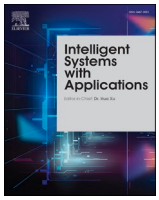
Intelligent Systems with Applications 22 (2024) 200345

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications



Visual representation learning using graph-based higher-order heuristic distillation for cell detection in blood smear images

Hyeokjin Kwon [a], Seonggyu Kim [a], Jihye Ha [b], Eun Jung Baek [b], Jong-Min Lee [a,c,*]

[a] Department of Electronic Engineering, Hanyang University, Seoul 04763, South Korea
[b] Department of Laboratory Medicine, Hanyang University College of Medicine, Guri-si, Gyeonggi-do 11923, South Korea
[c] Department of Biomedical Engineering, Hanyang University, Seoul 04763, South Korea

ARTICLE INFO

Keywords:
Lymphoma classification
Blood smear image
Self-supervised learning
Similarity-based distillation
Graph neural networks

ABSTRACT

Background and objective: In many real-world scenarios, including the blood smear domain, it is difficult for detection networks to achieve good performance because image annotation is usually time consuming and expensive. To address this issue, similarity-based distillation (SD) methods, considered the soft version of contrastive learning, are applied to learn a better visual representation without requiring any supervision of the downstream task. Motivated by our theoretical analysis, we treat standard SD methods as the maximization of common 1-hop neighboring key points between two queries in an attributed graph, where nodes represent query and key data points. However, such first-order graph heuristic methods that calculate the likelihood of an unseen link between target nodes by using up to 1-hop neighborhoods are normally limited by insufficient representation power and even lack of generalization ability.

Methods: Therefore, in this paper, we propose a novel higher-order heuristic distillation (H2D) method that distills knowledge about more general and powerful higher-order heuristic features based on a more than 1-hop relationship in the attributed graph. To do this, we utilize a graph neural network model to learn the higher-order heuristic features on the attributed graph constructed by query and key data representations and transfer the knowledge from the teacher to the student encoder.

Results: Our method outperforms the previous state-of-the-art SD methods in the cell detection task on the blood smear dataset as well on open databases (Pascal VOC and MS COCO). **Conclusions:** Our proposed model allow teacher encoder to transfer the knowledge about more general and powerful higher-order heuristic embeddings to the student and enables better learning for visual representation on cell detection task using blood smear images.

1. Introduction

Lymphomas are the most common hematologic malignancies worldwide (Siegel et al., 2021). Lymphoma could be further classified into plenty of distinct subtypes, which exhibit marked diversity in biological behavior and clinical outcomes (Swerdlow et al., 2016). For diagnosis of lymphomas, immunophenotyping, cytogenetics, molecular pathology results, and clinical features are needed in finalizing the diagnosis in lymphoma types (Swerdlow et al., 2016). Due to subtle differences in cell morphology between normal reactive lymphocytes and abnormal lymphocytes of various types of lymphomas, it is difficult to diagnose lymphoma by morphologic investigation using conventional microscope. Recently, rapid progress in digital imaging and deep learning-based diagnostic systems have made possible the development of automatic methods for digital image processing of blood cells (Kratz et al., 2019). Digital morphology analyzers can preclassify most of the normal blood cells in peripheral blood (PB) and rapidly being implemented routinely in diagnostic laboratories. However, it is still mainly used for differentiating the types of normal blood cells in PB, but it is still challenging to diagnose hematologic malignancies (Kratz et al., 2019). For diagnosing hematologic malignancy, there have been studies to diagnose leukemia using PB blood cell images, but there have been no studies on lymphoma (Boldú et al., 2021; Rehman et al., 2018). Despite the diagnosis of lymphoma in its early stages and the first smears can lead to immediate diagnosis and the quick initiation of the treatment, it is challenging to build a practically applicable deep learning-based

* Corresponding author at: Department of Biomedical Engineering, Electronic Engineering, Hanyang University, Seoul 04763, South Korea.
E-mail address: ljm@hanyang.ac.kr (J.-M. Lee).

https://doi.org/10.1016/j.iswa.2024.200345
Received 24 July 2023; Received in revised form 30 January 2024; Accepted 18 February 2024
Available online 23 February 2024

model for diagnosing lymphoma with PB images due to several drawbacks. Firstly, prediction accuracy of the diagnostic model depends highly on quality and amount of training image samples. However, variations in manual microscopy skills among the scientists and cell staining methods require plenty of training data (Kratz et al., 2019). Different staining methods may result in data heterogeneity that reduces sensitivity in diagnosing hematologic malignancy, by varying morphological features such as texture and color of individual PB cell images. Secondly, imbalanced incidence rate of different lymphomas limits generalization ability of the diagnostic model. For example, Meintker et al. performed an analysis of comparing 4 different hematology analyzers for several cell types except for basophils, which have very low percentage (lower than 0.2 %) (Meintker et al., 2013). It is known that the low number of these cells present in the samples can lead to imprecise prediction results in cell detection tasks (Kratz et al., 2019). Lastly, the diagnosis of specific subtypes is a challenging problem because they are non-discriminant as they share morphological characteristics. For example, when chromatin does not reveal typical clumped pattern in PB cell analysis, it is difficult to diagnosis the cell with chronic lymphocytic leukemia. Similarly, there remain ambiguities when discriminating between reactive lymphocyte and monocyte, or reactive atypical lymphocyte and lymphoma in laboratory.

With the rapid development of deep convolution neural networks, object detection models have been employed for analyses in many biological imaging applications (Chandradevan et al., 2020; Wang et al., 2022; Yu et al., 2019; Ji, 2023; Dong et al., 2023, 2022a, 2022b). However, the training of these models is typically hindered by a lack of well-annotated datasets, which are developed involving laborious and costly processes (Arruda et al., 2019). Moreover, class-imbalanced data, which usually have a long-tail distribution, cause the detection models to under-represent the data owing to the large number of tail classes; this inevitably and negatively impacts the training results (Wang et al., 2021). A potential way to alleviate this issue is to use unsupervised learning, which involves constructing an effective representation based on unlabeled data (Chen et al., 2020; He et al., 2020). Among the many different training methods based on unsupervised learning, we can view the most mainstream approach as contrastive learning, which exhibits promising results (Chen et al., 2020). Although this enables the model to learn useful representations without human supervision, careful treatment of negative pairs is necessary because they may be from the same class category as the positive pair (Grill et al., 2020). In such a scenario, the resulting representation can become far worse by forcing the model to increase the distance between many negative pairs from the same class category (Grill et al., 2020; Tejankar et al., 2021). In particular, an imbalanced cell class distribution in PB analysis further deteriorates the problem, which limits the application of contrastive learning methods to real-world datasets (Tejankar et al., 2021). A recently proposed similarity-based distillation (SD) approach mitigates this issue by relaxing the binary distinction of the contrastive learning framework with soft labeling (Tejankar et al., 2021). By optimizing the student encoder to replicate similarity distribution in the embedding space, these methods aim to transfer the knowledge from the teacher encoder in terms of the distributions between the query point and the other anchor (key) points (Abbasi Koohpayegani et al., 2020; Fang et al., 2021). Thus, the SD methods rely on the agreement between the similarity distribution from the teacher and the student, which makes the student encoder learn the visual representation by maximizing common neighboring key points between two query points.

Based on our theoretical results, we investigate the inherent mechanisms of the SD method and conclude that they result in a lack of expressive power of the learned visual representations. When assuming an attributed graph with unseen edges where nodes represent a set of query and key points, the SD method should be considered as a soft version of Common-Neighbors (CN), which measure the score of the unseen 'link' between the nodes based on the number of overlapped one-hop neighbors (Barabási & Albert, 1999; Zhang & Chen, 2018). Under the assumption that the query points (nodes) have an inherent link in the attributed graph, the student encoder is trained to embed visual representations that maximize the CN heuristic score between two query nodes by applying standard SD methods. However, it is shown that higher-order heuristic methods that can be obtained by considering more than one-hop relationships are more informative for link prediction, suggesting that the SD-based mechanism where the teacher transfers more informative knowledge using higher-order heuristics benefits more than the standard SD methods (Yun et al., 2021). Furthermore, predefined structural features, such as CN heuristics, lack the ability to express general graph structural features underlying different domains (Zhang & Chen, 2018; Yun et al., 2021).

To address these issues, we propose a novel higher-order heuristic-based distillation (H2D) method. In our mechanism, more general and informative higher-order heuristic features are learned by utilizing a graph neural network (GNN)-based model on an attributed graph consisting of query and key data points. Then, we train the student encoder by optimizing the agreement loss between the higher-order heuristic features from the two query nodes (points). We first perform an analysis using the faster R-CNN detector (Ren et al., 2015) with ResNet-50 as the backbone encoder on our local blood smear image dataset to evaluate our H2D for the cell detection task. Compared with the ordinary case of SD methods for image classification, we utilize a weakly supervised learning framework by defining 'tuple' as the input data sample to adapt to the object detection task (see Fig. 1). Our proposed model does not require any class information for bounding box annotating, enhancing generalizability of the model by making it easier to collect more training data. Thus, we expect that our model can alleviate the drawbacks of deep learning based PB image analysis (e.g., a lack of dataset, rare incidence rate, non-distinguishable class patterns). Moreover, we also evaluate our model trained with our H2D method using the widely used MS COCO (Lin et al., 2014) dataset by fine-tuning the model on the Pascal-VOC dataset (Everingham et al., 2010) to further assess generality. In summary, the key contributions of this paper are as follows: (1) We develop a theory unifying the CN heuristic method in graph link prediction tasks and the similarity-based distillation learning mechanism to address insufficient visual representation learning. (2) We present a new representation-learning framework, namely H2D, using more informative and general features based on the attributed graph, which consists of query and key data points to transfer that knowledge to the student. (3) We demonstrate that H2D outperforms recent state-of-the-art visual representation learning methods on our local blood smear dataset. H2D also outperforms the benchmarks in experiments using the MS COCO and Pascal VOC datasets.

2. Related works

2.1. Similarity-based distillation

SD aims to transfer knowledge from a teacher to a student in various ways. The objective of contrastive learning-based approaches such as MOCO (He et al., 2020) and SimCLR (Chen et al., 2020) can be considered as a special case of SD (Tejankar et al., 2021). Instead of assigning a soft label (similarity value) to the negative samples, they treated all key data points strictly as negative and generated a one-hot vector. In SEED (Fang et al., 2021) and Compress (Abbasi Koohpayegani et al., 2020), knowledge is extracted by a large and freezing teacher network and transferred to a small student. ISD (Tejankar et al., 2021) iteratively distills a slowly evolving teacher to a student with similarity-based knowledge by applying the momentum update framework. In the case of BYOL (Grill et al., 2020), the objective was to learn a visual representation by training a student to predict the teacher's embedding vector from an augmented view of the same image.

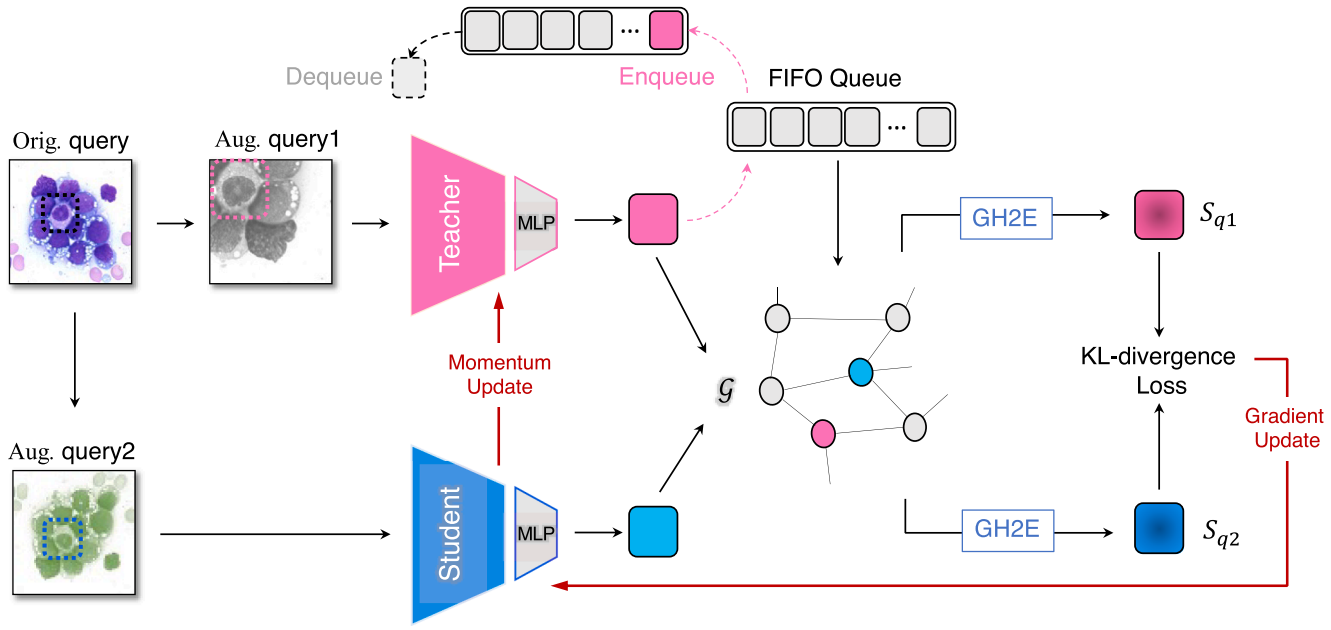


Fig. 1. Illustration of the proposed H2D method. We used a tuple consisting of bounding box information (dashed square) of an instance (a cell) and a corresponding image as the data sample. We fed them to the student (skyblue) and the teacher (pink) encoder, and applied ROI pooling with an MLP prediction layer to embed representations. By constructing the attributed graph $\mathcal{G}$ with two query and key embeddings from a FIFO queue, we further calculate the final query representations that have general higher-order heuristic information of the graph using the GH2E networks. Finally, the parameters of the student and teacher encoders were updated by optimizing the KL-divergence loss between the final query representations and by applying the momentum update rule (red arrows).

2.2. Link prediction and graph-based heuristics

Link prediction is a key problem in graph-structured data; it predicts the likelihood of unseen links between nodes in the graph and has many applications such as recommendation systems (Li et al., 2014), knowledge graph completion (Kazemi & Poole, 2018), and graph reconstruction (Oyetunde et al., 2017). Diverse heuristic methods that compute score functions to measure the likelihood of links based on structural information have been widely used (Yun et al., 2021). Link prediction heuristic methods can be categorized based on the range of neighboring nodes required to calculate link scores (Zhang & Chen, 2018). First- and second-order heuristic methods involve one- or two-hop neighborhood nodes, respectively. For example, the CN and preferential attachment (Barabási & Albert, 1999) methods calculate the score using overlapped one-hop neighborhood nodes. Adamic-Adar (Adamic & Adar, 2003) and resource allocation (Zhou et al., 2009) require knowledge of up to two-hop neighborhood nodes. Conventionally, higher-order heuristic methods such as the Katz index (Katz, 1953), PageRank (Brin & Page, 1998), and SimRank (Jeh & Widom, 2002) methods, which consider more than two-hop neighborhoods (usually up to the entire network), have shown significant improvements over first- and second-order heuristic methods for link prediction (Zhang & Chen, 2018).

2.3. Learning structural features of graph

In addition to heuristic-based methods, embedding-based methods that automatically learn more general and powerful higher-order features have been studied to predict link existence (Zhang & Chen, 2018). The Weisfeiler-Lehman Neural machine (Zhang & Chen, 2017) uses a fully connected network that encloses subgraphs for the link prediction of target nodes. SEAL (Zhang & Chen, 2018) also utilizes enclosing subgraphs with a GNN to classify whether two central nodes in a subgraph have a link. In addition, they emphasize the need to consider some node features (e.g., explicit and latent features) in link prediction and propose a double-radius node labeling strategy that encodes the node's role and position information. Conversely, neighborhood overlap-aware GNNs (Neo-GNNs) (Yun et al., 2021) generalize higher-order heuristic features by combining the structural neighborhood overlap-aware information learned from an adjacency matrix and input node features for link prediction.

3. Methods

3.1. Theorem on the relationship between SD and graph heuristic methods

In this section, we present the theoretical justification for understanding why standard SD methods can lead to insufficient visual representation. Furthermore, we provide insights into a novel SD framework that can learn more general and effective higher-order heuristics to transfer that knowledge to the student. First, we define a generalized CN score to support this.

Definition 3.1 (*Generalized CN*). For a weighted and homophilic graph $G = (V, E)$ where $v \in V$ and E represent nodes and a set of edges, respectively, a generalized CN score $\widetilde{f}_{CN}(i,j)$ for two nodes $(i,j) \in V$ has the following form.

$$\widetilde{f}_{CN}(i,j) = \sum_{k \in \{V \setminus (i,j)\}} \boldsymbol{q}_i(k)\boldsymbol{q}_j(k) \tag{1}$$

where, $\boldsymbol{q}_v \in \mathbb{R}^{|V\setminus(i,j)|}$ is the similarity-based probability distribution for node $v \in (i,j)$ and the other nodes from the complementary node set $\{V\setminus(i,j)\}$. The standard CN score $f_{CN}(i,j) = |\Gamma_i \cap \Gamma_j|$, where Γ_v is a set of one-hop neighboring nodes for node v, is a special case of the generalized CN score when the similarity distribution has binary entries $\boldsymbol{q}_v(u) \in \{0,1\}, \forall (u,v)$ which indicate whether the corresponding nodes (u,v) have an edge. Next, we describe the connection between standard SD methods and the generalized CN defined above.

Theorem 3.1. With the momentum update rule, the objective of the standard SD is equivalent to maximizing the generalized CN score in Definition 3.1 between two query nodes in the attributed graph with unseen edges, where nodes represent a set of query and key data points.

$$\widehat{\theta}_s \approx \underset{\theta_s}{\operatorname{argmax}} \widetilde{f}_{CN}(t,s) \tag{2}$$

where, θ_s ($\widehat{\theta_s}$) is the (optimal) trainable parameter of the student encoder and t, and s denote two query nodes, respectively. The proof is presented in supplementary material.

3.2. Discussion for the theorem

Our theoretical results show that the visual representation of the standard SD method is optimized by maximizing the generalized CN heuristic in Definition 3.1 between the two query nodes from the student and teacher, under the assumption that there exists an inherent link between the query nodes. However, this framework has two limitations. First, (generalized) CN methods may lead to unsatisfactory results because they only involve one-hop neighborhood nodes. It has been shown that higher-order heuristic-based methods that consider a wider range of neighborhood nodes are more effective in many domains (Zhang & Chen, 2018). Second, predefined heuristics such as CN may fail to generalize to various domains (Zhang & Chen, 2018). For example, it may be inappropriate to use heterophily, which indicates that the nodes from different classes are more likely to connect to one another, which is widely occurring in real-world data. To alleviate these issues, recent state-of-the-art embedding-based methods that automatically learn more general and powerful higher-order heuristics by using neural network models such as GNNs have shown prominent results in various domains (Yun et al., 2021). Thus, we propose a novel SD-based framework that provides knowledge about more effective higher-order heuristics for students by utilizing the GNN considering the structural information and node features simultaneously.

3.3. Proposed method: H2D

Our dataset consists of a set of N_{ins} (= the number of instances) tuples $\{\boldsymbol{b}_i, \boldsymbol{X}_i\},\ \forall\ i = 1, \ldots, N_{ins}$ where $\boldsymbol{b}_i$ and $\boldsymbol{X}_i$ denote the bounding-box coordinates and the corresponding image for a target instance ($i-th$ object), respectively. A query tuple {**b**,**X**} is augmented twice by applying independent random transforms, resulting in two different views of the sample: $\{\boldsymbol{b}_s, \boldsymbol{X}_s\}$ and $\{\boldsymbol{b}_t, \boldsymbol{X}_t\}$. We feed the images $(\boldsymbol{X}_s, \boldsymbol{X}_t)$ to the student and teacher encoders, respectively, and the ROI pooling method (Girshick, 2015) is applied to the output feature maps $(\boldsymbol{F}_s, \boldsymbol{F}_t)$ with the bounding box information $(\boldsymbol{b}_s, \boldsymbol{b}_t)$ to obtain their embeddings. With the embeddings, the fully connected layer computes the feature vectors with fixed length d, and they are further normalized into final embeddings, denoted z_s and z_t. Furthermore, to obtain a consistent and large set of key embeddings for better representation learning, we follow prior works with the first input and first output (FIFO) queue idea by reusing the encoded embedding vectors from the teacher network on preceding mini-batches (Abbasi Koohpayegani et al., 2020). For a FIFO data queue, we have a set of K key embedding vectors, $\{z_i\}_{i=1}^{K}$, where K is a user-defined hyperparameter.

3.3.1. Attributed graph construction

While the standard SD methods directly use the embeddings to calculate the similarity scores, we define an attributed graph G, where nodes correspond to queries, and key embeddings and edges encode attribute similarities between the nodes. First, we concatenate the key and query embeddings to obtain the node feature matrix $Z = [z_t|z_s|z_1|z_2|\ldots|z_K]^T \in \mathbb{R}^{(K+2)\times d}$. Given the node feature matrix Z, a symmetric similarity matrix $S \in \mathbb{R}^{(K+2)\times(K+2)}$ encoding the similarity scores between the nodes for G can be calculated as follow:

$$S = \sigma\left(ZZ^T\right) \tag{3}$$

where, $\sigma(\cdot)$ denotes the sigmoid function. Subsequently, an adjacency matrix $A \in \mathbb{R}^{(K+2)\times(K+2)}$ is defined by applying a threshold value of 0.5 to S. Finally, we remove the edge between two query nodes$(A_{ts}, A_{st} \leftarrow 0)$ to prevent our model from being exposed to the link existence information, resulting in a new adjacency matrix $A_c \in \mathbb{R}^{(K+2)\times(K+2)}$.

3.3.2. Graph-based higher-order heuristics embedding networks

Inspired by recent link prediction studies (Zhang & Chen, 2018; Yun et al., 2021), we developed a new graph-based higher-order heuristic embedding (GH2E) network to learn a better higher-order heuristic information based on the attributed graph G. The GH2E model consists of (1) structural embedding layers that extract graph structural information using an adjacency matrix A_c only, and (2) feature embedding layers that compute node feature representations based on A_c and node input features Z (see Fig. 2). First, the structural embedding layers generate structural feature scalars $\{h_i^{struct}\}_{i=1}^{K+2}$ for each node i based on local information. Specifically, we first calculate the hidden edge representations using a two-layer multilayer perceptron (MLP) and perform neighborhood aggregation:

$$h_i^{aggr} = \sum_{j \in \mathcal{N}_i} \left(A_{c,ij} W_1^{edge}\right) W_2^{edge} \tag{4}$$

where, $A_{c,ij} \in \{0, 1\},\ \forall\ (i, j)$ indicates whether the corresponding nodes have an edge, and $\mathcal{N}_i$ denotes a set of neighborhood nodes of node i. $W_1^{edge} \in \mathbb{R}^{1\times d}$ and $W_2^{edge} \in \mathbb{R}^{1\times d}$ are trainable parameters. Then, similar to edge embedding, we update the aggregated representation using a two-layer MLP:

$$h_i^{struct} = \left(h_i^{aggr} W_1^{node}\right) W_2^{node} \tag{5}$$

where, $W_1^{node} \in \mathbb{R}^{1\times d}$ and $W_2^{node} \in \mathbb{R}^{d\times 1}$ are trainable parameters. Now, we have a structural embedding vector $\boldsymbol{h}^{struct} = \{h_i^{struct}\}_{i=1}^{K+2}$ that contains the structural information for all nodes in graph G. Furthermore, the structural embedding matrix $H^{struct} \in \mathbb{R}^{(K+2)\times(K+2)}$ consisting of one-hot encoded node structure feature vectors, can be defined by constructing a diagonal matrix with the structural embedding vector $\boldsymbol{h}^{struct}$. We aggregate the structural information from multi-hop neighboring nodes to obtain the final structural information matrix $H \in \mathbb{R}^{(K+2)\times(K+2)}$ as follows:

$$H = f_{scale}\left(\sum_{l=1}^{L} \beta^{l-1} A_c^l H^{struct}\right) \tag{6}$$

where, L denotes the maximum range of the neighborhoods to be considered and β is a decaying parameter. f_{scale} is a two-layer MLP that controls the scale of output representation. Additionally, a message-passing-based GNN encoder consisting of L_{GNN} layers is applied to embed node feature representations. These GNN-based models have been applied to embed node-level hidden representations by considering higher-order interrelationships between nodes with a nonlinear update function (e.g., W_{GNN}^l in Eq. (4)) and an adjacency matrix representing a structure of the target graph data. Message propagation is performed by aggregating the updated node representations from local neighboring nodes for each node and layer, respectively. By recurrently repeating this process, higher-order nodal information from L-hop neighborhoods can be propagated to embed a target node representation. Formally, given the node feature matrix $Z^1 = Z$, the update rule for the GNN encoder in layer l is defined as follows:

$$Z^{l+1} = ReLU\left(A_c Z^l W_{GNN}^l\right) \tag{7}$$

where, $ReLU(\cdot)$ is a rectified linear unit activation function, $W_{GNN}^l \in \mathbb{R}^{d\times d},\ \forall\ l = 1, 2, \ldots, (L_{GNN} - 1)$ and $W_{GNN}^{L_{GNN}} \in \mathbb{R}^{d\times(K+2)}$ denote the trainable parameters in each layer. Finally, we calculate a convex combination between the structural embedding H and feature embedding $Z^{L_{GNN}}$ using trainable weight α to derive the output embedding matrix:

$$Q = \{\boldsymbol{q}_i\}_{i=1}^{K+2} \text{ as } Q = \alpha H + (1 - \alpha) Z^{L_{GNN}}. \tag{8}$$

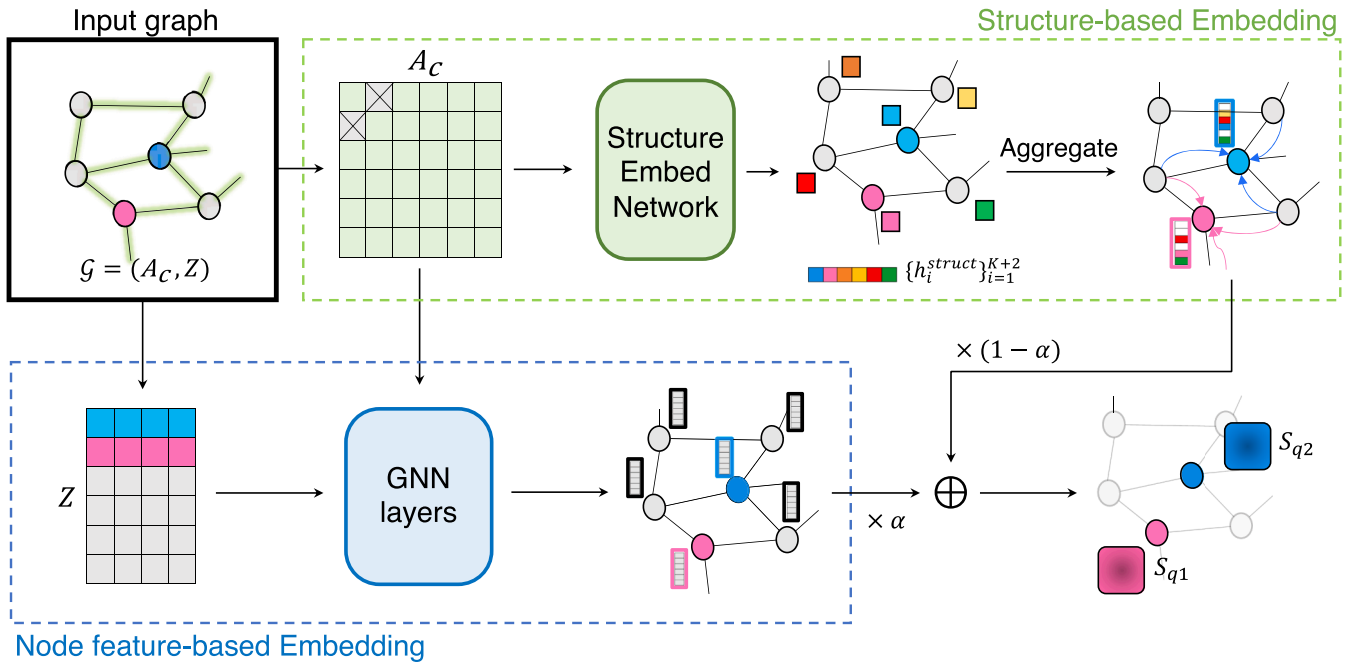


Fig. 2. The illustration of the proposed GH2E framework for an input graph G, including the query nodes for teacher (pink), and student (sky-blue) encoder, respectively. In structure-based embedding layers, hidden edge representations are aggregated to get h_i^{aggr} for each node i. Then a two-layer MLP is applied to calculate the structural information h_i^{struct} for each node i. The multi-hop aggregation of the structural information is performed by using the decayed multi-hop adjacency $\beta^{l-1}A_c^l$ in order to get the final structural information matrix (H) for each layer $l = 1, \ldots, L$, respectively. Moreover, GNN layers are applied to get feature-based node hidden representations $Z^{L_{GNN}}$. Finally, we merge these two representations (the structural information H, and the feature-based information $Z^{L_{GNN}}$), resulting in the output embedding matrix Q.

3.3.3. Loss function and momentum contrast

The output node embeddings Q from the GH2E are further updated to calculate the probability distribution $P = \{\boldsymbol{p}_i\}_{i=1}^{K+2}$ with a temperature parameter τ using a softmax function.

Algorithm 1
Pre-training with H2D.

input: Initialized teacher encoder R_{θ_t}, and student encoder R_{θ_s}. Initialized MLP layer f. Initialized K key vectors $\{z_i\}_{i=1}^{K}$. Initialized GH2E network f_e ,f_n, and f_{GNN} . Decay parameter β. Multi-hop layer depth L. Initialized weight α. Temperature parameter τ. Momentum parameter m. Random transformation function $\mathcal{T}$.
for sampled minibatch $\{\boldsymbol{b}_i, \boldsymbol{X}_i\}_{i=1}^{N}$ **do**
for all $i \in \{1, \ldots, N\}$ **do** *# drop the subscript i for simplicity in the loop*
 # feature extraction
$aug_s \sim \mathcal{T}$, $aug_t \sim \mathcal{T}$ *# draw two random transformations*
$\{\boldsymbol{b}_s, \boldsymbol{X}_s\} = aug_s(\boldsymbol{b}_i, \boldsymbol{X}_i)$, $\{\boldsymbol{b}_t, \boldsymbol{X}_t\} = aug_t(\boldsymbol{b}_i, \boldsymbol{X}_i)$ *# augmentation*
$\boldsymbol{F}_s = ROIpool(R_{\theta_s}(\boldsymbol{X}_s); \boldsymbol{b}_s)$, $\boldsymbol{F}_t = ROIpool(R_{\theta_t}(\boldsymbol{X}_t); \boldsymbol{b}_t)$ *# ROI pooling function ROIpool*$(\cdot)$
$z_s = \text{norm}(f(\boldsymbol{F}_s))$, $z_t = \text{norm}(f(\boldsymbol{F}_t))$
 # attributed graph construction
$Z = concat(z_s; z_t; \{z_i\}_{i=1}^{K})^T$ *# node feature matrix with concatenate function concat*$(\cdot)$
$A = threshold(sigmoid(ZZ^T))$ *# adjacency matrix* (Eq. (3).)
$A_c \leftarrow$*remove query edge of* A *#* $A_{ts}, A_{st} = 0$
 # GH2E
for all node $v \in \{1, \ldots, K+2\}$ **do** *# structure-based embedding*
$h_v^{struct} = f_n\left(\sum_{u \in \mathcal{N}_v} f_e(A_{c,vu})\right)$ *#* f_e *in* Eq. (4). f_n *in* Eq. (5).
end for
$H^{struct} \leftarrow diag(\{h_v^{struct}\}_{v=1}^{K+2})$
$H = MHaggr(H^{struct}; A_c, \beta, L)$ *# multi-hop aggregation MHaggr*$(\cdot)$ *in* Eq. (6).
$Z^{L_{GNN}} = f_{GNN}(A_c, Z)$ *# GNN layers in* Eq. (7).
$Q = \alpha H + (1-\alpha) Z^{L_{GNN}}$ *# in* Eq. (8).
 # calculate the probability distibution
$P = softMax(Q; \tau)$ *# in* Eq. (9).
end for
$\mathcal{L} = \text{KL}(\boldsymbol{p}_t | \boldsymbol{p}_s)$ *# mean KL divergence loss*
update all the networks except for the R_{θ_t} to minimize $\mathcal{L}$
update the teacher network using momentum update rule with m in Eq. (10).
FIFO queue update for the key embedding vector with z_t
end for
return Pre-trained student encoder F_{θ_s}.

$$\boldsymbol{p}_i(j) = -\log\left(\frac{\exp(\boldsymbol{q}_i(j))/\tau}{\sum_{k=1}^{K+2}\exp(\boldsymbol{q}_i(k))/\tau}\right) \tag{9}$$

where, $i,j = 1,2,...(K+2)$. To update the parameters θ_s of the student encoder, we optimized the following KL divergence loss between the final query embeddings: $L = KL(\boldsymbol{p}_t|\boldsymbol{p}_s)$. Because our FIFO queue makes it intractable to apply the ordinary gradient-based back propagation to the teacher encoder, we use the momentum contrast method, which updates the parameters with a moving-average scheme for the teacher:

$$\theta_t \leftarrow m\theta_t + (1-m)\theta_s \tag{10}$$

where, m is a user-defined momentum parameter, and θ_t are trainable parameters of the teacher (He et al., 2020).

4. Experiments

4.1. Implementation details

All our experiments were conducted using one NVIDIA Titan V 12GB GPU except for open data analysis (on four GPUs). We selected all hyper-parameters by adopting them from the literature or by performing a grid search algorithm. PyTorch and TorchVision frameworks were used (Paszke et al., 2019). All codes and models in this section are available at https://github.com/ForBlindReview1/H2D. Please see supplementary material for more details of implementation and dataset. The proposed H2D pretraining method is summarized in Algorithm 1.

4.2. Architecture

We used the popular Faster RCNN with feature-pyramidal networks (FPN) (Lin et al., 2017) as the detector model. The ResNet-50 was used as the backbone network, and we adopted it as a teacher and student encoder. The resulting feature maps from the FPN-ResNet-50 encoder consist of a 256-channel multi-scale output: $\{P_2, P_3, P_4, P_5\}$. To extract fixed-length feature vectors from the multi-scale feature maps of the FPN encoder for each instance (object) in the images, we used the ROI pooling method with a scale assignment strategy (Lin et al., 2017). We extracted a feature map with a fixed spatial extent of 7×7 using the ROI pooling method, resulting in an output feature vector with a size of 12,544 ($= 7 \times 7 \times 256$). Finally, we attached a prediction layer with a two-layer MLP to produce the $d-$ lengthed embedding vector, which was normalized by its L2-norm. To compare our proposed H2D method with state-of-the-art SD methods on weakly supervised pre-training and transfer learning, we adapted the unofficial implementation of the other state-of-the-art SD methods by replacing the standard single-scale feature encoder, which uses a global-average pooling layer with the multi-scale FPN encoder using the ROI pooling method.

5. Results and discussion

5.1. Comparison with state-of-the-art methods

To verify the effectiveness of our method, we compared the performance of H2D to the four state-of-the-art methods reported in Table 1. It can be concluded that the methods with standard contrastive learning (MOCO v2) and other knowledge distillation-based methods including SD (BYOL, Compress, ISD, and the proposed H2D) consistently surpass a baseline method through visual representation learning. Table 1 also shows that our H2D method consistently achieves the state-of-the-art results on cell detection tasks, which indicates that using graph-based higher-order heuristic learning is beneficial for visual representation learning. Examples of cell detection using our proposed framework are shown in Fig. 3.

Table 1
Results of comparison with state-of-the-art SD methods on a cell detection dataset (mean ± deviation). Rand. init. initializes the backbone from the scratch. Bold denote the best performances.

Method	Seed 1			Seed 2		
	mAP	mAP_{50}	*mAR*	*mAP*	mAP_{50}	*mAR*
Rand. Init.	84.9 ± 11.5	95.8 ± 8.2	89.4 ± 6.6	84.9 ± 11.5	95.8 ± 8.2	89.4 ± 6.6
MOCO-v2 (He et al., 2020)	86.5 ± 12.2	95.5 ± 10.0	89.6 ± 8.9	87.0 ± 11.1	96.1 ± 9.0	**90.6 ± 7.2**
BYOL (Grill et al., 2020)	86.6 ± 11.0	96.1 ± 8.1	90.5 ± 6.6	86.8 ± 10.3	96.1 ± 7.7	90.4 ± 6.8
Compress (Abbasi Koohpayegani et al., 2020)	87.0 ± 12.1	95.9 ± 10.3	90.3 ± 8.5	86.6 ± 11.1	96.3 ± 8.6	89.7 ± 8.0
ISD (Tejankar et al., 2021)	87.6 ± 11.8	96.4 ± 9.6	90.4 ± 8.5	87.0 ± 11.7	96.1 ± 8.9	90.4 ± 8.2
H2D (ours)	**87.6 ± 10.6**	**96.7 ± 7.8**	**90.9 ± 7.1**	**87.1 ± 11.1**	**96.6 ± 7.8**	90.6 ± 7.5

5.2. Effect of higher-order heuristics learning

To show that our GH2E can learn a general and powerful higher-order heuristic features compared with various existing graph heuristics, we performed pre-training with the modified GH2E, and compare the results to the original H2D. Given an attributed graph G with key, query nodes and their attributed adjacency A_c, our GH2E first extract hidden edge embedding by using the 2-layer MLP ($f_e(\cdot)$, Eq. (4)), then node-level encoder, $f_n(\cdot)$ (Eq. (5)), outputs a hidden structural node embedding by applying the two-layer MLP to the aggregated edge embeddings for node i:

$$h_i^{struct} = f_n\left(\sum_{j\in\mathcal{N}_i} f_e(A_{ij})\right), \tag{11}$$

Note that it is possible to learn a generalized higher-order heuristic features, and even the other heuristics can be generated by using this mechanism. For example, if $f_e(x) = x$ (identity function), and $f_n(x) = 1$ (constant function), and if we used only structural embedding for the output distribution ($\alpha = 1$), the KL-divergence objective of the standard SD mechanism can be treated as the maximization of standard CN score. Similarly, we replace the MLP layers with $f_e(x) = x$, and $f_n(x) = 1/\log(x)$ respectively to imitate the structural features for the AA, and additionally, set $L \geq 2$ and decay parameter $\beta < 1$ with same setting as standard CN to imitate the Katz, as the standard high-order heuristic. As shown in Table 2, we find that our proposed H2D benefits more than other existing heuristic methods. We further show the influence of the number of aggregation layers and decay parameter in supplementary material.

5.3. Effects of graph structure and node features

Our proposed H2D adopts the GH2E network, which calculates general higher-order heuristic information based on the graph structure and node features to learn rich visual representations. We conducted experiments on various configurations of the GH2E model to demonstrate how they influence the detection performance. As demonstrated in Table 3, H2D without structural information performs worse than the standard SD methods. This is in line with many recent link-prediction studies (Zhang & Chen, 2018; Yun et al., 2021), indicating that structural information is crucial for embedding meaningful heuristics. Interestingly, the H2D method considering only structural information (without node features) also yields poor performance, suggesting that using them together can improve representation learning.

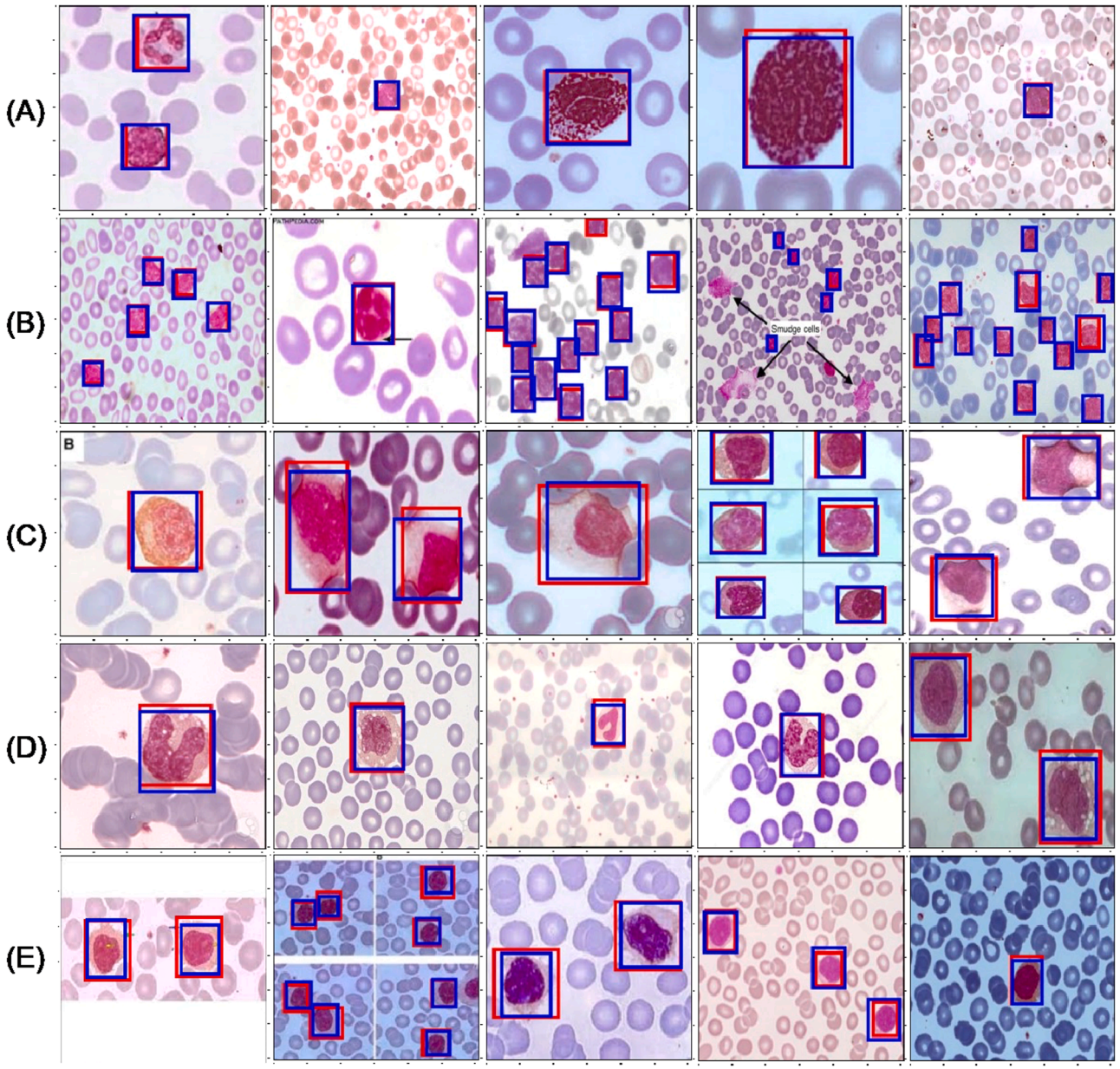


Fig. 3. Cell detection results using our proposed method, H2D, for example classes including (A) basophil, (B) chronic lymphocytic leukemia, (C) reactive lymphocyte, (D) monocyte, and (E) splenic marginal zone lymphoma. Blue and red boxes denote bounding boxes for ground truth, and prediction, respectively.

Table 2
Results of comparison with other heuristic methods on a cell detection dataset (mean ± standard deviation). Bold denote the best performances.

Method	*mAP*	mAP_{50}	*mAR*
CN	86.7 ± 11.7	95.7 ± 9.8	90.2 ± 7.8
AA	87.2 ± 12.0	95.9 ± 10.0	90.5 ± 7.1
Katz	87.4 ± 10.5	96.2 ± 8.2	90.8 ± 6.3
H2D (ours)	**87.6 ± 10.6**	**96.7 ± 7.8**	**90.9 ± 7.1**

Table 3
Effects of structural and feature embedding layers on H2D. Bold denote the best performances.

Method	Structure embedding	Feature embedding	mAP
H2D		o	86.5 ± 11.7
	o		86.7 ± 10.6
	o	o	**87.6 ± 10.6**

We also conducted experiments with various hyperparameters for the GH2E network and report the results in Table 4. For the node feature analysis, we can observe that the GNN consisting of two layers works the best. We suspect that this result might be caused by the over-smoothing issue, which means that the GNN causes node representations to converge to indistinguishable values with an increase in network layers (Li et al., 2019; Zhao et al., 2018). Moreover, for the case of structural information, considering up to 3-hop neighborhoods ($L = 3$) with a decay parameter β of one achieves the best performance, as shown in Fig. 4. To represent structural information for target node, our GH2E iteratively aggregate the hidden features of neighboring nodes. The effective range of nodes that a node's feature draws from is heavily

Table 4
Effect of number of GNN layers. Bold denote the best performances.

	1	*2*	*3*	*4*
mAP	86.6 ± 11.6	**87.6 ± 10.6**	86.8 ± 12.2	87.3 ± 10.0

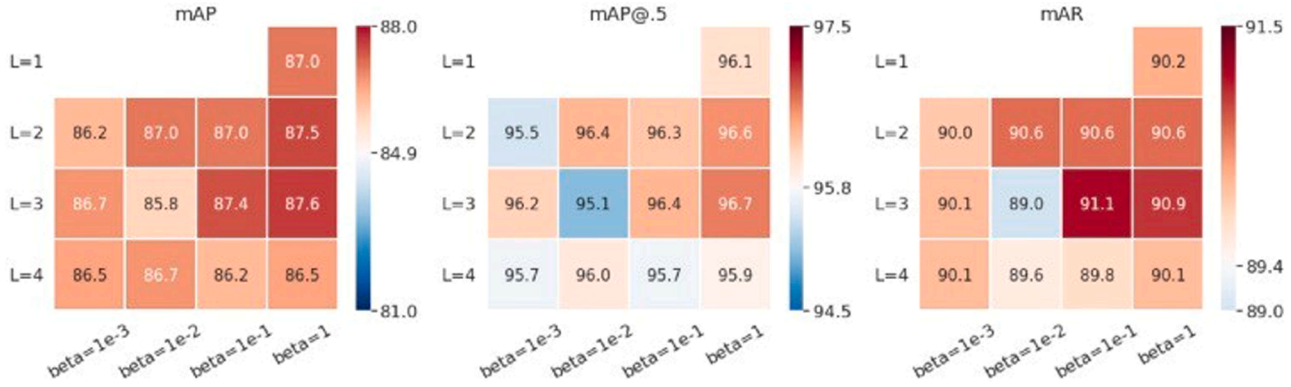


Fig. 4. Ablative studies of different aggregate ranges (L) and decay parameters (β) in the GH2E model.

affected by graph structure with the expansions of a random walk manner (Xu et al., 2018). Thus, depending on the structure of local subgraph, too rapid expansion of the effective range may lose information *via* averaging. We suspect that this may results in a performance drop when increasing L to 4.

5.4. Open dataset experiments

To test whether our proposed method can be successfully applied to more general cases, we conducted experiments using the MS COCO and Pascal VOC datasets. Details of the implementation are presented in supplementary material. We followed the same evaluation metric used in Section 5.1 and reported the results in Table 5. Our H2D achieves 47.8 mAP under IoU=[0.5:0.95], which is best performance among different methods (same result as the MOCOv2, and BYOL). And we observe that H2D brings clear improvement over the state-of-the-art methods in term of strict version of mAP (mAP_{75}, mAP under IoU=0.75), and mAR, respectively. We can conclude that these open-dataset results show the generality of the H2D method.

6. Conclusion

Despite sufficient progress, future work should explore other higher-order heuristic embedding networks because our GH2E requires $O(|E|)$ space and computational complexity, limiting the number of queue sizes K to a relatively small value. This differs from other SD methods that use a large dictionary queue ($K = 128$k). Fortunately, the small dictionary queue ($K = 64$ or $K = 96$) was successfully applied to the blood smear data, Pascal VOC, and MS COCO image datasets because they have small class categories ($N \leq 80$). However, to adapt to larger databases, such as ImageNet (1000 categories) (Russakovsky et al., 2015), there is a need for a more efficient higher-order heuristic embedding network.

In this study, we analyze the limitations of standard SD methods and propose a novel knowledge distillation method, namely H2D. Instead of simply calculating the existing higher-order heuristics or the similarity distributions, we allow H2D to transfer the knowledge about more general and powerful higher-order heuristic embeddings to the student by utilizing the GNN-based GH2E model on the constructed attributed graph. Our extensive experiments show that the proposed H2D method can learn rich visual representations compared to previous state-of-the-art SD methods. We hope that H2D will inspire future relational research, such as knowledge distillation and contrastive learning.

CRediT authorship contribution statement

Hyeokjin Kwon: Conceptualization, Methodology, Writing – original draft. **Seonggyu Kim:** . **Jihye Ha:** Data curation, Writing – review & editing. **Eun Jung Baek:** Data curation, Writing – review & editing. **Jong-Min Lee:** Supervision, Writing – review & editing.

Table 5
Results of the open-dataset experiments. bold denote the best performances.

Method	*mAP*	mAP_{50}	mAP_{75}	*mAR*
Rand. Init.	40.4	70.6	41.5	53.9
MOCO-v2	**47.8**	77.7	51.5	59.5
BYOL	**47.8**	**77.8**	51.5	59.5
Compress	47.6	77.6	51.1	59.3
ISD	47.7	77.7	51.5	59.5
H2D (ours)	**47.8**	77.6	**51.7**	**59.7**

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2020-0-01373, Artificial Intelligence Graduate School Program (Hanyang University)) and the Bio & Medical Technology Development Program of the National Research Foundation (NRF) funded by the Korean government (MSIT) (No. 2020M3E5D9080788).

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.iswa.2024.200345.

References

Abbasi Koohpayegani, S., Tejankar, A., & Pirsiavash, H. (2020). Compress: Self-supervised learning by compressing representations. In , *33. Proceedings of the advances in neural information processing systems* (pp. 12980–12992).
Adamic, L. A., & Adar, E. (2003). Friends and neighbors on the web. *Social Networks, 25* (3), 211–230.
Arruda, V. F., et al. (2019). Cross-domain car detection using unsupervised image-to-image translation: From day to night. In *Proceedings of the 2019 international joint conference on neural networks (IJCNN)*. IEEE.
Barabási, A.-L., & Albert, R. (1999). Emergence of scaling in random networks. *Science, 286*(5439), 509–512.
Boldú, L., et al. (2021). A deep learning model (ALNet) for the diagnosis of acute leukaemia lineage using peripheral blood cell images. *Computer Methods and Programs in Biomedicine, 202*, Article 105999.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems, 30*(1–7), 107–117.
Chandradevan, R., et al. (2020). Machine-based detection and classification for bone marrow aspirate differential counts: Initial development focusing on nonneoplastic cells. *Laboratory Investigation, 100*(1), 98–109.
Chen, T., et al. (2020). A simple framework for contrastive learning of visual representations. In *Proceedings of the international conference on machine learning*. PMLR.
Dong, Z., et al. (2022a). Multimodal neuromorphic sensory-processing system with memristor circuits for smart home applications. *IEEE Transactions on Industry Applications, 59*(1), 47–58.
Dong, Z., et al. (2022b). Memristor-based hierarchical attention network for multimodal affective computing in mental health monitoring. In *Proceedings of the IEEE consumer electronics magazine*.
Dong, Z., et al. (2023). ICNCS: Internal cascaded neuromorphic computing system for fast electric vehicle state of charge estimation. *IEEE Transactions on Consumer Electronics*.
Everingham, M., et al. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision, 88*, 303–338.
Fang, Z., et al., Seed: Self-supervised distillation for visual representation. arXiv preprint arXiv:2101.04731, 2021.
Girshick, R. (2015). Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision*.
Grill, J.-B., et al. (2020). Bootstrap your own latent-a new approach to self-supervised learning. In *, 33. Proceedings of the advances in neural information processing systems* (pp. 21271–21284).
He, K., et al. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.
Jeh, G., & Widom, J. (2002). Simrank: A measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining*.
Ji, X., et al. (2023). EMSN: An energy-efficient memristive sequencer network for human emotion classification in mental health monitoring. *IEEE Transactions on Consumer Electronics, 69*(4), 1005–1016.
Katz, L. (1953). A new status index derived from sociometric analysis. *Psychometrika, 18*(1), 39–43.
Kazemi, S. M., & Poole, D. (2018). Simple embedding for link prediction in knowledge graphs. In *Proceedings of the advances in neural information processing systems* (p. 31).
Kratz, A., et al. (2019). Digital morphology analyzers in hematology: ICSH review and recommendations. *International Journal of Laboratory Hematology, 41*(4), 437–447.
Li, G., et al. (2019). Deepgcns: Can GCNS go as deep as CNNS?. In *Proceedings of the IEEE/CVF international conference on computer vision*.
Li, J., et al. (2014). Recommendation algorithm based on link prediction and domain knowledge in retail transactions. *Procedia Computer Science, 31*, 875–881.
Lin, T.-Y., et al. (2014). Microsoft coco: Common objects in context. In *Proceedings of the computer vision–ECCV2014: 13th European conference*. Springer. September 6-12, 2014, Proceedings, Part V 13.
Lin, T.-Y., et al. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.
Meintker, L., et al. (2013). Comparison of automated differential blood cell counts from Abbott Sapphire, Siemens Advia 120, Beckman Coulter DxH 800, and Sysmex XE-2100 in normal and pathologic samples. *American Journal of Clinical Pathology, 139*(5), 641–650.
Oyetunde, T., et al. (2017). BoostGAPFILL: Improving the fidelity of metabolic network reconstructions through integrated constraint and pattern-based methods. *Bioinformatics, 33*(4), 608–611 (Oxford, England).
Paszke, A., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of the advances in neural information processing systems* (p. 32).
Rehman, A., et al. (2018). Classification of acute lymphoblastic leukemia using deep learning. *Microscopy Research and Technique, 81*(11), 1310–1317.
Ren, S., et al. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proceedings of the advances in neural information processing systems* (p. 28).
Russakovsky, O., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision, 115*, 211–252.
Siegel, R. L., et al. (2021). Cancer statistics, 2021. *CA: A Cancer Journal for Clinicians, 71*(1), 7–33.
Swerdlow, S. H., et al. (2016). The 2016 revision of the world health organization classification of lymphoid neoplasms. *Blood, The Journal of the American Society of Hematology, 127*(20), 2375–2390.
Tejankar, A., et al. (2021). Isd: Self-supervised learning by iterative similarity distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*.
Wang, C.-W., et al. (2022). Deep learning for bone marrow cell detection and classification on whole-slide images. *Medical Image Analysis, 75*, Article 102270.
Wang, T., et al. (2021). Adaptive class suppression loss for long-tail object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.
Xu, K., et al. (2018). Representation learning on graphs with jumping knowledge networks. In *Proceedings of the international conference on machine learning*. PMLR.
Yu, T.-C., et al. (2019). Automatic bone marrow cell identification and classification by deep neural network. *Blood, 134*, 2084.
Yun, S., et al. (2021). Neo-GNNS: Neighborhood overlap-aware graph neural networks for link prediction. In *, 34. Proceedings of the advances in neural information processing systems* (pp. 13683–13694).
Zhang, M., & Chen, Y. (2017). Weisfeiler-lehman neural machine for link prediction. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*.
Zhang, M., & Chen, Y. (2018). Link prediction based on graph neural networks. In *Proceedings of the advances in neural information processing systems* (p. 31).
Zhao, W., et al., When work matters: Transforming classical network structures to graph cnn. arXiv preprint arXiv:1807.02653, 2018.
Zhou, T., Lü, L., & Zhang, Y.-C. (2009). Predicting missing links *via* local information. *The European Physical Journal B, 71*, 623–630.