# HeNCler: Node Clustering in Heterophilous Graphs via Learned Asymmetric Similarity

**Anonymous authors**
Paper under double-blind review

## 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**uste**r**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.

## 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 (Zheng et al., 2024; Wu et al., 2021). The homophily assumption holds true in contexts such as social networks and citation graphs, where models like GCN (Kipf & Welling, 2017), GIN (Xu et al., 2019), and GraphSAGE (Hamilton et al., 2017) excel at tasks like node classification and graph prediction.

However, in heterophilous datasets, such as web page and transaction networks, edges often link nodes with differing labels. Models like GAT (Veličković et al., 2018) and various graph transformers (Ying et al., 2022; Dwivedi & Bresson, 2021) have demonstrated improved performance on these datasets. Their attention mechanisms learning edge importance provide a straightforward way to reduce the reliance on homophily for supervised tasks.

Our work specifically addresses unsupervised attributed node clustering tasks. Such tasks necessitate entirely unsupervised or self-supervised learning approaches. For instance, auto-encoder type models (Park et al., 2019; Pan et al., 2020) 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 (Hassani & Ahmadi, 2020; You et al., 2020) enhance node representation learning in homophilous settings only and lack a specific clustering objective. At the same time, several self-supervised methods have been developed to handle heterophilous graphs (Chen et al., 2022; Xiao et al., 2022; Yuan et al., 2023). For example, MUSE (Yuan et al., 2023) extracts semantic and contextual views for contrastive learning. However, these methods are designed for the general node representation learning task and lack a clustering objective.

In contrast, $S^3$GC (Devvrit et al., 2022) employs a self-supervised approach specifically designed for clustering. It however assumes homophily by leveraging random walk co-occurrences to infer proximity-based similarities. MinCutPool (Bianchi et al., 2020) and DMoN (Tsitsulin et al., 2023) introduce unsupervised losses linked to graph structure, with strong theoretical ties to spectral clustering and graph modularity, respectively. These methods are suited for undirected graphs only, and moreover rely on partitioning the adjacency matrix where effective clustering correlates with high intra-cluster and low inter-cluster similarity—a premise often invalid in heterophilous graphs.

This paper introduces HeNCler, a novel approach for node clustering in heterophilous graphs, illustrated in Figure 1. Existing works overlook the asymmetric relationships in heterophilous

Table 1: Qualitative comparison of HeNCler with several baselines. In the table, $|\mathcal{V}|$, $|\mathcal{B}|$, and $|\mathcal{E}|$ denote the total number of nodes, the mini-batch size, and the number of edges respectively.

| | BASELINES | | | | OURS |
|---|---|---|---|---|---|
| | MINCUTP. | DMON | $S^3$GC | MUSE | HENCLER |
| CAN HANDLE HETEROPHILY | ✗ | ✗ | ✗ | ✓ | ✓ |
| DIRECTED GRAPHS | ✗ | ✗ | ✓ | ✓ | ✓ |
| HAS CLUSTERING OBJECTIVE | ✓ | ✓ | ✓ | ✗ | ✓ |
| SPACE COMPLEXITY | $\mathcal{O}(\lvert\mathcal{V}\rvert^2)$ | $\mathcal{O}(\lvert\mathcal{V}\rvert + \lvert\mathcal{E}\rvert)$ | $\mathcal{O}(\lvert\mathcal{B}\rvert)$ | $\mathcal{O}(\lvert\mathcal{V}\rvert + \lvert\mathcal{E}\rvert)$ | $\mathcal{O}(\lvert\mathcal{B}\rvert)$ |
| TIME COMPLEXITY | $\mathcal{O}(\lvert\mathcal{V}\rvert + \lvert\mathcal{E}\rvert)$ | $\mathcal{O}(\lvert\mathcal{V}\rvert + \lvert\mathcal{E}\rvert)$ | $\mathcal{O}(\lvert\mathcal{V}\rvert)$ | $\mathcal{O}(\lvert\mathcal{V}\rvert + \lvert\mathcal{E}\rvert)$ | $\mathcal{O}(\lvert\mathcal{V}\rvert)$ |

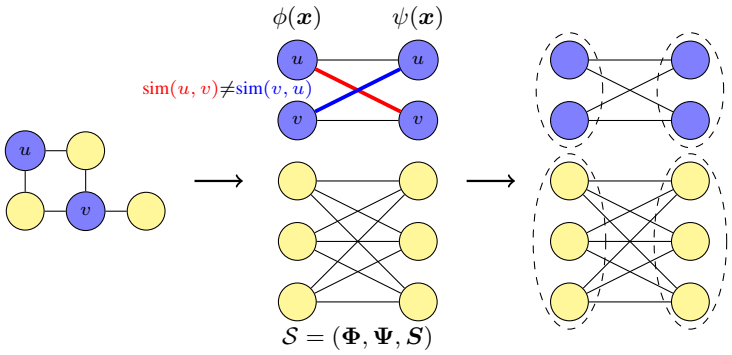


Figure 1: **HeNCler Overview**. Starting from a heterophilous graph, where nodes with the same label are not close to each other (left), HeNCler learns two sets of node representations, $\{\phi(\boldsymbol{x}_v)\}_{v\in\mathcal{V}}$ and $\{\psi(\boldsymbol{x}_v)\}_{v\in\mathcal{V}}$, forming a bipartite graph $\mathcal{S}$ (middle), where the similarity between nodes is defined as $S_{uv} = \text{sim}(u,v) = \phi(\boldsymbol{x}_u)^\top \psi(\boldsymbol{x}_v)$. Due to the clustering objective, nodes that should belong to the same cluster are positioned closer together in the learned graph. These clusters are then identified using spectral biclustering through wKSVD (right).

graphs, as shown in Table 1. 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.

**Contributions:** 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 is available in supplementary materials.

## 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 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 (Suykens, 2016) sets up a primal-dual framework, based on Lagrange duality, that formulates a variational principle in the primal formulation that corresponds to the matrix singular value decomposition (SVD) in the dual for linear feature maps. By employing non-linear feature mappings or asymmetric kernel functions, this framework allows for non-linear extensions of the SVD problem. The KSVD framework 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 (He et al., 2023). Interestingly, KSVD often outperforms the similar though symmetric kernel principal component analysis model on tasks where the asymmetry is not immediately apparent (Tao et al., 2024). A different connection is shown in Primal-Attention (Chen et al., 2023), where the authors demonstrate the relation between canonical self-attention, which is asymmetric, and KSVD. They show how to gain computational efficiency by considering a primal equivalent of the attention mechanism.

**Spectral clustering** generalizations have been proposed in many settings. Spectral graph biclustering (Dhillon, 2001) 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) (Alzate & Suykens, 2010) 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 will address.

**Restricted kernel machines** (RKM) (Suykens, 2017) possess primal and dual model formulations, based on the concept of conjugate feature duality. It is an energy-based framework for (deep) kernel machines, that shows relations with least-squares support vector machines (Suykens et al., 2002) and restricted Boltzmann machines (Salakhutdinov, 2015). The RKM framework encompasses many model classes, including classification, regression, kernel principal component analysis and KSVD, and allows for deep kernel learning (Tonin et al., 2021) and deep kernel learning on graphs (Achten et al., 2024). One possibility to represent the feature maps in RKMs is by means of deep neural networks, e.g., for unsupervised representation learning (Pandey et al., 2021; 2022). RKM models can work in either primal or dual setting, and with decomposition or gradient based algorithms (Achten et al., 2023).

**Homophilous node clustering** methods like MinCutPool (Bianchi et al., 2020) and DMoN (Tsitsulin et al., 2023) introduce unsupervised loss functions within a graph neural network framework. MinCutPool employs a relaxed version of the minimal cut loss applied to the adjacency matrix, while DMoN optimizes the modularity score of clustering labels with respect to the adjacency structure. Both of these methods rely on partitioning the adjacency matrix and inherently assume homophily. Additionally, due to their theoretical underpinnings, these losses are only applicable to undirected graphs. Beyond these adjacency partitioning-based approaches, self-supervised or contrastive methods have also been proposed (You et al., 2020; Hassani & Ahmadi, 2020; Devvrit et al., 2022). These methods typically use graph proximity as their supervision signal, which similarly assumes homophily. For example, $S^3$GC (Devvrit et al., 2022) employs a self-supervised loss based on random walk co-occurrences.

**Heterophilous node clustering** methods typically rely on self-supervised or contrastive techniques. Gong et al. (Gong et al., 2023) propose Sparse Graph Anomaly Detection (SparseGAD), a method that sparsifies graph structures to effectively reduce noise from irrelevant edges and enhance the detection of closely related nodes. This technique reveals underlying node dependencies, accommodating both homophilous and heterophilous relationships. Similarly, HGRL (Chen et al., 2022) employs

self-supervised learning on heterophilous graphs by utilizing graph augmentation techniques to capture global and higher-order structural information. MUSE (Yuan et al., 2023), on the other hand, constructs semantic and contextual views to capture both node-level and neighborhood information for contrastive learning, subsequently integrating these multi-view representations through a fusion controller.

While adjacency partitioning-based methods have demonstrated both theoretical and empirical success for homophilous graphs, they have not been effectively extended to heterophilous graph learning. On the other hand, self-supervised clustering approaches, though promising, often lack a clear clustering interpretation. 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 (Pandey et al., 2022). To capture long-range relational dependencies in heterophilous graphs, we utilize a KSVD loss, where a double feature mapping yields a learned asymmetric similarity matrix. To further enhance the cluster-ability of this matrix, we extend the loss function to a weighted KSVD (wKSVD) loss, which not only boosts clustering performance but also offers a spectral graph biclustering interpretation. We next introduce a general wKSVD framework, after which we introduce our HeNCler model that operates in the primal setting while jointly learning the feature mappings end-to-end.

## 3.1 KERNEL SPECTRAL BICLUSTERING WITH ASYMMETRIC SIMILARITIES

Consider a dataset with two, possibly different, 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:

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

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 Appendix A. 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 $S_{ij} = \phi(\boldsymbol{x}_i)^\top\psi(\boldsymbol{z}_j)$, $\boldsymbol{\Phi} = [\phi(\boldsymbol{x}_1)\dots\phi(\boldsymbol{x}_n)]^\top$, $\boldsymbol{\Psi} = [\psi(\boldsymbol{z}_1)\dots\psi(\boldsymbol{z}_m)]^\top$, and where $\boldsymbol{H}_e = [\boldsymbol{h}_{\boldsymbol{e}_1}...\boldsymbol{h}_{\boldsymbol{e}_n}]^\top$, and $\boldsymbol{H}_r = [\boldsymbol{h}_{\boldsymbol{r}_1}...\boldsymbol{h}_{\boldsymbol{r}_m}]^\top$ 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 case of Fenchel-Young inequality (Rockafellar, 1974):

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

$\forall \boldsymbol{e}_i, \boldsymbol{r}_j, \boldsymbol{h}_{\boldsymbol{e}_i}, \boldsymbol{h}_{\boldsymbol{r}_j} \in \mathbb{R}^s$, $\forall w_{1,i}, w_{2,j} \in \mathbb{R}_{>0}$, $\forall \boldsymbol{\Sigma} \in \mathbb{R}^{s\times s}_{\succ 0}$. The above inequalities can be verified by writing it in quadratic form: $\frac{1}{2}\begin{bmatrix}\boldsymbol{e}_i^\top & \boldsymbol{h}_{\boldsymbol{e}_i}^\top\end{bmatrix}\begin{bmatrix} w_{1,i}\boldsymbol{\Sigma}^{-1} & -\sqrt{w_{1,i}}\,\boldsymbol{I}_s \\ -\sqrt{w_{1,i}}\,\boldsymbol{I}_s & \boldsymbol{\Sigma}\end{bmatrix}\begin{bmatrix}\boldsymbol{e}_i \\ \boldsymbol{h}_{\boldsymbol{e}_i}\end{bmatrix} \geq 0, \quad \forall i$, with $\boldsymbol{I}_s$ the $s$-dimensional identity matrix, which follows immediately from the Schur complement form:

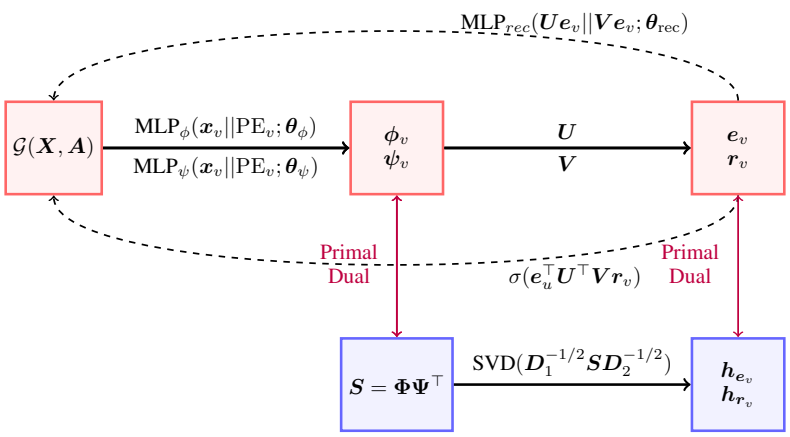


Figure 2: **The HeNCler model**. HeNCler operates in the primal setting (top of the figure in red) and uses a double multilayer perceptron (MLP) to map node representations to a feature space. The obtained representations $\boldsymbol{\phi}_v$ and $\boldsymbol{\psi}_v$ are then projected to latent representations $\boldsymbol{e}_v$ 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.

for a matrix $\boldsymbol{Q} = \left[\begin{smallmatrix} Q_1 & Q_2 \\ Q_2^\top & Q_3 \end{smallmatrix}\right]$, one has $\boldsymbol{Q} \succeq 0$ if and only if $\boldsymbol{Q}_1 \succ 0$ and the Schur complement $\boldsymbol{Q}_3 - \boldsymbol{Q}_2^\top \boldsymbol{Q}_1^{-1} \boldsymbol{Q}_2 \succeq 0$ (Boyd & Vandenberghe, 2004).

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 $\bar{J} \geq J$:

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

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

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

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

$$\begin{bmatrix} \boldsymbol{0} & \boldsymbol{W}_1^{1/2}\boldsymbol{S}\boldsymbol{W}_2^{1/2} \\ \boldsymbol{W}_2^{1/2}\boldsymbol{S}^\top\boldsymbol{W}_1^{1/2} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{H}_{\boldsymbol{e}} \\ \boldsymbol{H}_{\boldsymbol{r}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_{\boldsymbol{e}} \\ \boldsymbol{H}_{\boldsymbol{r}} \end{bmatrix} \boldsymbol{\Sigma}, \quad (6)$$

where $\boldsymbol{0}$ is an all-zeros matrix. Note that, by Lanczos' Theorem (Lanczos, 1958), the above eigenvalue problem is equivalent with (2), and that 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 2.** *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.

### 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 to different mappings that operate on the nodes given the entire graph $\mathcal{G} = (\mathcal{X}, \mathcal{A})$. Our method is visualized in Figure 2, where red indicates the primal setting of the framework and blue the dual.

In the preceding subsection, we showed that problem (1) has an equivalent dual problem corresponding to the graph bipartitioning problem, 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. This similarity graph $\mathcal{S}$ depends on the feature mappings $\phi(\cdot)$ and $\psi(\cdot)$, which for our method does not only depend on the node of interest, but also on the rest of the input graph and the learnable parameters. The mappings for node $v$ thus become $\phi(\boldsymbol{x}_v, \mathcal{G}; \boldsymbol{\theta_\phi})$ and $\psi(\boldsymbol{x}_v, \mathcal{G}; \boldsymbol{\theta_\psi})$ 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 behind our model is that we need to learn new similarities for clustering heterophilous graphs. The loss function is comprised of three terms: the wKSVD-loss, a node-reconstruction loss, and an edge-reconstruction loss:

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

where the trainable parameters of the model are in the the multilayer perceptron (MLP) feature maps ($\boldsymbol{\theta}_\phi$ and $\boldsymbol{\theta}_\psi$), the MLP node decoder ($\boldsymbol{\theta}_{\text{rec}}$), in the $\boldsymbol{U}$ and $\boldsymbol{V}$ projection matrices, and in 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:

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

The primal formulation of HeNCler (7) can be understood as follows: The first two terms aim to maximize the weighted variance of the learned node representations $\boldsymbol{e}$ and $\boldsymbol{r}$. The third and fourth terms act as regularizers, encouraging 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) (Dwivedi et al., 2022) 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 (Pandey et al., 2022), as well as for unsupervised node representation learning (Sun et al., 2021). 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:

$$\mathcal{L}_{\text{NodeRec}} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} ||\text{MLP}_{\text{rec}}(\boldsymbol{U}\boldsymbol{e}_v || \boldsymbol{V}\boldsymbol{r}_v; \boldsymbol{\theta}_{\text{rec}}) - \boldsymbol{x}_v||^2. \tag{8}$$

To reconstruct edges, we use a simple dot-product decoder $\sigma(\boldsymbol{e}_u^\top \boldsymbol{U}^\top \boldsymbol{V}\boldsymbol{r}_v)$ where $\sigma$ 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:

$$\mathcal{L}_{\text{EdgeRec}} = \frac{1}{|\mathcal{U}|} \sum_{(u,v) \in \mathcal{U}} \text{BCE}(\sigma(\boldsymbol{e}_u^\top \boldsymbol{U}^\top \boldsymbol{V}\boldsymbol{r}_v), \mathcal{E}_{uv}), \tag{9}$$

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: $(u, v) \in \mathcal{E}$.

**Optimizer, constraints, and cluster assignment** We use Adam (Kingma & Ba, 2015) for the training of all parameters. The batch normalization in the MLP's keeps the wKSVD-loss bounded and the constraints on the singular values is enforced with 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.

Table 2: Dataset statistics of the employed heterophilous graphs.

| Dataset | short | # Nodes | # Edges | # Classes | Directed | $\mathcal{H}(\mathcal{G})$ |
|---|---|---|---|---|---|---|
| Texas | tex | 183 | 325 | 5 | ✓ | 0.000 |
| Cornell | corn | 183 | 298 | 5 | ✓ | 0.150 |
| Wisconsin | wis | 251 | 515 | 5 | ✓ | 0.084 |
| Chameleon | cha | 2,277 | 31,371 | 5 | ✗ | 0.042 |
| Squirrel | squi | 5,201 | 198,353 | 5 | ✗ | 0.031 |
| Roman-empire | rom | 22,662 | 32,927 | 18 | ✗ | 0.021 |
| Minesweeper | mine | 10,000 | 39,402 | 2 | ✗ | 0.009 |
| Tolokers | tol | 11,758 | 519,000 | 2 | ✗ | 0.180 |

## 4 EXPERIMENTS

**Datasets** We assess the performance of HeNCler on heterophilous attributed graphs that are available in literature. we use Texas, Cornell, and Wisconsin (Pei et al., 2020)[1] , which are directed webpage networks where edges encode hyperlinks between pages. Next, we use Chameleon and Squirrel (Rozemberczki et al., 2021), which are undirected Wikipedia webpage networks where edges encode mutual links. We further assess our model on the undirected graphs: Roman-empire, Minesweeper, and Tolokers (Platonov et al., 2023), which are a graph representation of a Wikipedia article, a grid graph based on the minesweeper game, and a crowd-sourcing network respectively. We include

[1] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb

Table 3: 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.

| | | Baselines | | | | | Ours |
|---|---|---|---|---|---|---|---|
| Dataset | | KMeans | MinCutPool | DMoN | $S^3$GC | MUSE | HeNCler |
| tex | NMI | 4.97±1.00 | 11.60±2.19 | 9.06±2.11 | 11.56±1.46 | 39.23±4.91 | **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 | **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 | **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 | **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 | **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 | **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 | **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 | 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 | **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 | 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 | 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 | 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 | **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 | **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 | **73.66**±2.10 |

experimental results for additional homophilous datasets in Appendix B. The dataset statistics can be consulted in Table 2, where the class insensitive edge homophily ratio $\mathcal{H}(\mathcal{G})$ (Lim et al., 2021) is a homophily measure.

**Model selection and metrics** Model selection in this unsupervised setting is non-trivial, and the best metric depends on the task at hand. Therefore, this is not the scope of this paper and we assess our model agnostically to the model selection, and fairly w.r.t. to 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 (Bianchi et al., 2020) and DMoN (Tsitsulin et al., 2023), as well as $S^3$GC (Devvrit et al., 2022) and MUSE (Yuan et al., 2023), which represent the current state-of-the-art in homophilous and heterophilous node clustering, respectively. For HeNCler, we fix the hyperparameters to: 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 cfr. Table 2 for all methods. The experiments are run on a Nvidia V100 GPU.

**Experimental results** Table 3 presents the experimental results for heterophilous graphs. HeNCler consistently demonstrates superior performance, significantly outperforming KMeans, MinCutPool, DMoN, $S^3$GC, and MUSE, especially on the directed graphs. For undirected graphs, HeNCler also shows strong results, achieving the best performance in 5 out of 10 cases, compared to KMeans (1/10), MinCutPool (2/10), DMoN (1/10), $S^3$GC (0/10), and MUSE (1/10). These results highlight HeNCler's versatility and effectiveness in handling heterophilous graph structures.

**Ablation studies** We conduct several ablation studies, presented in Table 4. The 'Undirected' variant refers to a simplified, symmetric version of the model that uses a single MLP for both the $\phi(\cdot)$ and $\psi(\cdot)$ mappings, i.e., $\phi(\cdot) \equiv \psi(\cdot)$. In this version, the model loses its asymmetry. The

Table 4: Ablation study results. We report mean NMI and F1 scores for 10 runs (higher is better) for different model configurations. Best results are highlighted in bold.

| | tex | | corn | | cha | | rom | | tol | |
|---|---|---|---|---|---|---|---|---|---|---|
| Metric | NMI | F1 | NMI | F1 | NMI | F1 | NMI | F1 | NMI | F1 |
| Undirected | 27.58 | 65.20 | 18.12 | 53.69 | 19.91 | 44.08 | 33.17 | 33.57 | 6.33 | 73.89 |
| Reconstr only | 29.54 | 66.64 | 27.76 | 54.70 | 22.02 | 43.42 | **40.05** | 35.16 | 6.18 | 68.60 |
| wKSVD only | 31.64 | 62.83 | 20.63 | 47.12 | 22.60 | 42.98 | 35.99 | 35.30 | 4.42 | 68.45 |
| HeNCler | **43.65** | **71.39** | **41.52** | **63.40** | **23.89** | **44.14** | 36.99 | **35.43** | **6.73** | **73.66** |

'wKSVD only' and 'Reconstr only' variations reflect models that incorporate only the wKSVD loss ($\mathcal{L}_{\text{wKSVD}}$) and the reconstruction losses ($\mathcal{L}_{\text{NodeRec}} + \mathcal{L}_{\text{EdgeRec}}$), respectively. Interestingly, as shown in Table 4, even for undirected graphs, introducing asymmetry in HeNCler enhances clustering performance. Furthermore, all loss components are shown to contribute positively to HeNCler's overall performance. For a comprehensive analysis, including results across all datasets and standard deviations, we refer the reader to Table 7 in Appendix B.

## 5 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. Note that HeNCler uses KMeans to obtain cluster assignments. Therefore, the comparisons between HeNCler and KMeans, as shown in Tables 3 and 6 , 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, as shown in Table 3. Furthermore, our ablation study in Table 4 shows that, while most of the performance on undirected graphs stem from the graph learning component, HeNCler is additionally able to capture and learn meaningful asymmetric information. This capacity to extract valuable asymmetric insights from symmetric data is a common occurrence in KSVD frameworks (He et al., 2023; Tao et al., 2024). Importantly, thanks to the added performance boost from asymmetry, on top of the benefits from similarity learning, HeNCler outperforms state-of-the-art models, even when applied to undirected graphs.

We visualize the learned similarity matrix $\boldsymbol{S} = \boldsymbol{\Phi}\boldsymbol{\Psi}^\top$ for two datasets in Figure 3. These matrices are generally asymmetric, with the asymmetry particularly pronounced in the directed graph of the Wisconsin dataset. In contrast, the Roman-Empire dataset, which is represented by an undirected graph, exhibits less asymmetry in the learned similarity matrix. This demonstrates the adaptability of HeNCler to handle both directed and undirected graphs. Further, given the observable block structures, the learned similarities are meaningful w.r.t. to the ground truth node labels. Note however that our model operates in the primal setting and directly projects the learned mappings $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ to their final embeddings $\boldsymbol{e}$ and $\boldsymbol{r}$ using $\boldsymbol{U}$ and $\boldsymbol{V}$ respectively, avoiding quadratic space complexity and cubic time complexity of the SVD. This is the motivation of employing a kernel based method, and exploiting the primal-dual framework that comes with it. In fact, the matrices in Figure 3 are only constructed for the sake of this visualization.

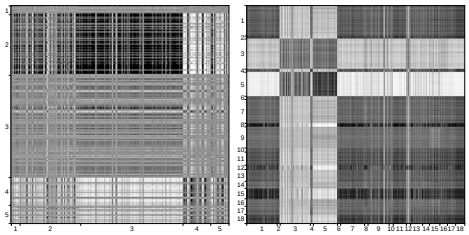

Figure 3: The learned matrix $\boldsymbol{S} = \boldsymbol{\Phi}\boldsymbol{\Psi}^\top$ for the Wisconsin (left) and Roman-empire (right) dataset. Rows and columns are grouped according to ground-truth node labels.

**Computational complexity** The space and time complexity of the current implementation of HeNCler are both linear w.r.t. the number of nodes $\mathcal{O}(|\mathcal{V}|)$. Whereas MinCutPool and DMoN need

all the node attributes in memory to calculate the loss w.r.t. the full adjacency matrix, HeNCler is easily adaptable to work with minibatches which reduces space complexity to the minibatch size $\mathcal{O}(|\mathcal{B}|)$. Although HeNCler relies on edge reconstruction, the edge sampling avoids quadratic complexity w.r.t. number of nodes, and is specifically designed to scale with the number of nodes, rather than the number of edges. Assuming the graphs are sparse, we add an overview of space and time complexity w.r.t. the number of nodes and edges for all methods in Table 1. A detailed table with measured computation times is provided in Appendix C.

## 6 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. In an end-to-end fashion, 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.

HeNCler is the first heterophilous node clustering model that does not rely on contrastive learning techniques. Future research could explore the integration of contrastive learning into HeNCler, potentially combining the strengths of both approaches. Another next step can be to investigate how to do the cluster assignments in a graph pooling setting (i.e., differentiable graph coarsening), to enable end-to-end learning for downstream graph prediction tasks.

## References

Sonny Achten, Arun Pandey, Hannes De Meulemeester, Bart De Moor, and Johan A. K. Suykens. Duality in Multi-View Restricted Kernel Machines. ICML Workshop on Duality for Modern Machine Learning, 2023. arXiv:2305.17251 [cs].

Sonny Achten, Francesco Tonin, Panagiotis Patrinos, and Johan A.K. Suykens. Unsupervised Neighborhood Propagation Kernel Layers for Semi-supervised Node Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(10):10766–10774, Mar. 2024.

Carlos Alzate and Johan A. K. Suykens. Multiway Spectral Clustering with Out-of-Sample Extensions through Weighted Kernel PCA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2):335–347, 2010.

Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. Spectral Clustering with Graph Neural Networks for Graph Pooling. In *International Conference on Machine Learning*, 2020.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787.

Jingfan Chen, Guanghui Zhu, Yifan Qi, Chunfeng Yuan, and Yihua Huang. Towards self-supervised learning on graphs with heterophily. In *ACM International Conference on Information & Knowledge Management*, pp. 201–211, 2022.

Yingyi Chen, Qinghua Tao, Francesco Tonin, and Johan Suykens. Primal-Attention: Self-attention through Asymmetric Kernel SVD in Primal Representation. In *Advances in Neural Information Processing Systems*, 2023.

Fnu Devvrit, Aditya Sinha, Inderjit Dhillon, and Prateek Jain. S3GC: Scalable Self-Supervised Graph Clustering. In *Advances in Neural Information Processing Systems*, 2022.

Inderjit S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001.

Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. AAAI Workshop on Deep Learning on Graphs: Methods and Applications, 2021. arXiv:2012.09699 [cs].

Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. In *International Conference on Learning Representations*, 2022.

Zheng Gong, Guifeng Wang, Ying Sun, Qi Liu, Yuting Ning, Hui Xiong, and Jingyu Peng. Beyond Homophily: Robust Graph Anomaly Detection via Neural Sparsification. In *International Joint Conference on Artificial Intelligence*, 2023.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*, 2017.

Kaveh Hassani and Amir Hosein Khas Ahmadi. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*, 2020.

Mingzhen He, Fan He, Lei Shi, Xiaolin Huang, and Johan A. K. Suykens. Learning With Asymmetric Kernels: Least Squares and Feature Interpretation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):10044–10054, 2023.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015.

Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*, 2017.

Cornelius Lanczos. Linear systems in self-adjoint form. *The American Mathematical Monthly*, 9(65): 665–679, 1958.

Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser-Nam Lim. Large Scale Learning on Non-Homophilous Graphs: New Benchmarks and Strong Simple Methods. In *Advances in Neural Information Processing Systems*, 2021.

Shirui Pan, Ruiqi Hu, Sai-Fu Fung, Guodong Long, Jing Jiang, and Chengqi Zhang. Learning graph embedding with adversarial training methods. *IEEE Transactions on Cybernetics*, 50(6): 2475–2487, June 2020. ISSN 2168-2275.

Arun Pandey, Joachim Schreurs, and Johan A.K. Suykens. Generative restricted kernel machines: A framework for multi-view generation and disentangled feature learning. *Neural Networks*, 135: 177–191, 2021.

Arun Pandey, Michaël Fanuel, Joachim Schreurs, and Johan A. K. Suykens. Disentangled representation learning and generation with manifold optimization. *Neural Computation*, 34(10):2009–2036, 09 2022. ISSN 0899-7667.

Jiwoong Park, Minsik Lee, Hyung Jin Chang, Kyuewang Lee, and Jin Young Choi. Symmetric graph convolutional autoencoder for unsupervised graph representation learning. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-GCN: Geometric Graph Convolutional Networks. In *International Conference on Learning Representations*, 2020.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? In *International Conference on Learning Representations*, 2023.

R. Tyrrell Rockafellar. *Conjugate Duality and Optimization*. SIAM, 1974.

Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-Scale attributed node embedding. *Journal of Complex Networks*, 9(1):1–22, 2021.

Ruslan Salakhutdinov. Learning Deep Generative Models. *Annual Review of Statistics and Its Application*, 2(1):361–385, 2015.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective Classification in Network Data. *AI magazine*, 29(3):93–93, 2008.

Dengdi Sun, Dashuang Li, Zhuanlian Ding, Xingyi Zhang, and Jin Tang. Dual-decoder graph autoencoder for unsupervised graph representation learning. *Knowledge-Based Systems*, 234: 107564, December 2021. ISSN 09507051.

Johan A. K. Suykens. SVD revisited: A new variational principle, compatible feature maps and nonlinear extensions. *Applied and Computational Harmonic Analysis*, 40(3):600–609, May 2016. ISSN 1063-5203.

Johan A. K. Suykens. Deep Restricted Kernel Machines Using Conjugate Feature Duality. *Neural Computation*, 29(8):2123–2163, 2017.

Johan A. K. Suykens, Tony Van Gestel, Jos De Brabanter, Bart De Moor, and Joos Vandewalle. *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002.

Qinghua Tao, Francesco Tonin, Alex Lambert, Yingyi Chen, Panagiotis Patrinos, and Johan A. K. Suykens. Learning in Feature Spaces via Coupled Covariances: Asymmetric Kernel SVD and Nyström method. International Conference on Machine Learning, 2024.

Francesco Tonin, Panagiotis Patrinos, and Johan A. K. Suykens. Unsupervised learning of disentangled representations in deep restricted kernel machines with orthogonality constraints. *Neural Networks*, 142:661–679, 2021.

Anton Tsitsulin, John Palowitch, Bryan Perozzi, and Emmanuel Müller. Graph Clustering with Graph Neural Networks. *Journal of Machine Learning Research*, 24(127):1–21, 2023.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *International Conference on Learning Representations*, 2018.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021.

Teng Xiao, Zhengyu Chen, Zhimeng Guo, Zeyang Zhuang, and Suhang Wang. Decoupled self-supervised learning for graphs. In *Advances in Neural Information Processing Systems*, 2022.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*, 2019.

Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting Semi-Supervised Learning with Graph Embeddings. In *International Conference on Machine Learning*, 2016.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do Transformers Really Perform Badly for Graph Representation? In *Advances in Neural Information Processing Systems*, 2022.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *Advances in Neural Information Processing Systems*, 2020.

Mengyi Yuan, Minjie Chen, and Xiang Li. MUSE: Multi-View Contrastive Learning for Heterophilic Graphs. In *ACM International Conference on Information and Knowledge Management*, pp. 3094–3103, 2023.

Xin Zheng, Yi Wang, Yixin Liu, Ming Li, Miao Zhang, Di Jin, Philip S. Yu, and Shirui Pan. Graph neural networks for graphs with heterophily: A survey, 2024. arXiv:2202.07082 [cs].

## A NOTE ON FEATURE MAP CENTERING

In the wKSVD framework, we assume that the feature maps are centered. More precisely, given two arbitrary mappings $\phi(\cdot)$ and $\psi(\cdot)$, the centered mappings are obtained by subtracting the weighted mean:

$$\phi_c(\boldsymbol{x}_i) = \phi(\boldsymbol{x}_i) - \frac{\sum_{k=1}^{n} w_{1,k}\, \phi(\boldsymbol{x}_k)}{\sum_{k=1}^{n} w_{1,k}},$$
$$\psi_c(\boldsymbol{z}_j) = \psi(\boldsymbol{z}_j) - \frac{\sum_{l=1}^{m} w_{2,l}\, \psi(\boldsymbol{z}_l)}{\sum_{l=1}^{m} w_{2,l}}.$$

Although we use the primal formulation in this paper, we next show how to obtain this centering in the dual for the sake of completeness. When using a kernel function or a given similarity matrix, one has no access to the explicit mappings and has to do an equivalently centering in the dual using:

$$\boldsymbol{S}_c = \boldsymbol{M}_1 \boldsymbol{S} \boldsymbol{M}_2^\top,$$

where $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ are the centering matrices:

$$\boldsymbol{M}_1 = \boldsymbol{I}_n - \frac{1}{\mathbf{1}_n^\top \boldsymbol{W}_1 \mathbf{1}_n} \mathbf{1}_n \mathbf{1}_n^\top \boldsymbol{W}_1$$
$$\boldsymbol{M}_2 = \boldsymbol{I}_m - \frac{1}{\mathbf{1}_m^\top \boldsymbol{W}_2 \mathbf{1}_m} \mathbf{1}_m \mathbf{1}_m^\top \boldsymbol{W}_2,$$

with $\boldsymbol{I}_n$ and $\mathbf{1}_n$ a $n \times n$ identity matrix and a $n$-dimensional all-ones vector respectively. We omit the subscript $c$ in the paper and assume the feature maps are always centered. Note that this can easily be achieved in the implementations by using the above equations.

## B ADDITIONAL EXPERIMENTS

**Homophilous experiments** Although our work primarily focuses on heterophilous graphs, we further evaluate our model on homophilous citation networks Cora, Citeseer, and PubMed (Sen et al., 2008; Yang et al., 2016). The dataset statistics can be consulted in Table 5. We employ the same experimental setup as for the heterophilous datasets and report the experimental results in Table 6. While S$^3$GC achieves the best overall performance due to its alignment with the homophily assumption, HeNCler outperforms adjacency partitioning methods like MinCutPool and DMoN. Additionally, HeNCler demonstrates competitive performance with MUSE, the state-of-the-art in heterophilous node clustering, further validating its robustness across different graph types.

Table 5: Dataset statistics of the employed homophilous graphs.

| Dataset | short | # Nodes | # Edges | # Classes | Directed | $\mathcal{H}(\mathcal{G})$ |
|---|---|---|---|---|---|---|
| Cora | cora | 2,708 | 5,278 | 7 | ✗ | 0.765 |
| CiteSeer | cite | 3,327 | 4,614 | 6 | ✗ | 0.627 |
| Pubmed | pub | 19,717 | 44,325 | 3 | ✗ | 0.664 |

**Comprehensive ablation study** We provide the full ablation study results in Table 7, including all datasets and standard deviations. We compare HeNCler with three simplified versions. 'Undirected' reflects an undirected variant of the model with a single MLP decoder. 'wKSVD only' and 'Reconstr only' is the model where only the wKSVD loss $\mathcal{L}_{wKSVD}$ and the reconstruction losses $\mathcal{L}_{\text{NodeRec}} + \mathcal{L}_{\text{EdgeRec}}$ are used respectively. We observe that HeNCler performs better than its undirected version, even for undirected graphs, and that all loss terms contribute to HeNCler's performance.

## C COMPUTATION TIMES

We trained MinCutPool, DMoN, and HeNCler for 300 iterations; and S$^3$GC for 30 iterations on a Nvidia V100 GPU, and report the computation times in Table 8. Figure 4 visualises these result w.r.t. the number of nodes in the graph, showing the linear time complexity of HeNCler and that it is insensitive to the number of edges. We conclude that HeNCler demonstrates fast computation times.

Table 6: Experimental results on homophilous 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.

| | | Baselines | | | | | Ours |
|---|---|---|---|---|---|---|---|
| Dataset | | KMeans | MinCutPool | DMoN | $S^3$GC | MUSE | HeNCler |
| cora | NMI | 35.0±3.21 | 49.0±2.24 | 51.7±1.63 | **53.62**±0.55 | 36.45±2.71 | 38.81±2.26 |
| | F1 | 36.0±2.12 | 47.1±1.78 | 51.8±2.02 | **60.12**±0.46 | 50.78±2.79 | 47.93±2.60 |
| cite | NMI | 19.9±2.90 | 29.5±3.21 | 30.3±1.09 | **43.56**±0.65 | 39.03±1.99 | 34.83±2.21 |
| | F1 | 39.4 ±3.07 | 47.1±1.21 | 57.4±3.42 | **64.12**±0.28 | 52.89±1.68 | 48.70±2.79 |
| pub | NMI | 31.4±2.18 | 21.4±1.46 | 25.7±2.46 | 31.01±2.35 | **36.09**±3.26 | 27.26±1.72 |
| | F1 | 59.2±2.32 | 44.5±2.47 | 34.3±2.05 | **69.12**±1.39 | 61.26±1.50 | 51.17±1.75 |

Table 7: Full Ablation study results. We report NMI and F1 scores for 10 runs (mean ± standard deviation in %) where higher is better. Best results are highlighted in bold.

| | | ablations | | | full model |
|---|---|---|---|---|---|
| dataset | | Undirected | Reconstr only | wKSVD only | HeNCler |
| tex | NMI | 27.58±4.75 | 29.54±2.27 | 31.64±2.14 | **43.65**±2.52 |
| | F1 | 65.20±2.06 | 66.64±1.83 | 62.83±3.91 | **71.39**±2.16 |
| corn | NMI | 18.12±2.57 | 27.76±3.29 | 20.63±5.92 | **41.52**±4.35 |
| | F1 | 53.69±0.98 | 54.70±1.88 | 47.12±2.61 | **63.40**±3.67 |
| wis | NMI | 25.08±3.54 | 34.65±1.86 | 39.86±4.63 | **47.13**±1.76 |
| | F1 | 57.13±1.34 | 62.28±1.58 | 63.60±2.46 | **68.30**±2.17 |
| cha | NMI | 19.91±0.48 | 22.02±0.25 | 22.60±0.57 | **23.89**±0.84 |
| | F1 | 44.08±1.79 | 43.42±1.63 | 42.98±0.37 | **44.14**±1.83 |
| squi | NMI | 9.59±0.21 | 9.59±0.27 | 9.56±0.19 | **9.67**±0.13 |
| | F1 | **55.43**±0.03 | 53.74±3.77 | 36.42±1.85 | 36.51±2.39 |
| rom | NMI | 33.17±1.25 | **40.05**±0.82 | 35.99±0.95 | 36.99±0.61 |
| | F1 | 33.57±2.15 | 35.16±1.34 | 35.30±0.97 | **35.43**±1.07 |
| mine | NMI | **0.08**±0.02 | 0.07±0.02 | 0.04±0.01 | 0.06±0.00 |
| | F1 | 76.15±2.25 | 76.05±2.16 | 73.77±3.40 | **76.48**±1.56 |
| tol | NMI | 6.33±0.94 | 6.18±0.67 | 4.42±0.54 | **6.73**±0.59 |
| | F1 | 73.89±4.00 | 68.60±5.95 | 68.45±7.57 | **73.66**±2.10 |

Table 8: Computation times in seconds.

| DATASET | BASELINES | | | OURS |
|---|---|---|---|---|
| | MINCUTP. | DMON | $S^3$GC | HENCLER |
| CHAMELEON | 8 | 20 | 89 | 24 |
| SQUIRREL | 14 | 86 | 105 | 49 |
| ROMAN-EMPIRE | 67 | 71 | 312 | 57 |
| AMAZON-RATING | 109 | 93 | 195 | 63 |
| MINESWEEPER | 55 | 27 | 98 | 21 |
| TOLOKERS | 71 | 198 | 100 | 35 |
| QUESTIONS | 340 | 215 | 217 | 125 |

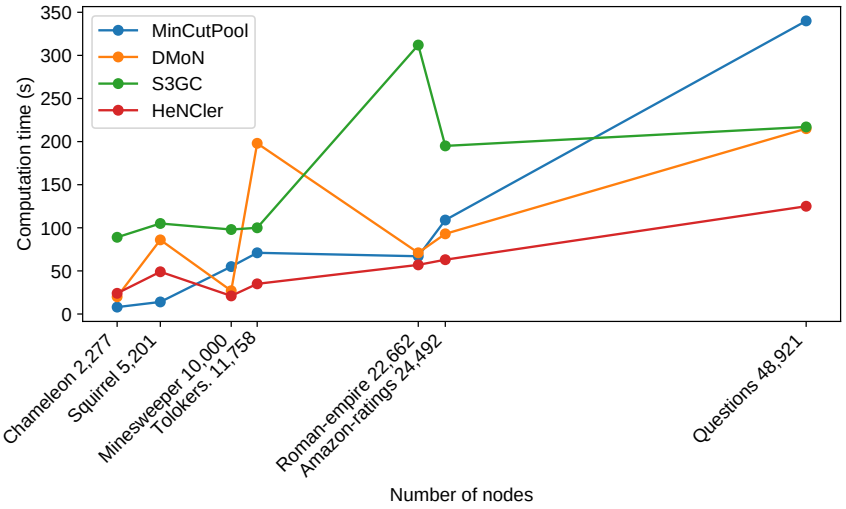


Figure 4: Computation times of MinCutPool, DMoN, $S^3$GC, and HeNCler w.r.t. the number of nodes of the datasets. We observe that HeNCler scales linearly with the number of nodes, and that it is not sensitive to the number of edges, as opposed to DMoN, showing a significant peak for the Tolokers dataset due the large number of edges in this graph.