HeNCler: Node Clustering in Heterophilous Graphs via Learned Asymmetric Similarity

Sonny Achten[1(✉)], Zander Op de Beeck[1], Francesco Tonin[2], Volkan Cevher[2], and Johan A. K. Suykens[1]

[1] ESAT-STADIUS, KU Leuven, Leuven, Belgium
{sonny.achten,zander.opdebeeck,johan.suykens}@kuleuven.be
[2] LIONS, EPFL, Lausanne, Switzerland
{francesco.tonin,volkan.cevher}@epfl.ch

Abstract. Clustering nodes in heterophilous graphs is challenging as traditional methods assume that effective clustering is characterized by high intra-cluster and low inter-cluster connectivity. To address this, we introduce HeNCler—a novel approach for **He**terophilous **N**ode **Cl**ust**er**ing. HeNCler *learns* a similarity graph by optimizing a clustering-specific objective based on weighted kernel singular value decomposition Our approach enables spectral clustering on an *asymmetric* similarity graph, providing flexibility for both directed and undirected graphs. By solving the primal problem directly, our method overcomes the computational difficulties of traditional adjacency partitioning-based approaches. Experimental results show that HeNCler significantly improves node clustering performance in heterophilous graph settings, highlighting the advantage of its asymmetric graph-learning framework.

Keywords: Heterophily · Clustering · Kernel SVD

1 Introduction

Graph neural networks (GNNs) have substantially advanced machine learning applications to graph-structured data by effectively propagating node attributes end-to-end. Typically, GNNs rely on the assumption of homophily, where nodes with similar labels are more likely to be connected [29]. The homophily assumption holds in contexts such as social networks and citation graphs. It underlies the effectiveness of message-passing GNNs like GCN [14] in tasks such as node classification and graph prediction. In homophilous graphs, local feature aggregation reinforces meaningful representations through neighborhood smoothing.

In heterophilous graphs, such as web or transaction networks, connected nodes often differ in labels, making local aggregation less effective. This requires capturing long-range dependencies by learning edge importance [3,8], which is more feasible in supervised tasks due to label guidance.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16072, pp. 55–68, 2026.
https://doi.org/10.1007/978-3-032-04552-2_8

Our work specifically targets attributed node clustering in a fully unsupervised setting, where neither labels nor external guidance are available—necessitating the use of unsupervised or self-supervised learning strategies. For instance, auto-encoder type models [15] are primarily focused on node representation learning rather than clustering, making them less suited for directly improving cluster-ability. Various self-supervised, contrastive learning techniques [12,27] enhance node representation learning in homophilous settings only. At the same time, other self-supervised methods have been developed to handle heterophilous graphs [5,26]. However, these methods are designed for the general node representation learning task and lack a clustering objective.

Existing node clustering methods [4,6,25] often rely on proximity-based objectives or partitioning of the adjacency matrix, assuming that effective clustering aligns with high intra-cluster and low inter-cluster similarity—a premise often invalid in heterophilous graphs. Moreover, these methods are typically limited to undirected graphs, disregarding valuable asymmetric information.

This paper introduces HeNCler, a novel approach for node clustering in heterophilous graphs, illustrated in Fig. 1. Existing work overlooks the asymmetric relationships in heterophilous graphs. HeNCler addresses this by using weighted kernel singular value decomposition (wKSVD) to induce a learned asymmetric similarity graph for both directed and undirected graphs. The dual problem of wKSVD aligns with asymmetric kernel spectral clustering, enabling the interpretation of similarities without homophily. By solving the primal problem directly, HeNCler overcomes computational difficulties and shows superior performance in node clustering tasks within heterophilous graphs.

Our contributions in this work can be summarized as follows:

- We introduce HeNCler, a kernel spectral biclustering framework designed to *learn* an induced *asymmetric* similarity graph suited for node clustering of heterophilous graphs, applicable to both directed and undirected graphs.
- We develop a primal-dual framework for a generic weighted kernel singular value decomposition (wKSVD) model.
- We show that the dual wKSVD formulation allows for biclustering of bipartite/asymmetric graphs, while we employ a computationally feasible implementation in the primal wKSVD formulation.
- We further generalize our approach with trainable feature mappings, using node and edge decoders, such that the similarity matrix to cluster is learned.
- We train HeNCler in the primal setting and demonstrate its superior performance on the node clustering task for heterophilous attributed graphs.
- Our implementation and supplementary materials are available on GitHub: https://github.com/sonnyachten/HeNCler/.

2 Preliminaries and Related Work

We use lowercase symbols (e.g., x) for scalars, lowercase bold (e.g., $\boldsymbol{x}$) for vectors and uppercase bold (e.g., $\boldsymbol{X}$) for matrices. A single entry of a matrix is

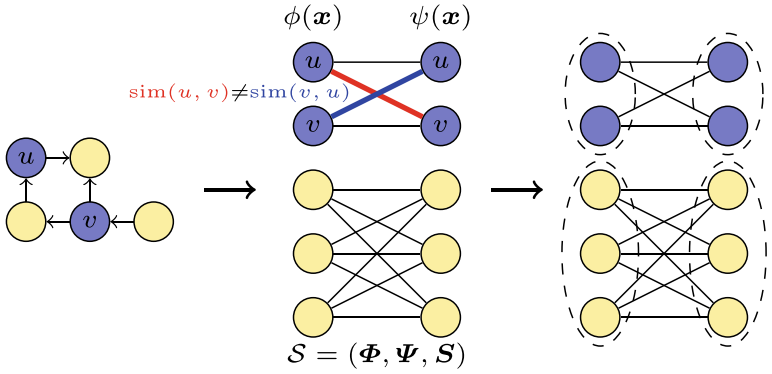


Fig. 1. HeNCler Overview. Starting from a heterophilous (directed) graph, where similar nodes are far apart (left), HeNCler learns two sets of node representations, $\{\phi(\boldsymbol{x}_v)\}_{v\in\mathcal{V}}$ and $\{\psi(\boldsymbol{x}_v)\}_{v\in\mathcal{V}}$, forming a bipartite graph $\mathcal{S}$ (middle), where the similarity between nodes is defined as $S_{uv} = \mathrm{sim}(u, v) = \phi(\boldsymbol{x}_u)^\top \psi(\boldsymbol{x}_v)$. The clustering objective brings related nodes closer in the learned graph, and clusters are then identified using spectral biclustering through wKSVD (right).

represented by X_{ij}. $\phi(\cdot)$ denotes a mapping and $\boldsymbol{\phi}_v = \phi(\boldsymbol{x}_v)$ represents the mapping of node v in the induced feature space. We represent a graph $\mathcal{G}$ by its vertices (i.e., nodes) $\mathcal{V}$ and edges $\mathcal{E}$, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, or by its node feature matrix and adjacency matrix $\mathcal{G} = (\boldsymbol{X}, \boldsymbol{A})$. For a bipartite graph, we have $\mathcal{G} = (\mathcal{I}, \mathcal{J}, \mathcal{E})$ or $\mathcal{G} = (\boldsymbol{X}_\mathcal{I}, \boldsymbol{X}_\mathcal{J}, \boldsymbol{S})$ where S_{ij} is the edge weight between nodes $i \in \mathcal{I}$ and $j \in \mathcal{J}$. Note that $\boldsymbol{S}$ is generally asymmetric and rectangular, and that the adjacency matrix of the bipartite graph is given by $\boldsymbol{A} = \left[\begin{smallmatrix} \mathbf{0} & \boldsymbol{S} \\ \boldsymbol{S}^\top & \mathbf{0} \end{smallmatrix}\right]$.

Kernel singular value decomposition (KSVD) [22] allows for non-linear extensions of the SVD problem and can be applied on data structures such as row and column features, directed graphs, and/or can exploit asymmetric similarity information such as conditional probabilities [13]. Interestingly, KSVD often outperforms the similar, though symmetric, kernel principal component analysis model on tasks where asymmetry is not immediately apparent [13,24].

Spectral clustering generalizations have been proposed in many settings. Spectral graph biclustering [7] formulates the spectral clustering problem of a bipartite graph $\mathcal{G} = (\mathcal{I}, \mathcal{J}, \boldsymbol{S})$ and shows the equivalence with the SVD of the normalized matrix $\boldsymbol{S}_n = \boldsymbol{D}_1^{-1/2} \boldsymbol{S} \boldsymbol{D}_2^{-1/2}$, where $D_{1,ii} = \sum_j S_{ij}$ and $D_{2,jj} = \sum_i S_{ij}$. Cluster assignments for nodes $\mathcal{I}$ and nodes $\mathcal{J}$ can be inferred from the left and right singular vectors respectively. Further, kernel spectral clustering (KSC) [2] proposes a weighted kernel principal component analysis in which the dual formulation corresponds to the random walks interpretation of the spectral clustering problem. KSC and the aforementioned spectral biclustering formulation lack asymmetry and a primal formulation respectively, which are limitations that our model addresses.

Restricted kernel machines (RKM) [23] possess primal and dual model formulations, based on the concept of conjugate feature duality. It is an energy-based framework for (deep) kernel machines. RKMs encompasses many model classes, including classification, regression, kernel principal component analysis, and KSVD, and allows deep kernel learning and deep kernel learning on graphs [1]. One possibility of representing feature maps in RKMs is by means of deep neural networks, e.g., for unsupervised representation learning [16].

Homophilous node clustering methods such as MinCutPool [4] and DMoN [25] introduce unsupervised loss functions in a graph neural network framework. MinCutPool employs a relaxed version of the minimal cut objective applied to the adjacency matrix, while DMoN optimizes the modularity score of clustering assignments with respect to the graph structure. Both methods rely on partitioning the adjacency matrix and inherently assume strong homophily. Moreover, due to their theoretical foundations, these losses are restricted to undirected graphs. Beyond adjacency partitioning, self-supervised and contrastive learning techniques have also been proposed [6,12,27]. These approaches typically use graph proximity as a supervision signal, again assuming a degree of homophily. For instance, S^3GC [6] leverages random walk co-occurrences to infer proximity-based similarities.

Heterophilous node clustering methods predominantly rely on contrastive or self-supervised techniques that are less dependent on local proximity. HoLe [11] addresses unsupervised graph structure learning by iteratively rewiring the graph using intermediate clustering results. This process increases effective homophily, thereby improving clustering performance on moderately heterophilous graphs. SparseGAD [10] sparsifies graph structures to effectively reduce noise from irrelevant edges and enhance the detection of closely related nodes. In contrast, methods like ours aim to learn an entirely new similarity graph, which is particularly valuable for strongly heterophilous settings. Other approaches, such as HGRL [5], employ self-supervised learning by using graph augmentation strategies to capture global and higher-order structural patterns. MUSE [28] builds semantic and contextual views for contrastive learning, integrating multi-view representations through a learned fusion controller.

While adjacency partitioning-based methods have shown strong theoretical and empirical performance on homophilous graphs, they do not extend naturally to heterophilous or directed settings. Conversely, self-supervised clustering approaches, though flexible, often lack explicit clustering objectives and/or still implicitly depend on homophily. In the following section, we introduce HeNCler, which bridges these gaps.

3 Method

Model Motivation. Our approach employs an RKM auto-encoder framework, which has been shown to be effective in unsupervised representation learning by jointly optimizing feature mappings and projection matrices within a kernel-based setting [16]. We propose a weighted KSVD (wKSVD) loss that employs double feature mappings to learn an asymmetric similarity matrix. This *learned*

similarity alleviates the homophily assumption by capturing long-range dependencies and enables the modeling of *asymmetric* relationships—making it particularly well-suited for directed graphs. Even in undirected graphs, this asymmetry can capture nuanced relational patterns. In fact, many graphs treated as undirected are inherently directed (e.g., citation networks). Modeling such asymmetries has been shown to be beneficial [13,24]. Additionally, the wKSVD loss admits a spectral graph biclustering interpretation, offering further theoretical insight. We first introduce the general wKSVD framework, followed by our HeNCler model, which operates in the primal setting and jointly learns the feature mappings in an end-to-end manner.

3.1 Kernel Spectral Biclustering with Asymmetric Similarities

Consider a dataset with two input sources $\{\boldsymbol{x}_i\}_{i=1}^n$ and $\{\boldsymbol{z}_j\}_{j=1}^m$, on which we want to define an unsupervised learning task. To this end, we introduce a weighted kernel singular value decomposition model (wKSVD), starting from the following primal optimization problem, which is a weighted variant of the KSVD formulation:

$$\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{e},\boldsymbol{r}} J \triangleq \mathrm{Tr}(\boldsymbol{U}^\top \boldsymbol{V}) - \frac{1}{2}\sum_{i=1}^{n} w_{1,i}\boldsymbol{e}_i^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{e}_i - \frac{1}{2}\sum_{j=1}^{m} w_{2,j}\boldsymbol{r}_j^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{r}_j$$
$$\text{s.t. } \{\boldsymbol{e}_i = \boldsymbol{U}^\top\phi(\boldsymbol{x}_i),\ \forall i = 1,\ldots,n; \quad \boldsymbol{r}_j = \boldsymbol{V}^\top\psi(\boldsymbol{z}_j), \forall j = 1,\ldots,m\}, \tag{1}$$

with projection matrices $\boldsymbol{U},\boldsymbol{V} \in \mathbb{R}^{d_f \times s}$; strictly positive weighting scalars $w_{1,i}, w_{2,j}$; latent variables $\boldsymbol{e}_i, \boldsymbol{r}_j \in \mathbb{R}^s$; diagonal and positive definite hyperparameter matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{s\times s}$; and centered feature maps $\phi(\cdot) : \mathbb{R}^{d_x} \mapsto \mathbb{R}^{d_f}$ and $\psi(\cdot) : \mathbb{R}^{d_z} \mapsto \mathbb{R}^{d_f}$. Details on centering of the feature maps are provided in the supplementary materials. The following derivation shows the equivalence with the spectral biclustering problem.

Proposition 1. *The solution to the primal problem (1) can be obtained by solving the singular value decomposition of*

$$\boldsymbol{W}_1^{1/2}\boldsymbol{S}\boldsymbol{W}_2^{1/2} = \boldsymbol{H}_e\boldsymbol{\Sigma}\boldsymbol{H}_r^\top, \tag{2}$$

where $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are diagonal matrices such that $W_{1,ii} = w_{1,i}$ and $W_{2,jj} = w_{2,j}$, $\boldsymbol{S} = \boldsymbol{\Phi}\boldsymbol{\Psi}^\top$ is an asymmetric similarity matrix where $S_{ij} = \phi(\boldsymbol{x}_i)^\top\psi(\boldsymbol{z}_j)$, $\boldsymbol{\Phi} = [\phi(\boldsymbol{x}_1)\ldots\phi(\boldsymbol{x}_n)]^\top$, $\boldsymbol{\Psi} = [\psi(\boldsymbol{z}_1)\ldots\psi(\boldsymbol{z}_m)]^\top$, and where $\boldsymbol{H}_e = [\boldsymbol{h}_{\boldsymbol{e}_1}\ldots\boldsymbol{h}_{\boldsymbol{e}_n}]^\top$, and $\boldsymbol{H}_r = [\boldsymbol{h}_{\boldsymbol{r}_1}\ldots\boldsymbol{h}_{\boldsymbol{r}_m}]^\top$ are the left and right singular vectors respectively; and by applying $\boldsymbol{r}_j = \boldsymbol{\Sigma}\boldsymbol{h}_{\boldsymbol{r}_j}/\sqrt{w_{2,j}}$ and $\boldsymbol{e}_i = \boldsymbol{\Sigma}\boldsymbol{h}_{\boldsymbol{e}_i}/\sqrt{w_{1,i}}$.

Proof. We now introduce dual variables $\boldsymbol{h}_{\boldsymbol{e}_i}$ and $\boldsymbol{h}_{\boldsymbol{r}_j}$ using a specific instance of the FenchelYoung inequality [19]:

$$\frac{1}{2}w_{1,i}\ \boldsymbol{e}_i^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{e}_i + \frac{1}{2}\boldsymbol{h}_{\boldsymbol{e}_i}^\top\boldsymbol{\Sigma}\boldsymbol{h}_{\boldsymbol{e}_i} \geq \sqrt{w_{1,i}}\ \boldsymbol{e}_i^\top\boldsymbol{h}_{\boldsymbol{e}_i},$$
$$\frac{1}{2}w_{2,j}\ \boldsymbol{r}_j^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{r}_j + \frac{1}{2}\boldsymbol{h}_{\boldsymbol{r}_j}^\top\boldsymbol{\Sigma}\boldsymbol{h}_{\boldsymbol{r}_j} \geq \sqrt{w_{2,j}}\ \boldsymbol{r}_j^\top\boldsymbol{h}_{\boldsymbol{r}_j}, \tag{3}$$

$\forall \boldsymbol{e}_i, \boldsymbol{r}_j, \boldsymbol{h}_{\boldsymbol{e}_i}, \boldsymbol{h}_{\boldsymbol{r}_j} \in \mathbb{R}^s$; $\forall w_{1,i}, w_{2,j} \in \mathbb{R}_{>0}$; and $\forall \boldsymbol{\Sigma} \in \mathbb{R}^{s \times s}_{\succ 0}$.[1]

By substituting the constraints of (1) and inequalities (3) into the objective function of (1), we obtain an objective in primal and dual variables as an upper bound on the primal objective $\bar{J} \geq J$:

$$\begin{aligned} \min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{h}_e,\boldsymbol{h}_r} \bar{J} \triangleq \operatorname{Tr}(\boldsymbol{U}^\top \boldsymbol{V}) - \sum_{i=1}^{n} \sqrt{w_{1,i}}\ \phi(\boldsymbol{x}_i)^\top \boldsymbol{U} \boldsymbol{h}_{\boldsymbol{e}_i} + \frac{1}{2} \sum_{i=1}^{n} \boldsymbol{h}_{\boldsymbol{e}_i}^\top \boldsymbol{\Sigma} \boldsymbol{h}_{\boldsymbol{e}_i} \\ - \sum_{j=1}^{m} \sqrt{w_{2,j}}\ \psi(\boldsymbol{z}_j)^\top \boldsymbol{V} \boldsymbol{h}_{\boldsymbol{r}_i} + \frac{1}{2} \sum_{j=1}^{m} \boldsymbol{h}_{\boldsymbol{r}_j}^\top \boldsymbol{\Sigma} \boldsymbol{h}_{\boldsymbol{r}_j}. \end{aligned} \tag{4}$$

Next, we formulate the stationarity conditions of problem (4):

$$\begin{aligned} &\frac{\partial \bar{J}}{\partial \boldsymbol{V}} = 0 \Rightarrow\ \boldsymbol{U} = \textstyle\sum_{j=1}^{m} \sqrt{w_{2,j}}\ \psi(\boldsymbol{z}_j) \boldsymbol{h}_{\boldsymbol{r}_j}^\top, \quad \frac{\partial \bar{J}}{\partial \boldsymbol{h}_{\boldsymbol{e}_i}} = 0 \Rightarrow\ \boldsymbol{\Sigma} \boldsymbol{h}_{\boldsymbol{e}_i} = \sqrt{w_{1,i}}\ \boldsymbol{U}^\top \phi(\boldsymbol{x}_i), \\ &\frac{\partial \bar{J}}{\partial \boldsymbol{U}} = 0 \Rightarrow\ \boldsymbol{V} = \textstyle\sum_{i=1}^{n} \sqrt{w_{1,i}}\ \phi(\boldsymbol{x}_i) \boldsymbol{h}_{\boldsymbol{e}_i}^\top, \quad \frac{\partial \bar{J}}{\partial \boldsymbol{h}_{\boldsymbol{r}_j}} = 0 \Rightarrow\ \boldsymbol{\Sigma} \boldsymbol{h}_{\boldsymbol{r}_j} = \sqrt{w_{2,j}}\ \boldsymbol{V}^\top \psi(\boldsymbol{z}_j), \end{aligned} \tag{5}$$

from which we then eliminate the primal variables $\boldsymbol{U}$ and $\boldsymbol{V}$. This yields:

$$\begin{bmatrix} \mathbf{0} & \boldsymbol{W}_1^{1/2} \boldsymbol{S} \boldsymbol{W}_2^{1/2} \\ \boldsymbol{W}_2^{1/2} \boldsymbol{S}^\top \boldsymbol{W}_1^{1/2} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{H}_e \\ \boldsymbol{H}_r \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_e \\ \boldsymbol{H}_r \end{bmatrix} \boldsymbol{\Sigma}, \tag{6}$$

where $\mathbf{0}$ is an all-zeros matrix. The above eigenvalue problem is equivalent with (2), and the stationarity conditions (5) provide the relationships between primal and dual variables, which concludes the proof. □

We have thus shown the connection between the primal (1) and dual formulation (2). Similarly to the KSVD framework, the wKSVD framework can be used for learning with asymmetric kernel functions and/or rectangular data sources. The spectral biclustering problem can now easily be obtained by choosing the weights $w_{1,i}$ and $w_{2,j}$ appropriately.

Corollary 1. *Given Proposition 1, and by choosing* $\boldsymbol{W}_1$ *and* $\boldsymbol{W}_2$ *to equal* $\boldsymbol{D}_1^{-1/2}$ *and* $\boldsymbol{D}_2^{-1/2}$*, where* $D_{1,ii} = \sum_j S_{ij}$ *and* $D_{2,jj} = \sum_i S_{ij}$*, we obtain the random walk interpretation* $\boldsymbol{D}_1^{-1/2} \boldsymbol{S} \boldsymbol{D}_2^{-1/2} = \boldsymbol{H}_e \boldsymbol{\Sigma} \boldsymbol{H}_r^\top$ *of the spectral graph bipartitioning problem for the bipartite graph* $\mathcal{S} = (\boldsymbol{\Phi}, \boldsymbol{\Psi}, \boldsymbol{S})$*.*

Moreover, the wKSVD framework is more general as, on the one hand, one can use a given similarity matrix (e.g. adjacency matrix of a graph) or (asymmetric) kernel function in the dual, or, on the other hand, one can choose to use explicitly defined (deep) feature maps in both primal or dual.

[1] Refer to the supplementary materials for verification of these inequalities.

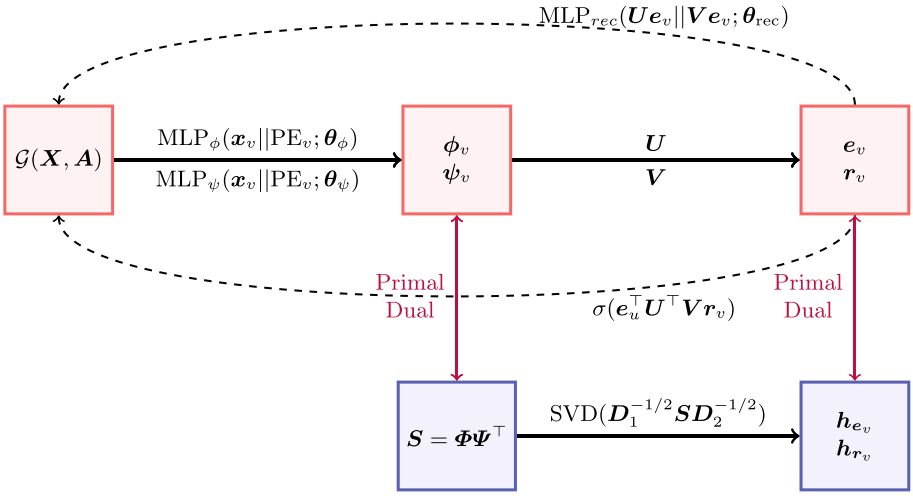


Fig. 2. The HeNCler model. HeNCler operates in the primal setting (top of the figure in red) and uses a double multilayer perceptron (MLP) to map node representations into a feature space. The obtained representations $\boldsymbol{\phi}_v$ and $\boldsymbol{\psi}_v$ are then projected into a latent space, yielding $\boldsymbol{e}_v$ and $\boldsymbol{r}_v$ respectively. The wKSVD loss ensures that these latent representations correspond to the dual equivalent (bottom of the figure in blue) i.e., a biclustering of the asymmetric similarity graph defined by $\boldsymbol{S}$. The node and edge reconstructions (dashed arrows) aid in the feature map learning. (Color figure online)

3.2 The HeNCler Model

HeNCler employs the wKSVD framework in a graph setting, where the dataset is a node set $\mathcal{V}$ and where the asymmetry arises from employing two different mappings that operate on the nodes given the entire graph $\mathcal{G} = (\boldsymbol{X}, \boldsymbol{A})$. Our approach is visualized in Fig. 2.

In the preceding subsection, we showed that the primal wKSVD formulation (1) has an equivalent dual problem corresponding to the graph bipartitioning problem. This equivalence holds when $w_{1,i}$ and $w_{2,j}$ are chosen to equal the square root of the inverse of the out-degree and in-degree of a similarity graph $\mathcal{S}$ respectively. The similarity graph $\mathcal{S}$ depends on the feature mappings $\phi(\cdot)$ and $\psi(\cdot)$, which for our method not only depend on the node of interest, but also on the rest of the input graph and the learnable parameters. The mappings for a node v thus become $\phi(\boldsymbol{x}_v, \mathcal{G}; \boldsymbol{\theta}_{\boldsymbol{\phi}})$ and $\psi(\boldsymbol{x}_v, \mathcal{G}; \boldsymbol{\theta}_{\boldsymbol{\psi}})$ and we will ease these notations to $\phi(\boldsymbol{x}_v)$ and $\psi(\boldsymbol{x}_v)$. The ability of our method to learn these feature mappings is an important aspect of our contribution, as a key motivation is that we need to learn new similarities for clustering heterophilous graphs.

HeNCler's loss function is comprised of three terms: the wKSVD-loss, a node-reconstruction loss, and an edge-reconstruction loss:

$$\mathcal{L}_{\mathrm{wKSVD}}(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\Sigma}, \boldsymbol{\theta}_{\phi}, \boldsymbol{\theta}_{\psi}) + \mathcal{L}_{\mathrm{NodeRec}}(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\theta}_{\phi}, \boldsymbol{\theta}_{\psi}, \boldsymbol{\theta}_{\mathrm{rec}}) + \mathcal{L}_{\mathrm{EdgeRec}}(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\theta}_{\phi}, \boldsymbol{\theta}_{\psi}),$$

where the trainable parameters of the model are in the multilayer perceptron (MLP) feature maps ($\boldsymbol{\theta}_\phi$ and $\boldsymbol{\theta}_\psi$), the MLP node decoder ($\boldsymbol{\theta}_{\text{rec}}$), the $\boldsymbol{U}$ and $\boldsymbol{V}$ projection matrices, and the singular values $\boldsymbol{\Sigma}$. All these parameters are trained end-to-end and we next explain the losses in more detail.

wKSVD-Loss. Instead of solving the SVD in the dual formulation, HeNCler leverages the primal formulation (1) of the wKSVD framework for greater computational efficiency. While equation (1) assumes that the feature maps $\phi(\cdot)$ and $\psi(\cdot)$ are fixed, HeNCler utilizes parametric functions $\phi(\cdot;\boldsymbol{\theta}_\phi)$ and $\psi(\cdot;\boldsymbol{\theta}_\psi)$, enabling it to learn new similarities between nodes. By incorporating regularization terms for these functions and defining the weighting scalars as $w_{1,v} = D_{1,vv}^{-1} = 1/\sum_u \phi(\boldsymbol{x}_v)^\top \psi(\boldsymbol{x}_u)$ and $w_{2,v} = D_{2,vv}^{-1} = 1/\sum_u \phi(\boldsymbol{x}_u)^\top \psi(\boldsymbol{x}_v)$, we derive the wKSVD-loss:

$$\mathcal{L}_{\text{wKSVD}} \triangleq -\sum_{v=1}^{|\mathcal{V}|} D_{1,vv}^{-1}\, \phi(\boldsymbol{x}_v)^\top \boldsymbol{U}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^\top \phi(\boldsymbol{x}_v) - \sum_{v=1}^{|\mathcal{V}|} D_{2,vv}^{-1}\, \psi(\boldsymbol{x}_v)^\top \boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{V}^\top \psi(\boldsymbol{x}_v)$$
$$+\text{Tr}(\boldsymbol{U}^\top \boldsymbol{V}) + \sum_{v=1}^{|\mathcal{V}|} \sqrt{D_{1,vv}^{-1}\, D_{2,vv}^{-1}}\, \phi(\boldsymbol{x}_v)^\top \psi(\boldsymbol{x}_v). \quad (7)$$

The primal formulation of HeNCler (7) consists of four terms with two distinct objectives. The first two terms promote the weighted variance of the learned node representations $\boldsymbol{e}$ and $\boldsymbol{r}$, encouraging informative embeddings. The third and fourth terms serve as regularizers that enforce asymmetry by penalizing the similarity between $\boldsymbol{U}$ and $\boldsymbol{V}$, and between $\phi(\boldsymbol{x}_v)$ and $\psi(\boldsymbol{x}_v)$, respectively.

For the two feature maps $\phi(\cdot)$ and $\psi(\cdot)$, we employ two MLPs: $\phi(\boldsymbol{x}_v, \mathcal{G}; \boldsymbol{\theta}_\phi) \equiv \text{MLP}_\phi(\boldsymbol{x}_v || \text{PE}_v; \boldsymbol{\theta}_\phi)$ and $\psi(\boldsymbol{x}_v, \mathcal{G}; \boldsymbol{\theta}_\psi) \equiv \text{MLP}_\psi(\boldsymbol{x}_v || \text{PE}_v; \boldsymbol{\theta}_\psi)$. We construct a random walks positional encoding (PE) [9] to embed the network's structure and concatenate this encoding with the node attributes. The MLPs have two linear layers with a LeakyReLU activation function in between, followed by a batch normalization layer. The singular values in $\boldsymbol{\Sigma}$ are jointly learned, constrained to lie between 0 and 1, with the additional condition that $\text{Tr}(\boldsymbol{\Sigma}^{-\frac{1}{2}}) = 1$.

Reconstruction Losses. Since the feature maps $\phi(\cdot)$ and $\psi(\cdot)$ need to be learned, an additional loss function beyond the above regularization term is required to effectively optimize the parameters of the MLPs. As the node clustering setting is completely unsupervised, we add a decoder network and a reconstruction loss. This technique has been proven to be effective for unsupervised learning in the RKM-framework [16], as well as for unsupervised node representation learning [21]. For heterophilous graphs, we argue that it is particularly important to also reconstruct node features and not only the graph structure.

For the node reconstruction, we first project the $\boldsymbol{e}$ and $\boldsymbol{r}$ variables back to feature space, concatenate these and then map to input space with another MLP. This MLP has also two layers and a leaky ReLU activation function. The hidden

layer size is set to the average of the latent dimension and input dimension. With the mean-squared-error as the associated loss, this gives:

$$\mathcal{L}_{\text{NodeRec}} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} ||\text{MLP}_{\text{rec}}(\boldsymbol{U}\boldsymbol{e}_v || \boldsymbol{V}\boldsymbol{r}_v; \boldsymbol{\theta}_{\text{rec}}) - \boldsymbol{x}_v||^2.$$

To reconstruct edges, we use a simple dot-product decoder $\sigma(\boldsymbol{e}_u^\top \boldsymbol{U}^\top \boldsymbol{V}\boldsymbol{r}_v)$ where σ is the sigmoid function. By using the $\boldsymbol{e}$ representation for source nodes and $\boldsymbol{r}$ for target nodes, this reconstruction is asymmetric and can reconstruct directed graphs. We use a binary cross-entropy loss:

$$\mathcal{L}_{\text{EdgeRec}} = \frac{1}{|\mathcal{U}|} \sum_{(u,v) \in \mathcal{U}} \text{BCE}(\sigma(\boldsymbol{e}_u^\top \boldsymbol{U}^\top \boldsymbol{V}\boldsymbol{r}_v), \mathcal{E}_{uv}),$$

where $\mathcal{U}$ is a node-tuple set, resampled every epoch, containing $2|\mathcal{V}|$ positive edges from $\mathcal{E}$ and $2|\mathcal{V}|$ negative edges from $\mathcal{E}^C$, and $\mathcal{E}_{uv} \in \{0, 1\}$ indicates whether an edge (u, v) exist: $(u, v) \in \mathcal{E}$.

Constraints and Cluster Assignments. The batch normalization in the MLPs keeps the wKSVD-loss bounded and the constraints on the singular values is enforced by a softmax function. Cluster assignments are obtained by KMeans clustering on the concatenation of learned $\boldsymbol{e}$ and $\boldsymbol{r}$ node representations.

HeNCler jointly learns the wKSVD projection matrices, $\boldsymbol{U}$ and $\boldsymbol{V}$, along with the feature map parameters, $\boldsymbol{\theta}_\phi$ and $\boldsymbol{\theta}_\psi$. The wKSVD loss improves the cluster-ability of the learned similarity graph, ensuring that $\boldsymbol{e}$ and $\boldsymbol{r}$ function as spectral biclustering embeddings. The two distinct feature maps enable asymmetric learning, effectively capturing potential asymmetric relationships in the data, while the reconstruction losses ensure robust and meaningful representation learning.

4 Experiments

Datasets. We assess the performance of HeNCler on heterophilous attributed graphs that are available in literature. We use Texas, Cornell, and Wisconsin [17], which are directed webpage networks where edges encode hyperlinks between pages. Next, we use Chameleon and Squirrel [20], which are undirected Wikipedia webpage networks where edges encode mutual links. We further evaluate our model on three undirected graphs: Roman-Empire, Minesweeper, and Tolokers [18]. These datasets represent, respectively, a graph-structured Wikipedia article, a grid graph inspired by the Minesweeper game, and a crowdsourcing interaction network. More details are provided in the supplementary materials.

Model Selection and Metrics. Model selection in this unsupervised setting is inherently challenging, as the most appropriate evaluation metric depends on the downstream task. Consequently, model selection is beyond the scope of this

paper. Instead, we evaluate our model in a task-agnostic manner and ensure a fair comparison with the baselines. We fix the hyperparameter configuration of the models across all datasets. We train for a fixed number of epochs and keep track of the evaluation metrics to report the best observed result. We repeat the training process 10 times and report average best results with standard deviations. We report the normalized mutual information (NMI) and pairwise F1-scores, based on the class labels.

Baselines and Hyperparameters. We compare our model against several methods, including a simple KMeans based on node attributes, adjacency partitioning-based approaches such as MinCutPool [4] and DMoN [25], as well as S^3GC [6], MUSE [28], and HoLe [11], which represent the current state-of-the-art in homophilous and heterophilous node clustering. For HeNCler, we fix the hyperparameters as follows: MLP hidden dimensions 256, output dimensions 128, latent dimension $s = 2 \times \#\text{classes}$, learning rate 0.01, and epochs 300. For the baselines, we used their code implementations and the default hyperparameter settings as proposed by the authors. The number of clusters to infer is set to the number of classes for all methods.

Experimental Results. Table 1 presents the experimental results for heterophilous graphs. HeNCler consistently demonstrates superior performance, significantly outperforming the baselines, especially on directed graphs. For undirected graphs, HeNCler also shows strong results, achieving the best performance in 5 out of 10 cases. These results highlight HeNCler's versatility and effectiveness in handling heterophilous graph structures.

Additional experiments are presented in the supplementary materials. We include an ablation study, some experiments on homophilous graphs, a visualization of the learned asymmetries, and a computational complexity analysis.

Discussion. A key motivation behind HeNCler is to learn a new graph representation where nodes belonging to the same cluster are positioned closer together, driven by the clustering objective. This results in spectral biclustering embeddings that exhibit improved cluster-ability. As HeNCler uses KMeans to obtain cluster assignments, the comparisons between HeNCler and KMeans demonstrate that our model enhances the cluster-ability of the node representations relative to the original input features.

The asymmetry in HeNCler eliminates the undirected constraints of traditional adjacency partitioning-based models, enabling superior performance on directed graphs. Furthermore, our ablation study in the supplementary materials shows that, while most of the performance on undirected graphs stems from the graph learning component, HeNCler is able to capture and learn additional meaningful asymmetric information. This capacity to extract valuable asymmetric insights from symmetric data is a common occurrence in KSVD frameworks [13,24]. Importantly, thanks to the added performance boost from asymmetry,

Table 1. Experimental results on heterophilous graphs. We report NMI and F1 scores for 10 runs (mean ± standard deviation), where higher values indicate better performance. The best results for each metric are highlighted in bold. OOM indicates an out-of-memory error on a 16GB GPU.

Dataset		Baselines						Ours
		KMeans	MinCutPool	DMoN	S^3GC	MUSE	HoLe	HeNCler
tex	NMI	4.97±1.00	11.60±2.19	9.06±2.11	11.56±1.46	39.23±4.91	7.51±0.37	**43.65**±2.52
	F1	59.27±0.83	55.26±0.56	47.76±4.79	43.69±2.74	65.96±3.52	35.30±0.44	**71.39**±2.16
corn	NMI	5.42±2.04	17.04±1.61	12.49±2.51	14.48±1.79	38.99±2.73	14.23±0.17	**41.52**±4.35
	F1	52.97±0.24	51.21±5.06	43.83±6.23	33.13±0.83	60.58±3.61	34.64±0.56	**63.40**±3.67
wis	NMI	6.84±4.39	13.38±2.36	12.56±1.23	13.07±0.61	39.71±2.22	11.89±0.42	**47.13**±1.76
	F1	56.16±0.58	55.63±2.96	45.72±7.85	31.71±2.25	58.94±3.09	37.05±0.02	**68.30**±2.17
cha	NMI	0.44±0.11	11.88±1.99	12.87±1.86	15.83±0.26	23.06±0.28	8.76±0.15	**23.89**±0.84
	F1	**53.23**±0.07	50.40±5.65	45.05±4.30	36.51±0.24	52.10±0.48	30.61±0.09	44.14±1.83
squi	NMI	1.40±2.12	6.35±0.32	3.08±0.38	3.83±0.11	8.30±0.23	4.99±0.09	**9.67**±0.13
	F1	54.05±2.72	**55.26**±0.57	49.21±2.74	35.08±0.18	50.07±5.99	28.71±0.36	36.51±2.39
rom	NMI	35.20±1.79	9.97±2.02	13.14±0.53	14.48±0.21	**40.50**±0.73	OOM	36.99±0.61
	F1	37.17±2.12	**42.19**±0.26	22.69±3.91	17.76±0.53	38.34±0.35		35.43±1.07
mine	NMI	0.02±0.02	6.16±2.17	**6.87**±2.91	6.53±0.17	0.06±0.01	6.46±0.07	0.06±0.00
	F1	73.63±3.58	71.76±8.86	70.42±9.47	48.78±0.63	75.77±2.24	63.01±0.13	**76.48**±1.56
tol	NMI	3.04±2.83	6.68±0.98	6.69±0.20	5.99±0.05	6.67±0.55	5.14±0.06	**6.73**±0.59
	F1	65.56±10.49	72.10±10.38	67.87±4.74	59.17±0.27	73.56±1.94	66.35±1.12	**73.66**±2.10

on top of the benefits from similarity learning, HeNCler outperforms state-of-the-art models, even when applied to undirected graphs.

While our experiments focus on benchmark heterophilous graphs, the design of HeNCler makes it a promising candidate for domains characterized by relational asymmetry and weak homophily. For example, in community detection within directed social or communication networks, HeNCler's ability to model asymmetric relationships could help capture influence flows that do not align with node similarity. In recommender systems, HeNCler could be applied to bipartite user-item graphs by using separate feature maps for each node type. This would enable the construction of a learned bipartite similarity graph that aligns user preferences with item characteristics in an unsupervised manner. Similarly, in biological networks such as gene regulatory systems, the model's capacity to handle directed, heterophilous interactions could prove valuable.

5 Conclusion and Future Work

We tackle three limitations of current node clustering algorithms that prevent these methods from effectively clustering nodes in heterophilous graphs: they assume homophily in their loss, they are only defined for undirected graphs and/or they lack a specific focus on clustering.

To this end, we introduce a weighted kernel SVD framework and harness its primal-dual equivalences. HeNCler relies on the dual interpretation for its theoretical motivation, while it benefits from the computational advantages of its implementation in the primal. It learns new similarities, which are asymmetric where necessary, and node embeddings resulting from the spectral biclustering interpretation of these learned similarities. As empirical evidence shows, our approach effectively eliminates the aforementioned limitations, significantly outperforming current state-of-the-art alternatives.

Future work could explore integrating contrastive learning into HeNCler, potentially combining the strengths of both paradigms. Another direction is to investigate cluster assignments in a graph pooling setting (i.e., differentiable graph coarsening), enabling end-to-end training for downstream graph-level tasks. Additionally, applying HeNCler to real-world domains such as recommender systems, biological networks, or directed social graphs–where asymmetric and heterophilous structures naturally occur–would be an important step toward validating its broader applicability.

Acknowledgments. ESAT-STADIUS has received funding from the European Research Council under the European Union's Horizon 2020 research and innovation program/ERC Advanced Grant E-DUALITY (787960). This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information. ESAT-STADIUS received funding from the Flemish Government (AI Research Program); iBOF/23/064; KU Leuven C1 project C14/24/103. Johan Suykens and Sonny Achten are also affiliated to Leuven.AI - KU Leuven institute for AI, B-3000, Leuven, Belgium.

LIONS-EPFL was supported by Hasler Foundation Program: Hasler Responsible AI (project number 21043), by the ARO under Grant Number W911NF-24-1-0048, and by the Swiss National Science Foundation (SNSF) under grant number 200021_205011.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Achten, S., Tonin, F., Patrinos, P., Suykens, J.A.K: Unsupervised neighborhood propagation kernel layers for semi-supervised node classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, no. 10, pp. 10766–10774 (2024)
2. Alzate, C., Suykens, J.A.K.: Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA. IEEE Trans. Pattern Anal. Mach. Intell. **32**(2), 335–347 (2010)
3. Bi, W., Du, L., Fu, Q., Wang, Y., Han, S., Zhang, D.: Make heterophilic graphs better fit GNN: a graph rewiring approach. IEEE Trans. Knowl. Data Eng. **36**(12), 8744–8757 (2024)
4. Bianchi, F.M., Grattarola, D., Alippi, C.: Spectral clustering with graph neural networks for graph pooling. In: International Conference on Machine Learning (2020)

5. Chen, J., Zhu, G., Qi, Y., Yuan, C., Huang, Y.: Towards self-supervised learning on graphs with heterophily. In: ACM International Conference on Information & Knowledge Management, pp. 201–211 (2022)
6. Devvrit, F., Sinha, A., Dhillon, I.S, Jain, P.: S3GC: scalable self-supervised graph clustering. In: Advances in Neural Information Processing Systems (2022)
7. Dhillon, I.S.: Co-clustering documents and words using bipartite spectral graph partitioning. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2001)
8. Dwivedi, V.P., Bresson, X.: A generalization of transformer networks to graphs. In: Methods and Applications, AAAI Workshop on Deep Learning on Graphs (2021)
9. Dwivedi, V.P., Luu, A.T., Laurent, T., Bengio, Y., Bresson, X.: Graph neural networks with learnable structural and positional representations. In: International Conference on Learning Representations (2022)
10. Gong, Z., et al.: Beyond homophily: robust graph anomaly detection via neural sparsification. In: International Joint Conference on Artificial Intelligence (2023)
11. Gu, M., et al.: Homophily-enhanced structure learning for graph clustering. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, pp. 577–586 (2023)
12. Hassani, K., Ahmadi, A.H.K.: Contrastive multi-view representation learning on graphs. In: International Conference on Machine Learning (2020)
13. He, M., He, F., Shi, L., Huang, X., Suykens, J.A.K.: Learning with asymmetric kernels: least squares and feature interpretation. IEEE Trans. Pattern Anal. Mach. Intell. **45**(8), 10044–10054 (2023)
14. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (2017)
15. Pan, S., Hu, R., Fung, S.F., Long, G., Jiang, J., Zhang, C.: Learning graph embedding with adversarial training methods. IEEE Trans. Cybern. **50**(6), 2475–2487 (2020)
16. Pandey, A., Fanuel, M., Schreurs, J., Suykens, J.A.K.: Disentangled representation learning and generation with manifold optimization. Neural Comput. **34**(10), 2009–2036 (2022)
17. Pei, H., Wei, B., Chang, K.C.C., Lei, Y., Yang, B.: Geom-GCN: geometric graph convolutional networks. In: International Conference on Learning Representations (2020)
18. Platonov, O., Kuznedelev, D., Diskin, M., Babenko, A., Prokhorenkova, L.: A critical look at the evaluation of GNNs under heterophily: are we really making progress? In: International Conference on Learning Representations (2023)
19. Rockafellar, R.T.: Conjugate Duality and Optimization. SIAM (1974)
20. Rozemberczki, B., Allen, C., Sarkar, R.: Multi-scale attributed node embedding. J. Complex Netw. **9**(1), 1–22 (2021)
21. Sun, D., Li, D., Ding, Z., Zhang, X., Tang, J.: Dual-decoder graph autoencoder for unsupervised graph representation learning. Knowl.-Based Syst. **234**, 107564 (2021)
22. Suykens, J.A.K.: SVD revisited: a new variational principle, compatible feature maps and nonlinear extensions. Appl. Comput. Harmon. Anal. **40**(3), 600–609 (2016)
23. Suykens, J.A.K.: Deep restricted kernel machines using conjugate feature duality. Neural Comput. **29**(8), 2123–2163 (2017)
24. Tao, Q., Tonin, F., Lambert, A., Chen, Y., Patrinos, P., Suykens, J.A.K.: Learning in feature spaces via coupled covariances: asymmetric kernel SVD and Nyström method. In: International Conference on Machine Learning (2024)

25. Tsitsulin, A., Palowitch, J., Perozzi, B., Müller, E.: Graph clustering with graph neural networks. J. Mach. Learn. Res. **24**(127), 1–21 (2023)
26. Xiao, T., Chen, Z., Guo, Z., Zhuang, Z., Wang, S.: Decoupled self-supervised learning for graphs. In: Advances in Neural Information Processing Systems (2022)
27. You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., Shen, Y.: Graph contrastive learning with augmentations. In: Advances in Neural Information Processing Systems (2020)
28. Yuan, M., Chen, M., Li, X.: MUSE: multi-view contrastive learning for heterophilic graphs. In: ACM International Conference on Information and Knowledge Management, pp. 3094–3103 (2023)
29. Zheng, X., et al.: Graph neural networks for graphs with heterophily: a survey. arXiv:2202.07082 (2024)