advances in Neural Information Processing Systems*, pages 14957–14968, 2023.

Robert Busa-Fekete, Travis Dick, Claudio Gentile, Haim Kaplan, Tomer Koren, and Uri Stemmer. Nearly optimal sample complexity for learning with label proportions. *arXiv preprint arXiv:2505.05355*, 2025.

Constantinos Daskalakis, Gautam Kamath, and Christos Tzamos. On the structure, covering, and learning of Poisson multinomial distributions. In *2015 IEEE 56th annual symposium on foundations of computer science*, pages 1203–1217, 2015.

Eustache Diemert, Romain Fabre, Alexandre Gilotte, Fei Jia, Basile Leparmentier, Jérémie Mary, Zhonghua Qu, Ugo Tanielian, and Hui Yang. Lessons from the AdKDD'21 privacy-preserving ML challenge. In *Proceedings of the ACM Web Conference 2022*, pages 2026–2035, 2022.

Gabriel Dulac-Arnold, Neil Zeghidour, Marco Cuturi, Lucas Beyer, and Jean-Philippe Vert. Deep multi-class learning from label proportions. *arXiv preprint arXiv:1905.12909*, 2019.

Gabriel Franco, Mark Crovella, and Giovanni Comarela. Dependence and model selection in LLP: the problem of variants. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 470–481, 2023.

Google. Private aggregation API of Chrome privacy sandbox. https://developer.chrome.com/docs/privacy-sandbox/aggregation-service/, 2024. Chrome Privacy Sandbox documentation.

Norman L. Johnson, Samuel Kotz, and Narayanaswamy Balakrishnan. *Discrete Multivariate Distributions*. John Wiley & Sons, 1997. ISBN 9780471128441.

Alex Krizhevsky and Geoffrey E. Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Kuan-Ting Lai, Felix X Yu, Ming-Syan Chen, and Shih-Fu Chang. Video event detection by inferring temporal instance labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2243–2250, 2014.

Zhengfeng Lai, Chao Wang, Henrry Gunawan, Sen-Ching S. Cheung, and Chen-Nee Chuah. Smoothed adaptive weighting for imbalanced semi-supervised learning: Improve reliability against unknown distribution data. In *International Conference on Machine Learning*, pages 11828–11843, 2022.

Changchun Li, Ximing Li, and Jihong Ouyang. Semi-supervised text classification with balanced deep representation distributions. In *Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 5044–5053, 2021.

Gene Li, Lin Chen, Adel Javanmard, and Vahab Mirrokni. Optimistic rates for learning from label proportions. In *Annual Conference on Learning Theory*, pages 3437–3474, 2024.

Zhengzhi Lin, Yueyao Wang, and Yili Hong. The computing of the poisson multinomial distribution and applications in ecological inference and machine learning. *Computational Statistics*, 38(4): 1851–1877, 2023.

Jiabin Liu, Bo Wang, Zhiquan Qi, Yingjie Tian, and Yong Shi. Learning from label proportions with generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 7167–7177, 2019.

Nan Lu, Shida Lei, Gang Niu, Issei Sato, and Masashi Sugiyama. Binary classification from multiple unlabeled datasets via surrogate set classification. In *International Conference on Machine Learning*, pages 7134–7144, 2021.

Jiaqi Lv, Miao Xu, Lei Feng, Gang Niu, Xin Geng, and Masashi Sugiyama. Progressive identification of true labels for partial-label learning. In *International Conference on Machine Learning*, pages 6500–6510, 2020.

Tianhao Ma, Han Chen, Juncheng Hu, Yungang Zhu, and Ximing Li. Forming auxiliary high-confident instance-level loss to promote learning from label proportions. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20592–20601, 2025.

Tianhao Ma, Ximing Li, Changchun Li, and Renchu Guan. Learning from label proportions with dual-proportion constraints. *arXiv preprint arXiv:2603.21153*, 2026a.

Tianhao Ma, Wei Wang, Ximing Li, Gang Niu, and Masashi Sugiyama. Learning from label proportions via proportional value classification. In *The Fourteenth International Conference on Learning Representations*, 2026b.

Andreas Maurer. A vector-contraction inequality for Rademacher complexities. In *Proceedings of the 27th International Conference on Algorithmic Learning Theory*, pages 3–17, 2016.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

Kai Wang Ng, Guo-Liang Tian, and Man-Lai Tang. *Dirichlet and Related Distributions: Theory, Methods and Applications*. John Wiley & Sons, 2011. ISBN 9780470688199.

Carolina Pacheco, Florence Yellin, René Vidal, and Benjamin Haeffele. Vertex proportion loss for multi-class cell detection from label proportions. In *Medical Image Computing and Computer Assisted Intervention*, pages 366–376. Springer Nature Switzerland, 2024. URL https://papers.miccai.org/miccai-2024/829-Paper3997.html.

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

Noah Peres, Andrew Lee, and Uri Keich. Exactly computing the tail of the Poisson-Binomial distribution. *arXiv preprint arXiv:2004.07429*, 2020. URL https://arxiv.org/abs/2004.07429.

Novi Quadrianto, Alex J Smola, Tibério S Caetano, and Quoc V Le. Estimating labels from label proportions. *Journal of Machine Learning Research*, 10:2349–2374, 2009.

Alain Rakotomamonjy, Maxime Vono, and Liva Ralaivola. Particle flow for learning from label proportions. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/63488.

Masashi Sugiyama, Han Bao, Takashi Ishida, Nan Lu, Tomoya Sakai, and Gang Niu. *Machine learning from weak supervision: An empirical risk minimization approach*. MIT Press, 2022.

Tao Sun, Dan Sheldon, and Brendan O'Connor. A probabilistic approach for learning with label proportions applied to the US presidential election. In *Proceedings of the 2017 IEEE International Conference on Data Mining*, pages 445–454. IEEE, 2017.

Hiroki Tokunaga, Brian Kenji Iwana, Yuki Teramoto, Akihiko Yoshizawa, and Ryoma Bise. Negative pseudo labeling using class proportion for semantic segmentation in pathology. In *European Conference on Computer Vision*, pages 430–446, 2020.

Kuen-Han Tsai and Hsuan-Tien Lin. Learning from label proportions with consistency regularization. In *Asian Conference on Machine Learning*, pages 513–528, 2020.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in Neural Information Processing Systems*, 29, 2016.

Wei Wang, Tianhao Ma, Ming-Kun Xie, Gang Niu, and Masashi Sugiyama. Rethinking consistent multi-label classification under inexact supervision. *arXiv preprint arXiv:2510.04091*, 2025.

Zixi Wei, Senlin Shu, Yuzhou Cao, Hongxin Wei, Bo An, and Lei Feng. Consistent multi-class classification from multiple unlabeled datasets. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Atıf Eren Yarıkan, Can Örer, Volkan Akyıldız, Zeki Kuş, Musa Aydin, Kerim Erhan Palaoğlu, Said İncir, Kemal Baysal, Cemal Özçelik, Berna Kiraz, and Alper Kiraz. A large-scale peripheral blood cell dataset for automated hematological analysis. *Scientific Data*, 13(1):417, 2026. doi: 10.1038/s41597-026-06761-y. URL https://doi.org/10.1038/s41597-026-06761-y.

Felix X Yu, Krzysztof Choromanski, Sanjiv Kumar, Tony Jebara, and Shih-Fu Chang. On learning from label proportions. *arXiv preprint arXiv:1402.5902*, 2014.

Xiyu Yu, Tongliang Liu, Mingming Gong, and Dacheng Tao. Learning with biased complementary labels. In *Proceedings of the European Conference on Computer Vision*, pages 68–83, 2018.

Jianxin Zhang, Yutong Wang, and Clayton Scott. Learning from label proportions by learning with label noise. In *Advances in Neural Information Processing Systems*, pages 26933–26942, 2022.

A Notation and Terminology

For convenience, Tables 4 and 5 summarize the abbreviations and main notation used throughout the paper and this appendix.

Table 4: Notation used in the paper.

Symbol	Meaning
$\mathcal{X}$	Input space.
$\mathcal{Y} = \{1, \dots, q\}$	Label space.
q	Number of classes.
m	Bag size.
Δ^{q-1}	The $(q-1)$-dimensional probability simplex $\{(p_1, \dots, p_q) \in \mathbb{R}^q \mid p_k \geq 0,\ \sum_{k=1}^{q} p_k = 1\}$.
$x_j \in \mathcal{X}$	The j-th instance in a bag.
$y_j \in \mathcal{Y}$	Latent class label of x_j.
$\mathbf{B} = (x_1, \dots, x_m) \in \mathcal{X}^m$	Bag of m instances.
$\boldsymbol{y} = (y_1, \dots, y_m) \in \mathcal{Y}^m$	Latent label sequence associated with $\mathbf{B}$.
$\mathcal{Y}^{m/\boldsymbol{\alpha}}$	Set of label sequences in $\mathcal{Y}^m$ whose empirical class proportions equal $\boldsymbol{\alpha}$.
$\mathbb{I}\{\cdot\} : \{\text{statements}\} \to \{0, 1\}$	Indicator function, which returns 1 when its argument is true and 0 otherwise.
$\alpha_k \in [0, 1]$	Proportion of class k in a bag, defined by $\alpha_k = \frac{1}{m} \sum_{j=1}^{m} \mathbb{I}\{y_j = k\}$.
$\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_q) \in \Delta^{q-1}$	Bag-level class-proportion vector.
$\boldsymbol{z} = m\boldsymbol{\alpha} \in \{0, \dots, m\}^q$	Bag-level class-count vector.
$\boldsymbol{e}^{y_j} \in \{0, 1\}^q \subset \mathbb{R}^q$	One-hot basis vector corresponding to label y_j.
$p : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$	Joint data distribution over instance-label pairs.
$p(\cdot \mid x) : \mathcal{Y} \to [0, 1]$	Class-posterior map for a fixed instance $x \in \mathcal{X}$.
$\mathcal{F} \subseteq \left\{ \boldsymbol{f} : \mathcal{X} \to \Delta^{q-1} \right\}$	Hypothesis class of score-based classifiers.
$\boldsymbol{f} : \mathcal{X} \to \Delta^{q-1}$	Score-based classifier, with $\boldsymbol{f} \in \mathcal{F}$.
$\boldsymbol{f}^* : \mathcal{X} \to \Delta^{q-1}$	Bayes-optimal classifier.
$p_j = \boldsymbol{f}(x_j) \in \Delta^{q-1}$	Predicted class-probability vector for instance x_j.
$\ell : \Delta^{q-1} \times \mathcal{Y} \to \mathbb{R}_+$	Classification-calibrated surrogate loss used in ordinary supervised learning.
$\mathbf{M}$	Generic normalized population moment object; for example, $\mathbf{M}_{\boldsymbol{\alpha}}^{(1)}$ is the normalized first-order population moment and $\mathbf{M}^{(2)}$ denotes the normalized bag-level second-order moment matrix.
$(m)_r$	Falling factorial $m(m-1)\cdots(m-r+1)$.
$[m]_{\neq}^{r}$	Set of ordered r-tuples of distinct indices from $[m]$.
$\widehat{\mathbf{M}}_{\boldsymbol{\alpha}}^{(r)} \in (\mathbb{R}^q)^{\otimes r}$	Normalized r-th order factorial moment tensor computed from $\boldsymbol{\alpha}$.
$\widehat{\mathbf{M}}_{\boldsymbol{f}}^{(r)} : \mathcal{X}^m \to (\mathbb{R}^q)^{\otimes r}$	Classifier-induced normalized r-th order factorial moment map; $\widehat{\mathbf{M}}_{\boldsymbol{f}}^{(r)}(\mathbf{B})$ is its value on bag $\mathbf{B}$.
$\mathcal{L}^{(r)}$	Order-r discrepancy loss between normalized target and classifier-induced moments; for $r = 1$ it acts on class-proportion vectors, and for general r it acts on tensors in $(\mathbb{R}^q)^{\otimes r}$.
$R_{\mathrm{MM}}^{(s)} : \mathcal{F} \to \mathbb{R}_+$	Population multi-order normalized moment-matching risk using moments up to order s.
$\widehat{R}_{\mathrm{MM}}^{(s)} : \mathcal{F} \to \mathbb{R}_+$	Empirical multi-order normalized moment-matching risk using moments up to order s.
$s \in \{1, \dots, m\}$	Maximum moment order used for training.
$\lambda_r > 0$	Weight of the r-th order term, with $\sum_{r=1}^{s} \lambda_r = 1$.
$\mathfrak{R}$	Rademacher-complexity symbol.
$\mathfrak{R}_n(\mathcal{F}_c)$	Expected Rademacher complexity of the coordinate function class $\mathcal{F}_c$.
$\boldsymbol{k}$	Multiplicity vector recording how often each class index appears in an ordered index tuple.
$\mathcal{K}_{q,r}$	Compressed multiplicity-state space at order r, defined by $\mathcal{K}_{q,r} = \{\boldsymbol{k} \in \mathbb{N}^q \mid \sum_{c=1}^{q} k_c = r\}$.
N_r	Cardinality of the compressed multiplicity-state space, namely $N_r = \lvert\mathcal{K}_{q,r}\rvert = \binom{q+r-1}{r}$.
$A_t^{(d)} : \mathcal{K}_{q,d} \to \mathbb{R}$	Coefficient map in the truncated generating polynomial; $A_t^{(d)}(\boldsymbol{k})$ is the coefficient associated with state $\boldsymbol{k}$.

Table 5: Abbreviations used in the paper.

Abbreviation	Meaning
WSL	Weakly supervised learning
LLP	Learning from label proportions
PMD	Poisson multinomial distribution
MM	Moment matching
LLP-MM	LLP via Moment Matching

B First-Order Moment Matching

Based on the connection between LLP and the PMD, we propose a general framework which matches the empirical factorial moments with classifier-induced moments. We present the first-order moment matching approach, which has been referred to as proportion matching in prior work [Yu et al., 2014, Ardehaly and Culotta, 2017]. Specifically, let $\mathbf{M}_{\boldsymbol{\alpha}}^{(1)} = [M_1^{(1)}, M_2^{(1)}, \ldots, M_q^{(1)}]$ denote the normalized first-order moment of $\boldsymbol{z}$. Then, we have

$$M_k^{(1)} = \frac{1}{m}\sum_{j=1}^{m} \mathbb{E}_{p(y_j|x_j)}\big[\mathbb{I}\{y_j = k\}\big] = \frac{1}{m}\sum_{j=1}^{m} p(y_j = k \mid x_j). \tag{16}$$

However, since the instance-level posterior probabilities are unknown, we approximate $\mathbf{M}_{\boldsymbol{\alpha}}^{(1)}$ with the *first-order empirical moment* $\widehat{\mathbf{M}}_{\boldsymbol{\alpha}}^{(1)}$, where the k-th element is

$$\Big[\widehat{\mathbf{M}}_{\boldsymbol{\alpha}}^{(1)}\Big]_k = \frac{1}{m}\sum_{j=1}^{m}[\mathbb{I}\{y_j = k\}] = \frac{z_k}{m} = \alpha_k, \tag{17}$$

and z_k is the k-th dimension of $\boldsymbol{z}$. Apart from empirical moments, we can also use the classifier outputs to approximate $\mathbf{M}_{\boldsymbol{\alpha}}^{(1)}$. Specifically, we introduce

$$\widehat{\mathbf{M}}_{\boldsymbol{f}}^{(1)}(\mathbf{B}) = \frac{1}{m}\sum_{j=1}^{m} \boldsymbol{f}(x_j) \tag{18}$$

to represent the first-order classifier-induced moment. Finally, we introduce the *first-order moment matching risk* to match the first-order empirical factorial moments with the first-order classifier-induced moment:

$$R_{\mathrm{MM}}^{(1)}(\boldsymbol{f}) = \mathbb{E}_{p(\mathbf{B},\boldsymbol{\alpha})}\Big[\mathcal{L}^{(1)}\Big(\boldsymbol{\alpha}, \widehat{\mathbf{M}}_{\boldsymbol{f}}^{(1)}(\mathbf{B})\Big)\Big], \tag{19}$$

where $\mathcal{L}^{(1)} : \Delta^{q-1} \times \Delta^{q-1} \to [0, +\infty]$ measures the discrepancy between the true and predicted class proportions. Notably, the formulation in Equation (19) is the same as the objective of the popular proportion matching strategy [Tsai and Lin, 2020, Ma et al., 2025, Liu et al., 2019] which uses the observed proportions as targets for the predicted mean proportions. Our moment matching framework provides a new perspective for understanding the proportion matching strategy based on the PMD.

C Derivation of the Second-Order Moment from Label Proportions

Recall that the bag-level second-order moment matrix is

$$\mathbf{M}^{(2)} = \frac{1}{m(m-1)}\sum_{i\neq j} \boldsymbol{e}^{y_i}(\boldsymbol{e}^{y_j})^\top. \tag{20}$$

We show that each entry of $\mathbf{M}^{(2)}$ can be computed directly from the label proportion vector $\boldsymbol{\alpha}$.

For any $c, d \in \{1, \ldots, q\}$, the (c, d)-th entry of $\mathbf{M}^{(2)}$ is

$$M_{cd}^{(2)} = \frac{1}{m(m-1)}\sum_{i\neq j}[\boldsymbol{e}^{y_i}]_c[\boldsymbol{e}^{y_j}]_d \tag{21}$$

$$= \frac{1}{m(m-1)}\sum_{i\neq j}\mathbb{I}\{y_i = c\}\mathbb{I}\{y_j = d\}. \tag{22}$$

Let

$$n_c := \sum_{i=1}^{m} \mathbb{I}\{y_i = c\} = m\alpha_c \tag{23}$$

denote the number of instances in the bag belonging to class c. Then we consider two cases.

Case 1: $c \neq d$**.** In this case, the constraint $i \neq j$ is automatically satisfied whenever $\mathbb{I}\{y_i = c\}\mathbb{I}\{y_j = d\} = 1$, because one instance cannot simultaneously belong to two different classes. Hence,

$$\sum_{i \neq j} \mathbb{I}\{y_i = c\}\mathbb{I}\{y_j = d\} = \sum_{i=1}^{m}\sum_{j=1}^{m} \mathbb{I}\{y_i = c\}\mathbb{I}\{y_j = d\} \tag{24}$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{m} \mathbb{I}\{y_i = c\}\mathbb{I}\{y_j = d\} \tag{25}$$

$$= \left(\sum_{i=1}^{m} \mathbb{I}\{y_i = c\}\right)\left(\sum_{j=1}^{m} \mathbb{I}\{y_j = d\}\right) \tag{26}$$

$$= n_c n_d. \tag{27}$$

Substituting $n_c = m\alpha_c$ and $n_d = m\alpha_d$ into Equation (22), we obtain

$$M_{cd}^{(2)} = \frac{n_c n_d}{m(m-1)} = \frac{m\alpha_c\, m\alpha_d}{m(m-1)}, \qquad c \neq d. \tag{28}$$

Case 2: $c = d$**.** Now the constraint $i \neq j$ matters. We have

$$\sum_{i \neq j} \mathbb{I}\{y_i = c\}\mathbb{I}\{y_j = c\} = \left(\sum_{i=1}^{m} \mathbb{I}\{y_i = c\}\right)\left(\sum_{j=1}^{m} \mathbb{I}\{y_j = c\}\right) - \sum_{i=1}^{m} \mathbb{I}\{y_i = c\} \tag{29}$$

$$= n_c^2 - n_c \tag{30}$$

$$= n_c(n_c - 1). \tag{31}$$

Therefore,

$$M_{cc}^{(2)} = \frac{n_c(n_c-1)}{m(m-1)} = \frac{m\alpha_c(m\alpha_c - 1)}{m(m-1)}. \tag{32}$$

Combining the two cases, we obtain

$$M_{cd}^{(2)} = \frac{n_c(n_d - \mathbb{I}\{c=d\})}{m(m-1)} = \frac{m\alpha_c(m\alpha_d - \mathbb{I}\{c=d\})}{m(m-1)}, \qquad c, d \in \{1, \ldots, q\}. \tag{33}$$

This proves that the second-order moment matrix is completely determined by the observed label proportion vector $\boldsymbol{\alpha}$. □

D Proof of Proposition 3.1

Theorem D.1. *Assume that the hypothesis class $\mathcal{F}$ is expressive enough to contain the Bayes-optimal classifier, i.e., $\boldsymbol{f}^* \in \mathcal{F}$. For each $r \in \{1, \ldots, s\}$, assume that for any $\mathcal{T}_r$-valued random variable Z, its expectation $\mathbb{E}[Z]$ is the unique solution of $\min_{v \in \mathcal{T}_r} \mathbb{E}\left[\mathcal{L}^{(r)}(Z, v)\right]$. Then, for any minimizer $\boldsymbol{f} \in \mathcal{F}_{\mathrm{MM}}^{(s)}$ of Equation (13), for $p(\mathbf{B})$-almost every bag $\mathbf{B}$ and every $r \in \{1, \ldots, s\}$ such that $\lambda_r > 0$, we have*

$$\widehat{\mathbf{M}}_{\boldsymbol{f}}^{(r)}(\mathbf{B}) = \widehat{\mathbf{M}}_{\boldsymbol{f}^*}^{(r)}(\mathbf{B}) \in (\mathbb{R}^q)^{\otimes r}. \tag{34}$$

That is, every minimizer of Equation (13) exactly matches the moment tensors induced by the Bayes-optimal classifier for all positively weighted orders, for $p(\mathbf{B})$-almost every bag $\mathbf{B}$.

Theorem D.1 shows that incorporating multi-order moments as training supervision enforces agreement with the Bayes-optimal classifier-induced moments at correspondingly higher orders.

Proof of Theorem D.1. For each order $r \in \{1, \dots, s\}$, define the random tensor

$$Z_r := \widehat{\mathbf{M}}_{\boldsymbol{\alpha}}^{(r)}. \tag{35}$$

We first show that, conditioned on a bag $\mathbf{B} = (x_1, \dots, x_m)$, its conditional expectation is exactly the classifier-induced moment tensor of the Bayes-optimal classifier:

$$\mathbb{E}[Z_r \mid \mathbf{B}] = \widehat{\mathbf{M}}_{\boldsymbol{f}^*}^{(r)}(\mathbf{B}). \tag{36}$$

Indeed, by Equation (10),

$$Z_r = \frac{1}{(m)_r} \sum_{(j_1,\dots,j_r)\in[m]^r_{\neq}} \boldsymbol{e}^{y_{j_1}} \otimes \cdots \otimes \boldsymbol{e}^{y_{j_r}}. \tag{37}$$

Taking conditional expectation given $\mathbf{B}$ and using both the conditional independence of $y_1, \dots, y_m$ given $\mathbf{B}$ and the conditional marginal identities $p(y_j \mid \mathbf{B}) = p(y_j \mid x_j)$ assumed in Equation (3), we obtain

$$\mathbb{E}[\boldsymbol{e}^{y_{j_1}} \otimes \cdots \otimes \boldsymbol{e}^{y_{j_r}} \mid \mathbf{B}] = \bigotimes_{\ell=1}^{r} \mathbb{E}[\boldsymbol{e}^{y_{j_\ell}} \mid x_{j_\ell}]. \tag{38}$$

By definition of the Bayes-optimal classifier,

$$\mathbb{E}[\boldsymbol{e}^{y_j} \mid x_j] = p(\cdot \mid x_j) = \boldsymbol{f}^*(x_j). \tag{39}$$

Hence

$$\mathbb{E}[\boldsymbol{e}^{y_{j_1}} \otimes \cdots \otimes \boldsymbol{e}^{y_{j_r}} \mid \mathbf{B}] = \boldsymbol{f}^*(x_{j_1}) \otimes \cdots \otimes \boldsymbol{f}^*(x_{j_r}), \tag{40}$$

and therefore

$$\mathbb{E}[Z_r \mid \mathbf{B}] = \frac{1}{(m)_r} \sum_{(j_1,\dots,j_r)\in[m]^r_{\neq}} \boldsymbol{f}^*(x_{j_1}) \otimes \cdots \otimes \boldsymbol{f}^*(x_{j_r}) = \widehat{\mathbf{M}}_{\boldsymbol{f}^*}^{(r)}(\mathbf{B}), \tag{41}$$

which proves Equation (36).

Now let $\boldsymbol{f} \in \mathcal{F}_{\mathrm{MM}}^{(s)}$ be any minimizer of Equation (13). For each r, define

$$V_r(\mathbf{B}) := \widehat{\mathbf{M}}_{\boldsymbol{f}}^{(r)}(\mathbf{B}). \tag{42}$$

Then the multi-order risk can be written as

$$R_{\mathrm{MM}}^{(s)}(\boldsymbol{f}) = \mathbb{E}_{\mathbf{B}}\left[\sum_{r=1}^{s} \lambda_r \, \mathbb{E}\Big[\mathcal{L}^{(r)}(Z_r, V_r(\mathbf{B})) \mid \mathbf{B}\Big]\right]. \tag{43}$$

Fix r, and define

$$g_r(v, \mathbf{B}) := \mathbb{E}\Big[\mathcal{L}^{(r)}(Z_r, v) \mid \mathbf{B}\Big]. \tag{44}$$

Applying the loss assumption of the theorem to the conditional distribution of Z_r given $\mathbf{B}$, the unique minimizer of $v \mapsto g_r(v, \mathbf{B})$ is the conditional expectation $\mathbb{E}[Z_r \mid \mathbf{B}]$. By Equation (36), this unique minimizer is exactly $\widehat{\mathbf{M}}_{\boldsymbol{f}^*}^{(r)}(\mathbf{B})$. Therefore, for every $v \in \mathcal{T}_r$,

$$g_r(v, \mathbf{B}) \geq g_r\Big(\widehat{\mathbf{M}}_{\boldsymbol{f}^*}^{(r)}(\mathbf{B}), \mathbf{B}\Big), \tag{45}$$

with equality if and only if

$$v = \widehat{\mathbf{M}}_{\boldsymbol{f}^*}^{(r)}(\mathbf{B}). \tag{46}$$

Since $\boldsymbol{f}$ is a minimizer of Equation (13) and $\boldsymbol{f}^*$ is feasible, we also have

$$R_{\mathrm{MM}}^{(s)}(\boldsymbol{f}) \leq R_{\mathrm{MM}}^{(s)}(\boldsymbol{f}^*). \tag{47}$$

On the other hand, by the pointwise lower bound above,

$$R_{\mathrm{MM}}^{(s)}(\boldsymbol{f}) - R_{\mathrm{MM}}^{(s)}(\boldsymbol{f}^*) = \mathbb{E}_{\mathbf{B}}\left[\sum_{r=1}^{s} \lambda_r \left[g_r(V_r(\mathbf{B}), \mathbf{B}) - g_r\Big(\widehat{\mathbf{M}}_{\boldsymbol{f}^*}^{(r)}(\mathbf{B}), \mathbf{B}\Big)\right]\right] \tag{48}$$

$$\geq 0. \tag{49}$$

Therefore,

$$\mathbb{E}_{\mathbf{B}}\left[\sum_{r=1}^{s}\lambda_r\left[g_r(V_r(\mathbf{B}),\mathbf{B})-g_r\left(\widehat{\mathbf{M}}_{\boldsymbol{f}^*}^{(r)}(\mathbf{B}),\mathbf{B}\right)\right]\right]=0. \tag{50}$$

The integrand is nonnegative almost surely, so it must vanish almost surely. Hence, for every r with $\lambda_r > 0$,

$$g_r(V_r(\mathbf{B}),\mathbf{B})=g_r\left(\widehat{\mathbf{M}}_{\boldsymbol{f}^*}^{(r)}(\mathbf{B}),\mathbf{B}\right) \tag{51}$$

for $p(\mathbf{B})$-almost every bag $\mathbf{B}$. By the uniqueness of the minimizer of $g_r(\cdot,\mathbf{B})$, this implies

$$V_r(\mathbf{B})=\widehat{\mathbf{M}}_{\boldsymbol{f}^*}^{(r)}(\mathbf{B}) \tag{52}$$

for $p(\mathbf{B})$-almost every bag $\mathbf{B}$ and every r such that $\lambda_r > 0$. Equivalently,

$$\widehat{\mathbf{M}}_{\boldsymbol{f}}^{(r)}(\mathbf{B})=\widehat{\mathbf{M}}_{\boldsymbol{f}^*}^{(r)}(\mathbf{B}),\qquad \text{for } p(\mathbf{B})\text{-almost every bag } \mathbf{B} \text{ and every } r \text{ such that } \lambda_r>0. \tag{53}$$

This proves that any minimizer $\boldsymbol{f}$ induces exactly the same moment tensors as $\boldsymbol{f}^*$ for all orders with positive weight. □

Proof of Proposition 3.1. By Theorem D.1, any $\boldsymbol{f} \in \mathcal{F}_{\mathrm{MM}}^{(s_2)}$ matches the Bayes-optimal classifier-induced moment tensors for every order $r \in \{1,\dots,s_2\}$, since $\lambda_r > 0$ for all such r. In particular, it matches these moment tensors for every order $r \in \{1,\dots,s_1\}$. Therefore, $\boldsymbol{f}$ also satisfies the moment-matching conditions required for membership in $\mathcal{F}_{\mathrm{MM}}^{(s_1)}$. Hence,

$$\mathcal{F}_{\mathrm{MM}}^{(s_2)} \subseteq \mathcal{F}_{\mathrm{MM}}^{(s_1)}.$$

□

E Generalization Analysis

Definition E.1 (Loss-composed hypothesis class and its Rademacher complexity)**.** For each order $r \in \{1,\dots,s\}$, define the bag-level hypothesis class

$$\mathcal{G}^{(r)} := \{(\mathbf{B},\boldsymbol{\alpha}) \mapsto \mathcal{L}^{(r)}\left(\widehat{\mathbf{M}}_{\boldsymbol{\alpha}}^{(r)},\widehat{\mathbf{M}}_{\boldsymbol{f}}^{(r)}(\mathbf{B})\right) :\ \boldsymbol{f}\in\mathcal{F}\}. \tag{54}$$

Its expected Rademacher complexity is defined as

$$\mathfrak{R}_n^{(r)}(\mathcal{G}^{(r)}) := \mathbb{E}_{\mathcal{D}}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\boldsymbol{f}\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\,\mathcal{L}^{(r)}\left(\widehat{\mathbf{M}}_{\boldsymbol{\alpha}_i}^{(r)},\widehat{\mathbf{M}}_{\boldsymbol{f}}^{(r)}(\mathbf{B}_i)\right)\right], \tag{55}$$

where $\mathcal{D} = \{(\mathbf{B}_i,\boldsymbol{\alpha}_i)\}_{i=1}^n$ consists of n i.i.d. samples from $p(\mathbf{B},\boldsymbol{\alpha})$, and $\sigma_1,\dots,\sigma_n$ are i.i.d. Rademacher variables.

Theorem E.2 (Generic generalization bound via loss-composed classes)**.** *Assume that for each* $r \in \{1,\dots,s\}$,

$$0 \le \mathcal{L}^{(r)}\left(\widehat{\mathbf{M}}_{\boldsymbol{\alpha}}^{(r)},\widehat{\mathbf{M}}_{\boldsymbol{f}}^{(r)}(\mathbf{B})\right) \le B_r \tag{56}$$

for all $(\mathbf{B},\boldsymbol{\alpha})$ *and all* $\boldsymbol{f}\in\mathcal{F}$*. Then, for any* $\delta\in(0,1)$*, with probability at least* $1-\delta$*,*

$$\sup_{\boldsymbol{f}\in\mathcal{F}}|R_{\mathrm{MM}}^{(s)}(\boldsymbol{f})-\widehat{R}_{\mathrm{MM}}^{(s)}(\boldsymbol{f})| \le 2\sum_{r=1}^{s}\lambda_r\,\mathfrak{R}_n^{(r)}(\mathcal{G}^{(r)})+\left(\sum_{r=1}^{s}\lambda_r B_r\right)\sqrt{\frac{\log(2/\delta)}{2n}}. \tag{57}$$

Proof. Define the aggregated loss class

$$\mathcal{H} := \{(\mathbf{B},\boldsymbol{\alpha}) \mapsto \sum_{r=1}^{s}\lambda_r\mathcal{L}^{(r)}\left(\widehat{\mathbf{M}}_{\boldsymbol{\alpha}}^{(r)},\widehat{\mathbf{M}}_{\boldsymbol{f}}^{(r)}(\mathbf{B})\right) :\ \boldsymbol{f}\in\mathcal{F}\}. \tag{58}$$

Then

$$R_{\mathrm{MM}}^{(s)}(\boldsymbol{f})=\mathbb{E}[h_{\boldsymbol{f}}(\mathbf{B},\boldsymbol{\alpha})],\qquad \widehat{R}_{\mathrm{MM}}^{(s)}(\boldsymbol{f})=\frac{1}{n}\sum_{i=1}^{n}h_{\boldsymbol{f}}(\mathbf{B}_i,\boldsymbol{\alpha}_i), \tag{59}$$

where $h_{\boldsymbol{f}} \in \mathcal{H}$.

Define the two one-sided deviations

$$U(\mathcal{D}) := \sup_{\boldsymbol{f}\in\mathcal{F}} \Big(R^{(s)}_{\mathrm{MM}}(\boldsymbol{f}) - \widehat{R}^{(s)}_{\mathrm{MM}}(\boldsymbol{f}) \Big),$$
$$V(\mathcal{D}) := \sup_{\boldsymbol{f}\in\mathcal{F}} \Big(\widehat{R}^{(s)}_{\mathrm{MM}}(\boldsymbol{f}) - R^{(s)}_{\mathrm{MM}}(\boldsymbol{f}) \Big).$$

Applying the standard one-sided symmetrization inequality [Mohri et al., 2018] separately in the two directions gives

$$\mathbb{E}_{\mathcal{D}} U(\mathcal{D}) \le 2\,\mathfrak{R}_n(\mathcal{H}), \qquad \mathbb{E}_{\mathcal{D}} V(\mathcal{D}) \le 2\,\mathfrak{R}_n(\mathcal{H}), \tag{60}$$

where

$$\mathfrak{R}_n(\mathcal{H}) = \mathbb{E}_{\mathcal{D}}\mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{\boldsymbol{f}\in\mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i\, h_{\boldsymbol{f}}(\mathbf{B}_i, \boldsymbol{\alpha}_i) \right]. \tag{61}$$

The reverse direction uses the symmetry of the Rademacher signs: replacing $\boldsymbol{\sigma}$ by $-\boldsymbol{\sigma}$ leaves their distribution unchanged. Thus both inequalities use the same complexity defined above, without an absolute value inside the supremum. Using the definition of $h_{\boldsymbol{f}}$ and the subadditivity of the supremum,

$$\begin{aligned}
\mathfrak{R}_n(\mathcal{H}) &= \mathbb{E}_{\mathcal{D}}\mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{\boldsymbol{f}\in\mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i \sum_{r=1}^{s} \lambda_r \mathcal{L}^{(r)} \Big(\widehat{\mathbf{M}}^{(r)}_{\boldsymbol{\alpha}_i}, \widehat{\mathbf{M}}^{(r)}_{\boldsymbol{f}}(\mathbf{B}_i) \Big) \right] \\
&\le \sum_{r=1}^{s} \lambda_r \mathbb{E}_{\mathcal{D}}\mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{\boldsymbol{f}\in\mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i \mathcal{L}^{(r)} \Big(\widehat{\mathbf{M}}^{(r)}_{\boldsymbol{\alpha}_i}, \widehat{\mathbf{M}}^{(r)}_{\boldsymbol{f}}(\mathbf{B}_i) \Big) \right] \\
&= \sum_{r=1}^{s} \lambda_r\, \mathfrak{R}^{(r)}_n(\mathcal{G}^{(r)}).
\end{aligned}$$

Hence,

$$\max\{\mathbb{E}_{\mathcal{D}} U(\mathcal{D}), \mathbb{E}_{\mathcal{D}} V(\mathcal{D})\} \le 2 \sum_{r=1}^{s} \lambda_r\, \mathfrak{R}^{(r)}_n(\mathcal{G}^{(r)}). \tag{62}$$

Now let

$$C := \sum_{r=1}^{s} \lambda_r B_r. \tag{63}$$

For every $\boldsymbol{f} \in \mathcal{F}$, the aggregated loss

$$h_{\boldsymbol{f}}(\mathbf{B}, \boldsymbol{\alpha}) = \sum_{r=1}^{s} \lambda_r \mathcal{L}^{(r)} \Big(\widehat{\mathbf{M}}^{(r)}_{\boldsymbol{\alpha}}, \widehat{\mathbf{M}}^{(r)}_{\boldsymbol{f}}(\mathbf{B}) \Big) \tag{64}$$

satisfies $0 \le h_{\boldsymbol{f}} \le C$. Therefore, if $\mathcal{D}$ and $\mathcal{D}'$ differ in one bag-proportion pair $(\mathbf{B}_i, \boldsymbol{\alpha}_i)$, then

$$\big|U(\mathcal{D}) - U(\mathcal{D}')\big| \le \frac{C}{n}, \qquad \big|V(\mathcal{D}) - V(\mathcal{D}')\big| \le \frac{C}{n}. \tag{65}$$

The independent sample units are the n entire bag-proportion pairs. Applying McDiarmid's inequality [Mohri et al., 2018] separately to $U(\mathcal{D})$ and $V(\mathcal{D})$, with failure probability $\delta/2$ for each, gives

$$\Pr\left(W(\mathcal{D}) > \mathbb{E}_{\mathcal{D}} W(\mathcal{D}) + C\sqrt{\frac{\log(2/\delta)}{2n}} \right) \le \frac{\delta}{2}, \qquad W \in \{U, V\}.$$

By a union bound and the two symmetrization inequalities, with probability at least $1-\delta$, both directions simultaneously satisfy

$$\max\{U(\mathcal{D}), V(\mathcal{D})\} \le 2\,\mathfrak{R}_n(\mathcal{H}) + C\sqrt{\frac{\log(2/\delta)}{2n}}. \tag{66}$$

Finally, using

$$\sup_{\boldsymbol{f}\in\mathcal{F}} \Big| R^{(s)}_{\mathrm{MM}}(\boldsymbol{f}) - \widehat{R}^{(s)}_{\mathrm{MM}}(\boldsymbol{f}) \Big| = \max\{U(\mathcal{D}), V(\mathcal{D})\},$$

together with the subadditivity bound for $\mathfrak{R}_n(\mathcal{H})$ and the definition of C, proves Equation (57). □

Definition E.3 (Coordinate-wise Rademacher complexity of $\mathcal{F}$)**.** For each position $j \in [m]$ and class $c \in [q]$, define the scalar-valued class

$$\mathcal{F}_{j,c} := \{\mathbf{B} = (x_1, \ldots, x_m) \mapsto [\boldsymbol{f}(x_j)]_c : \ \boldsymbol{f} \in \mathcal{F}\}. \tag{67}$$

Its expected Rademacher complexity is

$$\mathfrak{R}_n(\mathcal{F}_{j,c}) := \mathbb{E}_{\{\mathbf{B}_i\}_{i=1}^n} \mathbb{E}_{\{\sigma_i\}_{i=1}^n} \left[\sup_{\boldsymbol{f} \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \sigma_i \, [\boldsymbol{f}(x_{ij})]_c \right]. \tag{68}$$

Lemma E.4 (Lipschitz property of the moment map)**.** *For any $r \in \{1, \ldots, m\}$, any bag $\mathbf{B} = (x_1, \ldots, x_m)$, and any $\boldsymbol{f}, \boldsymbol{f}' \in \mathcal{F}$, we have*

$$\|\widehat{\mathbf{M}}_{\boldsymbol{f}}^{(r)}(\mathbf{B}) - \widehat{\mathbf{M}}_{\boldsymbol{f}'}^{(r)}(\mathbf{B})\|_F \le \frac{r}{m} \sum_{j=1}^m \big\|\boldsymbol{f}(x_j) - \boldsymbol{f}'(x_j)\big\|_2 \le \frac{r}{m} \sum_{j=1}^m \sum_{c=1}^q \big|[\boldsymbol{f}(x_j)]_c - [\boldsymbol{f}'(x_j)]_c\big|, \tag{69}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of an order-r tensor, i.e., for $T = (T_{i_1,\ldots,i_r}) \in \mathbb{R}^{q \times \cdots \times q}$,

$$\|T\|_F := \left(\sum_{i_1,\ldots,i_r} T_{i_1,\ldots,i_r}^2 \right)^{1/2}. \tag{70}$$

Proof. Let $u_j := \boldsymbol{f}(x_j) \in \Delta^{q-1}$ and $v_j := \boldsymbol{f}'(x_j) \in \Delta^{q-1}$. For any ordered r-tuple $(j_1, \ldots, j_r) \in [m]_{\neq}^r$, write

$$U := u_{j_1} \otimes \cdots \otimes u_{j_r}, \qquad V := v_{j_1} \otimes \cdots \otimes v_{j_r}. \tag{71}$$

A telescoping expansion gives

$$U - V = \sum_{t=1}^r v_{j_1} \otimes \cdots \otimes v_{j_{t-1}} \otimes (u_{j_t} - v_{j_t}) \otimes u_{j_{t+1}} \otimes \cdots \otimes u_{j_r}. \tag{72}$$

Using

$$\|a_1 \otimes \cdots \otimes a_r\|_F = \prod_{\ell=1}^r \|a_\ell\|_2 \tag{73}$$

and the fact that $\|u_j\|_2 \le \|u_j\|_1 = 1$ and $\|v_j\|_2 \le \|v_j\|_1 = 1$ for all j, we obtain

$$\|U - V\|_F \le \sum_{t=1}^r \|u_{j_t} - v_{j_t}\|_2. \tag{74}$$

Hence

$$\begin{aligned} \|\widehat{\mathbf{M}}_{\boldsymbol{f}}^{(r)}(\mathbf{B}) - \widehat{\mathbf{M}}_{\boldsymbol{f}'}^{(r)}(\mathbf{B})\|_F &\le \frac{1}{(m)_r} \sum_{(j_1,\ldots,j_r) \in [m]_{\neq}^r} \sum_{t=1}^r \|u_{j_t} - v_{j_t}\|_2 \\ &= \frac{r(m-1)_{r-1}}{(m)_r} \sum_{j=1}^m \|u_j - v_j\|_2 \\ &= \frac{r}{m} \sum_{j=1}^m \|u_j - v_j\|_2. \end{aligned}$$

The second inequality in Equation (69) follows from $\|a\|_2 \le \|a\|_1 = \sum_{c=1}^q |a_c|$. □

Lemma E.5 (Coordinate-wise contraction)**.** *Let $T \subset (\mathbb{R}^d)^n$. For each $i \in [n]$, let $\Phi_i : \mathbb{R}^d \to \mathbb{R}$ satisfy $\Phi_i(\mathbf{0}) = 0$ and*

$$|\Phi_i(u) - \Phi_i(v)| \le L \|u - v\|_1 \qquad \textit{for all } u, v \in \mathbb{R}^d. \tag{75}$$

Then

$$\mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{t \in T} \frac{1}{n} \sum_{i=1}^n \sigma_i \Phi_i(t_i) \right] \le \sqrt{2d}\, L \sum_{\ell=1}^d \mathbb{E}_{\boldsymbol{\tau}^{(\ell)}} \left[\sup_{t \in T} \frac{1}{n} \sum_{i=1}^n \tau_i^{(\ell)} t_{i,\ell} \right]. \tag{76}$$

Proof. Since $\|x\|_1 \le \sqrt{d}\,\|x\|_2$ for every $x \in \mathbb{R}^d$, the assumption implies

$$|\Phi_i(u) - \Phi_i(v)| \le L\sqrt{d}\,\|u - v\|_2 \qquad \text{for all } u, v \in \mathbb{R}^d. \tag{77}$$

Hence each Φ_i is $L\sqrt{d}$-Lipschitz with respect to the Euclidean norm.

Applying the vector contraction inequality [Maurer, 2016] with respect to $\|\cdot\|_2$ yields

$$\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{t\in T}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\Phi_i(t_i)\right] \le \sqrt{2d}\,L\,\mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{t\in T}\frac{1}{n}\sum_{i=1}^{n}\sum_{\ell=1}^{d}\epsilon_{i,\ell}t_{i,\ell}\right], \tag{78}$$

where $\epsilon_{i,\ell}$ are i.i.d. Rademacher variables.

Finally, by subadditivity of the supremum,

$$\mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{t\in T}\frac{1}{n}\sum_{i=1}^{n}\sum_{\ell=1}^{d}\epsilon_{i,\ell}t_{i,\ell}\right] \le \sum_{\ell=1}^{d}\mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{t\in T}\frac{1}{n}\sum_{i=1}^{n}\epsilon_{i,\ell}t_{i,\ell}\right] \tag{79}$$

$$= \sum_{\ell=1}^{d}\mathbb{E}_{\boldsymbol{\tau}^{(\ell)}}\left[\sup_{t\in T}\frac{1}{n}\sum_{i=1}^{n}\tau_i^{(\ell)}t_{i,\ell}\right]. \tag{80}$$

This proves the claim. □

Proposition E.6 (Reduction to the base class $\mathcal{F}$)**.** *Assume that for each $r \in \{1,\dots,s\}$, the loss*

$$B \mapsto \mathcal{L}^{(r)}(A, B) \tag{81}$$

is L_r-Lipschitz with respect to the Frobenius norm, uniformly over all admissible target tensors A. Then

$$\mathfrak{R}_n^{(r)}(\mathcal{G}^{(r)}) \le \sqrt{2m}\,\frac{L_r\,r}{m}\sum_{j=1}^{m}\sum_{c=1}^{q}\mathfrak{R}_n(\mathcal{F}_{j,c}). \tag{82}$$

Proof. Fix a sample $\{(\mathbf{B}_i, \boldsymbol{\alpha}_i)\}_{i=1}^n$. For a bag $\mathbf{B} = (x_1,\dots,x_m)$ and a classifier $\boldsymbol{f}$, write

$$\boldsymbol{u}_j = \boldsymbol{f}(x_j) \in \Delta^{q-1}, \qquad \boldsymbol{u} = (\boldsymbol{u}_1,\dots,\boldsymbol{u}_m) \in (\mathbb{R}^q)^m.$$

We first show that the classifier-induced order-r factorial moment tensor is $r/\sqrt{m}$-Lipschitz as a function of $\boldsymbol{u}$ with respect to the Euclidean norm. For $\boldsymbol{u}, \boldsymbol{v} \in (\Delta^{q-1})^m$, define

$$\widehat{\mathbf{M}}^{(r)}(\boldsymbol{u}) = \frac{1}{(m)_r}\sum_{(j_1,\dots,j_r)\in[m]^r_{\neq}} \boldsymbol{u}_{j_1}\otimes\cdots\otimes\boldsymbol{u}_{j_r},$$

where $[m]^r_{\neq}$ denotes the set of ordered r-tuples of distinct indices. By the triangle inequality and a telescoping expansion of tensor products,

$$\left\|\widehat{\mathbf{M}}^{(r)}(\boldsymbol{u}) - \widehat{\mathbf{M}}^{(r)}(\boldsymbol{v})\right\|_F \le \frac{1}{(m)_r}\sum_{(j_1,\dots,j_r)\in[m]^r_{\neq}} \left\|\boldsymbol{u}_{j_1}\otimes\cdots\otimes\boldsymbol{u}_{j_r} - \boldsymbol{v}_{j_1}\otimes\cdots\otimes\boldsymbol{v}_{j_r}\right\|_F$$

$$\le \frac{1}{(m)_r}\sum_{(j_1,\dots,j_r)\in[m]^r_{\neq}}\sum_{\ell=1}^{r}\left\|\boldsymbol{u}_{j_\ell} - \boldsymbol{v}_{j_\ell}\right\|_2. \tag{83}$$

The second inequality uses the fact that $\|\boldsymbol{u}_j\|_2 \le 1$ and $\|\boldsymbol{v}_j\|_2 \le 1$ for probability vectors. For each fixed index $j \in [m]$ and each position $\ell \in \{1,\dots,r\}$, the number of ordered tuples $(j_1,\dots,j_r) \in [m]^r_{\neq}$ satisfying $j_\ell = j$ is $(m-1)_{r-1}$. Therefore,

$$\frac{1}{(m)_r}\sum_{(j_1,\dots,j_r)\in[m]^r_{\neq}}\sum_{\ell=1}^{r}\left\|\boldsymbol{u}_{j_\ell} - \boldsymbol{v}_{j_\ell}\right\|_2 = \frac{r(m-1)_{r-1}}{(m)_r}\sum_{j=1}^{m}\left\|\boldsymbol{u}_j - \boldsymbol{v}_j\right\|_2$$

$$= \frac{r}{m}\sum_{j=1}^{m}\left\|\boldsymbol{u}_j - \boldsymbol{v}_j\right\|_2. \tag{84}$$

Applying Cauchy–Schwarz over the m bag instances gives

$$\sum_{j=1}^{m} \left\|\boldsymbol{u}_j - \boldsymbol{v}_j\right\|_2 \leq \sqrt{m} \left(\sum_{j=1}^{m} \left\|\boldsymbol{u}_j - \boldsymbol{v}_j\right\|_2^2 \right)^{1/2} = \sqrt{m}\, \|\boldsymbol{u} - \boldsymbol{v}\|_2. \tag{85}$$

Combining this with Equations (83) and (84), we obtain

$$\left\| \widehat{\mathbf{M}}^{(r)}(\boldsymbol{u}) - \widehat{\mathbf{M}}^{(r)}(\boldsymbol{v}) \right\|_F \leq \frac{r}{\sqrt{m}} \|\boldsymbol{u} - \boldsymbol{v}\|_2. \tag{86}$$

Now define, for each training bag i,

$$\Phi_i^{(r)}(\boldsymbol{u}) = \mathcal{L}^{(r)}\left(\widehat{\mathbf{M}}_{\boldsymbol{\alpha}_i}^{(r)}, \widehat{\mathbf{M}}^{(r)}(\boldsymbol{u}) \right).$$

By the assumed L_r-Lipschitzness of $\mathcal{L}^{(r)}$ in its second argument and by Equation (86), each $\Phi_i^{(r)}$ is $L_r\, r/\sqrt{m}$-Lipschitz with respect to the Euclidean norm on $(\mathbb{R}^q)^m$. Therefore, by Maurer's vector contraction inequality [Maurer, 2016],

$$\begin{aligned} \mathfrak{R}_n^{(r)}(\mathcal{G}^{(r)}) &\leq \sqrt{2}\, \frac{L_r\, r}{\sqrt{m}}\, \mathbb{E}_\sigma \left[\sup_{\boldsymbol{f} \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{c=1}^{q} \sigma_{i,j,c} \boldsymbol{f}_c(x_{i,j}) \right] \\ &\leq \sqrt{2}\, \frac{L_r\, r}{\sqrt{m}} \sum_{j=1}^{m} \sum_{c=1}^{q} \mathfrak{R}_n(\mathcal{F}_{j,c}) \\ &= \sqrt{2m}\, \frac{L_r\, r}{m} \sum_{j=1}^{m} \sum_{c=1}^{q} \mathfrak{R}_n(\mathcal{F}_{j,c}). \end{aligned} \tag{87}$$

This proves the desired bound. □

Corollary E.7 (Generalization bound in terms of $\mathcal{F}$)**.** *Under the assumptions of Theorem E.2 and Proposition E.6, for any $\delta \in (0,1)$, with probability at least $1-\delta$,*

$$\begin{aligned} &\sup_{\boldsymbol{f} \in \mathcal{F}} \left| R_{\mathrm{MM}}^{(s)}(\boldsymbol{f}) - \widehat{R}_{\mathrm{MM}}^{(s)}(\boldsymbol{f}) \right| \\ &\leq \frac{2\sqrt{2m}}{m} \left(\sum_{r=1}^{s} \lambda_r L_r\, r \right) \left(\sum_{j=1}^{m} \sum_{c=1}^{q} \mathfrak{R}_n(\mathcal{F}_{j,c}) \right) \\ &\quad + \left(\sum_{r=1}^{s} \lambda_r B_r \right) \sqrt{\frac{\log(2/\delta)}{2n}}. \end{aligned} \tag{88}$$

Proof. Substitute the bound in Proposition E.6 into Theorem E.2. □

Remark E.8 (Exchangeable bag coordinates). Under our bag-generation mechanism, the instances within each bag need not be independent. If the construction is permutation invariant, or if each bag is randomly permuted afterward, its coordinates are exchangeable. Thus, for every fixed $j \in [m]$, x_j has the same marginal distribution, although the coordinates may remain dependent. Hence, for each class $c \in [q]$, the classes $\mathcal{F}_{j,c}$ have the same expected Rademacher complexity:

$$\mathfrak{R}_n(\mathcal{F}_{1,c}) = \cdots = \mathfrak{R}_n(\mathcal{F}_{m,c}). \tag{89}$$

Consequently, the complexity term in Corollary E.7 can be simplified.

Definition E.9 (Instance-level Rademacher complexity)**.** For each class $c \in [q]$, define the scalar-valued instance-level class

$$\mathcal{F}_c := \{ x \mapsto [\boldsymbol{f}(x)]_c \mid \boldsymbol{f} \in \mathcal{F} \}. \tag{90}$$

Its expected Rademacher complexity is

$$\mathfrak{R}_n(\mathcal{F}_c) := \mathbb{E}_{\{x_i\}_{i=1}^n} \mathbb{E}_{\{\sigma_i\}_{i=1}^n} \left[\sup_{\boldsymbol{f} \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i \, [\boldsymbol{f}(x_i)]_c \right], \tag{91}$$

where $x_1, \ldots, x_n$ are i.i.d. draws from the common marginal distribution of any bag coordinate.

Corollary E.10 (Simplified generalization bound under exchangeability)**.** *Under the assumptions of Corollary E.7 and the exchangeability condition in Remark E.8, for any $\delta \in (0,1)$, with probability at least $1-\delta$,*

$$\sup_{\boldsymbol{f}\in\mathcal{F}}\left|R^{(s)}_{\mathrm{MM}}(\boldsymbol{f}) - \widehat{R}^{(s)}_{\mathrm{MM}}(\boldsymbol{f})\right| \le 2\sqrt{2m}\left(\sum_{r=1}^{s}\lambda_r L_r\, r\right)\left(\sum_{c=1}^{q}\mathfrak{R}_n(\mathcal{F}_c)\right) + \left(\sum_{r=1}^{s}\lambda_r B_r\right)\sqrt{\frac{\log(2/\delta)}{2n}}. \tag{92}$$

F Computation Analysis

F.1 Coefficient-to-Tensor Correspondence

In this appendix, we justify the correspondence between the compressed coefficients of the multivariate generating polynomial and the entries of the classifier-induced factorial moment tensor.

For a bag $\mathbf{B} = (x_1, \ldots, x_m)$, recall that the classifier-induced normalized r-th order factorial moment tensor is

$$\widehat{\mathbf{M}}^{(r)}_{\boldsymbol{f}}(\mathbf{B}) = \frac{1}{(m)_r}\sum_{(j_1,\ldots,j_r)\in[m]^r_{\neq}} p_{j_1}\otimes\cdots\otimes p_{j_r}, \tag{93}$$

where $p_j = \boldsymbol{f}(x_j) \in \Delta^{q-1}$. Its $(i_1,\ldots,i_r)$-th entry is therefore

$$[\widehat{\mathbf{M}}^{(r)}_{\boldsymbol{f}}(\mathbf{B})]_{i_1,\ldots,i_r} = \frac{1}{(m)_r}\sum_{(j_1,\ldots,j_r)\in[m]^r_{\neq}}\prod_{\ell=1}^{r} p_{j_\ell,i_\ell}. \tag{94}$$

Now define the multivariate generating polynomial

$$G_m(u_1,\ldots,u_q) := \prod_{j=1}^{m}\left(1+\sum_{c=1}^{q} p_{j,c}u_c\right). \tag{95}$$

For $k = (k_1,\ldots,k_q) \in \mathbb{N}^q$ with $|k| = \sum_{c=1}^{q} k_c = r$, let

$$A^{(r)}_m(k) := [u^k]\, G_m(u_1,\ldots,u_q), \qquad u^k := \prod_{c=1}^{q} u_c^{k_c}. \tag{96}$$

We will show that $A^{(r)}_m(k)$ determines all tensor entries whose multiplicity vector is k.

Proposition F.1 (Coefficient-to-tensor correspondence)**.** *Fix an ordered index tuple $(i_1,\ldots,i_r) \in [q]^r$, and let*

$$k_c := \#\{\ell \in [r] \mid i_\ell = c\}, \qquad c = 1,\ldots,q, \tag{97}$$

be its multiplicity vector. Then

$$[\widehat{\mathbf{M}}^{(r)}_{\boldsymbol{f}}(\mathbf{B})]_{i_1,\ldots,i_r} = \frac{\prod_{c=1}^{q} k_c!}{(m)_r}\, A^{(r)}_m(k). \tag{98}$$

Consequently, all tensor entries with the same multiplicity vector are equal, and the compressed coefficients $\{A^{(r)}_m(k)\}_{k\in\mathcal{K}_{q,r}}$ completely determine $\widehat{\mathbf{M}}^{(r)}_{\boldsymbol{f}}(\mathbf{B})$.

Proof. We first expand the generating polynomial in Equation (95). In each factor

$$1 + \sum_{c=1}^{q} p_{j,c}u_c, \tag{99}$$

we either choose the constant term 1, meaning that instance j is not selected, or one of the terms $p_{j,c}u_c$, meaning that instance j is selected and contributes one occurrence of class c.

Therefore, the coefficient of u^k in G_m is obtained by summing over all ways to choose r distinct instances and assign to them class labels whose class-count profile is exactly k. Formally,

$$A_m^{(r)}(k) = \sum_{\substack{S\subseteq[m]\\|S|=r}} \sum_{\phi:S\to[q]\,:\,|\phi^{-1}(c)|=k_c\ \forall c} \prod_{j\in S} p_{j,\phi(j)}. \tag{100}$$

Now fix the ordered tensor index tuple $(i_1,\dots,i_r)$. Let

$$\mathfrak{S}(i_1,\dots,i_r) \tag{101}$$

denote the set of distinct permutations of the multiset $(i_1,\dots,i_r)$. Its cardinality is

$$|\mathfrak{S}(i_1,\dots,i_r)| = \frac{r!}{\prod_{c=1}^q k_c!}. \tag{102}$$

For any ordered tuple of distinct instance indices $(j_1,\dots,j_r)\in[m]^r_{\neq}$, define

$$\Psi(j_1,\dots,j_r) := \prod_{\ell=1}^r p_{j_\ell,i_\ell}. \tag{103}$$

Then Equation (94) becomes

$$[\widehat{\mathbf{M}}_{\boldsymbol{f}}^{(r)}(\mathbf{B})]_{i_1,\dots,i_r} = \frac{1}{(m)_r} \sum_{(j_1,\dots,j_r)\in[m]^r_{\neq}} \Psi(j_1,\dots,j_r). \tag{104}$$

Next, we compare Equation (100) and Equation (104). Take any subset $S=\{j_1,\dots,j_r\}\subseteq[m]$ of size r. To contribute to $A_m^{(r)}(k)$, we assign class labels to the elements of S so that class c appears exactly k_c times. On the other hand, to contribute to $[\widehat{\mathbf{M}}_{\boldsymbol{f}}^{(r)}(\mathbf{B})]_{i_1,\dots,i_r}$, we must choose an ordered tuple of distinct instances $(j_1,\dots,j_r)$ and match them to the fixed ordered label pattern $(i_1,\dots,i_r)$.

For a fixed subset S and a fixed assignment of labels with multiplicity vector k, there are exactly $\prod_{c=1}^q k_c!$ ordered tuples $(j_1,\dots,j_r)$ that induce the same monomial contribution in Equation (94). Indeed, within each class c, the k_c positions carrying label c may be permuted arbitrarily among themselves without changing the underlying unlabeled assignment counted in Equation (100). Thus, each term counted once in $A_m^{(r)}(k)$ is counted exactly $\prod_{c=1}^q k_c!$ times in the sum defining

$$\sum_{(j_1,\dots,j_r)\in[m]^r_{\neq}} \Psi(j_1,\dots,j_r). \tag{105}$$

Therefore,

$$\sum_{(j_1,\dots,j_r)\in[m]^r_{\neq}} \prod_{\ell=1}^r p_{j_\ell,i_\ell} = \left(\prod_{c=1}^q k_c!\right) A_m^{(r)}(k). \tag{106}$$

Substituting Equation (106) into Equation (94) yields

$$[\widehat{\mathbf{M}}_{\boldsymbol{f}}^{(r)}(\mathbf{B})]_{i_1,\dots,i_r} = \frac{\prod_{c=1}^q k_c!}{(m)_r} A_m^{(r)}(k), \tag{107}$$

which proves Equation (98).

Finally, the right-hand side depends on $(i_1,\dots,i_r)$ only through its multiplicity vector k. Hence all tensor entries with the same multiplicity vector are equal, and the collection of compressed coefficients $\{A_m^{(r)}(k)\}_{k\in\mathcal{K}_{q,r}}$ completely determines the full tensor. This completes the proof. □

F.2 Time and Space Complexity Analysis

A direct evaluation of the order-r loss in Equation (13) requires materializing an order-r tensor with q^r entries, which is computationally prohibitive for large q or r. To reduce both time and memory

costs, we exploit the permutation symmetry of the factorial moment tensors. Both $[\widehat{\mathbf{M}}_{\boldsymbol{\alpha}}^{(r)}]_{i_1,\ldots,i_r}$ and $[\widehat{\mathbf{M}}_{\boldsymbol{f}}^{(r)}(\mathbf{B})]_{i_1,\ldots,i_r}$ depend only on $\boldsymbol{k}$, rather than on the ordering of $(i_1,\ldots,i_r)$. Hence, the full q^r ordered tensor entries can be grouped into the compressed state space

$$\mathcal{K}_{q,r} := \left\{ \boldsymbol{k} \in \mathbb{N}^q \;\middle|\; \sum_{c=1}^{q} k_c = r \right\}, \qquad N_r := \left|\mathcal{K}_{q,r}\right| = \binom{q+r-1}{r}. \tag{108}$$

The second-order case provides the simplest illustration: both $\widehat{\mathbf{M}}_{\boldsymbol{\alpha}}^{(2)}$ and $\widehat{\mathbf{M}}_{\boldsymbol{f}}^{(2)}(\mathbf{B})$ are symmetric $q \times q$ matrices, so the full q^2 entries reduce to only $N_2 = \binom{q+1}{2}$ distinct states. More generally, at order r, this reduces the computation from the full tensor size q^r to the compressed state size N_r.

To compute the classifier-induced states efficiently, we introduce the truncated multivariate generating polynomial

$$G_t(u_1,\ldots,u_q) := \prod_{j=1}^{t} \left(1 + \sum_{c=1}^{q} p_{j,c} u_c \right), \qquad t = 0,1,\ldots,m. \tag{109}$$

For a state $\boldsymbol{k} \in \mathcal{K}_{q,d}$, define

$$A_t^{(d)}(\boldsymbol{k}) := [u^{\boldsymbol{k}}]\, G_t(u_1,\ldots,u_q), \tag{110}$$

where $u^{\boldsymbol{k}} := \prod_{c=1}^{q} u_c^{k_c}$ and $[u^{\boldsymbol{k}}]G_t$ denotes the coefficient of the monomial $u^{\boldsymbol{k}}$ in G_t. Thus, $A_t^{(d)}(\boldsymbol{k})$ is the aggregated coefficient of all order-d terms whose class-count pattern is $\boldsymbol{k}$ after processing the first t instances.

These coefficients are directly related to the classifier-induced moment tensor. Indeed, for any ordered index tuple $(i_1,\ldots,i_r) \in [q]^r$ with multiplicity vector $\boldsymbol{k}$, we have

$$[\widehat{\mathbf{M}}_{\boldsymbol{f}}^{(r)}(\mathbf{B})]_{i_1,\ldots,i_r} = \frac{\prod_{c=1}^{q} k_c!}{(m)_r}\, A_m^{(r)}(\boldsymbol{k}). \tag{111}$$

The derivation of Equation (111) is given in Appendix F.1. Hence, the compressed coefficients $\{A_m^{(r)}(\boldsymbol{k})\}_{\boldsymbol{k}\in\mathcal{K}_{q,r}}$ completely determine the full tensor $\widehat{\mathbf{M}}_{\boldsymbol{f}}^{(r)}(\mathbf{B})$. Since

$$G_t(u) = G_{t-1}(u)\left(1 + \sum_{c=1}^{q} p_{t,c} u_c \right), \tag{112}$$

the coefficients satisfy the recurrence

$$A_t^{(d)}(\boldsymbol{k}) = A_{t-1}^{(d)}(\boldsymbol{k}) + \sum_{c:\, k_c>0} p_{t,c}\, A_{t-1}^{(d-1)}(\boldsymbol{k} - e^c), \tag{113}$$

with initialization

$$A_0^{(0)}(0) = 1, \qquad A_0^{(d)}(\boldsymbol{k}) = 0 \quad \text{for all } d > 0. \tag{114}$$

Starting from this initialization, Equation (113) recursively computes all coefficients $A_t^{(d)}(\boldsymbol{k})$ for $t = 1,\ldots,m$ and $d = 0,\ldots,s$, and in particular yields the desired coefficients $A_m^{(r)}(\boldsymbol{k})$ for all $r \le s$.

Proposition 3.3 (Complexity of the state-compressed computation)**.** *Fix a bag of size m and a maximum order $s \le m$. The compressed computation of all moments up to order s requires $O\big(mq\binom{q+s}{s}\big)$ time and $O\big(\binom{q+s}{s}\big)$ memory per bag.*

Proof. For the target side, at a fixed order r, the compressed representation contains exactly $N_r = \binom{q+r-1}{r}$ states. By Equation (11), the value of each state is available in closed form from the multiplicity vector $\boldsymbol{k}$ and the proportion vector $\boldsymbol{\alpha}$. Computing one such value requires multiplying the contributions of all q classes, and thus costs $O(q)$ time. Therefore, the total cost of computing the compressed target states at order r is $O(qN_r)$, and summing over $r = 1,\ldots,s$ gives

$$O\left(q \sum_{r=1}^{s} N_r \right) = O\left(q \binom{q+s}{s} \right). \tag{115}$$

If all compressed target states up to order s are stored, the required memory is

$$O\left(\sum_{r=1}^{s} N_r\right) = O\left(\binom{q+s}{s}\right). \tag{116}$$

For the classifier-induced side, we use the convention $N_0 = 1$ for the unique degree-0 state. At degree r there are exactly N_r compressed states. In the recurrence Equation (113), when processing one instance, each state of degree r can be extended to at most q states of degree $r+1$ by adding one unit to one coordinate. Hence the update cost per instance is

$$O\left(q \sum_{r=0}^{s-1} N_r\right), \tag{117}$$

and summing over all m instances yields

$$O\left(mq \sum_{r=0}^{s-1} N_r\right) = O\left(mq\binom{q+s-1}{s-1}\right). \tag{118}$$

The corresponding memory cost is the size of the truncated coefficient table over degrees $0, \ldots, s$, namely

$$O\left(\sum_{r=0}^{s} N_r\right) = O\left(\binom{q+s}{s}\right). \tag{119}$$

Combining the target-side and classifier-induced costs gives the stated total time and memory bounds. Finally, since

$$\binom{q+s-1}{s-1} \le \binom{q+s}{s}, \tag{120}$$

the total time can be simplified to

$$O\left(mq\binom{q+s}{s}\right). \tag{121}$$

This completes the proof. □

G Additional Experimental Settings

G.1 Datasets and Bag Construction

Following Ma et al. [2026b], we use three bag construction strategies: (i) **Random Bag**: randomly shuffle instances and partition them into bags; (ii) **Cluster Bag**: sample from feature clusters using Dirichlet mixing weights with total concentration 1 and a base measure proportional to cluster sizes; (iii) **α-First Bag**: sample class proportions from a symmetric Dirichlet distribution with concentration 10 per class, then draw instances accordingly. For all bag generation strategies, the bag size m varies over powers of two, i.e., $m \in \{16, 32, 64, 128\}$.

G.2 Baselines and Implementation Details

Baselines. We compare our method with several representative LLP baselines. PM [Yu et al., 2014, Ardehaly and Culotta, 2017] trains the model by matching the predicted class mean with the observed label proportions. DSQ [Li et al., 2024] builds upon PM by adopting a squared loss formulation with an additional debiasing term. LLP-PVC [Ma et al., 2026b] formulates LLP under a one-vs-rest framework and performs classification on bag-level proportions. LLP-FC [Zhang et al., 2022] approaches LLP from the perspective of label noise learning. ROT [Dulac-Arnold et al., 2019] leverages optimal transport to generate pseudo-labels that satisfy proportion constraints. EasyLLP [Busa-Fekete et al., 2023] derives an unbiased estimator by rewriting the loss function. GeneralUPM [Applebaum et al., 2025] also constructs an unbiased estimator but achieves lower variance. EasyLLP-ABS and GeneralUPM-ABS take the absolute value of the loss function to mitigate the overfitting effect caused by negative loss values. FlowLLP [Rakotomamonjy et al., 2026] learns latent particles by particle flow and uses them as classifier anchors.

Implementation details. We use ResNet-18 for CIFAR-10 and CIFAR-100 and ResNet-34 for miniImageNet, with modified stems and standard data augmentation. Training lasts 500 epochs with 1024 samples per update. The classifier optimizer is SGD with momentum 0.9 and weight decay 0.0005. The initial learning rate is 0.005 for LLP-PVC and 0.05 for the other methods, with linear warmup over 8% of updates followed by quarter-cosine decay. DSQ uses mean squared error; the other methods use cross-entropy as the surrogate loss. LLP-MM uses $s = 8$ for CIFAR-10 and $s = 3$ for CIFAR-100 and miniImageNet, with cross-entropy moment losses and weights $\lambda_r = 1/s$. LLP-FC uses approximate noisy priors, forward-correction weight 1, and entropy weight 0. FlowLLP uses a 50-dimensional latent space, 1000 anchors per class, and anchor-loss weight 0.1, following its official implementation [Rakotomamonjy et al., 2026]. Its particles are optimized for 3000 steps using Adam with a learning rate of 0.001. We report the highest finite test accuracy recorded in each run as mean $\pm$ sample standard deviation over three runs.

G.3 Compatibility with GAN-Based Representation Learning

LLP-GAN [Liu et al., 2019] combines GAN-based representation learning with first-order proportion matching. Replacing its proportion loss with our order-eight moment-matching objective improves mean accuracy in all 12 CIFAR-10 settings in Table 6, with gains of up to 13.37 percentage points. However, compared with the standalone results, MM+GAN underperforms LLP-MM in most settings, while LLP-GAN does not consistently outperform PM. Both GAN-based variants generally exhibit larger standard deviations than their standalone counterparts, especially with large α-First bags. This variability may be related to instability in adversarial training. Implementation details follow.

Table 6: Best test accuracy on CIFAR-10 with GAN-based representation learning (mean $\pm$ std over three runs, %). The best result in each paired comparison is bold.

Bag	Method	16	32	64	128
Random	LLP-GAN	92.96 ± 0.09	88.96 ± 0.35	78.07 ± 1.31	58.14 ± 2.94
	MM+GAN	$\mathbf{93.56 \pm 0.19}$	$\mathbf{91.18 \pm 0.22}$	$\mathbf{80.69 \pm 1.17}$	$\mathbf{67.41 \pm 1.41}$
Cluster	LLP-GAN	92.81 ± 0.15	90.37 ± 0.16	85.16 ± 0.77	76.27 ± 0.91
	MM+GAN	$\mathbf{93.62 \pm 0.05}$	$\mathbf{91.79 \pm 0.16}$	$\mathbf{87.17 \pm 0.38}$	$\mathbf{76.53 \pm 1.93}$
α-First	LLP-GAN	91.04 ± 0.59	74.75 ± 0.55	69.52 ± 4.12	66.25 ± 2.23
	MM+GAN	$\mathbf{93.39 \pm 0.08}$	$\mathbf{88.12 \pm 1.09}$	$\mathbf{77.09 \pm 4.17}$	$\mathbf{73.13 \pm 5.62}$

Implementation details. For the CIFAR-10 comparison in Table 6, LLP-GAN and MM+GAN share the same modified-stem ResNet-18 discriminator, generator, and bag assignments. Both are trained for 500 epochs with 1024 real images per update and unit weights for the bag and adversarial losses. The discriminator uses SGD with a learning rate of 0.05, momentum 0.9, and weight decay 0.0005, with five warmup epochs followed by standard cosine decay; the generator uses Adam with a learning rate of 0.0003, $(\beta_1, \beta_2) = (0.5, 0.999)$, and 100-dimensional noise. MM+GAN replaces only the proportion loss with equally weighted cross-entropy moment losses up to order $s = 8$, retaining the adversarial and generator feature-matching objectives. We report the highest test accuracy recorded during training as mean $\pm$ sample standard deviation over three runs.

G.4 Semi-Real-World Implementation Details

Bag construction. Cells from each known patient form one bag. We retain all 219 known-patient training bags and partition the 3,204 cells with unknown patient IDs into 26 bags of 123–124 cells using a fixed shuffle, independently of cell labels. The resulting 245 training bags have sizes 1–207, with a mean of 104.98; all 56 test bags are unchanged. Bag proportions are computed after grouping, and only proportions are used for training.

Training. We use ImageNet-pretrained ResNet-18 with its standard stem. Images use 224×224 inputs and ImageNet normalization, with horizontal and vertical flips during training. The classifier optimizer is Adam with a learning rate of 0.001. Training lasts 100 epochs, including five warmup epochs. Each update uses four complete bags, or five for GeneralUPM and GeneralUPM-ABS, with forward chunks of 32 images. LLP-MM uses orders $s \in \{3, 5, 8\}$; orders exceeding the bag size are omitted and the remaining weights are renormalized. FlowLLP uses 50 pretraining and 50 joint-training epochs. Method-specific loss settings follow Appendix G.2.

Evaluation. We select the checkpoint with the highest test Macro-F1 in each run and evaluate all metrics at that checkpoint. Results are reported as mean $\pm$ sample standard deviation over three runs.

Metrics. Accuracy is the fraction of correctly classified cells, and Balanced Accuracy is the mean class-wise recall. Macro-F1 averages class-wise F1 scores equally, whereas Weighted-F1 weights them by class frequency in the test set.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction accurately summarize the paper's methodological, theoretical, and empirical contributions without overstating the scope beyond the stated LLP setting and assumptions.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Sections 2.2, 3.4, and 5 for assumptions and limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Section 3.4 states the results; Appendices D-F provide the proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4 and Appendix G describe the datasets, baselines, evaluation protocols, model and training setup, and hyperparameter choices needed to reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes].

Justification: The experiments use publicly available datasets. The public repository linked in the abstract provides code, environment setup, data preparation instructions, and scripts for reproducing the main experimental results.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes].

Justification: Sections 4.1 and 4.3 and Appendix G specify the data splits, bag construction, optimizers, hyperparameters, training schedules, and evaluation protocols used in the experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes].

Justification: Tables 1-3 and 6 report mean and standard deviation across three runs, with the summary statistic identified in each caption. Appendix G describes the corresponding training and evaluation protocols.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Figure 2 reports hardware and per-epoch runtime, and Appendix G gives training schedules. Complete resource usage for each experiment and the total compute are not reported.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper does not delve into the broader societal impacts.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: Assets are cited, but their licenses are not fully documented.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The public LLP-MM repository linked in the abstract includes a README with setup, data preparation, and run instructions, together with an MIT license.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects, so IRB approval or equivalent review is not applicable.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as an important, original, or non-standard component.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.