2025 IEEE International Conference on Robotics and Automation (ICRA)
May 19-23, 2025. Atlanta, USA

LamPro: Multi-prototype representation learning for enhanced visual pattern recognition

Ji Qi[1], Wei Sun[2], Qihe Huang[2], Zhengyang Zhou[2*] and Yang Wang[2]

***Abstract*— Visual pattern recognition usually plays important roles in robotics and automation society where the pattern recognition relies on representation learning. Existing representation learning often neglects two important issues, the diversity of intra-class representation and under-exploited label utilization, especially the negative feedback during training process. Fortunately, prototype learning potentially raises label utilization and encourages intra-class diversity. In this paper, we investigate the intra-class diversity and effective updates in prototype learning for enhanced visual pattern recognition. Specifically, we propose a Label-aware multi-Prototype learning, LamPro, by incorporating the label awareness into both prototype formation and update to improve the representation quality. Firstly, we design a supervised contrastive learning to achieve class-discriminative representations. Secondly, we randomly initialize multiple prototypes and update the nearest prototype upon the arrival of instance, to preserve intra-class diversity. Thirdly, we propose a novel Label-guided Adaptive Updating. We separate the prototype updates from the representation optimization and exploit the label indexes to directly implement the prediction feedback. To correct the model optimization directions, we identify the negative feedback, and correct the prototype updates via queries of labels. Finally, we design a memory-based counter to alternately update these deviated prototypes. Experiments verify the effectiveness of our label-aware and joint multi-prototype updating strategies.**

I. Introduction

Visual pattern recognition is often of great significance for operations of robots [1], [2]. Among artificial intelligence solutions, representation learning based on neural networks has contributed to enable high-quality recognition and pattern extraction on both graph and image signals [3], [4]. However, the model performances do not only depend on the designed neural architectures, but also heavily rely on the data quality. Actually, the increases of data volumes has not improved their quality but introducing computation burdens and noise. Thus, emerging real-world datasets tend to exhibit three characteristics, i.e., data volume explosion, outlier noise, and intra-class pattern diversity, which inherently challenge the learning models [5]. To reduce the impacts of data volume explosion and prominent outliers, prototype learning has been introduced [6]. Generally, prototype learning aims to select samples with high signal-noise ratios within one class and summarize the commonality of these samples into representative representations, therefore it can be exploited to filter noise and improve both learning and inference efficiency. Recently, researchers have found that prototypes can not only contribute the learning efficiency improvement and data refinement, but also can be explored to facilitate model designs, enabling more challenging tasks such as few-shot learning [7]. Therefore, prototypes can be incorporated into deep models to improve representation quality.

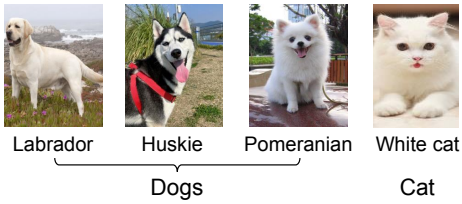

Fig. 1. Illustration of intra-class diversity

For general representation learning, there are two issues, i.e., representation diversity and under-exploited label utilization, hindering better representations. First, we raise a new feature of intra-class diversity, which is an inevitable issue in real-world datasets. For instance, in Figure 1, all three photos fall into the same category of dog but with different patterns in their shapes and colors. Thus, models without considering the intra-class representation diversity fail to form irregular but compact boundary shapes for gaining tolerances of unseen instances within seen classes. Second, the ground-truth labels are still under exploited. Although various label-aware [8], [9] and label distribution learning [10], [11] have been developed by inserting label learning objectives, we argue that such objective-based optimization is still suboptimal as it progressively adjusts the learnable parameters with minor steps and can be influenced by the randomness of gradient descent. Actually, a natural learning law reveals that the negative feedback will impress the learner much more and teach him a lesson [12]. Therefore, the learning process calls for a more direct way of label exploitation, especially emphasizing the negative feedback, to facilitate the rectification on model updates.

The common issues in existing learning methods elicit us to encourage irregular but compact class representation boundaries and develop efficient model updating strategy with informative negative feedback. Fortunately, prototype learning, which enjoys the representation summarization capacity and flexible updating, is introduced to pattern recognition and potentially to tackle above two issues [13], [14]. Prototype learning can be generally classified into two aspects, single prototype and multi-prototype. For single-prototype ones, they are first proposed to summarize the commonality

*Zhengyang Zhou is the corresponding author.
[1]Ji Qi is with China Mobile (Suzhou) Software Technology Co., Ltd, China. qiji@cmss.chinamobile.com
[2] Q Huang, W Sun, Z Zhou and Y Wang are with the University of Science and Technology of China (USTC), Suzhou, Jiangsu, China. zzy0929@ustc.edu.cn

2025 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3315-4139-2/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICRA55743.2025.11127977

of intra-class instances and enable dynamic prototype updating, which open an avenue to efficiently capture more robust patterns and reduce the memory in inference [15], [16], [17]. But given intra-class diversity, the single prototype strategy inherently falls short in forming rational but irregular feature spaces for accommodating the intra-class diversity. To this end, an emerging literature [18] proposes a multi-prototype solution to encourage intra-class diversity by assigning multiple prototypes for each class. However, the reported results in [18] reveal that such solution cannot fully uncover the potential edge of multi-prototype mechanism. In this way, we speculate the reason lies in that the prototypes are updated via gradient backpropagation and trapped into small updating step size, which deteriorate the diversity preserving capacity in multi-prototype learning. Therefore, a more direct updating strategy, along with effective label utilization on prototype updates is required.

To this end, exploiting the multi-prototype scheme to enable representation diversity and high-quality label utilization is both promising and challenging. We can summarize the challenges below, 1) How to exploit prototypes to construct the irregular but compact class boundary that possesses inter-class separation and intra-class diversity? 2) How to fully exploit the labels to establish active feedback for timely prototype update and effective model adjustment?

To tackle above two challenges, we shed light on a Label-aware multi-Prototype learning scheme, LamPro, seamlessly incorporating the label awareness into both prototype formation and updates. In order to achieve the separation among classes, we borrow the idea of contrastive learning and construct a contrastive loss regularization term into the loss function. To preserve the intra-class diversity, we first randomly initialize multiple prototypes for each class and propose a simple yet effective updating strategy to improve the prototype granularity from class-level to pattern-level. Particularly, for each arrived instance, we generally update the geometrically nearest prototype to maximally preserve the original shape of overall class boundary, thus suppressing the interference between two far-away intra-class instances. To this end, the well-learned prototypes in each class can jointly establish the irregular but compact class boundary. Secondly, to enable effective updates and negative feedback emphasis, we take advantage of informative labels and propose a Label-guided Adaptive Updating strategy. First, we separate the representation learning and prototype update to avoid intractable gradient propagation with discrete prototype labels and ensure a more flexible and direct prediction feedback for models. To inform whether the model optimizes towards correct directions, we improve the updating strategies upon different prediction feedback. When positive feedback is identified by labels, we update the nearest prototype to the arrived new instance, which can be viewed as the beneficial prototype dominating the results. However, such nearest updating strategy can potentially bring in a deviation issue, i.e., the prototypes will potentially deviate away from their intrinsic representation space due to the unsatisfactory initialization and imbalanced feature distributions of arriving

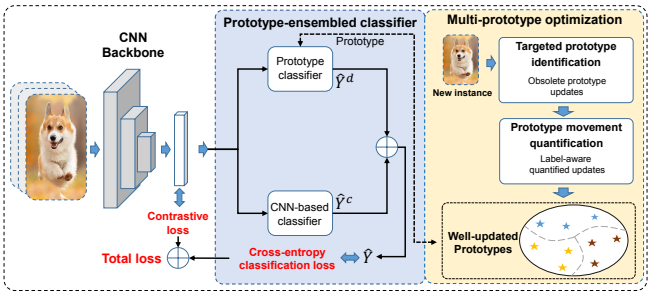

Fig. 2. Model architecture of LamPro

samples. We designate such deviated prototype as obsolete prototype, and design a memory-based updating counter to alternately update these obsolete prototypes until they get rid of such role. When negative feedback arrives, we exploit the labels to identify the correct prototypes and select the geometrically nearest prototype for corresponding instance-level updating, thus enabling the high efficient and negative feedback-aware updating. Extensive experiments on six datasets have demonstrated the competitive performances of LamPro against baselines, showing the superiority for visual recognition on both graphs and images and supporting the robot automation.

II. Related Work

Visual pattern recognition is of great significance for automatic cruise in robotic fields [19], [20]. The open-world visual signals can be generally divided into graphs and images where representation learning has made contributions. Existing representation learning mostly focuses on modifying learning objectives [21], [22]. For instance, [23] proposes to integrate cross-entropy and contrastive loss where cross-entropy encourages the intra-class diversity while contrastive loss simultaneously amplifies inter-class diversity and minimizes intra-class diversity. Further, [24] devises a triplet loss to learn representation in the compact Euclidean space via instance-level similarity, and such well-learned representation has been exploited in tasks from recognition, verification to clustering. To promote the summarized common representation, prototype learning is proposed from the neural network-based Learning Vector Quantization (LVQ) [16]. Prototypes, derived by samples with high signal-noise ratios, summarize the commonality of good samples into representative representations thus filtering noise and facilitating the inference. There are two popular ways for prototype updating, one develops condition and rule-based methods to regularize the updating [25] while another devises various optimization objectives to guide the updates [16]. Even so, we argue that these objective or condition-based updating strategies are not straightforward to receive the prediction feedback and the single prototype for each class directly neglects the intra-class diversity in real-world datasets.

III. METHOD

A. Problem definition

We realize our label-aware multi-prototype learning architecture upon image classification tasks. Given the training set $\mathbb{S} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{|S|}$ where $\boldsymbol{x}_i$ is the image and y_i is the class label. Let $\mathbb{U} = \{\boldsymbol{x}_i^u, y_i^u\}_{i=1}^{|U|}$ be the unseen set left for testing. We aim to design a multi-prototype construction function $f(\boldsymbol{x}_i, y_i; \boldsymbol{\theta}_f)$ to derive a prototype set $\mathbb{P} = \{\boldsymbol{P}_j^k\}(j = 1, ..., C, k = 1, ..., K)$ where C is the total number of classes in training set while K is number of prototypes assigned for each class, and a label-aware updating strategy $g(\mathbb{P}; \boldsymbol{\theta}_g)$ to dynamically achieve the latest prototype. We can finally obtain the predicted $\widehat{y}_i^u$ by the prototype assigment $Assign$. The process is formulated as, $\mathbb{S} \xrightarrow{f \circ g} P_i^k, \widehat{y}_i^u = Assign(f \circ g; \boldsymbol{\theta}_f, \boldsymbol{\theta}_g)$.

B. Framework overview

To simultaneously model the intra-class diversity and enable label awareness in representation learning, we propose our LamPro, a label-aware multi-prototype learning strategy. The overview of proposed LamPro is illustrated in Figure 2. As illustrated, our LamPro consists of two main components, a prototype-ensembled classifier $f_d(\cdot)$ to explicitly incorporate the prototype-based solution for final prediction, and a multi-prototype optimization module for fine-grained and timely prototype updates.

C. Prototype-ensembled classifier

To model the intra-class diversity, we propose to exploit a multi-prototype scheme to accommodate different patterns within the same class. However, in prototype learning, the initialized prototypes only take the role of placeholders for pattern learning and will be progressively updated via subsequent prototype optimization. Thus, we leverage an ensembled classifier, which is composed of a prototype-driven classifier and an CNN feature extractor driven classifier, to jointly achieve final prediction, where both powers of prototypes and CNN-based representation can be maximumly exploited. Then we can elaborate these two branches.

Assume there are totally C classes in a given task and K patterns within each class, we first initialize $C \times K$ prototypes, where these prototypes will be gradually updated with strategies introduced in Sec. III-D. Actually, the CNN-based classifier and prototype initialization are mutually dependent. Given an input image $\boldsymbol{x}$, we first abstract its feature representation via a CNN-based extractor $f_\theta(\boldsymbol{x})$, and then we can exploit the CNN-based extractor to determine the appropriate prototypes. Specifically, we choose the Euclidean distance as the affinity measurement and calculate the distances between extracted features and all prototypes via computing the inverse of Euclidean similarities, then we select the closest prototype to the extracted feature map $f_\theta(\boldsymbol{x})$ as the most probable prototype where the corresponding class is the predicted class. To obtain the prediction probability, we further impose the Softmax function to the derived distances $d_j(\boldsymbol{x})$ to normalize the probability to $(0, 1)$. The implementation can be summarized as Eq. (1)-(2),

$$d_j(x) = \frac{1}{\min\limits_{k=1}^{K} \left\| f_\theta(\boldsymbol{x}_i) - \boldsymbol{P}_j^k \right\|_2^2} \tag{1}$$

$$\widehat{Y}_j^p = \mathrm{Softmax}\{d_j(\boldsymbol{x})\} \tag{2}$$

$\widehat{Y}_j^p$ represents the probability that sample $\boldsymbol{x}$ is classified into class j through the prototype-driven classifier. Detoning $\boldsymbol{P}_y^k$ as the prototype of ground-truth, we can derive that our learning objective will force to minimize $\min\limits_{k=1}^{K} \left\| f_\theta(\boldsymbol{x}_i) - \boldsymbol{P}_y^k \right\|_2^2$ and increase $\sum\limits_{j \neq y} \min\limits_{k=1}^{K} \left\| f_\theta(\boldsymbol{x}_i) - \boldsymbol{P}_j^k \right\|_2^2$. It is worth noting that this process is exactly ingenious and can realize an iteratively mutual enhancement. Specifically, the prototypes will be determined and optimized by CNN extractor while the representations will also tend to approach to the prototypes during the learning process. After then, benefiting from the prototypes, we can summarize the representations from diverse samples into compact prototypes where the noise can be explicitly filtered out and the representation can be significantly improved. Therefore, we can sufficiently exploit the prototypes to enhance CNN-based representations.

Even though, if the model is totally guided by the prototype at initialized steps, the results may not be satisfactory as the prototypes are not well updated. To get rid of this situation, we ensemble a fully connected layer-based classifier, which is directly cascaded by the CNN-based extractor and map the representation to a probability value. To this end, our learning process becomes an ensembled classifier where the importance between two classifiers are fused with α,

$$\widehat{Y} = \alpha \widehat{Y}^d + (1 - \alpha)\, \widehat{Y}^c \tag{3}$$

where $\widehat{Y}^d$ refers to the predicted probability given by prototypes, while $\widehat{Y}^c$ is the prediction of CNN-MLP classifier. For the adjustable parameter α, we will first impose a small α close to 0 and let it progressively increase during the training process. When α is close to 0, the CNN-based classifier can dominate the final results as the initialized prototypes are still embryonic and unreliable. With the training process of our model, the prototypes gets reliable, and we further increase α to let the model gradually enjoy the advantages of prototype learning. At the end stage of training, we shrink the proportion of MLP classifier to zero and fully utilize prototypes prediction with $\alpha = 1$ where it can sufficiently achieve inter-class separation and intra-class diversity.

Benefits. The benefits of this gradually learning process are as follows. First, we can take advantage of MLP to expedite the model convergence and obtain better representation extraction, while secondly, our model can win a buffering time to progressively update prototypes into a reliable template during the steps with small α.

D. Multi-prototype optimization

Actually, data in nature mostly satisfy Gaussian distribution, with the majority concentrated on the expectation

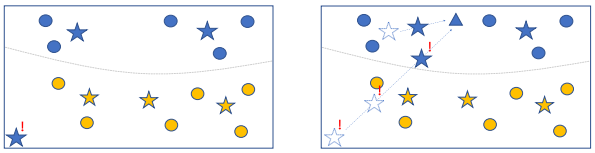

Fig. 3. Illustration of updating process. The left figure represents the situation of obsolete prototypes, and the right figure represents the process of prototype adjustment. In the figures, the circle represents the sample representation, the star represents the prototype, the triangle is a new representation instance and the prototype with an exclamation mark in the upper right corner indicates the obsolescence.

while only minority of them scattered across edges. To this end, in classification tasks, the virtual class boundary is often critically determined by the minority of edge data which is far away from concentration points. Thus, how to construct an irregular class space from complex data distribution has always been a challenge. In this work, we exploit the both summarization and anti-nosie capacity of prototypes, and construct multiple prototypes for each class to formalize the irregular class space. We then elaborate the solution to dynamically updating prototypes through receiving real-time incoming representations, where the core objective for prototype representation is to achieve the inter-class separation and intra-class diversity. In fact, our prototype optimization process can be generally categorized into three-fold, the contrastive-based optimization, determining the updated prototype and calculating how much to move for each updated prototype.

1) Contrastive-based inter-class separation: First, in order to achieve inter-class separation, we borrow the idea of unsupervised contrastive learning and propose a regularization term to preserve such separation,

$$I_c = \frac{\sum_{i=1}^{N}\sum_{j=i+1}^{N} \mathbb{I}(y_i = y_j) e^{\mathrm{sim}(f(x_i), f(x_j))/\tau}}{\sum_{i=1}^{N}\sum_{j=i+1}^{N} e^{\mathrm{sim}(f(x_i), f(x_j))/\tau}} \tag{4}$$

$$\mathcal{L}_{\mathrm{cov}} = -\frac{1}{C}\sum_{c=1}^{C} \log I_c \tag{5}$$

N is the size of a mini-batch, and $\mathbb{I}$ is an indicator function, e.g., the indicator $\mathbb{I}(\cdot)$ becomes 1 if and only if sample $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ belong to same label, otherwise $\mathbb{I}$ is 0. Specifically, $\mathrm{sim}(\cdot)$ indicates the similarity between representations, and τ is the hyperparameter of temperature. This contrastive-based regularization guarantees the representations between different classes separated as much as possible while representations between same classes compact enough.

2) Multi-prototype updater: We propose a label-aware prototype updater that adaptively refines multiple prototypes for each class, addressing intra-class diversity. The updater assigns new instances to prototypes and operates in two stages: targeted prototype identification and prototype movement quantification.

Targeted prototype identification. When a new instance arrives, the updater selects the closest prototype for updating to maintain intra-class diversity. However, inactive prototypes can become obsolete, drifting from the representation space. To prevent this, we use a memory-based counter to track the number of instances assigned to each prototype. Prototypes with fewer than a threshold number δ of instances are deemed obsolescent and are re-integrated into the representation space until they regain sufficient sample representation.

Prototype movement quantification. In this stage, we quantify how much a prototype should move during each update. We emphasize the importance of negative feedback (misclassifications) in forming class boundaries, giving greater significance to incorrectly classified instances. This label-aware approach ensures prototypes adjust according to both correct and incorrect predictions, refining class boundaries without impacting the classifier. When the prototype-based classifier makes the right decision, we will impose a moving-average based update process to corresponding prototype. Given the k-th prototype within i-th class $\boldsymbol{P}_j^k$ and an instance representation $f_\theta(\boldsymbol{x}_j)$ for $\boldsymbol{x}_j$, we can obtain a linear combination of both prototype $\boldsymbol{P}_j^k$ and $f_\theta(\boldsymbol{x}_j)$,

$$\boldsymbol{P}_j^k = (1-\lambda)\,\boldsymbol{P}_j^k + \lambda f_\theta(\boldsymbol{x}_j) \tag{6}$$

where λ controls the update volume. When the prototype-based classifier provides an erroneous decision, it not only manifests the low quality of existing prototype representation, but also indicates the inferior and irrational representation space for this class. Fortunately, the first issue can be fixed by gradient descent with our learning objective. For the second issue, we take advantage of the sample representation of misclassified one to adjust the class representation space, which emphasizes the incorrect feedback and closes the gap between the irrational prototype and corresponding class representation space. Specifically, we push away the prototypes away from the sample representation while bring the prototype closer to correct class representation space. We can modify Eq.(6) to formalize this updating process as,

$$\boldsymbol{P}_j^k = (1-\eta\lambda)\,\boldsymbol{P}_j^k - \eta\lambda f_\theta(\boldsymbol{x}_j) \tag{7}$$

where $\boldsymbol{P}_j^k$ is the prototype with erroneous classification, and $\boldsymbol{x}_j$ is the new instance. We especially impose a positive integer η to accommodate the movement scale, and drastically push away the prototype from the incorrect representation space with $-\eta\lambda$ where parameters η, λ satisfy $0 \leqslant \eta\lambda \leqslant 1$. With this solution, we can put more emphasis on the update volume of misclassified samples to correct classification and simultaneously push away the incorrect prototypes.

E. Learning objective

Our proposal is operated on classification tasks and our goal is to achieve higher classification accuracy and lower classification loss. Here, we have chosen the commonly used Mean Squared Error (MSE) as our loss function, and we have added a contrastive loss regularization term to achieve inter-class separation and intra-class compactness. Given totally

TABLE I

THE STATISTICS OF DATASETS

Image dataset	Number	Size	Channel	Class
CIFAR-10	60000	32×32	3	10
CIFAR-100	60000	32×32	3	100
Caltech101	9146	300×200	3	101
Graph dataset	Node	Edge	Feature	Class
Cora	2708	5249	1433	7
Citeseer	3327	3703	3703	6
Pubmed	19717	44338	500	3

TABLE II

PERFORMANCE ON IMAGE DATESETS (BEST RESULTS ARE IN BOLD)

	CIFAR-10	CIFAR-100	Caltech 101
ResNet-18	90.00±0.63	70.27±0.45	62.89±0.85
ResNet-18 (k=1)	91.25±0.88	71.27±0.36	63.12±0.79
ResNet-18 (k=20)	**93.56±0.37**	**72.63±0.35**	**64.51±0.69**
ResNet-50	92.21±0.67	74.83±0.61	68.65±1.25
ResNet-50 (k=1)	93.10±0.65	75.28±0.51	69.29±0.94
ResNet-50 (k=20)	**94.12±0.39**	**76.93±0.32**	**70.58±0.94**
CvT-7	78.81±0.42	62.34±0.56	52.35±0.89
CvT-7(k=1)	80.12±0.25	63.25±0.52	53.89±0.67
CvT-7(k=20)	**81.26±0.36**	**63.98±0.29**	**54.59±0.85**

TABLE III

PERFORMANCE ON GRAPH DATESETS

	Cora	Citeseer	Pubmed
GCN	79.82±0.53	69.32±0.43	78.13±0.43
GCN (K=1)	80.21±0.12	69.65±0.34	78.22±0.22
GCN (K=20)	**83.26±0.19**	**72.71±0.13**	**81.13±0.08**
DeepWalk	67.03±0.23	43.32±0.34	64.45±0.77
DeepWalk (K=1)	67.13±0.15	43.01±0.12	64.79±0.54
DeepWalk (K=20)	**69.87±0.43**	**47.41±0.54**	**67.76±0.34**
GAT	81.33±0.27	71.23±0.55	78.02±0.32
GAT (K=1)	82.67±0.34	72.31±0.21	78.11±0.65
GAT (K=20)	**85.23±0.52**	**73.22±0.23**	**82.43±0.31**

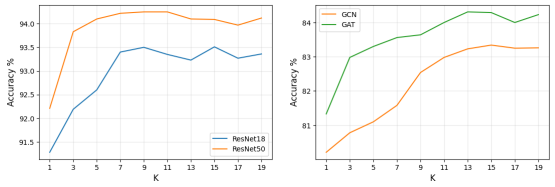

Fig. 4. The impact of the number of prototypes on classification results, where the left figure is image classification models, and the right figure is graph node classification models.

N samples, our final loss function can be derived by,

$$\mathcal{L}_{cls} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y}_i) \tag{8}$$

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{cov}} \tag{9}$$

IV. EXPERIMENTS

We adopt six datasets including graphs and images which are common signals in robotic visual tasks, and foundation models for image and graph learning are as our baselines.

A. Datasets

As open-world visual signals can be generally divided into graphs and images, we evaluate our methods on both image and graph datasets. Three prevailing image datasets for recognition tasks include **CIFAR-10**, **CIFAR-100** [26], **Caltech-101** [27], while three widely adopted real-world graph datasets include **Cora** [28], **Citeseer** [29] and **Pubmed** [30]. The detailed statistics are presented in Table I.

B. Baselines and backbones

We select six backbones to evaluate our solution. We employ two classical deep learning architectures for image classification and three graph-based learning frameworks for graph-level classification. We incorporate our solution with these backbones to analyze the performance variations as well as conduct the ablation studies.

ResNet: A class of deep CNN models, with residual connections. In our work, we employ ResNet-18 and ResNet-50 as our backbone for evaluation [31]. **CvT:** A transformer-based architecture that organizes the transformer blocks with convolution blocks, to realize the image classification [32], where CvT-7 is exploited. **GCN:** A classic graph-structured data modeling baseline, where it uses GCN layer to aggregate information from node's local neighborhood and update node embedding [33]. **DeepWalk:** A node embedding strategy by using random walks in a neural network, which samples nodes and treats the walks as sentences to learn distributed node representations via a skip-gram model [34]. **GAT:** It incorporates self-attention mechanisms into GCNs to learn node embeddings. It leverages node-wise relationship to assign attention scores to each neighbor, for information aggregation [35].

To verify the effectiveness of our LamPro, we incorporate our adaptive prototype updating with above backbones. More specifically, since the improvement of prediction performance become less prominent as K gets larger than 10, we then let the number of prototypes K vary ranging from 1 to 20 for multi-prototype testing.

C. Results

The performance of different methods on image datasets are summarized in Table II while performance on graphs are in Table III. Due to the space limitation, we only report the scenario of $K = \{0, 1, 20\}$. The reported results explicitly illustrate three backbones those are with and without the prototype updating, where it shrinks to a single prototype when $K = 1$. We have following three observations.

Performance comparison. Compared among baselines those are with prototypes and without prototypes, prototype-based representation learning can generally compete non-prototype baselines by 1.64% to 4.41%. Specifically, we find that the performances of multiple prototypes can reveal prominent advantages against the single prototype, e.g., performances at K = 20 are better than performances at K = 1, which verifies the motivation of intra-class diversity and multi-prototype solution. Among them, ResNet-18 has an improvement of nearly 3.5% on CIFAR-10, and other benchmarks also have a general improvement of approxi-

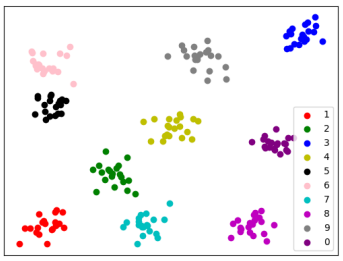

Fig. 5. Prototypes visualization of ResNet-50 on the CIFAR-10.

mately 2%. This not only confirms the existence of multiple pattrens in the same class but also demonstrates the reliability of our proposed multi-prototype update strategy.

Effects of different K. We let the parameter of different number of prototypes K vary on the same model and visualize the prediction results in Figure 4. We gradually increase the value of K and find that the performance will improve accordingly. Typically, we choose ResNet-18 and ResNet-50 on the CIFAR-10 dataset while GCN and GAT on Cora dataset for visualization. We find that on the image dataset, when the number of prototypes reaches 7, the improvement of model performance reaches its limit, while in graph node classification, the improvement slows down when the number of K larger than 13.

Difference on two types of tasks. The reasons behind such differences are attributed to more complex patterns within graphs. For example, the Cora dataset is a citation network consisting of scientific publications from different subjects, where it contains 2,708 scientific papers, and each of them is represented as a binary bag-of-words feature vector, indicating the presence or absence of words in the corresponding document. Such complexity can be reflected by the local neighboring aggregation of GNN, consisting of both neighbors and node itself. To this end, graphs are deservedly with more diverse patterns.

D. Representation visualization

We utilize dimensionality reduction and visualization techniques to map prototypes into a two-dimensional space for illustration. We select a typical task, i.e., representation learning with ResNet-50 on CIFAR-10, in Figure 5. We can clearly see that not only there exists enough distance spaces between different classes, but each class also has a clear boundary. Compared with single-prototype based representation, our method has more space to accommodate the diversity for different intra-class samples. At this time, the diversity representation space provided by multiple prototypes can possess stronger representation capacity. The visualized results demonstrate the quality of our multi-prototype learning and further confirm the interpretability when LamPro is adapted to vital visual recognition tasks in automatic robots.

TABLE IV

THE IMPACTS OF λ ON PERFORMANCE.

Impacts on images	CIFAR-10	CIFAR-100
ResNet-50 (λ=0.001)	93.1	76.01
ResNet-50 (λ=0.005)	**94.2**	76.43
ResNet-50 (λ=0.01)	94.12	**77.12**
ResNet-50 (λ=0.05)	93.56	77.03
ResNet-50 (λ=0.1)	93.43	76.43
Impacts on graphs	Cora	Citeseer
GCN (λ=0.001)	81.78	70.76
GCN (λ=0.005)	82.98	71.65
GCN (λ=0.01)	**83.43**	**72.65**
GCN (λ=0.05)	83.23	72.34
GCN (λ=0.1)	82.14	71.32

E. Hyperparameter setting

For updating quantity λ, we conduct a series of analysis experiments, where results are presented in Table IV. In our experiment settings, we set the prototypes number uniformly to 20 and select five different values of λ, i.e., $\{0.001, 0.005, 0.01, 0.05, 0.1\}$ for studies. Based on the experimental results, we find the values of λ cannot achieve optimal performances when it is either too large or too small. Specifically, when λ becomes too large, the prototype gets overly dependent on the representations of subsequent instances, which causes insufficient preservation of previous representation information and leads to suboptimal inductive ability. Conversely, when λ is too small, the updating quantity of the prototype is inadequate, resulting in slow updating rates and unsatisfactory performance. Finally, we choose a median value and set $\lambda = 0.01$. In the settings, the hyperparameter η is searched from $\{10, 11, ..., 25\}$ and we observe that the best performance is achieved when $\eta = 17$. We also set multiple test values for the sample quantity threshold δ of the obsolete prototype, and through experiments, we found that $\delta = 20$ yields the best performance.

V. CONCLUSION

In this paper, we address the intra-class diversity and effective prototype updates in prototype learning, to consolidate the deep representation capacity. To this end, we design a multi-prototype learning scheme, LamPro via improving the label exploitation. Specifically, LamPro accommodates the irregular intra-class boundary and representation diversity by assigning multiple prototypes for each class. To enable effective prototype updates, a label-guided adaptive updating strategy is proposed, which separates the updates of representation and prototypes, and actively identifies the negative prediction feedback for immediate correction of model optimization. The empirical results and interpretable visualizations have verified the effectiveness and interpretability of LamPro for visual recognition.

ACKNOWLEDGMENT

This paper is partially supported by Jiangsu Natural Science Foundation (No.BK20240460), the National Natural Science Foundation of China (No.62072427, No.12227901), and the grant from State Key Laboratory of Resources and Environmental Information System.

REFERENCES

[1] J. Lu, F. Liu, C. Girerd, and M. C. Yip, "Image-based pose estimation and shape reconstruction for robot manipulators and soft, continuum robots via differentiable rendering," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 560–567, IEEE, 2023.

[2] N. Kumar, H.-M. Chao, B. D. D. S. Tassari, E. Sabinson, I. D. Walker, and K. E. Green, "Design of two morphing robot surfaces and results from a user study on what people want and expect of them, towards a" robot-room"," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11239–11244, IEEE, 2024.

[3] Z. Han, C. Fang, and X. Ding, "Discriminative prototype learning in open set face recognition," in *2010 20th International Conference on Pattern Recognition*, pp. 2696–2699, IEEE, 2010.

[4] C.-L. Liu, "One-vs-all training of prototype classifier for pattern classification and retrieval," in *2010 20th International Conference on Pattern Recognition*, pp. 3328–3331, IEEE, 2010.

[5] C. Zhang, J. Jin, J. Frey, N. Rudin, M. Mattamala, C. Cadena, and M. Hutter, "Resilient legged local navigation: Learning to traverse with compromised perception end-to-end," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 34–41, IEEE, 2024.

[6] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[7] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[8] H. Chen, Y. Xu, F. Huang, Z. Deng, W. Huang, S. Wang, P. He, and Z. Li, "Label-aware graph convolutional networks," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 1977–1980, 2020.

[9] Y. Liu, W. Zhang, and Y. Yu, "Aggregating crowd wisdom with side information via a clustering-based label-aware autoencoder," in *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp. 1542–1548, 2021.

[10] N. Cao, T. Zhang, and H. Jin, "Partial multi-label optimal margin distribution machine.," in *IJCAI*, pp. 2198–2204, 2021.

[11] J. Wang and X. Geng, "Classification with label distribution learning.," in *IJCAI*, pp. 3712–3718, 2019.

[12] A. Franco, A. Lumini, and D. Maio, "A new approach for relevance feedback through positive and negative samples," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 4, pp. 905–908, IEEE, 2004.

[13] B. Saleh, A. M. Elgammal, and J. Feldman, "Incorporating prototype theory in convolutional neural networks.," in *IJCAI*, pp. 3446–3453, 2016.

[14] F. Aiolli, A. Sperduti, and Y. Singer, "Multiclass classification with multi-prototype support vector machines.," *Journal of Machine Learning Research*, vol. 6, no. 5, 2005.

[15] C.-L. Liu and M. Nakagawa, "Evaluation of prototype learning algorithms for nearest-neighbor classifier in application to handwritten character recognition," *Pattern Recognition*, vol. 34, no. 3, pp. 601–615, 2001.

[16] A. Sato and K. Yamada, "A formulation of learning vector quantization using a new misclassification measure," in *Proceedings. Fourteenth international conference on pattern recognition (Cat. No. 98EX170)*, vol. 1, pp. 322–325, IEEE, 1998.

[17] A. Li, P. Yuan, and Z. Li, "Semi-supervised object detection via multi-instance alignment with global class prototypes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9809–9818, 2022.

[18] H.-M. Yang, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Robust classification with convolutional prototype learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3474–3482, 2018.

[19] T.-H. Lin, C.-T. Chang, B.-H. Yang, C.-C. Hung, and K.-W. Wen, "Ai-powered shotcrete robot for enhancing structural integrity using ultra-high performance concrete and visual recognition," *Automation in Construction*, vol. 155, p. 105038, 2023.

[20] J. Zheng, J. Zhang, K. Yang, K. Peng, and R. Stiefelhagen, "Materobot: Material recognition in wearable robotics for people with visual impairments," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2303–2309, IEEE, 2024.

[21] S. Jung, J. Lee, X. Meng, B. Boots, and A. Lambert, "V-strong: Visual self-supervised traversability learning for off-road navigation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1766–1773, IEEE, 2024.

[22] M. Zahid and F. T. Pokorny, "Cloudgripper: An open source cloud robotics testbed for robotic manipulation research, benchmarking and data collection at scale," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12076–12082, IEEE, 2024.

[23] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," *Advances in neural information processing systems*, vol. 27, 2014.

[24] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

[25] S. Geva and J. Sitte, "Adaptive nearest neighbor pattern classification," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 318–322, 1991.

[26] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[27] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 4, pp. 594–611, 2006.

[28] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: extraction and mining of academic social networks," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 990–998, 2008.

[29] C. L. Giles, K. D. Bollacker, and S. Lawrence, "Citeseer: An automatic citation indexing system," in *Proceedings of the third ACM conference on Digital libraries*, pp. 89–98, 1998.

[30] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, pp. 93–93, 2008.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[32] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "Cvt: Introducing convolutions to vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22–31, 2021.

[33] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, vol. 6, no. 1, pp. 1–23, 2019.

[34] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710, 2014.

[35] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.