Learning in Feature Spaces via Coupled Covariances: Asymmetric Kernel SVD and Nyström method

Qinghua Tao [*1] **Francesco Tonin** [*2] **Alex Lambert** [1] **Yingyi Chen** [1] **Panagiotis Patrinos** [1] **Johan A.K. Suykens** [1]

Abstract

In contrast with Mercer kernel-based approaches as used e.g. in Kernel Principal Component Analysis (KPCA), it was previously shown that Singular Value Decomposition (SVD) inherently relates to asymmetric kernels and asymmetric Kernel Singular Value Decomposition (KSVD) has been proposed. However, the existing formulation to KSVD cannot work with infinite-dimensional feature mappings, the variational objective can be unbounded, and needs further numerical evaluation and exploration towards machine learning. In this work, *i)* we introduce a new asymmetric learning paradigm based on coupled covariance eigenproblem (CCE) through covariance operators, allowing infinite-dimensional feature maps. The solution to CCE is ultimately obtained from the SVD of the induced asymmetric kernel matrix, providing links to KSVD. *ii)* Starting from the integral equations corresponding to a pair of coupled adjoint eigenfunctions, we formalize the asymmetric Nyström method through a finite sample approximation to speed up training. *iii)* We provide the first empirical evaluations verifying the practical utility and benefits of KSVD and compare with methods resorting to symmetrization or linear SVD across multiple tasks.

1. Introduction

Feature mappings can transport the data in a Hilbert space of a typically higher dimension. They are intimately linked through inner products with reproducing kernels (Aronszajn, 1950) and thus often associated with symmetric learning. One can for example think of kernel principal components analysis (KPCA, Schölkopf et al. (1998)) where one tries to find orthonormal directions in the feature space that maximize the variance associated to a symmetric Gram matrix, or kernel canonical correlation analysis (KCCA, Lai & Fyfe (2000)) where the maximization of a correlation based on two different views of the data leads to an optimization problem governed by two symmetric Gram matrices.

In many real-world applications however, there is an inherent degree of asymmetry. Among others, directed graphs of citation networks (Ou et al., 2016), biclustering (Kluger et al., 2003), attention in Transformers (Wright & Gonzalez, 2021; Chen et al., 2023) typically involve an asymmetry that cannot be captured when working with reproducing kernels. Often the asymmetric matrices are first symmetrized before applying some matrix decomposition such as singular value decomposition (SVD, Strang (2006); Golub & Van Loan (2013)) so that only one set of eigenvectors is obtained.

As a fundamental linear algebra tool, SVD can process arbitrary non-symmetric matrices and jointly learns both left and right singular vectors, e.g., embeddings of source and target nodes (Estrada, 2012). However, SVD alone lacks flexibility for nonlinear feature learning. Suykens (2016) propose asymmetric kernel SVD (KSVD), a variational principle based on least-square support vector machines (LSSVMs) that leads to the matrix SVD and mentions that nonlinear extensions can be obtained when the SVD is applied to an asymmetric kernel matrix rather than the given data matrix. However, their formulation only allows finite-dimensional feature mappings to induce the kernel and its variational objective is unbounded unless the regularization hyperparameters are properly selected. Yet, Suykens (2016) does not provide numerical evaluations on the practical utility and applications of KSVD, leaving this topic largely unexplored. While infinite-dimensional feature maps are common in all kernel methods, including the asymmetric ones, e.g., Wright & Gonzalez (2021) focus on the understandings of the asymmetric dot-product attention kernel resulting from the queries and keys through a pair of Banach spaces in the supervised setting, little literature addresses learning with generic asymmetric kernel machines with infinite-dimensional maps. Differently, our work provides a

[*] Equal contribution [1] ESAT-STADIUS, KU Leuven, Belgium [2] LIONS, EPFL, Switzerland (most of the work was done at ESAT-STADIUS, KU Leuven). Correspondence to: Qinghua Tao, Francesco Tonin <qinghua.tao@esat.kuleuven.be, francesco.tonin@epfl.ch>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

new asymmetric learning paradigm for unsupervised feature learning based on the CCE allowing two generic datasets.

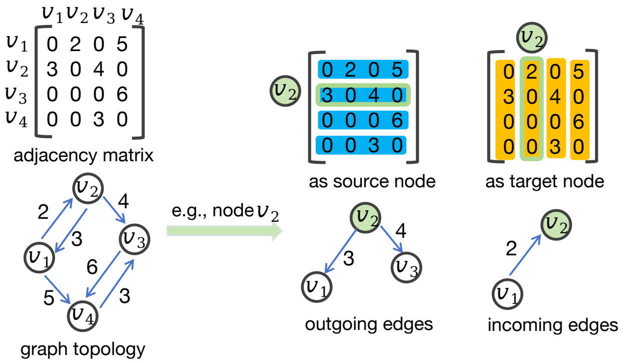
Kernel methods additionally suffer from efficiency, as they require processing a kernel matrix that is quadratic in the sample size. Many approaches have been proposed to improve the efficiency, among which the Nyström method has been widely applied (Williams & Seeger, 2000; Zhang et al., 2008; Gittens & Mahoney, 2016; Meanti et al., 2020; Xiong et al., 2021). The Nyström method of subsampling arises from the approximate eigendecomposition of an integral operator associated with a symmetric kernel (Baker, 1981), which restricts the existing Nyström method to only Mercer kernels. In (Drineas et al., 2005; Nemtsov et al., 2016; Xiong et al., 2021), Nyström-like methods for matrix compression or approximation are discussed by directly applying the symmetric Nyström method to estimate left and right singular vectors, yet ignoring the asymmetry constraints. In (Michaeli et al., 2016), though the asymmetric Nyström method is mentioned in the proposed nonparametric KCCA method, it still leverages the existing symmetric Nyström method in implementation for the eigenvectors of two symmetric positive definite kernels and can only deal with square matrices. Hence, the analytical framework of the Nyström method to asymmetric kernel machines remains to be formalized and is of particular interest for the efficient computation of KSVD.

The research question that we tackle in this paper is "*How can we learn directions in the feature space in an asymmetric way while controlling the computational complexity of our method ?*"

The technical contributions of this work are summarized as:

- We first present a new asymmetric learning paradigm based on *coupled covariances eigenproblem* (CCE) allowing infinite-dimensional feature maps. We show that its solution leads to the KSVD problem associated with a specific asymmetric similarity matrix that blends in two feature maps.
- We leverage the integral equations involving the pair of adjoint eigenfunctions related to the continuous analog of SVD and derive an extension to the Nyström method able to handle asymmetric kernels, which can be used to speed up KSVD training without significant decrease in accuracy of the solution.
- We conduct extensive experiments to demonstrate the performance of the CCE asymmetric learning scheme in unsupervised feature extraction and different downstream tasks with real-world datasets. The efficacy of the proposed Nyström method is also verified to efficiently compute the KSVD.

Note that we do not claim to propose the KSVD algorithm, as it was already sketched in the letter by Suykens (2016). Rather, we give a novel asymmetric learning problem based on two covariance operators in the feature space, whose solution coincides with a KSVD with infinite-dimensional feature maps, a case that was not previously possible.



$$\begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 0 & 2 & 0 & 5 \\ v_2 & 3 & 0 & 4 & 0 \\ v_3 & 0 & 0 & 0 & 6 \\ v_4 & 0 & 0 & 3 & 0 \end{array}$$

Figure 1. Illustrative example of asymmetric similarity. In a directed graph, each node can act as the source or the target. Given the adjacency matrix $[a(v_i, v_j)]_{i,j=1}^N$, its rows relate to the outgoing edges, while the columns relate to the incoming edges. The connections between nodes are directional, s.t. $a(v_i, v_j) \neq a(v_j, v_i)$, $i \neq j$.

2. Learning in Feature Spaces with Asymmetry

We begin this section by reviewing in Section 2.1 the concept of asymmetric similarity that is critical to this work, before introducing in Section 2.2 the Coupled Covariances Eigenproblem (CCE) that allows us to learn in feature spaces with asymmetry as the solution is ultimately obtained from the SVD of an asymmetric similarity matrix. We conclude in Section 2.3 with some remarks about related work.

2.1. Asymmetric Similarity

Typically, a kernel $\hat{\kappa}\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is induced by a *single* feature map $\hat{\phi}$ on a *single* data set whose samples lie in a space $\mathcal{X}$ and is symmetric. However, in practice, asymmetric similarities are widely used such as in directed graphs (where similarity is directional) as exemplified in Fig. 1. Each node acts as source and target and is associated with two feature vectors x_i, z_i, possibly from different spaces, for its source and target role, respectively. One can thus extract two sets of features for each node, one related to the nodes to which it points and one for the nodes that point to it. In general, an asymmetric kernel $\kappa\colon \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ describes a similarity between elements from *two different* spaces $\mathcal{X}, \mathcal{Z}$. Despite the utility of asymmetry, classical Mercer-kernel methods, *e.g.* KPCA, only deal with symmetric similarities induced by a single feature map, and thus one has to resort

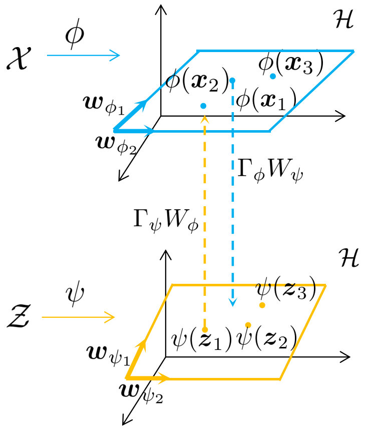


Figure 2. Schematic of our construction. $\mathcal{X}$, $\mathcal{Z}$ from A are mapped to a possibly infinite-dimensional space $\mathcal{H}$. We propose to consider coupled scalar products $\psi(z_j)$ onto w_l^ϕ and $\phi(x_i)$ onto w_l^ψ. $\mathcal{H}$ is shown separately for clarity.

KPCA	**CCE**
$\Sigma_\phi w_\phi = \lambda_\phi w_\phi$	$\Sigma_\phi w_\psi = \lambda w_\phi$
$\Sigma_\psi w_\psi = \lambda_\psi w_\psi$	$\Sigma_\psi w_\phi = \lambda w_\psi$

Figure 3. Overview comparison of KPCA and CCE.

to symmetrizing an asymmetric similarity matrix K, which can be done by considering $(K^\top + K)/2$, $KK^\top$, or $K^\top K$.

Compared to the literature on Mercer kernels, asymmetric kernels are less studied. They have been mostly applied in supervised learning, e.g., regression (Mackenzie & Tieu, 2004; Wu et al., 2010; Kuruwita et al., 2010) and classification (Muñoz et al., 2003; Koide & Yamashita, 2006; Tsuda, 1998). Some works do not resort to symmetrization: (He et al., 2023) applies two feature mappings to the given samples and maintains an asymmetric kernel in the LSSVM classifier. Chen et al. (2023) applies the variational objective from (Suykens, 2016) as an auxiliary regularization loss to the model for low-rank self-attention in Transformers are built as the asymmetric similarity between queries and keys. Relaxations of the Mercer conditions have also been generalized to learning in reproducing kernel Banach spaces (Zhang et al., 2009; Xu & Ye, 2019) and Kreĭn spaces (Oglic & Gaertner, 2018). Other related but orthogonal approaches include (Neto & Rodrigues, 2023) for robust SVD estimation with Gaussian norm in the original space, and (Vasilescu, 2009) for tensor data where SVD is applied to the symmetric kernel in each mode.

2.2. Coupled Covariances Eigenproblem

The goal of this section is to gradually define and solve the Coupled Covariances Eigenproblem (CCE). Our goal is to provide a new tool able to learn in (infinite-dimensional) feature spaces and take advantage of asymmetry.

Notation. Given a bounded linear operator Γ between Hilbert spaces, its adjoint is referred to as Γ^*. The Frobenius norm of a matrix is denoted by $\|\cdot\|_{\mathrm{F}}$. The identity matrix of size r is I_r. Set the spaces $\mathcal{X} = \mathbb{R}^m$ and $\mathcal{Z} = \mathbb{R}^n$. We assume access to two sets of samples $\{x_i\}_{i=1}^n \in \mathcal{X}^n$ and $\{z_j\}_{j=1}^m \in \mathcal{Z}^m$. We consider two mappings $\phi \colon \mathcal{X} \to \mathcal{H}$ and $\psi \colon \mathcal{Z} \to \mathcal{H}$ whose outputs lie in a common feature space $\mathcal{H}$. We moreover assume that the feature maps are centered. In practice, given the training samples, one can realize the centering by the translated feature maps $\tilde{\phi}(x) = \phi(x) - \frac{1}{n}\sum_{i=1}^n \phi(x_i)$ and $\tilde{\psi}(z) = \psi(z) - \frac{1}{m}\sum_{j=1}^m \psi(z_i)$, and then the similarity matrix of interest $[\tilde{G}]_{ij} = \langle \tilde{\phi}(x_i), \tilde{\psi}(z_j)\rangle$ can be computed straightforwardly, e.g., $\tilde{G} = (I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)G(I_m - \frac{1}{m}\mathbf{1}_m\mathbf{1}_m^\top)$.

Construction of the Subspaces in $\mathcal{H}$. In CCE, the goal is to learn a pair of r directions in the feature space $\mathcal{H}$ that solve a coupled eigenvalues problem. The sough-after directions are collected in vectors $W_\phi \in \mathcal{H}^r, W_\psi \in \mathcal{H}^r$ as follows:

$$W_\phi = [w_1^\phi, \dots, w_r^\phi], \qquad W_\psi = [w_1^\psi, \dots, w_r^\psi].$$

Denote by $\Sigma_\phi, \Sigma_\psi \in \mathcal{L}(\mathcal{H})$ the empirical covariance operators described by

$$\Sigma_\phi = \frac{1}{n}\sum_{i=1}^n \phi(x_i)\phi(x_i)^*, \quad \Sigma_\psi = \frac{1}{m}\sum_{j=1}^m \psi(z_j)\psi(z_j)^*.$$

While performing KPCA would result in solving two eigenvalue problems independently for both covariance operators and using the top r eigenvectors of each to compute interesting directions, we propose to intricate the learned directions in the feature space by solving the following CCE problem:

Definition 2.1 (CCE). Find $W_\phi \in \mathcal{H}^r, W_\psi \in \mathcal{H}^r$ such that

$$\Sigma_\phi W_\psi = \Lambda W_\phi, \qquad \Sigma_\psi W_\phi = \Lambda W_\psi, \tag{1}$$

for some diagonal matrix $\Lambda \in \mathbb{R}^{r\times r}$ with positive values.

Even if $\mathcal{H}$ is infinite-dimensional, we can parameterize the directions W_ϕ, W_ϕ using matrices. Indeed, given that a solution exists, it holds that for any $l \in \{1, \dots, r\}$

$$\Sigma_\phi w_l^\psi = \frac{1}{n}\sum_{i=1}^n \langle \phi(x_i), w_l^\psi\rangle \phi(x_i) = \lambda_l w_l^\phi.$$

Thus all directions $\{w_l^\phi\}_{l=1}^r$ lie in Span $\{\phi(x_i)\}_{i=1}^n$. Consequently, we can parameterize the directions W_ϕ over the $\{\phi(x_i)\}_{i=1}^n$ by a matrix of coefficients $B_\phi \in \mathbb{R}^{n\times r}$. A similar argument holds for the directions W_ψ over the

$\{\psi(z_j)\}_{j=1}^m$ with coefficients $B_\psi \in \mathbb{R}^{m\times r}$ so that for all $l \in \{1,\dots,r\}$

$$w_l^\phi = \sum_{i=1}^n b_{il}^\phi \phi(x_i), \qquad w_l^\psi = \sum_{j=1}^m b_{jl}^\psi \psi(z_j). \tag{2}$$

Projection Operators. Let $\Gamma_\phi \colon \mathcal{H}^r \to \mathbb{R}^{n\times r}$ and $\Gamma_\psi \colon \mathcal{H}^r \to \mathbb{R}^{m\times r}$ be linear operators acting on some directions $W \in \mathcal{H}^r$ in the following way:

$$[\Gamma_\phi W]_{il} = \frac{1}{\sqrt{n}}\langle \phi(x_i), w_l\rangle, \quad [\Gamma_\psi W]_{jl} = \frac{1}{\sqrt{m}}\langle \psi(z_j), w_l\rangle.$$

These operators compute the inner products between the chosen directions and the feature maps associated with the data. As Γ_ϕ and Γ_ψ are bounded linear operators they admit adjoint operators whose action can be made explicit: for any $B \in \mathbb{R}^{n\times r}$, $\Gamma_\phi^* B = \frac{1}{\sqrt{n}}[\sum_{i=1}^n b_{il}\phi(x_i)]_{l=1}^r \in \mathcal{H}^r$ and Γ_ψ^* can be treated similarly. This observation allows us to rewrite Equation 2 under the form

$$W_\phi = \Gamma_\phi^* B_\phi, \qquad W_\psi = \Gamma_\psi^* B_\psi.$$

We also remark that the covariance operators Σ_ϕ and Σ_ψ can be expressed using these projection operators, so that Equation 1 can be reformulated using matrix variables B_ϕ, B_ψ as

$$\Gamma_\phi^*\Gamma_\phi\Gamma_\psi^* B_\psi = \Gamma_\phi^* B_\phi \Lambda, \quad \Gamma_\psi^*\Gamma_\psi\Gamma_\phi^* B_\phi = \Gamma_\psi^* B_\psi \Lambda. \tag{3}$$

Asymmetric Kernel Matrix. The operators $\Gamma_\psi\Gamma_\phi^*$ and $\Gamma_\phi\Gamma_\psi^*$ are of particular interest and their action can be described by related matrices as formalized in the following.

Proposition 2.2. *Let $G \in \mathbb{R}^{n\times m}$ such that $g_{ij} = \frac{1}{\sqrt{nm}}\langle \phi(x_i), \psi(z_j)\rangle$. For all $B_\phi \in \mathbb{R}^{n\times r}$ and $B_\psi \in \mathbb{R}^{m\times r}$, it holds that*

$$\Gamma_\psi\Gamma_\phi^* B_\phi = G^\top B_\phi, \qquad \Gamma_\phi\Gamma_\psi^* B_\psi = G B_\psi.$$

This proposition resembles the celebrated *kernel trick* but induces an asymmetry in what is an equivalent of the Gram matrix associated with an asymmetric kernel $\kappa(x,z) = \langle \phi(x), \psi(z)\rangle$. This kernel permits to avoid the explicit computation of the feature mappings.

Because most classical kernel functions require that the two inputs have compatible dimensions, there are a few challenges associated with the computation of $\kappa(x,z)$ when $\mathcal{X}$ and $\mathcal{Z}$ are different by nature. In this case, we can transform the two inputs x, z into the same dimension through a compatible linear transformation $C \in \mathcal{L}(\mathcal{Z},\mathcal{X})$. For Euclidean spaces we can find matrices C, such that $C^\top x$ is compatible with z in dimensions, and then apply existing (symmetric) kernel functions thereafter.

Remark 2.3 (Dimensionality Compatibility Matrix). We consider different alternatives to attain the compatibility matrix C as follows: a_0) the pseudo-inverse of the tackled data matrix; however, it can be computationally unstable and expensive, thus we propose the following a_1-a_3.

a_1) PCA projection on x_i; it finds the projection directions capturing the most variance of data samples (Jolliffe, 1986). a_2) randomizing the projection C; the random linear transformation has been shown to retain the main patterns of the data matrix (Larsen & Nelson, 2017). a_3) learnable C w.r.t. the downstream tasks; it gives the optimal C by optimizing the downstream task objective, e.g., classification loss.

a_2 is very computationally efficient while learning the optimal C in a_3 can take more computation, up to the task and its optimizer, e.g. SGD optimizer with backpropagated C as experimented in Section 4.3. Note that a_0-a_2 can be applied under general unsupervised setups for feature learning, while a_3 is commonly used when considering end-to-end training for the downstream tasks under supervised setups.

Solution to the CCE. Solving the CCE gives rise to a generalized shifted eigenvalue problem, as shown in the following proposition.

Proposition 2.4. *Let $G \in \mathbb{R}^{n\times m}$ be the asymmetric kernel matrix from Proposition 2.2. The directions $(W_\phi, W_\psi) \in \mathcal{H}^r$ respectively parameterized by the matrices $(B_\phi, B_\psi) \in \mathbb{R}^{n\times r}\times\mathbb{R}^{m\times r}$ are solution to the CCE problem if and only if (B_ϕ, B_ψ) are solution to the generalized shifted eigenvalue problem*

$$\begin{aligned} G^\top G B_\psi &= G^\top B_\phi \Lambda, \\ G G^\top B_\phi &= G B_\psi \Lambda, \end{aligned} \tag{4}$$

where $\Lambda \in \mathbb{R}^{r\times r}$ is a positive diagonal matrix.

According to Lanczos' decomposition theorem (Lanczos, 1958), Problem 4 can be solved by taking for B_ϕ, B_ψ the top-r left and right singular vectors of the matrix G.

Proposition 2.5. *Let B_ϕ^{svd} (resp. B_ψ^{svd}) be top-r left (resp. right) singular vectors of G. Then*

$$W_\phi = \Gamma_\phi^* B_\phi^{svd}, \qquad W_\psi = \Gamma_\psi^* B_\psi^{svd}$$

is a solution to the CCE.

We have shown that solving the CCE reduces to an KSVD problem, with an asymmetric similarity matrix that involves both feature maps. Once the directions are learned, if we are given some new data $x \in \mathcal{X}$ or $z \in \mathcal{Z}$ we can compute the projected feature scores

$$[\langle \phi(x), w_l^\psi\rangle]_{l=1}^r, \qquad [\langle \psi(z), w_l^\phi\rangle]_{l=1}^r,$$

and use these in downstream tasks.

CCE versus 2KPCA. The proposed CCE problem gives a new understanding of the set of directions of interest in the feature space, namely W_ϕ and W_ψ from Proposition 2.5, arising from the SVD of the asymmetric kernel matrix G and the feature maps ϕ, ψ. We note that these directions can also be interpreted as the principal directions associated to the covariance operators of two symmetrized kernels in two separate KPCA problems arising from feature maps $x \mapsto \Sigma_\psi^{1/2}\phi(x)$ and $z \mapsto \Sigma_\phi^{1/2}\psi(z)$, respectively. In the dual, this corresponds to taking the SVDs of $GG^\top$ and $G^\top G$, which is equivalent to taking the SVD of G. We refer to this interpretation as 2KPCA. From a computational standpoint, performing 2KPCA or CCE yields the same singular vectors. However, they are significantly different in the modelling from the following perspectives.

- In 2KPCA, one takes the principal components associated to kernels built via complicated entanglement of ϕ and ψ. In CCE, the empirical covariances associated to both feature maps appear free from any other factor.
- The coupling between the two input variables within the feature maps of 2KPCA is realized through the square root of the other covariance, while in CCE the coupling of the input variables naturally arises by crossing the learned directions in Definition 2.1.
- For principal component extraction, we need to compute the projections on the singular vectors W_ϕ and W_ψ in $\mathcal{H}$, which are essential in downstream tasks to extract the principal components of test points. This can be easily accomplished in CCE with explicit directions, while it is not as clear in 2KPCA.

2.3. Related work

We now discuss research areas that are tangent to our topic: asymmetric kernel SVD (KSVD) and symmetric kernel approaches such as KPCA or KCCA.

Asymmetric Kernel SVD. Given a data matrix $A \in \mathbb{R}^{n\times m}$, (Suykens, 2016) regards it w.r.t. either the collection of rows $\{A[i,:] \triangleq x_i \in \mathcal{X}\}_{i=1}^n$ or the collection of columns $\{A[:,j] \triangleq z_j \in \mathcal{Z}\}_{j=1}^m$. In the example in Fig. 1, $\mathcal{X}$ denotes the outgoing edges of source nodes, while $\mathcal{Z}$ denotes the incoming edges of the target. (Suykens, 2016) proposes a variational principle for SVD with two linear mappings $\phi(x_i) = C_1^\top x_i, \psi(z_j) = C_2 z_j$ with compatibility transformations C_1, C_2 on the rows and columns of A. Provided that the compatibility condition $AC_1C_2A = A$ holds, the stationary solutions correspond to the SVD of A. The two mappings can be extended to construct the $n \times m$ matrix $G_{ij} = k(\phi(x_i), \psi(z_j))$, where k is a kernel function allowing to be nonlinear and asymmetric. Stationary solutions are then linked to the SVD of G when the regularization hyperparameters are fixed as the top singular values of G. The KSVD algorithm therefore finds singular vectors of features non-linearly related to the input variables through the SVD of the non-symmetric rectangular matrix K.

Symmetric Kernel Approaches with Covariances. Our new construction makes it easier to compare KSVD with other common algorithms based on finding the best approximation of some covariance quantity, which instead work with symmetric kernels in contrast to our work. KPCA applies a nonlinear feature mapping ϕ to a set of data samples x_i and considers projections ${a_{\phi_1}}^\top\phi(x_i)$ for maximal variances w.r.t. a *single* covariance cov $(\Phi a_{\phi_1}, \Phi a_{\phi_1})$. KPCA can also be tackled through a symmetric PSD kernel $k_\phi := \phi^\top(\cdot)\phi(\cdot)$ (Schölkopf et al., 1998), while KSVD works with two covariances coupled by each other. We note that doing two KPCA with $\phi(x_i)$ and $\psi(z_j)$ lead to two decoupled covariances and lead to two symmetric kernels $\phi^\top(\cdot)\phi(\cdot)$ and $\psi^\top(\cdot)\psi(\cdot)$ w.r.t. x_i and z_j, respectively. This is significantly different from KSVD, as shown in Figure 3, as KSVD is associated with two coupled covariances and essentially works with an asymmetric kernel $\phi^\top(\cdot)\psi(\cdot)$. KCCA deals with samples from two data sources and only considers projections of each data source. KSVD seeks maximal variances of two sets of projections from a single matrix. Specifically, KCCA considers projections ${a_{\phi_1}}^\top\phi(x_i)$ and ${a_{\psi_1}}^\top\psi(z_i)$ and couples them in a *single* covariance cov $(\Phi a_{\phi_1}, \Psi a_{\psi_1})$. In our formulation, we consider ${a_{\phi_1}}^\top\psi(z_j)$ and ${a_{\psi_1}}^\top\phi(x_i)$ leading to *two* covariances $\mathrm{cov}(\Psi a_{\phi_1}, \Psi a_{\phi_1})$, $\mathrm{cov}(\Phi a_{\psi_1}, \Phi a_{\psi_1})$. KCCA leads to *two separate symmetric PSD kernels* $k_\phi := \phi^\top(\cdot)\phi(\cdot)$, $k_\psi := \psi^\top(\cdot)\psi(\cdot)$, while KSVD couples two feature mappings inducing a *single asymmetric kernel* $\kappa := \phi^\top(\cdot)\psi(\cdot)$. Our construction is therefore key to allow for asymmetric kernels w.r.t. KCCA and contrasts with earlier KSVD constructions, where drawing parallels with related approaches such as KCCA was notably challenging due to the lack of a covariance and subspace interpretation.

3. Nyström Method for Asymmetric Kernels

We adapt the celebrated Nyström method to asymmetric kernels, the goal being to speed up the computation of the left and right singular vectors of G from Section 2. The existing Nyström method approximates eigenfunctions of the integral operator associated with a symmetric kernel (Williams & Seeger, 2000). Schmidt (1907) discusses the treatment of the integral equations with an asymmetric kernel for the continuous analog of SVD (Stewart, 1993). In this section, we base our formulation upon the pair of adjoint eigenfunctions originally studied in (Schmidt, 1907), namely singular functions, and start from the corresponding integral equations (Baker, 1981) to formally derive the asymmetric Nyström method in a similar spirit with the widely adopted symmetric Nyström method (Williams & Seeger, 2000).

Adjoint Eigenfunctions With an asymmetric kernel $\kappa(x,z)$, $u_s(x)$ and $v_s(z)$ satisfying

$$\begin{aligned}\lambda_s u_s(x) &= \int_{\mathcal{D}_z} \kappa(x,z) v_s(z)\, p_z(z) dz, \\ \lambda_s v_s(z) &= \int_{\mathcal{D}_x} \kappa(x,z) u_s(x)\, p_x(x) dx\end{aligned} \tag{5}$$

are called a pair of adjoint eigenfunctions corresponding to the eigenvalue λ_s with $\lambda_1 \geq \lambda_2 \geq \ldots \geq 0$, where $p_x(x)$ and $p_z(z)$ are the probability densities over $\mathcal{D}_x$ and $\mathcal{D}_z$. Note that (Schmidt, 1907) works with the reciprocal of λ_s, which is called a singular value by differentiating from the eigenvalues of symmetric matrices (Stewart, 1993). The integral equations in (5) do not specify the normalization of the adjoint eigenfunctions, which correspond to the left and right singular vectors with finite sample approximation, while generally in SVD the singular values are solved as orthonormal. Thus, to correspond the results of the adjoint eigenfunctions to the orthonormal singular vectors in SVD, the scalings determining the norms are implicitly included in (5). For normalization, we incorporate three scalings $l_{\lambda_s}, l_{u_s}, l_{v_s}$ for $\lambda_s, u_s(x), v_s(z)$, respectively, into (5), such that $l_{\lambda_s} \lambda_s l_{u_s} u_s(x) = \int_{\mathcal{D}_z} \kappa(x,z) l_{v_s} v_s(z)\, p_z(z) dz$ and $l_{\lambda_s} \lambda_s l_{v_s} v_s(z) = \int_{\mathcal{D}_x} \kappa(x,z) l_{u_s} u_s(x)\, p_x(x) dx$.

Nyström Approximation for the Adjoint Eigenfunctions Given the i.i.d. samples $\{x_1, \ldots, x_n\}$ and $\{z_1, \ldots, z_m\}$, similar to (Williams & Seeger, 2000), from the probability densities $p_x(x), p_z(z)$ over $\mathcal{D}_x, \mathcal{D}_z$, the two integral equations in (5) over $p_x(x)$ and $p_z(z)$ are approximated by an empirical average:

$$\begin{aligned}\lambda_s u_s(x) &\approx \frac{l_{v_s}}{m l_{\lambda_s} l_{u_s}} \sum_{j=1}^{m} \kappa(x, z_j) v_s(z_j), \\ \lambda_s v_s(z) &\approx \frac{l_{u_s}}{n l_{\lambda_s} l_{v_s}} \sum_{i=1}^{n} \kappa(x_i, z) u_s(x_i),\end{aligned} \tag{6}$$

where $s = 1, \ldots, r$, which corresponds to the rank-r compact SVD on a kernel through the Lanczos' decomposition theorem (Lanczos, 1958):

$$\begin{aligned}G^{(n,m)} V^{(n,m)} &= U^{(n,m)} \Lambda^{(n,m)}, \\ (G^{(n,m)})^\top U^{(n,m)} &= V^{(n,m)} \Lambda^{(n,m)},\end{aligned} \tag{7}$$

where $G^{(n,m)} \in \mathbb{R}^{n \times m}$ is the asymmetric kernel matrix with entries $G_{ij} = \kappa(x_i, z_j)$ and $r \leq \min\{n, m\}$, $V^{(n,m)} = [v_1^{(n,m)}, \ldots, v_r^{(n,m)}] \in \mathbb{R}^{m \times r}, U^{(n,m)} = [u_1^{(n,m)}, \ldots, u_r^{(n,m)}] \in \mathbb{R}^{n \times r}$ are column-wise orthonormal and contain the singular vectors, and $\Lambda^{(n,m)} = \operatorname{diag}\{\lambda_1^{(n,m)}, \ldots, \lambda_r^{(n,m)}\}$ denotes the positive singular values. To match (6) against (7), we first require the scalings on the right side of the two equations in (6) to be consistent, i.e., $l_{v_s}/(m l_{\lambda_s} l_{u_s}) \triangleq l_{u_s}/(n l_{\lambda_s} l_{v_s})$, which yields $l_{v_s} = (\sqrt{m}/\sqrt{n})\, l_{u_s}$ and $l_{v_s}/(m l_{\lambda_s} l_{u_s}) \triangleq l_{u_s}/(n l_{\lambda_s} l_{v_s}) = 1/\left(\sqrt{mn} l_{\lambda_s}\right)$.

When running all samplings x_i and z_j in (6) to match (7), we arrive at: $u_s(x_i) \approx \sqrt{\sqrt{mn} l_{\lambda_s}} U_{is}^{(n,m)}$, $v_s(z_j) \approx \sqrt{\sqrt{mn} l_{\lambda_s}} V_{js}^{(n,m)}$, $\lambda_s \approx (1/(\sqrt{mn} l_{\lambda_s})) \lambda_s^{(n,m)}$. The Nyström approximation to the s-th pair of adjoint eigenfunctions with an asymmetric kernel $\kappa(x,z)$ is obtained for $s = 1, \ldots, r$:

$$\begin{aligned}u_s^{(n,m)}(x) &\approx (\sqrt{\sqrt{mn} l_{\lambda_s}} / \lambda_s^{(n,m)}) \sum\nolimits_{j=1}^{m} \kappa(x, z_j) V_{js}^{(n,m)}, \\ v_s^{(n,m)}(z) &\approx (\sqrt{\sqrt{mn} l_{\lambda_s}} / \lambda_s^{(n,m)}) \sum\nolimits_{i=1}^{n} \kappa(x_i, z) U_{is}^{(n,m)},\end{aligned} \tag{8}$$

which are also called the out-of-sample extension to evaluate new samples, where the norms of $u_s^{(n,m)}, v_s^{(n,m)}$ are up to the scaling l_{λ_s}. In (8), it explicitly formalizes the approximated adjoint functions (left and right singular vectors) with the asymmetric kernel κ (G).

Nyström Approximation to Asymmetric Kernel Matrices With the asymmetric Nyström approximation derived in (8), we can apply CCE to a subset of the data with sample size $n < N$ and $m < M$ to approximate the adjoint eigenfunctions at all samplings $\{x_i\}_{i=1}^N$ and $\{z_j\}_{j=1}^M$. We assume the kernel matrix to approximate from KSVD is $G \in \mathbb{R}^{N \times M}$ and denote $\tilde{\lambda}_s^{(N,M)}, \tilde{u}_s^{(N,M)}$, and $\tilde{v}_s^{(N,M)}$ as the Nyström approximation of the singular values, and left and right singular vectors of G, respectively. We then utilize the Nyström method to approximate the singular vectors of G through the out-of-sample extension (8):

$$\begin{aligned}\tilde{u}_s^{(N,M)} &= (\sqrt{\sqrt{mn} l_{\lambda_s}} / \lambda_s^{(n,m)}) G_{N,m} v_s^{(n,m)}, \\ \tilde{v}_s^{(N,M)} &= (\sqrt{\sqrt{mn} l_{\lambda_s}} / \lambda_s^{(n,m)}) G_{n,M}^\top u_s^{(n,m)},\end{aligned} \tag{9}$$

with $\tilde{\lambda}_s^{(N,M)} = (1/\sqrt{mn} l_{\lambda_s}) \lambda_s^{(n,m)}$ for $s = 1, \ldots, r$, where $u_s^{(n,m)}, v_s^{(n,m)}$ are the left and right singular vectors to the s-th nonzero singular value $\lambda_s^{(n,m)}$ of an $n \times m$ sampled submatrix $G_{n,m}$, $G_{N,m} \in \mathbb{R}^{N \times m}$ is the submatrix by sampling m columns of G, and $G_{n,M} \in \mathbb{R}^{n \times M}$ is by sampling n rows of G. More remarks on the developed asymmetric Nyström method and comparisons to the existing symmetric one are provided in Appendix A.2.2.

4. Numerical Experiments

This section aims to give a comprehensive empirical evaluation of SVD in feature spaces with asymmetric kernels in the formulation discussed above. In existing works, the potential benefits in applications remain largely unexplored w.r.t. advantages of asymmetric kernels. The following experiments do not claim that asymmetric kernels are always superior to symmetric ones as it can be problem-dependent. We consider a variety of tasks, including representation learning in directed graphs, biclustering, and downstream classification/regression on general data. A key aspect of our setup is that we can use the solutions B_ϕ, B_ψ to express the nonlinear embeddings without explicitly computing the feature mappings $\{\phi(x_i)\}_{i=1}^n$, $\{\psi(z_j\}_{j=1}^m$, which in our derivation, and differently from previous work, are allowed to be infinite-dimensional. The effectiveness of our new asymmetric Nyström method is also evaluated.

4.1. Directed Graphs

Setups Unsupervised node representation learning extracts embeddings of nodes from graph topology alone. We consider five benchmark directed graphs (Sen et al., 2008; Yang et al., 2016). KSVD is compared with its closely related baselines, i.e., PCA, SVD, and KPCA, and also with node embedding algorithms DeepWalk (Perozzi et al., 2014), a well-known random walk-based approach, HOPE (Ou et al., 2016), which preserves the asymmetric node roles with two embedding spaces using network centrality measures, and also Directed Graph Autoencoders (DiGAE) (Kollias et al., 2022). All compared methods are unsupervised and require only the adjacency matrix; note that this is different from the common setup of graph neural networks (Wu et al., 2022) that use additional node attributes on top of graph topology and operate in semi-supervised setups.

We evaluate the downstream applications of node classification and graph reconstruction. With (K)PCA and DeepWalk, we only obtain one set of embeddings. With SVD, KSVD, HOPE, and DiGAE two sets of embeddings are obtained and then concatenated. As the adjacency matrix is square, there is no compatibility issue. We compute ℓ_1, ℓ_2 norms (lower is better (↓)) for graph reconstruction and Micro- and Macro-F1 scores (higher is better (↑)) for node classification using an LSSVM classifier averaged over 10 trials on the extracted 1000 components following (Ou et al., 2016; He et al., 2023). KPCA employs the RBF kernel and KSVD employs the asymmetric kernel

$$\kappa_{\text{SNE}}(x,z) = \frac{\exp(-\|x-z\|_2^2/\gamma^2)}{\sum_{z'\in\mathcal{Z}}\exp(-\|x-z'\|_2^2/\gamma^2)},$$

also known as the SNE kernel (Hinton & Roweis, 2002), which can be seen as an asymmetric extension of RBF , and conduct 10-fold cross-validation for the kernel parameter in the same range. Detailed experimental setups are provided in Appendix C.

Results In Table 1 for node downstream classification, the results indicate consistent improvements over both SVD and KPCA, verifying the effectiveness of employing nonlinearity (to SVD) and asymmetric kernels (to KPCA). The graph reconstruction task reflects how well the extracted embeddings preserve the node connection structure. The adjacency matrix is reconstructed with the learned embeddings and then compared to the ground truth with ℓ_1, ℓ_2 norms. Asymmetric kernels greatly improve SVD, further illustrating the significance of using nonlinearity. KPCA achieves better performance than SVD, showing that considering the asymmetry alone, i.e., SVD, is not enough and nonlinearity is of great importance. Although DeepWalk, HOPE, and DiGAE are designed specifically for graphs, the simpler KSVD shows competitive performance, demonstrating great potential in representation learning for directed graphs.

4.2. Biclustering

Setups Biclustering simultaneously clusters samples and features of the data matrix, e.g., cluster documents and words. SVD has long been a common method by clustering rows and columns through right/left singular vectors. KPCA can be applied either to the rows or the columns at a time, due to its symmetry. We apply k-means to the extracted embeddings from SVD, KPCA, and KSVD. We also compare with the biclustering methods EBC (Percha & Altman, 2015), based on ensemble, and the recently proposed BCOT (Fettal et al., 2022), based on optimal transport. In the considered benchmarks (Fettal et al., 2022), the rows relate to documents, where the NMI metric can be used. The columns relate to terms, where the Coherence index is used (Dhillon et al., 2003). Other settings are as in Section 4.1 and we use a_1 for the compatibility matrix.

Results In Table 2, KSVD outputs considerably better clustering compared to KPCA, which can only perform clustering on a single data view at a time. Despite the KSVD algorithm not being specialized for this task, it consistently achieves competitive or superior performance compared to BCOT and EBC, both specifically designed for biclustering. This experiment further emphasizes the significance of asymmetric feature learning and its potential to boost the performance of downstream tasks in applications.

4.3. General Data

Setups Since asymmetric kernels are more general than symmetric ones, the features learned with asymmetric kernels can help boost performance in generic feature extraction. We evaluate KSVD on general data from UCI (Dua & Graff, 2017). First, we extract embeddings with kernel methods, and then apply a linear classifier/regressor and report results on test data (20% of the dataset). Besides SNE, we employ RBF and note that the resulting kernel matrix G in (4) is still asymmetric, as the kernel is applied to two different sets $\mathcal{X}$ and $\mathcal{Z}$, i.e. $\kappa(x_i, z_j) \neq \kappa(x_j, z_i)$. Data matrices are generally non-square, so we need the dimensionality compatibility C as in Remark 2.3. C is realized by $A^\dagger$ in

Table 1. Results on the node embedding downstream tasks with directed graph datasets.

Dataset	Node classification								Graph reconstruction							
	F1 Score (↑)	PCA	KPCA	SVD	KSVD	DeepW	HOPE	DiGAE	Norm (↓)	PCA	KPCA	SVD	KSVD	DeepW	HOPE	DiGAE
Cora	Micro	0.757	0.771	0.776	**0.792**	0.741	0.750	0.783	ℓ_1	556.0	349.0	622.0	**14.0**	19.0	15.0	26.0
	Macro	0.751	0.767	0.770	**0.784**	0.736	0.473	0.776	ℓ_2	41.2	37.9	41.7	**17.4**	17.6	18.1	20.9
Citeseer	Micro	0.648	0.666	0.667	**0.678**	0.624	0.642	0.663	ℓ_1	138.0	46.0	176.0	**25.0**	**25.0**	26.0	25.0
	Macro	0.611	0.635	0.632	**0.640**	0.587	0.607	0.627	ℓ_2	21.3	16.0	24.6	14.3	14.4	**13.3**	16.4
Pubmed	Micro	0.765	0.754	0.766	0.773	0.759	0.771	**0.781**	ℓ_1	1937.0	171.0	1933.0	**170.0**	171.0	171.0	171.0
	Macro	0.736	0.715	0.738	0.743	0.737	0.741	**0.749**	ℓ_2	128.0	31.9	118.1	23.8	**19.4**	23.8	27.9
TwitchPT	Micro	0.681	0.681	0.694	**0.712**	0.637	0.685	0.633	ℓ_1	1780.0	766.0	1839.0	**756.0**	864.0	1108.0	759.0
	Macro	0.517	0.531	0.543	**0.596**	0.589	0.568	0.593	ℓ_2	196.3	172.4	192.1	**140.3**	146.5	158.2	79.7
BlogCatalog	Micro	0.648	0.663	0.687	**0.710**	0.688	0.704	0.697	ℓ_1	5173.0	766.0	5166.0	**764.0**	810.0	3709.0	771.0
	Macro	0.643	0.659	0.673	**0.703**	0.679	0.697	0.690	ℓ_2	429.4	99.9	410.5	**94.2**	104.0	286.7	202.6

Table 2. Biclustering results of documents w.r.t. NMI and of terms w.r.t. Coherence (Coh).

Method	ACM		DBLP		Pubmed		Wiki	
	NMI	Coh	NMI	Coh	NMI	Coh	NMI	Coh
SVD	0.58	0.21	0.09	-0.06	0.31	0.42	0.39	0.42
KPCA	0.59	0.28	0.26	0.17	0.29	0.51	0.46	0.57
KSVD	**0.68**	**0.32**	**0.28**	0.21	**0.33**	0.54	**0.48**	**0.64**
BCOT	0.38	0.27	0.27	**0.22**	0.16	0.54	**0.48**	**0.64**
EBC	0.62	0.20	0.15	0.21	0.19	**0.56**	0.47	0.63

previous work (Suykens, 2016); we denote this approach a_0. We compare a_0 with our proposed approaches a_1, a_2 in unsupervised settings, and with our a_3 with learnable C, optimized by SGD on the downstream task objective.

Results In Table 3, KSVD maintains the best overall results with all alternatives a_0-a_3, showing promising potentials of applying asymmetric kernels on general data for downstream tasks. Under unsupervised setups, the alternatives a_1-a_2 for C all lead to comparable performance to the expensive pseudo-inverse a_0. For fair comparisons with learnable C, we also evaluate KPCA with optimized C, i.e., we use $\hat{\kappa}(C^\top x, C^\top x)$ in KPCA. With a_3, asymmetric kernels consistently outperform KPCA, while, for KPCA, a learnable C only provides marginally improved or comparable results. The matrix C can be viewed as a transformation for dimensionality compatibility providing additional degrees of freedom to learn enhanced embeddings.

4.4. Asymmetric Nyström Method

We evaluate the proposed asymmetric Nyström method against other standard solvers on problems of different sizes. We compare with three common SVD solvers: truncated SVD (TSVD) from the ARPACK library, the symmetric Nyström (Sym. Nys.) applied to $GG^\top$ and $G^\top G$ employing the Lanczos Method (Lehoucq et al., 1998) for the SVD subproblems, and randomized SVD (RSVD) (Halko et al., 2011). For all used solvers, we use the same stopping criterion based on achieving a target tolerance ε. The accuracy of a solution $\tilde{U} = [\tilde{u}_1, \ldots, \tilde{u}_r], \tilde{V} = [\tilde{v}_1, \ldots, \tilde{v}_r]$, is evaluated as the weighted average $\eta = \frac{1}{r}\sum_{i=1}^r w_i(1 - |u_i^\top \frac{\tilde{u}_i}{\|\tilde{u}_i\|}|) + \frac{1}{r}\sum_{i=1}^r w_i(1 - |v_i^\top \frac{\tilde{v}_i}{\|\tilde{v}_i\|}|)$, with $w_i = \lambda_i$ and $U = [u_1, \ldots, u_r], V = [v_1, \ldots, v_r]$ the left and right singular vectors of G from its rank-r truncated SVD. The stopping criterion for all methods is thus $\eta \leq \varepsilon$. This criterion is meaningful in feature learning tasks as the aim is to learn embeddings of the given data as scalar products with the singular vectors, rather than approximating the full kernel matrix. We use random subsampling for all Nyström methods and increase the number of subsamples m to achieve the target ε, where we use $m = n$ as the kernel matrices are square; we employ the SNE kernel and set $r = 20$.

Table 4 shows the algorithm running time at tolerance level $\varepsilon = 10^{-1}$. We also show the speedup w.r.t. RSVD, i.e., $t^{(\text{RSVD})}/t^{(\text{Ours})}$, where $t^{(\text{RSVD})}, t^{(\text{Ours})}$ denote the training time of RSVD and our asymmetric Nyström solver. Our solver shows to be the fastest and our improvement is more significant with larger problem sizes. In Appendix B.2, we present the results at tolerance level $\varepsilon = 10^{-2}$, also verifying our advantages. Further, we consider that a solver's performance may depend on the singular spectrum of the kernel. We vary the bandwidth γ of the SNE kernel on Cora to assess how the singular value decay of the kernel matrix affects performance, where an increased γ leads to spectra with faster decay, and vice versa.

In Fig. 4, we vary γ and show the required subsamples m to achieve the given tolerance and the runtime speedup w.r.t. RSVD. Our method shows overall speedup to RSVD, and our asymmetric Nyström requires significantly fewer subsamples on matrices with faster singular spectrum decay, showing greater speedup in this scenario. In Fig. 5, the node classification F1 score (Macro) is reported for several values of subsamples m, where KSVD employs the asymmetric Nyström method and KPCA uses the symmetric Nyström on the same RBF kernel. It shows superior performances of the asymmetric method at all considered m without significant accuracy decrease due to the subsampling. Additional results are provided in Appendix B.

Table 3. Downstream task results on general UCI datasets, where the best results are in bold.

Dataset	Metric	KPCA (RBF)	KPCA (RBF Learnable C)	KSVD (RBF)				KSVD (SNE)			
				a_0	a_1	a_2	Learnable C (a_3)	a_0	a_1	a_2	Learnable C (a_3)
Diabetes	ACC (↑)	0.67	0.68	0.66	0.67	0.67	**0.71**	0.66	0.65	0.67	**0.69**
Ionosphere	ACC (↑)	0.67	0.68	0.68	0.69	0.69	**0.70**	0.70	0.67	0.68	**0.72**
Liver	ACC (↑)	0.71	0.72	0.74	0.70	0.71	**0.76**	0.71	0.72	0.70	**0.75**
Cholesterol	RMSE (↓)	59.36	53.14	54.53	52.80	55.24	**45.91**	49.11	47.83	48.12	**44.40**
Yacht	RMSE (↓)	15.85	14.05	14.90	14.74	16.57	**12.98**	15.41	14.18	13.89	**13.05**
Physicochemical-protein	RMSE (↓)	5.96	5.96	5.94	5.96	6.01	**5.90**	5.96	6.03	6.00	**5.93**

Table 4. Runtime for multiple KSVD problems at tolerance $\epsilon = 10^{-1}$; the lowest solution time is in bold.

Task	N	Time (s)				
		TSVD	RSVD	Sym. Nys.	Ours	Speedup
Cora	2708	0.841	0.274	0.673	**0.160**	1.71×
Citeseer	3312	0.568	0.290	0.214	**0.136**	2.14×
PubMed	19717	9.223	4.577	44.914	**0.141**	32.51×

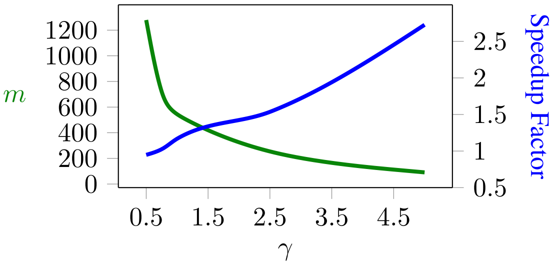


Figure 4. **Varying singular spectrum.** Number of samples m (green) to achieve a fixed tolerance and the speedup factor w.r.t. RSVD (blue) on Cora when the spectrum of G varies (larger γ leads to faster decay).

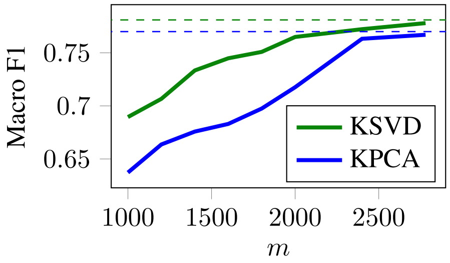


Figure 5. **Effect of m.** Performance on Cora at different m by asymmetric Nyström. Dashed lines indicate the exact solution.

5. Conclusion

This work presents a novel learning scheme for asymmetric learning in feature spaces. We establish that the solution to the coupled covariances eigenproblem (CCE) can be obtained by performing SVD on an asymmetric kernel matrix, providing a new perspective on KSVD grounded in covariance operators. In addition, the resulting computations can be sped up on large-scale problems, thanks to the formally derived asymmetric Nyström method. Numerical results show the potential of the retained asymmetry and nonlinearity realized in KSVD and the effectiveness of the developed asymmetric Nyström method. The insights and methodologies in this work pave the way for further exploration of asymmetric kernel methods in machine learning.

Acknowledgements

This work is jointly supported by ERC Advanced Grant E-DUALITY (787960), iBOF project Tensor Tools for Taming the Curse (3E221427), KU Leuven Grant CoE PFV/10/002, and Grant FWO G0A4917N, EU H2020 ICT-48 Network TAILOR (Foundations of Trustworthy AI - Integrating Reasoning, Learning and Optimization), and Leuven.AI Institute. This work was also supported by the Research Foundation Flanders (FWO) research projects G086518N, G086318N, and G0A0920N; Fonds de la Recherche Scientifique — FNRS and the Fonds Wetenschappelijk Onderzoek — Vlaanderen under EOS Project No. 30468160 (SeLMA). We thank the anonymous reviewers for constructive comments.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Aronszajn, N. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

Baker, C. T. The numerical treatment of integral equations. *SIAM Review*, 23(2):266, 1981.

Chang, C.-C. and Lin, C.-J. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

Chen, Y., Tao, Q., Tonin, F., and Suykens, J. A. Primal-attention: Self-attention through asymmetric kernel svd in primal representation. *Advances in Neural Information Processing Systems*, 2023.

Dhillon, I. S., Mallela, S., and Modha, D. S. Information-theoretic co-clustering. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Dining*, pp. 89–98, 2003.

Drineas, P., Mahoney, M. W., and Cristianini, N. On the Nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6(12), 2005.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Estrada, E. *The structure of complex networks: theory and applications*. Oxford University Press, USA, 2012.

Fettal, C., Nadif, M., et al. Efficient and effective optimal transport-based biclustering. *Advances in Neural Information Processing Systems*, 35:32989–33000, 2022.

Gittens, A. and Mahoney, M. W. Revisiting the Nyström method for improved large-scale machine learning. *Journal of Machine Learning Research*, 17(1):3977–4041, 2016.

Golub, G. H. and Van Loan, C. F. *Matrix Computations*. JHU press, 2013.

Halko, N., Martinsson, P. G., and Tropp, J. A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

He, M., He, F., Shi, L., Huang, X., and Suykens, J. A. K. Learning with asymmetric kernels: Least squares and feature interpretation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45:10044–10054, 2023.

Hinton, G. and Roweis, S. T. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems*, volume 15, pp. 833–840, 2002.

Jolliffe, I. T. *Principal Component Analysis*. Springer, 1986.

Kluger, Y., Basri, R., Chang, J. T., and Gerstein, M. Spectral biclustering of microarray data: coclustering genes and conditions. *Genome Research*, 13(4):703–716, 2003.

Koide, N. and Yamashita, Y. Asymmetric kernel method and its application to Fisher's discriminant. In *International Conference on Pattern Recognition*, volume 2, pp. 820–824, 2006.

Kollias, G., Kalantzis, V., Idé, T., Lozano, A., and Abe, N. Directed graph auto-encoders. In *AAAI Conference on Artificial Intelligence*, volume 36, pp. 7211–7219, 2022.

Kuruwita, C., Kulasekera, K., and Padgett, W. Density estimation using asymmetric kernels and bayes bandwidths with censored data. *Journal of Statistical Planning and Inference*, 140(7):1765–1774, 2010.

Lai, P. L. and Fyfe, C. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10(05):365–377, 2000.

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

Larsen, K. G. and Nelson, J. Optimality of the Johnson-Lindenstrauss lemma. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science*, pp. 633–638, 2017.

Lehoucq, R. B., Sorensen, D. C., and Yang, C. *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*. SIAM, 1998.

Mackenzie, M. and Tieu, A. K. Asymmetric kernel regression. *IEEE transactions on neural networks*, 15(2): 276–282, 2004.

Meanti, G., Carratino, L., Rosasco, L., and Rudi, A. Kernel methods through the roof: handling billions of points efficiently. *Advances in Neural Information Processing Systems*, 33:14410–14422, 2020.

Michaeli, T., Wang, W., and Livescu, K. Nonparametric canonical correlation analysis. In *International Conference on Machine Learning*, pp. 1967–1976. PMLR, 2016.

Muñoz, A., Martín de Diego, I., and Moguerza, J. M. Support vector machine classifiers for asymmetric proximities. In *International Conference on Artificial Neural Networks*, pp. 217–224. Springer, 2003.

Nemtsov, A., Averbuch, A., and Schclar, A. Matrix compression using the nyström method. *Intelligent Data Analysis*, 20(5):997–1019, 2016.

Neto, E. d. A. L. and Rodrigues, P. C. Kernel robust singular value decomposition. *Expert Systems with Applications*, 211:118555, 2023.

Oglic, D. and Gaertner, T. Learning in reproducing kernel Krein spaces. In *International Conference on Machine Learning*, 2018.

Ou, M., Cui, P., Pei, J., Zhang, Z., and Zhu, W. Asymmetric transitivity preserving graph embedding. In *ACM International Conference on Knowledge Discovery and Data Mining*, pp. 1105–1114, 2016.

Percha, B. and Altman, R. B. Learning the structure of biomedical relationships from unstructured text. *PLoS Computational Biology*, 11(7):e1004216, 2015.

Perozzi, B., Al-Rfou, R., and Skiena, S. Deepwalk: Online learning of social representations. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 701–710, 2014.

Schmidt, E. Zur theorie der linearen und nichtlinearen integralgleichungen. *Mathematische Annalen*, 63(4):433–476, 1907.

Schölkopf, B., Smola, A., and Müller, K.-R. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10(5):1299–1319, July 1998. ISSN 0899-7667. doi: 10.1162/089976698300017467.

Schölkopf, B., Mika, S., Burges, C. J., Knirsch, P., Muller, K.-R., Ratsch, G., and Smola, A. J. Input space versus feature space in kernel-based methods. *IEEE transactions on Neural Networks*, 10(5):1000–1017, 1999.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., and Eliassi-Rad, T. Collective classification in network data. *AI Magazine*, 29(3):93–93, 2008.

Stewart, G. W. On the early history of the singular value decomposition. *SIAM Review*, 35(4):551–566, 1993.

Strang, G. *Linear algebra and its applications.* Belmont, CA: Thomson, Brooks/Cole, 2006.

Suykens, J. A. SVD revisited: A new variational principle, compatible feature maps and nonlinear extensions. *Applied and Computational Harmonic Analysis*, 40(3): 600–609, 2016.

Tsuda, K. Support vector classifier with asymetric kernel function. In *European Symposium on Arti- ficial Neural Networks*, pp. 183–188, 1998.

Vasilescu, M. A. O. *A Multilinear (Tensor) Algebraic Framework for Computer Graphics, Computer Vision and Machine Learning*. PhD thesis, University of Toronto, 2009.

Williams, C. and Seeger, M. Using the Nyström method to speed up kernel machines. In Leen, T., Dietterich, T., and Tresp, V. (eds.), *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2000.

Wright, M. A. and Gonzalez, J. E. Transformers are deep infinite-dimensional non-mercer binary kernel machines. *arXiv preprint arXiv:2106.01506*, 2021.

Wu, L., Cui, P., Pei, J., Zhao, L., and Guo, X. Graph neural networks: foundation, frontiers and applications. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4840–4841, 2022.

Wu, W., Xu, J., Li, H., and Oyama, S. Asymmetric kernel learning. *Technical Report, Microsoft Research*, 2010.

Xiong, Y., Zeng, Z., Chakraborty, R., Tan, M., Fung, G., Li, Y., and Singh, V. Nyströmformer: A Nyström-based algorithm for approximating self-attention. In *AAAI Conference on Artificial Intelligence*, volume 35, pp. 14138–14148, 2021.

Xu, Y. and Ye, Q. *Generalized Mercer kernels and reproducing kernel Banach spaces*, volume 258. American Mathematical Society, 2019.

Yang, Z., Cohen, W., and Salakhudinov, R. Revisiting semi-supervised learning with graph embeddings. In *International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 40–48, New York, New York, USA, 2016. PMLR.

Zhang, H., Xu, Y., and Zhang, J. Reproducing kernel Banach spaces for machine learning. *Journal of Machine Learning Research*, 10(12), 2009.

Zhang, K., Tsang, I. W., and Kwok, J. T. Improved nyström low-rank approximation and error analysis. In *International Conference on Machine Learning*, pp. 1232–1239, 2008.

A. Further comparisons with related work

A.1. KSVD and discussions with related work

Our main interest in this work is to derive a new formulation for KSVD, to promote more insights into nonlinear feature learning with considerations to asymmetry. We start from a new asymmetric learning paradigm based on coupled covariances eigenproblem (CCE) and show that the solutions to CCE leads to the KSVD problem associated with a specific asymmetric similarity matrix that blends in two feature maps. Our formulations involve two covariance operators allowing to work with infinite-dimensional feature mappings with induced asymmetric kernels, aiming to provide a vigorous formalization equipped with interpretations w.r.t. both the covariance matrix and kernel matrix. KSVD attains an asymmetric kernel matrix G simultaneously coupling two sets of mapping information, which is intrinsically different from KPCA. Through this work, we would also like to convey that although the solutions of PCA and KPCA can be computed numerically by the linear algebra tool of SVD, PCA is essentially different from SVD, and so is KPCA from KSVD.

The solution of KSVD leads (4) in terms of an asymmetric kernel G instead of the given data matrix A, and is therefore related to the compact SVD as a solution to (4). (Suykens, 2016) revisits the compact matrix SVD with a variational principle under the setups of least squares support vector machines (LSSVM), where the dual solution leads to a shifted eigenvalue problem regarding the given data matrix A. It focuses on the (linear) matrix SVD; although it mentions the possibility with nonlinearity by transforming A into some asymmetric kernel matrix of the same size, it cannot deal with infinite-dimensional feature spaces nor nor connect to the covariances, where it neither formalizes the derivations to the kernel trick, nor mentions possible applications with any experimental evaluations. In (Chen et al., 2023), the asymmetric self-attention is remodelled for low-rank properties through the finite-dimensional feature mappings with neural networks.The queries and keys are regarded as two data sources and directly tackle the self-attention by applying the variational objective proposed in (Suykens, 2016) as an auxiliary regularization loss into the optimization objective, which is iteratively minimized to approach zero and cannot provide the singular vectors nor singular values. In the early work of Schmidt (Schmidt, 1907), the shifted eigenvalue problem is also discussed w.r.t. the integral equations regarding a pair of adjoint eigenfunctions in the continuous cases with function spaces. Hence, we can see that there can be multiple frameworks that can lead to a solution in the form resembling a shifted eigenvalue problem either on the given data matrix or an asymmetric kernel matrix as derived in KSVD, whereas different goals are pertained in the addressed scenarios and the methodologies are also varied with different optimization objectives and interpretations.

Moreover, to get the terminology of KSVD clearer, we additionally discuss the differences to a few other existing works that share some similarities in naming the methodology. In (Neto & Rodrigues, 2023), it considers a new algorithm for SVD that incrementally estimates each set of robust singular values and vectors by replacing the Euclidean norm with the Gaussian norm in the objective. Different from kernel-based methods, (Neto & Rodrigues, 2023) operates in the original space, not in the feature space, where the kernel is only used in the objective for the estimator and the data are not processed with any nonlinearity in the feature space. Despite the similarity in names, the tasks and methodologies in (Neto & Rodrigues, 2023) and KSVD are intrinsically different. In (He et al., 2023), it presents how to apply asymmetric kernels with LSSVMs for supervised classification with both input samples and their labels, and is derived with finite-dimensional feature spaces. In particular, unlike our construction with $\mathcal{X}$ and $\mathcal{Z}$, (He et al., 2023) can only consider a single data set under the context of its supervised task, exploring the supervised learning for the row data and possibly missing full exploitation of the asymmetry residing in the data. Accordingly, the asymmetry in (He et al., 2023) only comes from the choice of the asymmetric kernel function, while our asymmetry also comes from jointly handling two different sets. In (Vasilescu, 2009), KPCA is extended to tensor data to analyze the factors w.r.t. each mode of the tensor, where SVD is applied to solve the eigendecomposition of the KPCA problem in each mode and the left singular vectors (i.e., eigenvectors) are obtained as the nonlinear factor for each mode. (Vasilescu, 2009) still only considers the symmetry in feature learning but extend it to higher-order tensors. Hence, the data processing, the kernel-based learning scheme, the optimization framework, and also the task are all different from the ones considered in the present work.

A.2. Asymmetric Nyström method and related work

A.2.1. Background

The existing Nyström method starts from the numerical treatment of an integral equation with a symmetric kernel function $\hat{\kappa}(\cdot,\cdot)$ such that $\lambda u(x) = \int_a^b \hat{\kappa}(x,z)u(x)\,dx$, i.e., the continuous analogue to the eigenvalue problem, where the quadrature technique can be applied to formulate the discretized approximation (Baker, 1981). Concerning the more general cases with multivariate inputs, the probability density function and the empirical average technique of finite sampling have been utilized to compute the approximated eigenfunctions that correspond to the eigenvectors (Baker, 1981; Schölkopf et al., 1999). To better illustrate the differences to the established asymmetric Nyström, we provide more details on the symmetric Nyström method for reference, based on the derivations from (Williams & Seeger, 2000).

Given the i.i.d. samples $\{x_1, \ldots, x_q\}$ from the probability density $p_x(x)$ over $\mathcal{D}_x$, an empirical average is used to approximate the integral of the eigenfunction with a symmetrick kernel:

$$\lambda_s u_s(x) = \int_{\mathcal{D}_x} \hat{\kappa}(x, z) u_s(x) p_x(x)\, dx \approx \frac{1}{q} \sum\nolimits_{i=1}^{q} \hat{\kappa}(x, x_i) u_s(x_i), \tag{10}$$

where u_s is said to be an eigenfunction of $\hat{\kappa}(\cdot, \cdot)$ corresponding to the eigenvalues with $\lambda_1 \geq \lambda_2 \geq \ldots \geq 0$. By running x in (10) at $\{x_1, \ldots, x_q\}$, an eigenvalue problem is motivated, such that $G^{(q)} U^{(q)} = U^{(q)} \Lambda^{(q)}$, where $G^{(q)} \in \mathbb{R}^{q \times q}$ is the Gram matrix with $G_{ij}^{(q)} = \hat{\kappa}(x_i, x_j)$ for $i, j = 1, \ldots, N$, $U^{(q)} = [u_1^{(q)}, \ldots, u_q^{(q)}] \in \mathbb{R}^{q \times q}$ is column orthonormal and the diagonal matrix $\Lambda^{(q)} \in \mathbb{R}^{q \times q}$ contains the eigenvalues such that $\lambda_1^{(q)} \geq \ldots \geq \lambda_q^{(q)} \geq 0$. In this case, the approximation of eigenvalues and eigenfunction from the integral equation (10) arrives at:

$$\lambda_s \approx \frac{\lambda_s^{(q)}}{q}, \quad u_s(x_i) \approx \sqrt{q} U_{i,s}^{(q)}, \tag{11}$$

which can be plugged back to (10), leading to the Nyström approximation to the i-th eigenfunction:

$$u_s(x) \approx \frac{\sqrt{q}}{\lambda_s^{(q)}} \sum_{i=1}^{q} \hat{\kappa}(x, x_i) U_{i,s}^{(q)}, \tag{12}$$

with $\forall s : \lambda_s^{(q)} > 0$. With the Nyström technique in (12), one can use different sampling sets to approximate the integral (10). Thus, given a larger-scale Gram matrix $G^{(N)} \in \mathbb{R}^{N \times N}$, for the first p eigenvalues and eigenfunctions, a subset of training data $q \triangleq n < N$ can be utilized to attain their approximation at all N points for the kernel matrix $G^{(N)}$ with (11):

$$\tilde{\lambda}_s^{(N)} \triangleq \frac{N}{n} \lambda_s^{(n)}, \quad \tilde{u}_s^{(N)} \triangleq \sqrt{\frac{n}{N}} \frac{1}{\lambda_s^{(n)}} G_{N,n} u_s^{(n)}, \tag{13}$$

where $\tilde{\lambda}_s^{(N)}$ and $\tilde{u}_s^{(N)}$ are the Nyström approximation of the eigenvalues and eigenvectos of $G^{(N)}$. Here $u_s^{(n)}$ are eigenvectors corresponding to the s-th eigenvalues $\lambda_s^{(n)}$ of an $n \times n$ submatrix $G_{n,n}$ and $G_{N,n}$ is the submatrix by sampling n columns of $G^{(N)}$.

A.2.2. Discussions

We provide the following remarks elaborating on the existing Nyström method w.r.t. the eigenvalue problem for Mercer kernels and our extended Nyström method w.r.t. the SVD problem for asymmetric kernels.

1. **Integral equations.** As shown in Section A.2.1 above, the existing Nyström method starts from a single integral equation with a symmetric kernel $\hat{\kappa}(\cdot, \cdot)$, corresponding to an eigenvalue problem in the discretized scenarios (Baker, 1981; Williams & Seeger, 2000). Thus, the existing Nyström method is derived only for Mercer kernels with symmetry constraints on the tackled matrix. Differently, the proposed asymmetric Nyström method deals with an asymmetric kernel $\kappa(\cdot, \cdot)$ and starts from a pair of adjoint eigenfunctions, which jointly determine an SVD problem in the discretized scenarios (Schmidt, 1907; Stewart, 1993). In (Drineas et al., 2005; Nemtsov et al., 2016), the matrix compression is discussed with Nyström-like methods to general matrices. However, the method in (Nemtsov et al., 2016) is formulated to approximate subparts of the left and right singular vectors, and still applies the symmetric Nyström method to heuristically approximate the asymmetric submatrix twice for the corresponding subparts; (Drineas et al., 2005) directly applies the symmetric Nyström method and resembles its formulas to approximate left and right singular vectors of general matrices, ignoring the asymmetry constraints. Rather than working with singular vectors, (Xiong et al., 2021) utilizes the technique in (Drineas et al., 2005) to the submatrix blocks to approximate a surrogate attention matrix in Transformers for computation efficiency. Hence, the analytical framework of the asymmetric Nyström method has not been formally formulated yet. In our paper, the explicit rationale of leveraging the Nyström technique is

provided for the asymmetric matrices through the finite sample approximation to the pair of adjoint eigenfunction, which incorporates the asymmetry constraints on the tackled matrix, so that from analytical and practical aspects it becomes viable to directly apply the asymmetric Nyström method to the cases that pertain the asymmetric nature.

2. **Special case with symmetry.** In the derivations on the finite sample approximation, three scalings l_{λ_s}, l_{u_s}, and l_{v_s} are introduced to the singular values λ_s, right singular vectors $u_s(x)$, and left singular vectors $v_s(z)$ in Eq. (7) in Section 4 in the paper, for the considerations on their norms. Meanwhile, the constant coefficients in the two equations in Eq. (8) in the paper are required to be the same in scalings to proceed the derivations that match the SVD problem. In the existing symmetric Nyström method, the scaling issue of the approximated eigenfunction does not appear with $l_{\lambda_s}\lambda l_{u_s}u(x) = \int \hat{\kappa}(x,z)l_{u_s}u(x)p_x(x)\,dx$, as the scaling l_{u_s} is cancelled out in the two sides of this equation, i.e., the Eq. (10) above. Thus, in (10) it implicitly sets the scaling of the eigenvalue as $l_{\lambda_s} = 1$ (Williams & Seeger, 2000), while in (13) l_{λ_s} is set as $1/N$ in the application of the Nyström method to speedup the eigenvalue problem on a larger Gram matrix $G^{(N)}$.

   Note that, for feature learning, we only need to find the singular vectors in Eq. (10) or (11) in the paper, which are taken as embeddings of the given data for downstream tasks. The computation of the singular values can be omitted, so that we can simply implement the scaling through normalization in practice. The numerical computation of the approximated kernel matrix is also not necessary for the considered feature learning tasks. When considering the special case where the kernel matrix G in KSVD is square ($N = M$) and symmetric ($G = G^\top$), the numbers of samplings to the rows and column are the same ($n = m$), and the scaling l_{λ_s} is set the same, the asymmetric Nyström method boils down to the existing Nyström method.

3. **Another alternative derivation.** We consider an asymmetric kernel function $\kappa(x,y)$, and define the induced kernel operator and its adjoint by

$$\begin{aligned}(Gg)(x) &= \mathbb{E}_{p_x(x)}[\kappa(x,Y)g(Y)],\\ (G^* f)(y) &= \mathbb{E}_{p_y(y)}[\kappa(X,y)f(X)],\end{aligned} \tag{14}$$

   for L^2-integrable functions f and g, where we denote the two datasets in the matrix form by arranging the samples row-wisely in X and Y, respectively. Then, the left and right s-th singular functions $u_s(\cdot)$ and $v_s(\cdot)$ of the kernel operator $\kappa(x,y)$ satisfy

$$\begin{aligned}(G^* u_s)(y) &= \lambda_s v_s(y),\\ (Gv_s)(x) &= \lambda_s u_s(x).\end{aligned} \tag{15}$$

   Given n samples $x_1,\dots,x_n$ drawn from $p_x(x)$ and m samples $y_1,\dots,y_m$ drawn from $p_y(y)$, the relations can be approximated as

$$\begin{aligned}v_s(y) &= \frac{1}{\sigma_s}(G^* u_s)(y) \approx \frac{1}{n\lambda_s}\sum_{i=1}^{n}\kappa(x_i,y)u_s(x_i),\\ u_s(x) &= \frac{1}{\sigma_s}(Gv_s)(x) \approx \frac{1}{m\lambda_s}\sum_{j=1}^{m}\kappa(x,y_j)v_s(y_j).\end{aligned} \tag{16}$$

   As $G = [\kappa(x_i,y_j)] = U\Lambda V^T \in \mathbb{R}^{n\times m}$, we then scale the kernel matrix by $1/\sqrt{mn}$, and the left and right singular vectors by $1/\sqrt{n}$ and $1/\sqrt{m}$, respectively, yielding the approximated estimates of the pair of adjoint eigenfunctions:

$$\begin{aligned}v_s^{(n,m)}(y) &\approx \frac{1}{n\lambda_s}\sum_{i=1}^{n}\kappa(x_i,y)U_{is}^{(n,m)},\\ u_s^{(n,m)}(x) &\approx \frac{1}{m\lambda_s}\sum_{j=1}^{m}\kappa(x,y_j)V_{js}^{(n,m)},\end{aligned} \tag{17}$$

   such that

$$\begin{aligned}v_s(y) &\approx \frac{1}{n\lambda_s}\sum_{i=1}^{n}\kappa(x_i,y)u_s(x_i) \Rightarrow \frac{\sqrt{mn}}{N\lambda_s}\sum_{i=1}^{n}\kappa(x_i,y_j)\sqrt{n}U_{is} \approx \sqrt{m}V_{js} \Rightarrow \frac{1}{\lambda_s}\sum_{i=1}^{n}\kappa(x_i,y_j)U_{is} \approx V_{js},\\ u_s(x) &\approx \frac{1}{m\lambda_s}\sum_{j=1}^{m}\kappa(x,y_j)v_s(y_j) \Rightarrow \frac{\sqrt{mn}}{m\lambda_s}\sum_{j=1}^{m}\kappa(x_i,y_j)\sqrt{m}V_{js} \approx \sqrt{N}U_{is} \Rightarrow \frac{1}{\lambda_s}\sum_{j=1}^{m}\kappa(x_i,y_j)V_{js} \approx U_{is},\end{aligned} \tag{18}$$

which indeed correspond to matrix SVD in (7).

Note that while this alternative above can also derive the asymmetric Nyström method, it is different from the techniques presented in Section 3. In contrast, the derivation in (3) starts from the integral equations of the pair of adjoint eigenfunctions with asymmetric kernels. One of our goals is to align and compare w.r.t. the symmetric Nyström in (Williams & Seeger, 2000), which is widely adopted in machine learning, which views the Nyström approximation (Baker, 1981) originally from the integral equations with symmetric kernels, as presented in Section A.2 in the Appendix, where thorough comparisons on the connections and differences are discussed.

B. Additional numerical results

B.1. Additional ablations on KSVD

To further study the effect of simultaneous nonlinearity and asymmetry in KSVD, we design the following experiment. We first make some non-linear encoding in a preprocessing step to the samples x_i (i.e., rows of the given data matrix A) and then compute SVD, and compare the downstream classification/regression results with SVD on the asymmetric kernel matrix. Specifically, we consider polynomial features with degree 2 of the samples x_i as $\varphi(x_i)$ and then apply SVD to $\varphi(A) = [\varphi(x_1), \ldots, \varphi(x_N)]^\top$ as $\varphi(A) = U_A \Sigma_A V_A^\top$ and use U_A as the learned embeddings. Correspondingly, KSVD employs the polynomial kernel of degree 2 $k_{\text{poly}}(x, z) = (x^\top z + 1)^2$ and applies SVD to the asymmetric kernel matrix $G_{ij} = k(x_i, z_j)$ and we use the singular vectors B_ϕ as the learned embeddings for fair comparisons. The embeddings are then fed to a linear classifier/regressor for the downstream classification/regression tasks as in Section 4.3 in the main paper.

Table 5. Ablation study on SVD applied after nonlinear preprocessing v.s. KSVD. Higher values (↑) are better for AUROC and lower values (↓) are better for RMSE.

Method	AUROC (↑)			RMSE (↓)		
	Diabetes	Ionosphere	Liver	Cholesterol	Yacht	Physicochemical-protein
Nonlinear+SVD	0.6296	0.7292	0.7032	**49.0867**	15.0002	5.9517
KSVD	**0.7607**	**0.8374**	**0.7100**	49.1592	**14.6489**	**5.4583**

This experiment shows the additional benefit brought by the construction on row space and column space, as $\mathcal{X}, \mathcal{Z}$ in our derivations, and with the asymmetric kernel trick, instead of simply applying SVD to a matrix which is attained by applying some nonlinear transformation to the rows of the data matrix A. In fact, our experiments show that KSVD is an effective tool to learn more informative embeddings when the given data physically present asymmetric similarities as in Sections 4.1 and 4.2 in the main paper, and it also shows better performance for general datasets as experimented in Section 4.3 in the main paper.

B.2. Additional results on the asymmetric Nyström method

In Figure 6 in the main paper, the node classification F1 score is reported for multiple number of subsamplings m, where KSVD (green line) employs the asymmetric Nyström method and KPCA (blue line) uses the symmetric Nyström, both employing the RBF kernel. Note that, as explained in the main paper, the resulting kernel matrix G in KSVD maintains the asymmetry even with the (symmetric) RBF function, as the kernel is applied to two different inputs, i.e., $\mathcal{X}$ and $\mathcal{Z}$. Note that the data matrix is square, so we can set $m = n$ for the subsamplings of the asymmetric Nyström. In addition, we provide the corresponding Micro F1 scores on Cora and also add the evaluations on Citeseer and Pubmed. The asymmetric Nyström-based kernel method KSVD shows superior performances at all considered m compared to KPCA without significant decrease in accuracy of the solution due to the subsampling.

In Table 6, we provide extensional results on Table 4 for the tolerance levels $\varepsilon = 10^{-1}$ and 10^{-2}, showing the training time and the speedup w.r.t. RSVD, i.e. $t^{(\text{RSVD})}/t^{(\text{Ours})}$, where $t^{(\text{RSVD})}, t^{(\text{Ours})}$ is the training time of RSVD and our asymmetric Nyström solver, respectively. Our solver maintain the fastest than the compared solvers and our improvement is more significant with larger problem sizes.

We further experiment on large-scale datasets with millions of samples and features in Table 7 below, showing the classification performance (AUROC) of KPCA/KSVD with RBF with subsampling $m = 1000$, where N is the number of samples and M is the number of variables. We employ alternative a_2 for the compatibility matrix C. In Table 7, KSVD

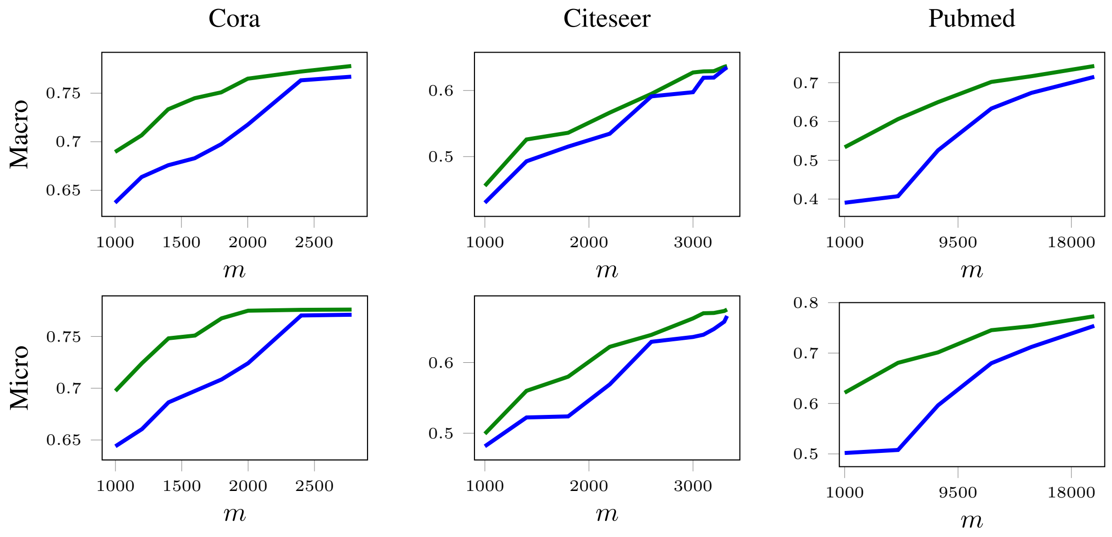


Figure 6. F1 Scores at different numbers of subsampling m with the asymmetric and symmetric Nyström method. Green line: KSVD, blue line: KPCA.

achieves the best performance also in real-world large datasets, further verifying the effectiveness and scalability.

Table 6. Runtime for multiple KSVD problems at different tolerances; the lowest solution time is in bold.

Task	N	Time (s) for $\varepsilon = 10^{-1}$					Time (s) for $\varepsilon = 10^{-2}$				
		TSVD	RSVD	Sym. Nys.	Ours	Speedup	TSVD	RSVD	Sym. Nys.	Ours	Speedup
Cora	2708	0.841	0.274	0.673	**0.160**	1.71×	0.841	0.313	0.681	**0.225**	1.39×
Citeseer	3312	0.568	0.290	0.214	**0.136**	2.14×	0.568	0.396	0.425	**0.239**	1.66×
PubMed	19717	9.223	4.577	44.914	**0.141**	32.51×	9.223	5.209	53.297	**0.590**	8.83×

Table 7. Classification results with AUROC metric (↑) on large-scale real-world datasets from (Chang & Lin, 2011).

Dataset	N	M	KPCA	KSVD
AmazonCat-13K	1,186,239	203,882	0.51	**0.55**
Avazu	14,596,137	1,000,000	0.52	**0.67**
Criteo	45,840,617	1,000,000	0.54	**0.63**

C. Experimental details

Details of the experimental setups are provided below. Experiments in Sections 4.1 and 4.2 are implemented in MATLAB 2023b, and Python 3.7 is used in Section 4.3. Experiments are run on a PC with an Intel i7-8700K and 64GB RAM, and experiments in Section 4.3 use a single NVIDIA GeForce RTX 2070 SUPER GPU.

C.1. Feature learning experiments

In the experiments, we conduct 10-fold cross validation for determining kernel hyperparameters with grid searches in the same range for fair comparisons. The employed nonlinear kernels in the experiments are $\hat{\kappa}_{\text{RBF}}(x,z) = \exp(-\frac{\|x-z\|_2^2}{\gamma^2})$ and $\kappa_{\text{SNE}}(x,z) = \frac{\exp(-\|x-z\|_2^2/\gamma^2)}{\sum_{z' \in \mathcal{Z}} \exp(-\|x-z'\|_2^2/\gamma^2)}$ with hyperparameter γ. In the node classification experiments, we denote $A =: X = [x_1, \ldots, x_N]^\top$ as the asymmetric adjacency matrix with X_{ij} as the directed similarity between node i and node j. KPCA is conducted for feature extraction in the following way: we compute symmetric kernel matrix $\hat{G}$ s.t. $\hat{G}_{ij} = \hat{k}(x_i, x_j)$, with (symmetric) RBF kernel $\hat{k}$, and its top 1000 eigenvectors are taken as the extracted features taken as input to the

Table 8. Descriptions of the tested directed graph datasets.

Datasets	Cora	Citeseer	Pubmed	TwitchPT	BlogCatalog
# Nodes	2708	3327	19717	1,912	10,312
# Edges	5429	4732	44338	64,510	333,983
# Classes	7	6	3	2	39

Table 9. Descriptions of the tested datasets for biclustering (Fettal et al., 2022).

Datasets	ACM	DBLP	Pubmed	Wiki
# Documents	3025	4057	19717	2405
# Terms	1870	334	500	4973
# Document clusters	3	4	3	17
# Term clusters	18	2	3	23

LSSVM classifier, following (He et al., 2023). PCA is conducted similarly by taking the linear kernel $\hat{k}(x_i, x_j) = x_i^\top x_j$. For all methods, we employ an LSSVM classifier with regularization parameter set to 1 and we utilize the one-vs-rest scheme. We use the original implementations of the authors for all baselines and the best parameters reported in their papers. Graph reconstruction is a typical task in node representation learning and is helpful to evaluate how well the learned representations preserve neighborhood information in embedding space. Graph reconstruction reconstructs all existing edges by reconstructing the full adjacency matrix from embedding space. In this task, with the feature embeddings extracted by all tested methods, we recover the matrix that reflects the edges between nodes and then the connections between each node. For a given node v with the out-degree k_v, the closest k_v nodes to v in feature space are searched to reconstruct the adjacency matrix. The ℓ_1, ℓ_2 norms between X and its reconstruction are evaluated.

In biclustering tasks, the closely related baseline methods, i.e., SVD and KPCA, are compared with KSVD, where the kernel setup is the same as above. Specifically, we apply SVD and KSVD on the data matrix with attained left and right singular vectors and then k-means is adopted for performing the biclustering task with extracted features, where we use the scikit-learn in Python to implement k-means. We note that as KPCA only works with symmetric kernels, so KPCA is applied twice. We also compare with the biclustering method EBC (Percha & Altman, 2015) based on ensemble and the recently proposed BCOT method (Fettal et al., 2022) based on optimal transport. We follow the data setups and evaluations in (Fettal et al., 2022) with official sources in `https://github.com/chakib401/BCOT`: the rows relate to the clustering of documents, where the ground truth can be compared through the popular clustering metric NMI; the columns relate to the clustering of terms, where the Coherence index is used (Dhillon et al., 2003). For the compatibility matrix C, we use alternative a_1. The results of BCOT are taken from its orignal paper (Fettal et al., 2022), and EBC are ran by its official codes provided in `https://github.com/blpercha/ebc` with threshold 10^{-4}. On the tested datasets, we provide their descriptions in Table 8 and Table 9. The results of BCOT are from its paper (Fettal et al., 2022), and EBC are ran by official codes with threshold 10^{-4}.

C.2. Nyström experiments

In this part, we evaluate the efficiency of the proposed asymmetric Nyström method with comparisons to other standard solvers. The accuracy of a solution $\tilde{U} = [\tilde{u}_1, \dots, \tilde{u}_r], \tilde{V} = [\tilde{v}_1, \dots, \tilde{v}_r]$, is evaluated as the weighted average $\eta = \frac{1}{r}\sum_{i=1}^{r} w_i(1 - |u_i^\top \frac{\tilde{u}_i}{\|\tilde{u}_i\|}|) + \frac{1}{r}\sum_{i=1}^{s} w_i(1 - |v_i^\top \frac{\tilde{v}_i}{\|\tilde{v}_i\|}|)$, with $w_i = \lambda_i$, where r is the rank of the low-rank approximation, $U = [u_1, \dots, u_r], V = [v_1, \dots, v_r]$ are the left and right singular vectors of G from its rank-r compact SVD with singular values $\lambda_1 \geq \cdots \geq \lambda_r$. We compare our method with three common SVD solvers: truncated SVD (SVD) from the ARPACK library, Symmetric Nyström (Williams & Seeger, 2000) applied to $GG^\top$ and $G^\top G$, and randomized SVD (RSVD) (Halko et al., 2011). We employ the Lanczos method at rank r (Lehoucq et al., 1998) for the SVD subproblem of symmetric Nyström, and we employ RSVD at rank r for the SVD subproblem of asymmetric Nyström. Truncated SVD is run to machine precision for comparison. For a given tolerance ε, we stop training when $\eta < \varepsilon$, with η being the accuracy of a solution.

In Table 4 in the paper, we evaluate multiple tolerances, i.e., $\varepsilon = 10^{-1}, 10^{-2}$. In particular, for RSVD, we increase the number of oversamples until the target tolerance is reached. For the Nyström methods, we increase the number of subsamples m until the target tolerance is reached. We use random subsampling for all Nyström methods. The tolerance used in Figures 4 and 5 is $\varepsilon = 10^{-2}$. In Table 4, the SNE kernel bandwidth is set as $\gamma = k\sqrt{M\gamma_x}$, with γ_x the variance of the training data and data-dependent k ($k = 1$ for Cora and Citeseer, $k = 0.5$ for Pubmed); e.g., for Cora $\gamma_x = 0.0002$ and $\gamma = k\sqrt{M\gamma_x} \approx 0.74$. This gives an indication on the scaling w.r.t γ in Figure 4. In Figure 4, we consider that a solver's performance may depend on the singular spectrum of the kernel matrix, so we vary γ as shown in the horizontal axis in Figure 4, where an increased γ leads to spectra with faster decay, and assess training time. Our approach shows overall speedup compared to RSVD, and our asymmetric Nyström requires significantly fewer subsamples on the matrices with faster decay of the singular spectrum, showing greater speedup w.r.t. RSVD in this scenario. In the experiments of Fig. 6 in this Appendix and of Figure 5 in the main body, we compare the node classification performance of KPCA using symmetric Nyström against KSVD using our proposed asymmetric Nyström. We use the RBF kernel for both KPCA and KSVD, with γ tuned via 10-fold cross validation. Note that KSVD achieves higher performance at all considered subsamplings m, even if both methods use the RBF kernel. Similarly, even when symmetric kernel functions are chosen, the resulting G matrix in the KSVD solution w.r.t. (2.4) in the paper still maintains the asymmetry, as the two inputs of the kernel is applied to $\mathcal{X}$ and $\mathcal{Z}$, respectively.

C.3. General data experiments

In the experiments on general datasets, we consider three common classification datasets, including Diabetes of size 768, Ionosphere of size 351, Liver of size 583, and three commonly used regssion datasets, including Cholesterol of size 303, Yacht of size 308, and Physicochemical-protein of size 45730. Note that, though only the embeddings for samples are needed in prediction, i.e., the right singular vectors in KSVD and the eigenvectors in KPCA, the embeddings by KSVD are learned on an asymmetric kernel with two feature maps, while in KPCA they are learned with a symmetric kernel relating to a single feature map. To implement a learnable C matrix in KSVD, i.e., the alternative a_3 in Remark 3.2 in the paper, we utilize the backpropagation learning scheme with stochastic gradient descent (SGD) based optimizers for minimizing the loss in the downstream tasks. Correspondingly, we set C matrix as learnable parameters that can be backpropagated and optimized by SGD-based optimizer in an end-to-end manner. To make C learnable, we set GV as the learned features on the data samples to the downstream classifier/regressor, where V is chosen as the top-4 right singular vectors of G. In this manner, gradient can be backpropagated, where V is alternatively updated through the SVD on G. To be specific, we adopt an iterative training scheme for conducting SVD on the asymmetric kernel matrix G and updating other parameters: *i)* for input X and Z, which is given as $Z := X^\top C$ in this case, we compute the asymmetric kernel matrix $G := [\kappa(x, z)]$, $x \in X$, $z \in Z$, and then conduct SVD on G to obtain V s.t. $GV = U\Lambda$. *ii)* As C can only be backpropagated through G, we detach the gradient of V computed in previous step and fix it, we then forward X, Z to update G and send the projected features of samples from KSVD, i.e., GV, to the classification or regression head with the computed loss (cross-entropy loss or the mean squared error loss), and update all the parameters except V. In other experiments using KPCA or fixed C, i.e., a_0, a_1, a_2, we also train these methods with SGD-based optimizers, which makes our KSVD comparable to the learnable C case in a_3 for fair and consistent evaluations. Here, the difference lies in that we only need to update the classification/regression head, as the projected features of all samples (GV) is fixed with the given input data.

We adopt SGD as the optimizer for the linear classification or regression head, where the learning rate is set to 10^{-3} for all experiments except Cholesterol (10^{-1}) and Physicochemical-protein (10^{-4}). We choose the first 4 right singular vectors, i.e., $GV_{[:,:4]}$, to feed forward to the classification or regression head. When RBF kernel is used, γ^2 is selected as 1e7 in most cases except for Physicochemical-protein dataset, which is with 1e6. When SNE kernel is used, γ^2 is selected as 1e5 in most cases except for Ionosphere dataset with 1e6, Liver dataset with 1e4. Moreover, since Physicochemical-protein is a larger dataset, we utilize batch-training mode where we fix the batch size to be 500. All experiments are run for 2000 iterations.

D. Algorithm for C

Algorithm 1 details the realization of the compatibility matrix discussed in Section 2.2 in the main paper. Below, we consider the case $M > N$, where we construct the projection matrix $C_x \in \mathbb{R}^{M\times N}$ such that $XC_x \in \mathbb{R}^{N\times N}$. If $N > M$, we rather construct $C_z \in \mathbb{R}^{N\times M}$ such that $ZC_z \in \mathbb{R}^{M\times M}$. The construction of C_z mirrors the algorithm for C_x with the appropriate changes. In the case of square matrix with $N = M$, $C = I_N$, with I_N the identity matrix of size $N \times N$.

Algorithm 1 Compatibility Matrix Realization.

```
Input: $\mathcal{X} = \{x_i \in \mathbb{R}^M\}_{i=1}^N$
Define $X = [x_1, \dots, x_N]^\top$
if projection on $x_i$ then
    $C_x = \arg\min_C \left\| X - XCC^\top \right\|_\mathrm{F}^2$ {Alternative $a_1$}
else if randomized projection then
    $C_x = \mathrm{randn}(M, N)$ {Alternative $a_2$}
else if pseudoinverse then
    $C_x = \left((XX^\top)^\dagger X\right)^\top$ {Alternative $a_0$}
else if learnable then
    $C_x$ is learned by optimizing the downstream task objective. {Alternative $a_3$}
end if
Return: $C_x$
```

E. Proof of Proposition 2.2

Proof. Let $B_\phi \in \mathbb{R}^{n \times r}$.

$$\begin{aligned}
[\Gamma_\psi \Gamma_\phi^* B_\phi]_{jl} &= \frac{1}{\sqrt{m}} \langle \psi(z_j), \frac{1}{\sqrt{n}} \sum_{i=1}^n b_{il}^\phi \phi(x_i) \rangle \\
&= \sum_{i=1}^n \frac{1}{\sqrt{nm}} \langle \phi(x_i), \psi(z_j) \rangle b_{il}^\phi \\
&= [G^\top B_\phi]_{jl}
\end{aligned}$$

The proof for $\Gamma_\phi \Gamma_\psi^*$ is similar. □

F. Proof of Proposition 3.3

Proof. Apply on the left respectively Γ_ϕ and Γ_ψ to both equations from Equation (3) combined with Proposition 3.2. □

G. Proof of Proposition 3.4

Proof. Perform the substitution of the proposed W_ϕ, W_ψ in the CCE problem with the knowledge that $B_\phi^{\mathrm{svd}}, B_\psi^{\mathrm{svd}}$ come from the SVD of G. □