Multi-Label Classification with Incremental and Decremental Features

Mingdie Jiang[1*], Quanjiang Li[1*], Tingjin Luo[1†], Yiping Song[1], Chenping Hou[1]

[1]College of Science, National University of Defense Technology, Changsha 410073, Hunan, China
jiangmingdie20@nudt.edu.cn, liquanjiang@nudt.edu.cn, tingjinluo@hotmail.com, songyiping@pku.edu.cn, hcpnudt@hotmail.com

Abstract

Feature dynamics have emerged as a critical topic about open-environment learning due to the instability of feature availability. While traditional feature evolution targets single-label tasks, multi-label learning is essential to accommodate the exploding annotation spaces. However, multi-label classification with incremental and decremental features is a crucial yet underexplored problem, which poses the challenge of preserving feature representations and label correlations from historical instances and simultaneously adapting to newly arriving streaming data. To address these issues, we propose a two-stage, one-pass learning approach termed MLID. It attempts to compress the informative content of vanished features into the domain of survived ones, facilitate the propagation of label dependencies via low-rank regularization of the classifier and incorporate augmented features to construct an adaptive classification mechanism. Besides, we design optimization strategies for each stage and provide theoretical guarantees of convergence. Moreover, we establish the generalization error bound of MLID and demonstrate that the compactness of the trace norm and the reuse of models based on effective features can enhance the generalization performance. Finally, we extend it to multi-shot case and extensive experimental results validate the superiority of our MLID.

Introduction

In the era of big data, practical tasks in open environments are increasingly affected by evolving data collection methods, resulting in continuous data evolution over time. Dynamically changing data contradicts the static assumptions of traditional machine learning, such as stationary feature space and fixed data distribution (Zhou 2022). A particularly evident scenario is dynamic feature learning, where the feature space evolves with certain features fading away while new ones emerge (Hou and Zhou 2018). For instance, news platforms continually refresh keywords and named entities (Zhou, Zhong, and Li 2014). Similarly, in environmental monitoring, meteorological conditions and pollution sources exhibit temporal dynamics (Vaseashta et al. 2007; Ho et al. 2005). Consequently, sensors must be periodically updated with models that adapt to these dynamic feature environments to ensure the continued effectiveness of predictions.

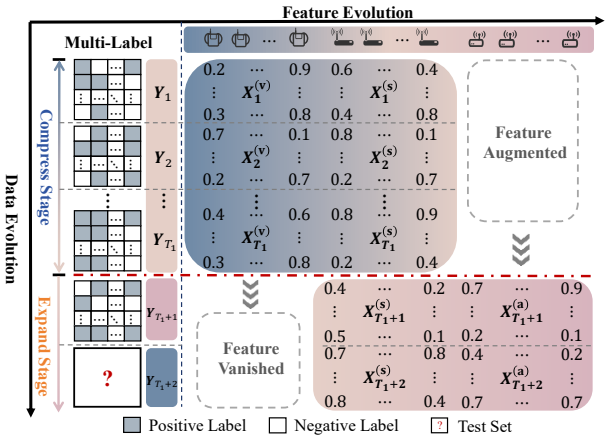


Figure 1: Simultaneous evolution of features and instances in one-shot setting for multi-label classification.

Traditional dynamic feature problems are primarily designed for multi-class classification, while real-world applications increasingly demand richer representations. Due to the complexity of object attributes and the need for comprehensive semantic descriptions, multi-label classification (MLC), where each instance can be associated with multiple labels, has become widespread (Li et al. 2024, 2025; Li, Luo, and Liao 2025; Luo et al. 2025). Furthermore, incorporating undeniable label relationships into the modeling process leads to improved predictive performance, stronger semantic consistency and greater adaptability in complex application domains (Wang and Sukthankar 2013; Katakis, Tsoumakas, and Vlahavas 2008; Sun et al. 2014; Rubin et al. 2012). For instance, labels such as ozone levels, PM2.5 concentrations, high temperature and strong winds often exhibit significant interdependencies. They commonly appear in tandem and influence each other through latent correlations.

Several existing methods for dynamic feature tasks, such as OPID (Hou and Zhou 2018), FESL (Hou, Zhang, and Zhou 2021) and FLLS (Gu, Qian, and Hou 2022), have achieved promising results through techniques like feature completion and model reuse. However, these approaches

*These authors contributed equally.
†Corresponding author.

are primarily designed single-label tasks. When MLC is decomposed into independent binary classification problems, higher-order label correlations are often neglected. The oversight hinders the model's ability to capture underlying semantic dependencies, thereby reducing predictive performance. Moreover, conventional MLC algorithms are typically developed for static data under closed-world assumptions, relying on repeated access to historical data. Such assumptions render them unsuitable for handling the evolving nature of streaming data. The core challenge in dynamic feature learning for MLC lies in enabling the model to adapt to streaming data environments while retaining informative representations of obsolete features, preserving label correlations and adaptively integrating newly emerging features.

Motivated by above observations, we propose a novel method Multi-Label Classification with Incremental and Decremental Features named **MLID**, which is a unified one-pass learning framework with two stages that integrates regression loss with low-rank regularization under consistency constraints. The framework is designed to preserve semantic representations of vanished features and maintain label dependencies, while adaptively incorporating augmented features. In C-Stage, we apply low-rank matrix factorization to compress informative historical representations. In E-Stage, we employ a proximal gradient descent algorithm to perform alternating optimization, enabling adaptive classification through weighted regression over newly introduced features. Additionally, we have theoretically demonstrated the effectiveness of each stage's optimization strategy in driving the convergence of the objective function. Utilizing the trace norm ensures tighter convex relaxation and by reusing effective feature information to reduce the unknown space, we further validate the model's generalization error. Extensive experiments indicate that MLID delivers precise and timely predictions in dynamic streaming data and evolving feature spaces. In summary, our main contributions are as follows:

- To the best of our knowledge, MLID is the first work to address MLC in streaming data with simultaneously evolving instances and feature spaces.
- MLID concurrently aligns feature projections and learns low-rank label representation, efficiently compressing feature-label correlations. The adaptive framework enables one-pass MLC via historical knowledge transfer.
- We propose an efficient optimization algorithm with theoretical guarantees, where trace norm regularization and knowledge inheritance enhance generalization.
- Extensive experimental results on diverse benchmarks validate MLID consistently outperforms state-of-the-art MLC methods in both one-shot and multi-shot scenarios.

MLID Framework

Notation

In our problem setting, data arrives in a continuous streaming fashion over time. We partition the temporal progression into two sequential stages based on the evolution of instances and the feature space. Specifically, the time interval, $t \in [1, T_1]$, is defined as the Compression Stage (C-Stage), followed by the Expansion Stage (E-Stage). During the evolution, the feature space dynamically transitions and is categorized into vanished, survived and augmented features. An illustrative overview of the process is provided in Fig. 1. At each time step, the incoming data can be decomposed into $\mathbf{X}_t^{(\mathrm{v})} \in \mathbb{R}^{n_t \times d^{(\mathrm{v})}}$, $\mathbf{X}_t^{(\mathrm{s})} \in \mathbb{R}^{n_t \times d^{(\mathrm{s})}}$ and $\mathbf{X}_t^{(\mathrm{a})} \in \mathbb{R}^{n_t \times d^{(\mathrm{a})}}$, corresponding to vanished, survived and augmented features, respectively. Here, n_t denotes the number of instances in the t-th mini-batch, while $d^{(\mathrm{v})}$, $d^{(\mathrm{s})}$ and $d^{(\mathrm{a})}$ represent the dimensionality of each feature type. The label matrix is denoted by $\mathbf{Y}_t \in \mathbb{R}^{n_t \times c}$, where c is the number of classes. Labels are binary, with $\mathbf{Y}_t(k, l) = 1$ indicating that the k-th instance at t is associated with the l-th label, and 0 otherwise. Beginning from $T_1 + 1$, the system enters E-Stage, during which the feature space undergoes dynamic evolution. Crucially, at each time step, only the data available at the current moment is retained and all previously observed data are discarded. Given only the current input tuple $\bar{\mathbf{X}}_{T_1+1} \triangleq [\mathbf{X}_{T_1+1}^{(\mathrm{s})}, \mathbf{X}_{T_1+1}^{(\mathrm{a})}, \mathbf{Y}_{T_1+1}]$, the goal is to accurately predict the label matrix $\hat{\mathbf{Y}}_{T_1+2}$.

One-shot MLID

C-Stage In C-stage, the feature space consists of both vanished and survived features. Due to the streaming setting, past observations are inaccessible. However, the survived features persist across stages, providing a crucial bridge for knowledge transfer. Our goal is to train a model solely based on the survived features while preserving the essential classification knowledge derived from the complete feature set. A natural yet effective strategy for achieving this is to enforce prediction consistency between classifiers trained on different feature compositions. Specifically, we introduce consistency constraints such that the classifier operating on $\mathbf{X}_t^{(\mathrm{s})}$ remains aligned with the classifier trained on the joint feature representation $\tilde{\mathbf{X}}_t \triangleq [\mathbf{X}_t^{(\mathrm{v})}, \mathbf{X}_t^{(\mathrm{s})}]$. Beyond feature-level alignment, we further aim to exploit the rich semantic structure encoded in the label space. Modeling label correlations leverages structural dependencies among labels to reduce model complexity and enhance generalization. To capture this, we assume that label prediction functions share latent components, allowing us to model inter-label correlations through low-rank constraints on the classifier parameters. Consequently, our overall objective integrates feature alignment term, label correlation regularization term and standard regularization term. Let $\tilde{\mathcal{H}}$ and $\mathcal{H}^{(\mathrm{s})}$ denote the hypothesis spaces corresponding to the full feature set and the survived feature subset, respectively. We then learn two classifiers, $\tilde{h} \in \tilde{\mathcal{H}}$ and $\tilde{h}^{(\mathrm{s})} \in \mathcal{H}^{(\mathrm{s})}$, with an explicit consistency constraint between them. Based on these formulations, we define the loss $\mathcal{L}_c$ as follows:

$$\begin{aligned}
&\min_{\tilde{h}\in\tilde{\mathcal{H}},\tilde{h}^{(\mathrm{s})}\in\mathcal{H}^{(\mathrm{s})}} \mathcal{L}_c(\tilde{h}, \tilde{h}^{(\mathrm{s})}) = \sum_{t=1}^{T_1} \mathcal{L}_t(\tilde{h}, \tilde{h}^{(\mathrm{s})}) \\
&= \sum_{t=1}^{T_1} \tilde{\ell}\Big(\tilde{h}(\tilde{\mathbf{X}}_t), \mathbf{Y}_t\Big) + \tilde{\ell}^{(\mathrm{s})}\Big(\tilde{h}^{(\mathrm{s})}(\mathbf{X}_t^{(\mathrm{s})}), \mathbf{Y}_t\Big) \\
&+ \alpha\tilde{\mathcal{D}}\Big(\tilde{h}, \tilde{h}^{(\mathrm{s})}\Big) + \lambda\tilde{\mathcal{F}}\big(\tilde{h}, \tilde{h}^{(\mathrm{s})}\big),
\end{aligned} \tag{1}$$

where $\mathcal{L}_t$ denotes the loss at t. The terms $\tilde{\ell}$ and $\tilde{\ell}^{(\mathrm{s})}$ represent the prediction losses on the full feature set and the survived subset during C-Stage. $\tilde{\mathcal{D}}$ enforces classifier consistency with regularization to enhance model stability, while $\tilde{\mathcal{F}}$ regularizes inter-label dependencies by modeling shared latent structures. Hyperparameters α and λ control the strength of these regularizers. For simplicity, we employ linear classifiers for all $t \in [1, T_1]$. Under this setting, the overall objective in Eq. (1) is reformulated as

$$\begin{aligned}\min_{\tilde{\mathbf{W}},\mathbf{W}^{(\mathrm{s})}} \; & \|\mathbf{X}_t\tilde{\mathbf{W}} - \mathbf{Y}_t\|_F^2 + \|\mathbf{X}_t^{(\mathrm{s})}\mathbf{W}^{(\mathrm{s})} - \mathbf{Y}_t\|_F^2 \\ & + \alpha\|\mathbf{X}_t\tilde{\mathbf{W}} - \mathbf{X}_t^{(\mathrm{s})}\mathbf{W}^{(\mathrm{s})}\|_F^2 + \beta\big(\|\tilde{\mathbf{W}}\|_F^2 + \|\mathbf{W}^{(\mathrm{s})}\|_F^2\big) \\ & + \lambda\big(\|\tilde{\mathbf{W}}\|_* + \|\mathbf{W}^{(\mathrm{s})}\|_*\big),\end{aligned} \tag{2}$$

where $\alpha > 0$ controls the consistency constraint. β and λ regularize to mitigate overfitting and enforce structural coherence. We adopt a regression-based formulation to efficiently optimize feature-related loss in the streaming context. To model inter-label dependencies, we impose a low-rank structure on the classifier parameter matrix using the trace norm $\|\cdot\|_*$, which is the tightest convex surrogate of the rank function (Fazel, Hindi, and Boyd 2001), encouraging shared latent representations across labels.

E-stage Given the persistence of survived features across both stages, we explore the possibility of extending the classifier $\tilde{h}_*^{(\mathrm{s})}$ to adapt to the expanded feature space encountered in E-Stage. However, such a direct transfer fails to exploit the augmented features and the semantic structure embedded in the label space, constraining its adaptability and performance. To overcome these limitations, we adopt a stacking-based transfer strategy that enables both the inheritance and enhancement of the classifier learned during C-Stage (Breiman 1996; Zhou 2025). Specifically, we reuse the C-Stage classifier $\tilde{h}_*^{(\mathrm{s})}$ on the input $\mathbf{X}_{T_1+1}^{(\mathrm{s})}$, enabling the model to indirectly preserve information from vanished features even after the feature space has evolved. The process yields a compact representation $\mathbf{Z}_{T_1+1}^{(\mathrm{s})} = \tilde{h}_*^{(\mathrm{s})}\big(\mathbf{X}_{T_1+1}^{(\mathrm{s})}\big)$. We then concatenate this representation with the augmented features $\bar{\mathbf{Z}}_{T_1+1} \triangleq [\mathbf{Z}_{T_1+1}^{(\mathrm{s})}, \mathbf{X}_{T_1+1}^{(\mathrm{a})}]$. In E-Stage, the label space often exhibits intricate dependencies that reflect dynamic shifts in feature-label associations. These correlations are critical in identifying latent patterns and adapting to changes in the feature distribution. To effectively exploit this structure, we adopt an ensemble-based optimization framework, where two classifiers are unified via a joint objective that integrates feature-level information and inter-label correlation modeling. Accordingly, the objective of E-Stage integrates feature representation term and label correlation regularizer, jointly enabling knowledge transfer and robust generalization under evolving feature spaces, which can be written as

$$\begin{aligned}\min_{\bar{h}^{(\mathrm{s})},\bar{h},w_1,w_2} \mathcal{L}_e = \min_{\bar{h}^{(\mathrm{s})},\bar{h},w_1,w_2} \; & \bar{\ell}\big(w_1\bar{h}^{(\mathrm{s})}(\mathbf{Z}_{T_1+1}^{(\mathrm{s})}) \\ & + w_2\bar{h}(\bar{\mathbf{Z}}_{T_1+1}), \mathbf{Y}_{T_1+1}\big) + \gamma R(\bar{h}^{(\mathrm{s})}, \bar{h}),\end{aligned} \tag{3}$$

where $w_1, w_2 \geq 0$ with $w_1 + w_2 = 1$. They balance the contributions of two linear regressors, which are $\bar{h}^{(\mathrm{s})}$ on the compact survived feature representation and $\bar{h}$ on the extended feature space. The ensemble formulation jointly optimizes both classifiers under label-structure regularization, fully exploiting the shifted feature space to enhance robustness and generalization in evolving environments. Under this formulation, Eq. (3) can be rewritten as

$$\begin{aligned}\min_{\bar{\mathbf{V}},\mathbf{V}^{(\mathrm{s})},w_1,w_2} \; & \|w_1\mathbf{Z}_{T_1+1}^{(\mathrm{s})}\mathbf{V}^{(\mathrm{s})} + w_2\bar{\mathbf{Z}}_{T_1+1}\bar{\mathbf{V}} - \mathbf{Y}_{T_1+1}\|_F^2 \\ & + \gamma\big(\|\mathbf{V}^{(\mathrm{s})}\|_* + \|\bar{\mathbf{V}}\|_*\big),\end{aligned} \tag{4}$$

where $w_1 + w_2 = 1, w_1 > 0, w_2 > 0$.

Multi-shot MLID

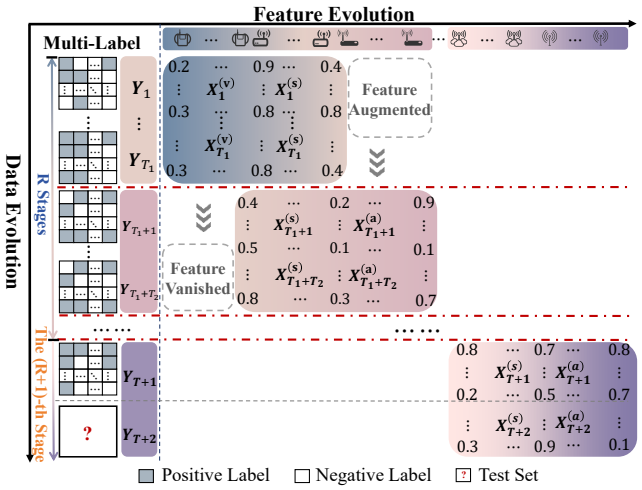


Figure 2: Simultaneous evolution of features and instances in multi-shot case for multi-label classification.

In real-world scenarios, feature spaces often change irregularly and repeatedly. To address this, we generalize our one-shot MLID framework to a multi-shot setting, introducing M^{2}LID. The notation for this setting is summarized in Fig. 2. Assume that training is divided into R distinct stages, with the t-th stage comprising T_t mini-batches. Denote the total number of training batches be $T = \sum_{t=1}^{R} T_t$. Within the M^{2}LID framework, we consider two distinct tasks.

Task 1. Given the training data $\{\mathbf{X}_{T+1}^{(\mathrm{s})}, \mathbf{X}_{T+1}^{(\mathrm{a})}, \mathbf{Y}_{T+1}\}$ and test data $\{\mathbf{X}_{T+2}^{(\mathrm{s})}, \mathbf{X}_{T+2}^{(\mathrm{a})}\}$ from the $(R+1)$-th stage, our goal is to predict the label matrix $\hat{\mathbf{Y}}_{T+2}$. Since only the immediately preceding stage shares overlapping survived features with the current one, we apply the MLID method to the last two stages for effective knowledge transfer.

Task 2. Perform prediction at any batch within any of the R training stages. To facilitate this, we propose a strategy that compresses knowledge from survived features into the augmented features. Consider any $t \in [T_1 + 1, T_1 + j]$ for some $j < T_2$. We substitute $\mathbf{X}_t$ and $\mathbf{X}_t^{(\mathrm{s})}$ in Eq. (2) with a unified representation $\bar{\mathbf{X}}_t \triangleq [\mathbf{X}_{T_1+j}^{(\mathrm{s})}, \mathbf{X}_{T_1+j}^{(\mathrm{a})}]$ and $\mathbf{X}_t^{(\mathrm{a})}$, respectively. Then we apply the learned weight matrix $\mathbf{W}_*^{(\mathrm{a})}$ to

$\mathbf{X}^{(\mathrm{a})}_{T_1+t}$ and $\mathbf{X}^{(\mathrm{a})}_{T_1+t+1}$ to obtain their compact representations $\mathbf{Z}^{(\mathrm{a})}_{T_1+t}$ and $\mathbf{Z}^{(\mathrm{a})}_{T_1+t+1}$. Similarly, $\mathbf{X}^{(\mathrm{s})}_{T_1+t}$ and $\mathbf{X}^{(\mathrm{s})}_{T_1+t+1}$ can be projected into their compact forms $\mathbf{Z}^{(\mathrm{s})}_{T_1+t}$ and $\mathbf{Z}^{(\mathrm{s})}_{T_1+t+1}$. By substituting these representations into Eq. (4) and following the same optimization procedure as before, we can compute the prediction $\hat{\mathbf{Y}}_{T_1+t+2}$.

Optimization

In this section, we offer a comprehensive introduction to C-Stage, which is grounded in low-rank matrix factorization. And then we will further demonstrate how the proximal gradient descent in E-Stage effectively integrates knowledge.

Optimization in C-Stage. Although $\mathbf{X}_t$ and $\mathbf{X}^{(\mathrm{s})}_t$ have inclusion relationships, they correspond to different feature spaces. Consequently, the objective function in Eq. (2) can be efficiently addressed through an alternating optimization scheme with respect to $\mathbf{W}^{(\mathrm{s})}$ and $\tilde{\mathbf{W}}$. Focusing on the optimization of $\mathbf{W}^{(\mathrm{s})}$, Eq. (2) can be reformulated as

$$\begin{aligned}\min_{\mathbf{W}^{(\mathrm{s})}} &||\mathbf{X}^{(\mathrm{s})}_t\mathbf{W}^{(\mathrm{s})} - \mathbf{Y}_t||^2_F + \alpha||\mathbf{X}_t\tilde{\mathbf{W}} - \mathbf{X}^{(\mathrm{s})}_t\mathbf{W}^{(\mathrm{s})}||^2_F \\ &+ \beta||\mathbf{W}^{(\mathrm{s})}||^2_F + \lambda||\mathbf{W}^{(\mathrm{s})}||_*.\end{aligned} \tag{5}$$

However, the non-differentiability of the trace norm poses a challenge. To circumvent this, we adopt the low-rank matrix factorization strategy (Yu et al. 2014). For example, we represent $\mathbf{W}^{(\mathrm{s})}$ by two low-rank matrices, i.e., $\mathbf{W}^{(\mathrm{s})} = \mathbf{U}^{(\mathrm{s})}\mathbf{H}^{(\mathrm{s})}$. The rank of $\mathbf{W}^{(\mathrm{s})}$ is thereby implicitly controlled by the dimensions of subspace. Under the factorization, the trace norm regularization term transforms into a differentiable surrogate, given by $\|\mathbf{W}^{(\mathrm{s})}\|_* = \frac{1}{2}\big(\|\mathbf{U}^{(\mathrm{s})}\|^2_F + \|\mathbf{H}^{(\mathrm{s})}\|^2_F\big)$. The reformulation enables efficient gradient-based optimization. To solve the objective, we iteratively update $\mathbf{U}^{(\mathrm{s})}$ and $\mathbf{H}^{(\mathrm{s})}$ in an alternating fashion. Denoting $\mathbf{A}^{(\mathrm{s})}_t = [\mathbf{X}^{(\mathrm{s})}_t, \sqrt{\alpha}\mathbf{X}^{(\mathrm{s})}_t, \sqrt{\beta}\mathbf{I}]^T$ and $\mathbf{B}^{(\mathrm{s})}_t = [\mathbf{Y}_t, \sqrt{\alpha}\mathbf{X}_t\tilde{\mathbf{W}}, \mathbf{0}]^T$, Eq. (5) can be rewritten as

$$\begin{aligned}\min_{\mathbf{U}^{(\mathrm{s})},\mathbf{H}^{(\mathrm{s})}} &\left\|\mathbf{A}^{(\mathrm{s})}_t\mathbf{U}^{(\mathrm{s})}\mathbf{H}^{(\mathrm{s})} - \mathbf{B}^{(\mathrm{s})}_t\right\|^2_F \\ &+ \frac{\lambda}{2}\Big(\left\|\mathbf{U}^{(\mathrm{s})}\right\|^2_F + \left\|\mathbf{H}^{(\mathrm{s})}\right\|^2_F\Big).\end{aligned} \tag{6}$$

If $\mathbf{U}^{(\mathrm{s})}$ is fixed, each column of $\mathbf{H}^{(\mathrm{s})}$ denoted by $\boldsymbol{h}^{(\mathrm{s})}_j$ can be optimized. And when $\mathbf{H}^{(\mathrm{s})}$ is given, $\tilde{\boldsymbol{a}}_{ij} = \boldsymbol{a}_i \otimes \boldsymbol{h}_j$ with $|\Omega|$ data points in $\{(\mathbf{B}_{ij}, \hat{\boldsymbol{a}}_{ij}) : (i,j) \in \Omega\}$, according to Eq. (6), we have

$$g(\boldsymbol{u}^{(\mathrm{s})}) = \sum_{(i,j)\in\Omega} \ell_u\left(\mathbf{B}_{ij}, \big(\boldsymbol{u}^{(\mathrm{s})}\big)^T\tilde{\boldsymbol{a}}_{ij}\right) + \frac{1}{2}\lambda\|\boldsymbol{u}^{(\mathrm{s})}\|^2_2, \tag{7}$$

where $u^{(\mathrm{s})}_* := \boldsymbol{vec}\big(\mathbf{U}^{(\mathrm{s})_*}\big) = \arg\min_{\boldsymbol{u}^{(\mathrm{s})}\in\mathbb{R}^{d^{(s)}m}} g(\boldsymbol{u}^{(\mathrm{s})})$. m is a known low rank. The gradient matrix for $g(u^{(\mathrm{s})})$ is

$$\begin{aligned}&\nabla g(\boldsymbol{u}^{(\mathrm{s})}) \\ =&\boldsymbol{vec}\left(\big(\mathbf{A}^{(\mathrm{s})}_t\big)^T\mathbf{A}^{(\mathrm{s})}_t\mathbf{U}^{(\mathrm{s})}\mathbf{M} - \big(\mathbf{A}^{(\mathrm{s})}_t\big)^T\mathbf{B}_t\mathbf{H}^{(\mathrm{s})}\right) + \lambda\boldsymbol{w},\end{aligned} \tag{8}$$

where $\mathbf{M}^{(\mathrm{s})} = {\mathbf{H}^{(\mathrm{s})}}^T\mathbf{H}^{(\mathrm{s})}$.

Optimization in E-Stage. Although directly optimizing the objective in Eq. (4) is intractable due to its complex structure, it can be equivalently transformed through a series of analytical derivations into a more tractable formulation given by

$$\begin{aligned}\min_{\bar{\mathbf{V}},\mathbf{V}^{(\mathrm{s})},w_1,w_2} &\|\mathbf{Z}^{(\mathrm{s})}_{T_1+1}\mathbf{V}^{(\mathrm{s})} + \bar{\mathbf{Z}}_{T_1+1}\bar{\mathbf{V}} - \mathbf{Y}_{T_1+1}\|^2_F \\ &+ \gamma\big(\frac{1}{w_1}\|\mathbf{V}^{(\mathrm{s})}\|_* + \frac{1}{w_2}\|\bar{\mathbf{V}}\|_*\big).\end{aligned} \tag{9}$$

Here, $w_1 + w_2 = 1$ and $w_1 > 0, w_2 > 0$. Given the separability between the optimization of classifiers and balance coefficients, we adopt an alternating optimization scheme that allows each subproblem to be solved efficiently.

(1) Fix $\bar{\mathbf{V}}, \mathbf{V}^{(\mathrm{s})}$ and optimize w_1 and w_2. Eq. (9) can be transformed into the following form:

$$\min_{w_1,w_2} \gamma\big(\frac{1}{w_1}\|\mathbf{V}^{(\mathrm{s})}\|_* + \frac{1}{w_2}\|\bar{\mathbf{V}}\|_*\big). \tag{10}$$

By substituting $w_2 = 1 - w_1$ into the above equation, take the derivative of the objective function with respect to w_1 and set it to zero. We obtain that

$$\begin{cases} w_1 = \frac{\sqrt{\|\mathbf{V}^{(\mathrm{s})}\|_*}}{\sqrt{\|\mathbf{V}^{(\mathrm{s})}\|_*}+\sqrt{\|\bar{\mathbf{V}}\|_*}}, \\ w_2 = 1 - w_1 = \frac{\sqrt{\|\bar{\mathbf{V}}\|_*}}{\sqrt{\|\mathbf{V}^{(\mathrm{s})}\|_*}+\sqrt{\|\bar{\mathbf{V}}\|_*}}. \end{cases} \tag{11}$$

(2) Fix w_1, w_2 and optimize $\bar{\mathbf{V}}, \mathbf{V}^{(\mathrm{s})}$. Subsequently, IRNN(Lu et al. 2014, 2015) is applied to iteratively refine the closed-form classifier solution through alternating optimization. The forecast for the subsequent time instance in $T_1 + 2$ is computed as

$$\begin{aligned}&\mathbf{Z}^{(\mathrm{s})}_{T_1+2} = \tilde{h}^{(\mathrm{s})}_*\big(\mathbf{X}^{(\mathrm{s})}_{T_1+2}\big), \\ &\bar{\mathbf{Z}}_{T_1+1} \triangleq [\mathbf{Z}^{(\mathrm{s})}_{T_1+2}, \mathbf{X}^{(\mathrm{a})}_{T_1+2}], \\ &\hat{\mathbf{Y}}_{T_1+2} = w_1\mathbf{Z}^{(\mathrm{s})}_{T_1+1}\mathbf{V}^{(\mathrm{s})}_* + w_2\bar{\mathbf{Z}}_{T_1+1}\bar{\mathbf{V}}_*.\end{aligned} \tag{12}$$

Theoretical Analysis

Generalization Error Bound In this section, we analyze the generalization ability of MLID. The theoretical result highlights two key insights. On the one hand, trace norm yields stricter generalization bounds than Frobenius norm. On the other hand, model reuse under effective feature assumptions enhances performance. We introduce essential theorems as below:

Theorem 1 *Let $\mathcal{H}$ be the family of the hypothesis set, and denote the hypothesis on whole data after feature changing as h. Suppose the loss function is L-Lipschitz. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over a sample of size n_t, the following inequality holds for all $h \in \mathcal{H}$,*

$$\mathcal{L}(\hat{h}) - \hat{\mathcal{L}}(\hat{h}) \leq \frac{4Lc}{\sqrt{n_t}}M\Lambda + \mathcal{O}\left(c\sqrt{\frac{\log\frac{1}{\delta}}{n_t}}\right), \tag{13}$$

where $\Lambda = \max\{\|\boldsymbol{x}\|_F \mid \boldsymbol{x} \in \mathcal{X}\}$ represents the radius of feature domain and $\|\boldsymbol{w}\|_* \leq M$. M represents the radius of linear hypothesis space. $\boldsymbol{w}$ is the hypothesis coefficient corresponding to the linear classifier h and $r(\boldsymbol{w}) \leq r_e$.

Theorem 1 establishes a generalization error bound when the model is trained exclusively on the data from E-stage. It is particularly noteworthy that the use of trace norm as a regularization term not only enforces semantic consistency among label correlations, but also provides provably tighter upper bounds than those derived from the matrix ℓ_2-norm or Frobenius norm in Eq. (14).

$$\begin{aligned}\mathbb{E}\left[\sup_{h_e\in\mathcal{H}}\langle w,\sum_{i=1}^{n_t}\sigma_i h(x_i)\rangle\right] &\leq \mathbb{E}\left[\sup_{h_e\in\mathcal{H}}\|w\|_*\|\sum_{i=1}^{n_t}\sigma_i x_i\|_F\right]\\ &\leq \mathbb{E}\left[\sup_{h_e\in\mathcal{H}} r_e\|\sum_{i=1}^{n_t}\sigma_i x_i\|_2\right]\\ &\leq r_e.\end{aligned} \tag{14}$$

Theorem 2 *Let $\bar{\mathcal{H}}$ be the family of the hypothesis set, and denote the hypothesis returned in E-Stage as $\bar{h}_e$. Suppose $\mathcal{L}_e$ is L-Lipschitz. Then, for any $\delta > 0$, with probability at least $1-\delta$ over a sample of size n_t, the following inequality holds for all $h_e \in \bar{\mathcal{H}}$,*

$$\begin{aligned}&\mathcal{L}_e(\hat{h}_e) - \hat{\mathcal{L}}_e(\hat{h}_e)\\ &\leq \frac{4Lc}{\sqrt{n_t}}\Lambda_z(\bar{M}+\sqrt{M_e^2-\bar{M}^2}) + \mathcal{O}\left(c\sqrt{\frac{\log\frac{1}{\delta}}{n_t}}\right),\end{aligned} \tag{15}$$

where $\Lambda_z = \max\{\|\boldsymbol{z}\|_F \mid \boldsymbol{z}\in\mathcal{Z}\}$ represents the radius of feature domain after model reuse. $\|\boldsymbol{w}_e\|_* \leq M_e$. M_e represents the radius of linear hypothesis space and $\|\boldsymbol{w}_2\|_* \leq \bar{M}$.

Theorem 2 presents the generalization error bound for MLID. Compared to $\mathcal{H}$, the hypothesis function space $\mathcal{H}_e$ has been constrained due to the adding of $\mathbf{W}^{(\mathrm{s})}$. Thus, under the assumption that the features are effective for MLC, we naturally have $M_e \leq M, \Lambda_z \leq \Lambda$ and $\bar{M} + \sqrt{(M_e)^2 - \bar{M}^2} \leq M_e \leq M$. Therefore, model reuse yields a more favorable upper bound for the generalization error.

Convergence Analysis In this section, we analyze the convergence behavior of loss functions. Specifically, the loss function in C-stage is efficiently updated for low-rank matrix factorization parameters via gradient descent. Since it satisfies the Lipschitz condition, the properties of convex optimization ensure that the iterative process will converge to the global optimum with appropriate step size selection. To prove the convergence of E-Stage in Theorem 3, the following lemmas are necessary.

Lemma 1 *(Lu et al. 2014) $f : \mathbb{R}^{m\times n} \to \mathbb{R}^+$ is a smooth function of type $C^{1,1}$, i.e., the gradient is Lipschitz continuous. $\|\nabla f(\mathbf{X}) - \nabla f(\mathbf{Y})\|_F \leq L(f)\|\mathbf{X}-\mathbf{Y}\|_F$. For any $\mathbf{X},\mathbf{Y} \in \mathbb{R}^{m\times n}$, $L(f) > 0$ is called Lipschitz constant of ∇f and $f(\mathbf{X})$ is possibly nonconvex. $F(\mathbf{X}) \to \infty$ iff $\|\mathbf{X}\|_F \to \infty$.*

Lemma 2 *(Lu et al. 2014) Let $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ be a continuously differentiable function with Lipschitz continuous gradient and Lipschitz constant $L(f)$. Then, for any $\mathbf{X},\mathbf{Y} \in \mathbb{R}^{m\times n}$ and $\mu \geq L(f)$, $f(\mathbf{X}) \leq f(\mathbf{Y}) + \langle \mathbf{X}-\mathbf{Y}, \nabla f(\mathbf{Y})\rangle + \frac{\mu}{2}\|\mathbf{X}-\mathbf{Y}\|_F^2$.*

Theorem 3 *Assume that $F(\mathbf{V}^{(\mathrm{s})}) = \gamma\|\mathbf{V}^{(\mathrm{s})}\|_* + f\big(\mathbf{V}^{(\mathrm{s})}\big)$ represents the optimization problem in Eq. (4), which satisfies the assumptions in Lemma 1 and Lemma 2. The sequence $\{\mathbf{V}_k^{(\mathrm{s})}\}$ generated in Algorithm satisfies:*

(1) $F(\mathbf{V}_k^{(\mathrm{s})})$ is monotonically decreasing. $F(\mathbf{V}_k^{(\mathrm{s})}) - F(\mathbf{V}_{k+1}^{(\mathrm{s})}) \geq \frac{\mu - L(f)}{2}\|\mathbf{V}_k^{(\mathrm{s})} - \mathbf{V}_{k+1}^{(\mathrm{s})}\|_F^2 \geq 0$;

(2) $\lim_{k\to\infty}(\mathbf{V}_k^{(\mathrm{s})} - \mathbf{V}_{k+1}^{(\mathrm{s})}) = 0$ and $\{\mathbf{V}_k^{(\mathrm{s})}\}$ is bounded.

Experiment

Configuration

Datasets and Evaluation metrics. To evaluate the effectiveness of our approach, we conduct experiments on seven widely used benchmark datasets spanning diverse application domains: Genbase (Diplaris et al. 2005), Scene (Boutell et al. 2004), Yeast (Elisseeff and Weston 2001), Corel5k (Turnbull et al. 2008), Bibtex (Katakis, Tsoumakas, and Vlahavas 2008) and Mediamill (Snoek et al. 2006). Performance is assessed using six standard MLC metrics: Ranking Loss (RL), Recall, Average Precision (AP), Normalized Discounted Cumulative Gain (NDCG), Hamming Loss (HL), and the adapted Area Under the Curve (AUC). For clarity, we report 1-RL and 1-HL. So that higher values consistently indicate better performance.

Implementation Details. For consistency, we assume a uniform distribution of samples across categories in both one-shot and multi-shot scenarios for all training and testing batches. The R training phases are conducted with a fixed total number of samples, while the number of points per batch is systematically varied. In the multi-shot setting, the data is divided into three distinct stages to facilitate clearer exposition. In addition to the configurations described above, the following settings are further applied. (1) For each scene, all batches are divided into two stages and each comprising an equal number of data points; (2) The complete feature set is partitioned into four equal segments based on their original order. In particular, the first segment comprises features withheld from Stage 1, while the final segment includes those exclusively introduced in Stage 3.

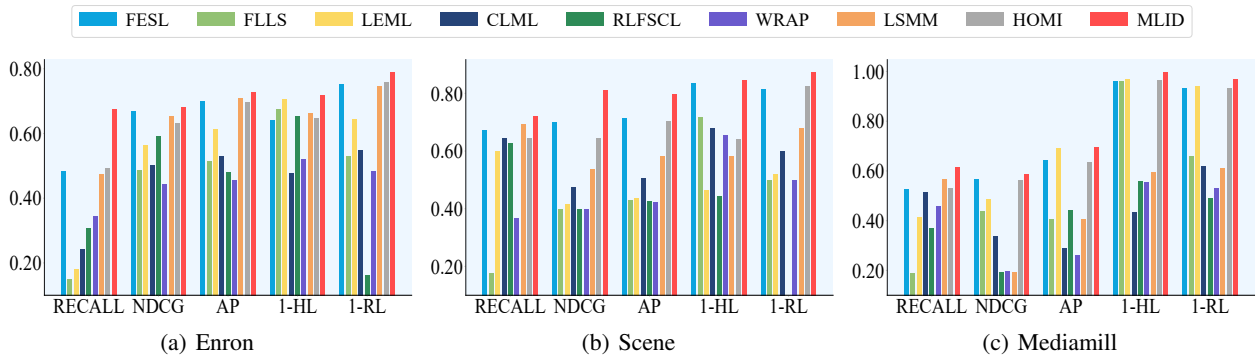
Comparison Methods. We compare MLID with eight methods, i.e., FESL, FLLS, LEML, CLML (Li et al. 2022), RLFSCL (Liu et al. 2023), WRAP (Yu and Zhang 2021), LSMM (Mao, Wang, and Zhang 2023) and HOMI (Si et al. 2023). Notably, FESL and FLLS represent the latest and influential approaches for streaming data. The remaining baselines are designed for MLC under closed-environment and are evaluated only on E-Stage. For fairness, all hyperparameters are configured according to the settings recommended in their original publications or official implementations.

Comparative Study

To assess the effectiveness of our approach, we conducted experiments with eight advanced methods on seven datasets. Table 1 summarizes the AUC results of various methods across multiple datasets, while performance under additional evaluation metrics is illustrated in Fig. 3. Additional experimental results are provided in the Appendix for completeness. Based on these results, the report highlight several

Data set	n_i	FESL	FLLS	LEML	CLML	RLFSCL	WRAP	LSMM	HOMI	MLID
Genbase	8	.8811 (.1540)	.7557 (.1352)	.5519 (.1199)	.3268 (.1382)	.2251 (.0193)	.4908 (.1295)	.9380 (.0351)	.8823 (.0875)	**.9499 (.0334)**
	16	.8883 (.1359)	.6920 (.1393)	.5981 (.1097)	.5670 (.1626)	.2337 (.0206)	.5290 (.0972)	.9367 (.0291)	.9091 (.0260)	**.9611 (.0166)**
	30	.8467 (.1305)	.6977 (.1304)	.6287 (.0796)	.6608 (.1263)	.2529 (.0474)	.5413 (.1043)	.9271 (.0359)	.8978 (.0281)	**.9684 (.0159)**
	60	.8781 (.1224)	.7343 (.0782)	.5381 (.0758)	.6623 (.1342)	.3451 (.0435)	.5049 (.0676)	.9466 (.0204)	.8885 (.0147)	**.9747 (.0126)**
Enron	20	.7566 (.0341)	.5343 (.0551)	.6454 (.0798)	.5512 (.0327)	.4531 (.1402)	.4843 (.0635)	.7456 (.0562)	.7602 (.0107)	**.7887 (.0262)**
	40	.7546 (.0081)	.5408 (.0439)	.7234 (.0181)	.5646 (.0280)	.3376 (.0127)	.5042 (.0313)	.7484 (.0443)	.7654 (.0109)	**.7840 (.0225)**
	80	.7772 (.0283)	.5062 (.0528)	.7226 (.0134)	.5625 (.0142)	.3673 (.0101)	.4719 (.0230)	.7468 (.0440)	.7673 (.0090)	**.7845 (.0237)**
	160	.7860 (.0250)	.5380 (.0592)	.7280 (.0262)	.5908 (.0149)	.4309 (.0124)	.4917 (.0498)	.7342 (.0533)	.7664 (.0154)	**.7914 (.0212)**
Scene	30	.8133 (.0462)	.5127 (.0246)	.5204 (.0197)	.6003 (.0241)	.5149 (.0180)	.4999 (.0388)	.6786 (.0297)	.8241 (.0161)	**.8731 (.0111)**
	60	.8496 (.0186)	.5487 (.0253)	.5243 (.0201)	.6543 (.0219)	.5207 (.0125)	.5079 (.0210)	.7988 (.0283)	.8321 (.0354)	**.8776 (.0180)**
	120	.8549 (.0169)	.5315 (.0396)	.5161 (.0119)	.7386 (.0201)	.5119 (.0174)	.5064 (.0345)	.5036 (.0345)	.8206 (.0264)	**.8902 (.0098)**
	240	.8759 (.0177)	.5231 (.0314)	.5243 (.0087)	.7995 (.0150)	.5093 (.0124)	.4957 (.0498)	.7781 (.0500)	.8286 (.0284)	**.8834 (.0093)**
Yeast	30	.6080 (.0147)	.5102 (.0136)	.5808 (.0238)	.5363 (.0118)	.5443 (.4659)	.4659 (.0117)	.6160 (.0253)	.7531 (.0058)	**.7617 (.0054)**
	60	.6104 (.0125)	.4978 (.0246)	.5617 (.0293)	.5296 (.0180)	.5489 (.0121)	.4964 (.0818)	.6149 (.0240)	.7462 (.0095)	**.7729 (.0090)**
	120	.6180 (.0137)	.5179 (.0233)	.5758 (.0179)	.5451 (.0134)	.5720 (.0115)	.4658 (.0365)	.6103 (.0215)	.7518 (.0107)	**.7670 (.0109)**
	240	.5179 (.0233)	.6126 (.0106)	.5778 (.0168)	.5866 (.0164)	.5889 (.0096)	.4688 (.0446)	.6050 (.0254)	.7503 (.0113)	**.7642 (.0114)**
Corel5k	60	.5457 (.0091)	.4133 (.0136)	.5260 (.0126)	.4257 (.0174)	.4065 (.0058)	.5038 (.0268)	.6818 (.0080)	.7760 (.0082)	**.7817 (.0054)**
	120	.5424 (.0168)	.4125 (.0187)	.5204 (.0152)	.4333 (.0099)	.4059 (.0081)	.4937 (.0166)	.6966 (.0318)	.7755 (.0141)	**.7839 (.0122)**
	240	.5506 (.0156)	.4194 (.0112)	.5287 (.0121)	.4340 (.0164)	.4090 (.0116)	.5027 (.0119)	.6956 (.0300)	.7328 (.0082)	**.7589 (.0104)**
	480	.5496 (.0168)	.4132 (.0105)	.5299 (.0119)	.4421 (.0098)	.4086 (.0101)	.5025 (.0222)	.6036 (.0260)	.7529 (.0084)	**.7621 (.0096)**
Bibtex	85	.7865 (.0051)	.4845 (.0168)	.5788 (.0072)	.6044 (.0149)	.4851 (.0166)	.5039 (.0167)	.5277 (.0189)	.8082 (.0061)	**.8113 (.0144)**
	175	.7362 (.0076)	.4794 (.0163)	.5855 (.0317)	.6884 (.0130)	.4888 (.0087)	.4947 (.0447)	.5053 (.0213)	.7830 (.0106)	**.7943 (.0130)**
	350	.7488 (.0059)	.5060 (.0339)	.5848 (.0069)	.7338 (.0144)	.4947 (.0077)	.5005 (.0137)	.4989 (.0113)	.7907 (.0085)	**.7982 (.0103)**
	700	.7733 (.0134)	.4923 (.0183)	.6008 (.0515)	.7369 (.0112)	.4929 (.0101)	.4996 (.0178)	.4996 (.0130)	.8058 (.0060)	**.8100 (.0084)**
Mediamill	50	.9304 (.0013)	.6599 (.0677)	.8417 (.0013)	.6181 (.0205)	.4885 (.0716)	.5305 (.0718)	.6121 (.0743)	.9338 (.0008)	**.9369 (.0039)**
	100	.9307 (.0013)	.6216 (.0768)	.8410 (.0033)	.6550 (.0161)	.5693 (.0010)	.5294 (.0797)	.5660 (.0010)	.9336 (.0005)	**.9367 (.0033)**
	200	.9320 (.0016)	.6246 (.0677)	.8426 (.0007)	.6725 (.0160)	.5693 (.0010)	.5054 (.1203)	.4773 (.0943)	.9339 (.0007)	**.9358 (.0038)**
	400	.9335 (.0010)	.6196 (.0739)	.8422 (.0032)	.6964 (.0060)	.5745 (.0009)	.4435 (.0603)	.5004 (.0889)	.9338 (.0006)	**.9354 (.0031)**

Table 1: The one-shot testing AUC (mean std) between MLID and the compared methods on 7 data sets with different number of training and testing examples.

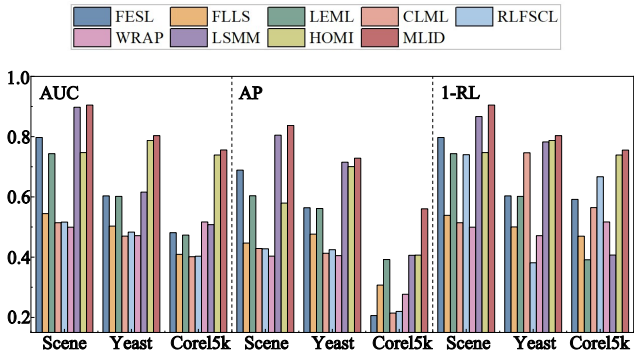


(a) Enron (b) Scene (c) Mediamill

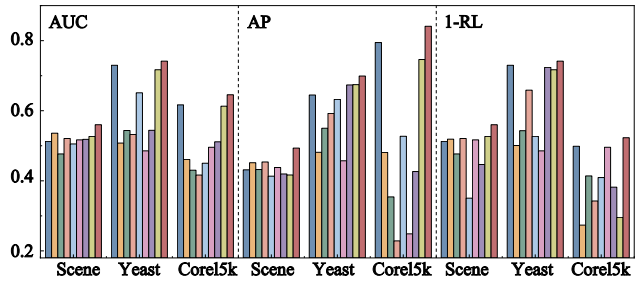
Figure 3: Performance of various metrics across methods and datasets with n_i fixed as the minimum batch number.

key observations. (1) The proposed method consistently outperforms baseline approaches across a broad range of evaluation metrics and conditions. Its substantial gains in AUC and AP underscore its robustness and adaptability, particularly in open-world scenarios characterized by dynamic feature variations; (2) Under identical experimental settings, MLID consistently outperforms all comparative MLC methods. The improvement is primarily attributed to the integration of auxiliary learning for model reuse, which effectively addresses the limitations of static frameworks. The advantage is especially pronounced in the E-Stage, where data scarcity poses substantial challenges; (3) MLID achieves competitive performance in streaming data processing, outperforming strong baselines such as FESL and FLLS. These results highlight the critical role of modeling label correlations, which capture latent semantic structures often neglected by conventional MLC methods; (4) Across a range of diverse n_i settings, MLID consistently outperforms all baseline methods. Additionally, our approach maintains stability and competitive results across all scenarios.

As shown in Fig. 4, M^{2}LID outperforms the comparison methods across all evaluation metrics, with MLID achieving even greater improvements, which is primarily attributed to the one-shot setting, which provides more training samples

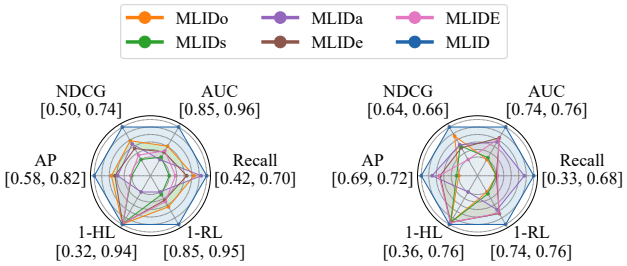


(a) The performance of various methods in predicting $\hat{\mathbf{Y}}_{T+2}$.



(b) The performance of various methods in predicting $\hat{\mathbf{Y}}_{T_1+t}$.

Figure 4: AUC, AP and 1-RL between M^{2}LID and the compared methods on three data sets with n_i fixed as the minimum batch number in two tasks.



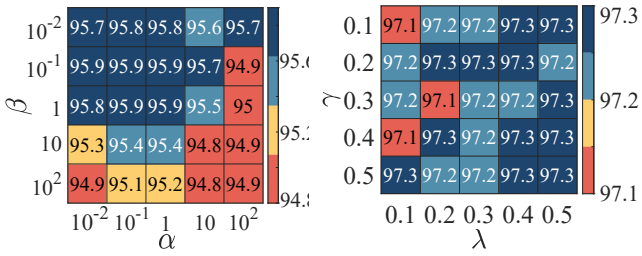
(a) Genbase (b) Yeast

Figure 5: Performance on various metrics across datasets in the ablation study with n_i fixed at the minimum batch size.

and a higher proportion of preserved features. In Fig. 4(b), M^{2}LID achieves slightly better performance on Task 2, owing to the additional compression applied in Stage 2 which supports the effectiveness of our strategy in this context.

Ablation Study

To systematically evaluate the contributions of individual components in the model reuse framework, we conduct an ablation study within MLID by comparing five representative variants. Specifically, a linear classifier trained solely on E-Stage dataset is denoted as MLIDe. To isolate the effect of feature enhancement within E-Stage, we construct MLIDs and MLIDa. To assess the role of feature evolution,



(a) The influence of α and β. (b) The influence of λ and γ.

Figure 6: Parameter sensitivity analysis on Genbase in one-shot setting.

MLIDo is trained on features excluding both vanished and augmented features. Finally, we introduce MLIDE to examine the benefit of recovering vanished features. MLIDE is an ensemble model that combines predictions from a classifier trained on survived features in C-Stage and another trained on the full E-Stage feature set. The results shown in Fig. 5 reveal that MLIDs, MLIDa and MLIDe consistently underperform compared to MLID. Notably, incorporating classifiers trained in C-Stage significantly enhances E-Stage performance, especially under data-scarce conditions. The results underscore the importance of model inheritance and feature reuse. Furthermore, MLID outperforms MLIDo by jointly handling vanished and augmented features, and surpasses MLIDE in most cases, which demonstrates the necessity of compressing obsolete features for effective transfer.

Parameter Sensitivity

To evaluate parameter sensitivity, we group the hyperparameters into (α, β), (λ, γ) and analyze their joint effects. As shown in Fig. 6, α and β are selected from $\{0.001, 0.01, 0.1, 1, 10, 100\}$ while λ and γ range over $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. The heatmaps indicate that performance improves as α and β approach 0.001. Meanwhile, λ approaches 0.5. Importantly, MLID consistently delivers strong performance across a wide range of values, demonstrating robustness and low sensitivity to parameter settings.

Conclusion

To address MLC under dynamic feature scenarios involving incremental and decremental features, the paper proposes MLID, which is a two-stage, one-pass learning framework. Our method effectively compresses the essential information and label dependencies of vanished features into functions over the survived features, and further extends them with augmented features to enable adaptive MLC. We additionally develop corresponding optimization strategies with theoretical convergence guarantees and derive generalization error bounds for each stage. Moreover, MLID extends its applicability from one-shot case to dynamic environments with arbitrary feature evolution, supporting predictions at any time step. Extensive experimental results demonstrate the superiority of the proposed approach. In future work, we will explore the challenge of combining dynamic feature scenarios with limited annotations.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 62376281, the NSF for Distinguished Young Scholars under Grant No. 62425607, and the Key NSF of China under Grant No. 62136005.

References

Boutell, M. R.; Luo, J.; Shen, X.; and Brown, C. M. 2004. Learning Multi-label Scene Classification. *Pattern recognition*, 37(9): 1757–1771.

Breiman, L. 1996. Stacked Regressions. *Machine learning*, 24(1): 49–64.

Diplaris, S.; Tsoumakas, G.; Mitkas, P. A.; and Vlahavas, I. 2005. Protein Classification with Multiple Algorithms. In *Advances in Informatics: 10th Panhellenic Conference on Informatics, PCI 2005, Volas, Greece, November 11-13, 2005. Proceedings 10*, 448–456. Springer.

Elisseeff, A.; and Weston, J. 2001. A Kernel Method for Multi-labelled Classification. *Advances in neural information processing systems*, 14.

Fazel, M.; Hindi, H.; and Boyd, S. P. 2001. A Rank Minimization Heuristic with Application to Minimum Order System Approximation. In *Proceedings of the 2001 American control conference.(Cat. No. 01CH37148)*, volume 6, 4734–4739. IEEE.

Gu, S.; Qian, Y.; and Hou, C. 2022. Incremental feature spaces learning with label scarcity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(6): 1–26.

Ho, C. K.; Robinson, A.; Miller, D. R.; and Davis, M. J. 2005. Overview of Sensors and Needs for Environmental Monitoring. *Sensors*, 5(1): 4–37.

Hou, B.-J.; Zhang, L.; and Zhou, Z.-H. 2021. Learning With Feature Evolvable Streams. *IEEE Transactions on Knowledge and Data Engineering*, 33(6): 2602–2615.

Hou, C.; and Zhou, Z.-H. 2018. One-Pass Learning with Incremental and Decremental Features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(11): 2776–2792.

Katakis, I.; Tsoumakas, G.; and Vlahavas, I. 2008. Multilabel Text Classification for Automated Tag Suggestion. *ECML PKDD discovery challenge*, 75: 2008.

Li, J.; Li, P.; Hu, X.; and Yu, K. 2022. Learning Common and Label-specific Features for Multi-label Classification with Correlation Information. *Pattern recognition*, 121: 108259.

Li, Q.; Luo, T.; Jiang, M.; Jiang, Z.; Hou, C.; and Li, F. 2025. Semi-Supervised Multi-View Multi-Label Learning with View-Specific Transformer and Enhanced Pseudo-Label. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 18430–18438.

Li, Q.; Luo, T.; Jiang, M.; Liao, J.; and Jiang, Z. 2024. Deep Incomplete Multi-View Network Semi-Supervised Multi-Label Learning with Unbiased Loss. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 9048–9056.

Li, Q.; Luo, T.; and Liao, J. 2025. Theory-Inspired Deep Multi-View Multi-Label Learning with Incomplete Views and Noisy Labels. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 20706–20715.

Liu, Z.; Tang, C.; Abhadiomhen, S. E.; Shen, X.-J.; and Li, Y. 2023. Robust Label and Feature Space Co-learning for Multi-label Classification. *IEEE Transactions on Knowledge and Data Engineering*, 35(11): 11846–11859.

Lu, C.; Tang, J.; Yan, S.; and Lin, Z. 2014. Generalized Nonconvex Nonsmooth Low-rank Minimization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4130–4137.

Lu, C.; Tang, J.; Yan, S.; and Lin, Z. 2015. Nonconvex Nonsmooth Low Rank Minimization via Iteratively Reweighted Nuclear Norm. *IEEE Transactions on Image Processing*, 25(2): 829–839.

Luo, T.; Li, Q.; Jiang, M.; and Hou, C. 2025. Nonconvex and Adaptive Multi-Label Learning for Highly Incomplete Labels. *Knowledge-Based Systems*, 114784.

Mao, J.; Wang, W.; and Zhang, M.-L. 2023. Label Specific Multi-Semantics Metric Learning for Multi-Label Classification: Global Consideration Helps. In *IJCAI*, 4055–4063.

Rubin, T. N.; Chambers, A.; Smyth, P.; and Steyvers, M. 2012. Statistical Topic Models for Multi-label Document Classification. *Machine learning*, 88(1): 157–208.

Si, C.; Jia, Y.; Wang, R.; Zhang, M.-L.; Feng, Y.; and Qu, C. 2023. Multi-label Classification with High-rank and High-order Label Correlations. *IEEE Transactions on Knowledge and Data Engineering*, 36(8): 4076–4088.

Snoek, C. G.; Worring, M.; Van Gemert, J. C.; Geusebroek, J.-M.; and Smeulders, A. W. 2006. The Challenge Problem for Automated Detection of 101 Semantic Concepts in Multimedia. In *Proceedings of the 14th ACM international conference on Multimedia*, 421–430.

Sun, F.; Tang, J.; Li, H.; Qi, G.-J.; and Huang, T. S. 2014. Multi-label Image Categorization with Sparse Factor Representation. *IEEE Transactions on Image Processing*, 23(3): 1028–1037.

Turnbull, D.; Barrington, L.; Torres, D.; and Lanckriet, G. 2008. Semantic Annotation and Retrieval of Music and Sound Effects. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2): 467–476.

Vaseashta, A.; Vaclavikova, M.; Vaseashta, S.; Gallios, G.; Roy, P.; and Pummakarnchana, O. 2007. Nanostructures in Environmental Pollution Detection, Monitoring, and Remediation. *Science and Technology of Advanced Materials*, 8(1-2): 47.

Wang, X.; and Sukthankar, G. 2013. Multi-label Relational Neighbor Classification Using Social Context Features. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 464–472.

Yu, H.-F.; Jain, P.; Kar, P.; and Dhillon, I. 2014. Large-scale Multi-label Learning with Missing Labels. In *International conference on machine learning*, 593–601. PMLR.

Yu, Z.-B.; and Zhang, M.-L. 2021. Multi-label Classification with Label-specific Feature Generation: A wrapped approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9): 5199–5210.

Zhou, E.; Zhong, N.; and Li, Y. 2014. Extracting News Blog Hot Topics Based on the W2T Methodology. *World Wide Web*, 17: 377–404.

Zhou, Z.-H. 2022. Open-environment Machine Learning. *National Science Review*, 9(8).

Zhou, Z.-H. 2025. *Ensemble Methods: Foundations and Algorithms*. CRC press.