Kernel-driven Self-Supervision for Multi-View Learning over Graphs

Alireza Sadeghi, Konstantinos D. Polyzos, and Georgios B. Giannakis

Department of Electrical and Computer Engineering, University of Minnesota, USA

2024 58th Asilomar Conference on Signals, Systems, and Computers | 979-8-3503-5405-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/IEEECONF60004.2024.10942820

***Abstract*—Self-supervised (SeSu) learning is a powerful subclass of unsupervised methods that aims to alleviate the need for large-scale annotated datasets to successfully train data-hungry machine learning models. To this end, SeSu methods learn contextualized embeddings from unlabeled data to efficiently tackle downstream tasks. Despite their success, most existing SeSu approaches are heuristic, and typically fail to exploit multiple views of data available for the problem at hand. This becomes particularly challenging when non-linear dependencies among multiple views or data samples exist, often emerging in applications such as learning over large-scale graphs. In this context, the present paper builds upon kernel-based learning framework to introduce principled SeSu approaches. Specifically, in lieu of the well-celebrated Representer theorem, this work posits that the optimal function for addressing the downstream problem resides in a Reproducing Kernel Hilbert space. The proposed SeSu approach then learns "low-dimensional" embeddings to approximate the feature map associated with the optimal underlying kernel. By judiciously combining the learned embeddings from multiple views of data, this paper demonstrates that a wide range of downstream problems over graphs can be efficiently solved. Numerical tests using synthetic and real graph datasets showcase the merits of the proposed approach relative to competing alternatives.**

***Index Terms*—Self-supervised learning, kernel-based learning, multi-view data, semi-supervised learning over graphs.**

I. Introduction

Millions of connected devices and large-scale networks continuously generate vast amounts of data. While this data holds great promise for training deep neural network models, a significant challenge arises from the fact that most of it remains *unlabeled* due to the high costs of annotation. Traditional data analytics, while effective for small-scale datasets, struggle to scale with such sheer volume and high dimensionality of data. This necessitates innovative approaches capable of efficiently processing large-scale, high-dimensional, and *predominantly unlabeled* data while maintaining affordable computational complexity.

Self-supervised learning (SeSu) is an innovative paradigm that addresses the challenges of limited labeled data by leveraging abundant *unlabeled* data and *predefined reference models* [11], [9]. Unlike traditional supervised learning, which depends on costly and labor-intensive manual annotations, SeSu utilizes unlabeled datasets to *pre-train* expressive feature extractors. These pre-trained models generate *low-dimensional*, context-rich representations of the input data [11], [7], [18]. The learned representations serve as a foundation for *fine-tuning* lightweight models with affordable computation using a minimal amount of labeled samples, enabling effective learning in data-scarce scenarios.

Prior work. SeSu has recently demonstrated remarkable success across diverse domains, including computer vision [4], natural language processing [25], [5], and graph learning [8], making it a promising approach for label-efficient and data-driven solutions. Its abiligty to harness the wealth of unlabeled data effectively positions SeSu as a key enabler for advancing machine learning in a sample-efficient manner. Critical to SeSu is the *reference model* selection, with recently popular approaches including predicting masked data, such as in auto-regressive masked language models [5] or bidirectional ones [25], reconstructing input samples, or contrasting similar and dissimilar data points as in the contrastive learning paradigm [4]. Such approaches enable pre-trained models to learn meaningful representations without explicit supervision. Once pre-trained on these tasks, the learned representations can be used and further fine-tuned with minimal labeled data, significantly reducing the need to annotate samples. Despite their remarkable success in pre-training [25], [5], [19], most existing SeSu approaches remain heuristic, particularly in specifying the reference model and leveraging unlabeled data [7].

This paper builds on kernel-driven learning methods [23], [12], [16], [21] to introduce *principled* SeSu algorithms with enhanced prediction performance. Relative to prior art, our contributions can be summarized as follows

- A novel kernel-driven SeSu approach is advocated to fully leverage unlabeled data, and learn expressive embeddings during a pre-training step. The learned embeddings can be used to efficiently carry out a down-stream learning task using only a few labeled data samples.
- An extended unified kernel-driven SeSu framework is proposed for multi-view learning over graphs, treating nodal features and the adjacency matrix as distinct but complementary views of graph data.
- Numerical tests were conducted on synthetic and real graph datasets, demonstrating significant performance gains in prediction tasks, outperforming competing alternatives.

II. Single-view self-supervised learning

Consider a generic learning task over a graph $\mathcal{G} := \{\mathcal{V}, \mathbf{A}, \mathbf{X}\}$ with vertex set $\mathcal{V} := \{1, \ldots, N\}$ collecting N

This work was supported by NSF grants 2212318, 2220292, 2126052, 2102312, 2103256, and 2312547. Emails: sadeghi@umn.edu, polyz003@umn.edu, and georgios@umn.edu.

nodes, and an $N \times N$ adjacency matrix $\mathbf{A}$ capturing the node connectivity pattern, with $\mathbf{a}_n = \mathbf{A}[:, n]$ representing the adjacency of node n. The matrix $\mathbf{X} \in \mathbb{R}^{F \times N}$ captures the $F \times 1$-dimensional nodal features, with $\mathbf{x}_n = \mathbf{X}[:, n]$ representing the nodal feature on node n. The objective here is to learn a real-valued function $f : \mathcal{V} \times \mathbf{X} \to \mathbb{R}$, where the labels y_n per node obey the input-output relationship

$$y_n = f(\mathbf{a}_n, \mathbf{x}_n) + \epsilon_n, \forall n \tag{1}$$

with ϵ_n denoting the observation noise; see [15], [24], [13], [14]. For the time being, we assume f is only a function of $\{\mathbf{a}_n\}_{n=1}^N$, and we will revise the generic problem (1) while incorporating nodal features $\{\mathbf{x}_n\}_{n=1}^N$ from a multi-view data analysis perspective later.

The objective here is to perform a *semi-supervised* learning task over the sought graph. That is, in the given graph, the labels $\{y_n\}_{n\in\mathcal{O}}$ are given only for a *small* subset of observed nodes, where $\mathcal{O}$ represents the index set of such nodes, and $\mathcal{U}$ denotes the index set of *unobserved* ones. Given a large unlabeled dataset $\{\mathbf{a}_n\}_{n\in\mathcal{U}}$ and a small labeled one $\{(\mathbf{a}_n, y_n)\}_{n\in\mathcal{O}}$, with $|\mathcal{O}| \ll |\mathcal{U}|$, the task is to learn an expressive function f as in (1). To address such ill-posed problems, common remedies include regularization, such as the Laplacian [3], or using graph neural networks (GNNs) [10], [26] to propagate information from observed to unobserved nodes. The former approach is challenged in selecting appropriate regularization, while the latter fails when features are unavailable. Here, we propose an alternative method to address these issues, as outlined next.

A. Kernel-driven Self-Supervision

To formalize our approach, let $f : \mathcal{V} \to \mathbb{R}$ belong to a reproducing kernel Hilbert space (RKHS), which is a class of functions $\mathcal{H} := \{f | f(\mathbf{a}) = \sum_{n=1}^N \alpha_n \kappa(\mathbf{a}, \mathbf{a}_n)\}$ induced by a kernel $\kappa(\mathbf{a}, \mathbf{a}_n) : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ which measures the similarity between any two node with their adjacency patterns $\mathbf{a}$ and $\mathbf{a}_n$. Among possible choices, a popular choice for κ is the Gaussian kernel given by $\kappa(\mathbf{a}, \mathbf{a}_n) := \exp\left[-\|\mathbf{a} - \mathbf{a}_n\|_2^2/(2\sigma^2)\right]$ with width σ. A kernel is reproducing if it satisfies $\langle \kappa(\mathbf{a}, \cdot), \kappa(\mathbf{a}_n, \cdot)\rangle_{\mathcal{H}} = \kappa(\mathbf{a}, \mathbf{a}_n)$, which in turn induces the RKHS norm $\|f\|_{\mathcal{H}}^2 := \sum_n \sum_{n'} \alpha_n \alpha_{n'} \kappa(\mathbf{a}_n, \mathbf{a}_{n'})$. Having $f \in \mathcal{H}$, the objective is to learn f by solving

$$\min_{f\in\mathcal{H}} \sum_{n\in\mathcal{O}} \mathcal{L}(f(\mathbf{a}_n), y_n) + \lambda\, \Omega\left(\|f\|_{\mathcal{H}}^2\right) \tag{2}$$

where $\mathcal{L}(\cdot)$ is the loss, and the regularizer $\Omega(\cdot)$ is an increasing function, with hyperparameter $\lambda > 0$ that controls overfitting. A pitfall of the learning task in (2) is that it falls short of leveraging abundant *unlabeled* data $\{\mathbf{a}_n\}_{n\in\mathcal{U}}$. The typical remedy for solving (2) is to invoke the Representer theorem to find a simple finite-dimensional closed-form optimal solution [22]

$$\hat{f}(\mathbf{a}) = \sum_{n\in\mathcal{O}} \alpha_n \kappa(\mathbf{a}, \mathbf{a}_n) := \boldsymbol{\alpha}^\top \mathbf{k}(\mathbf{a}) \tag{3}$$

where $\boldsymbol{\alpha} := [\alpha_1, \ldots, \alpha_{|\mathcal{O}|}]^\top \in \mathbb{R}^{|\mathcal{O}|}$ collects the learnable coefficients, and the $|\mathcal{O}| \times 1$ kernel vector is represented by $\mathbf{k}(\mathbf{a}) := [\kappa(\mathbf{a}, \mathbf{a}_1), \ldots, \kappa(\mathbf{a}, \mathbf{a}_{|\mathcal{O}|})]^\top$. Substituting (2) into the RKHS norm, we obtain $\|f\|_{\mathcal{H}}^2 := \sum_n \sum_{n'} \alpha_n \alpha_{n'} \kappa(\mathbf{a}_n, \mathbf{a}_{n'}) = \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}$ where the $|\mathcal{O}| \times |\mathcal{O}|$ kernel matrix $\mathbf{K}$ has entries $[\mathbf{K}]_{n,n'} := \kappa(\mathbf{a}_n, \mathbf{a}_{n'})$. Using this, our problem boils down to the following one over $\boldsymbol{\alpha}$

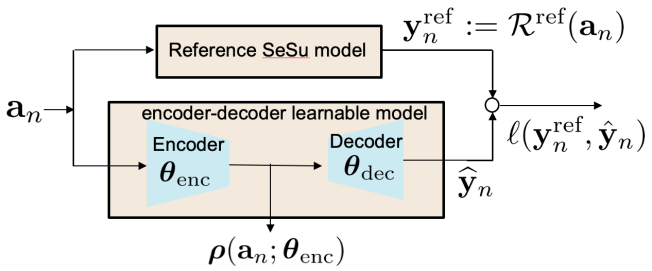

Fig. 1: Kernel-driven (reference) Self-supervision.

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^N} \sum_{n\in\mathcal{O}} \mathcal{L}\big(\boldsymbol{\alpha}^\top \mathbf{k}(\mathbf{a}_n), y_n\big) + \lambda\, \Omega\big(\boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}\big) \tag{4}$$

where $\mathbf{k}^\top(\mathbf{a}_n)$ is the nth row of the matrix $\mathbf{K}$. While a scalar y_n is used here for brevity, coverage extends readily to vectors $\{\mathbf{y}_n\}_{n=1}^N$.

The learning task formulated under (4) has certain limitations. The main drawback is that it relies solely on labeled data samples, requires a pre-selected kernel function κ, and assumes data pairs $\{\mathbf{a}_n, y_n\}_{n\in\mathcal{O}}$ are available in batch form. Additionally, the dimension of the learnable parameter $\boldsymbol{\alpha}$ grows with $|\mathcal{O}|$, where the computational complexity required to solve this is often prohibitive, scaling with $|\mathcal{O}|^3$. These challenges necessitate the development of efficient, scalable, and adaptive methods capable of operating effectively in dynamic environments, particularly in contexts involving large graphs with partially labeled or even completely unlabeled data. Addressing these limitations could offer the full potential of kernel-based methods for broader applications, including those of online learning, semi-supervised graph inference, and real-time decision-making over complex networks.

To address these challenges, we reformulate the functional learning problem in (2) as a parametric one, ensuring that the dimensionality of the optimization variables remains fixed regardless of the number of available samples. More importantly, the proposed approach effectively leverages all available unlabeled data, thereby enhancing overall performance. Our novel reformulation allows us to leverage powerful tools from convex optimization and online learning in vector spaces, facilitating efficient and scalable solutions.

To fully leveraging the abundant unlabeled data, we build our approach on SeSu learning, to pre-train a *generalizable* feature extractor, that can be fine-tuned later using only *a few* labeled samples. Our novel approach advocates a two-step learning process, including a pre-training step, followed by fine-tuning. Critical to our proposed kernel-driven SeSu framework pre-training step, where an encoder with weights $\boldsymbol{\theta}_{\text{enc}}$ and a decoder with weights $\boldsymbol{\theta}_{\text{dec}}$ are pre-trained using only *unlabeled* data $\{\mathbf{a}_n\}_{n\in\mathcal{U}}$; see Fig. 1 for an illustration. The key here is the user-specified *reference model* $\mathcal{R}^{\text{ref}}(\cdot)$, which synthetically generates "pseudo-labels" for the unlabeled data as $\mathbf{y}_n^{\text{ref}} := \mathcal{R}^{\text{ref}}(\mathbf{a}_n)$, for $n \in \mathcal{U}$. The output of the decoder, $\hat{\mathbf{y}}_n$, aims to reconstruct the pseudo-label given

its input $\boldsymbol{\rho}(\mathbf{a}_n;\boldsymbol{\theta}_{\text{enc}}) \in \mathbb{R}^d\,(d < |\mathcal{O}|)$, which is a low-dimensional embedding for the input $\mathbf{a}_n$ generated by the encoder. The encoder-decoder architecture is trained end-to-end using back-propagation via the least-squares reconstruction loss $\ell(\boldsymbol{\theta}_{\text{enc}},\boldsymbol{\theta}_{\text{dec}}) = \sum_{n\in\mathcal{U}\cup\mathcal{O}} \|\hat{\mathbf{y}}_n(\boldsymbol{\theta}_{\text{enc}},\boldsymbol{\theta}_{\text{dec}}) - \mathbf{y}_n^{\text{ref}}\|_2^2$.

Remark 1. Although SeSu leverages unlabeled data, it can also incorporate labeled data during pre-training. Hence, the reconstruction loss above considers the union of the unlabeled set $\mathcal{U}$ and labeled set $\mathcal{O}$ to utilize all available information.

This work advocates a *kernel-driven reference model* where $\mathcal{R}^{\text{ref}}(\mathbf{a}) := [\kappa(\mathbf{a},\mathbf{a}_1),\ldots,\kappa(\mathbf{a},\mathbf{a}_{|\mathcal{U}|})]$ which guides the pre-training process. Such a reference model offers all the benefits of kernel-based learning [20], [23], [22]. In practice, however, the dimensionality of the kernel-driven $\mathcal{R}^{\text{ref}}(\mathbf{a}) \in \mathbb{R}^{|\mathcal{U}|\times 1}$ can be prohibitive when $|\mathcal{U}| \gg 1$. To address this challenge, a remedy is to sample a subset $\mathcal{S} := \{\mathbf{a}_1^{(s)},\ldots,\mathbf{a}_{|\mathcal{S}|}^{(s)}\}$ from $\{\mathbf{a}_n\}_{n\in\mathcal{U}}$, thereby reducing the dimensionality to $\mathcal{R}^{\text{ref}}(\mathbf{a}) = [\kappa(\mathbf{a},\mathbf{a}_1^{(s)}),\ldots,\kappa(\mathbf{a},\mathbf{a}_{|\mathcal{S}|}^{(s)})] \in \mathbb{R}^{|\mathcal{S}|\times 1}$ with $|\mathcal{S}| \ll |\mathcal{U}|$. The intuition here is that the kernel-driven reference model $\mathcal{R}^{\text{ref}}(\mathbf{a})$ produces embeddings that provide non-linear approximations of the kernel, that is

$$\kappa(\mathbf{a}_n,\mathbf{a}_{n'}) \approx \boldsymbol{\rho}^\top(\mathbf{a}_n;\boldsymbol{\theta}_{\text{enc}})\boldsymbol{\rho}(\mathbf{a}_{n'};\boldsymbol{\theta}_{\text{enc}}) \tag{5}$$

which provides more expressive feature representations compared to e.g., random features (RF) [20], [23], and accommodates a broader class of kernels, beyond shift-invariant ones. Here, $\boldsymbol{\rho}(\mathbf{a},\boldsymbol{\theta}_{\text{enc}}) \in \mathbb{R}^d$ is a learned *low-dimensional* parametric embedding of input $\mathbf{a}$, with associated encoder weights $\boldsymbol{\theta}_{\text{enc}}$, which is trained using unlabeled data and preselected reference model $\mathcal{R}^{\text{ref}}(\cdot)$. This allows for learning a reduced-dimensional model f in the transformed feature space, resulting in more sample-efficient learning procedures. Compared with alternatives [25], [4], [7], [8], the kernel-driven SeSu can learn a function with certain optimality guarantees as we shall see later. Leveraging the encoder with *pre-trained* $\widehat{\boldsymbol{\theta}}_{\text{enc}}$, one can obtain the following linear function approximant

$$\begin{aligned}\hat{f}(\mathbf{a}) = \sum_{n\in\mathcal{O}} \alpha_n \kappa(\mathbf{a},\mathbf{a}_n) &\approx \sum_{n\in\mathcal{O}} \alpha_n \boldsymbol{\rho}^\top(\mathbf{a};\widehat{\boldsymbol{\theta}}_{\text{enc}})\boldsymbol{\rho}(\mathbf{a}_n;\widehat{\boldsymbol{\theta}}_{\text{enc}}) \\ &:= \boldsymbol{\theta}^\top \boldsymbol{\rho}(\mathbf{a};\widehat{\boldsymbol{\theta}}_{\text{enc}}).\end{aligned} \tag{6}$$

where the d-dimensional $\boldsymbol{\theta} := \sum_{n\in\mathcal{O}} \alpha_n \boldsymbol{\rho}(\mathbf{a}_n;\widehat{\boldsymbol{\theta}}_{\text{enc}})$ is the weighted sum of parametric embeddings of *labeled* data. Instead of directly finding $\{\alpha_n\}_{n\in\mathcal{O}}$, we rely on the parameter vector $\boldsymbol{\theta} := \sum_{n\in\mathcal{O}} \alpha_n \boldsymbol{\rho}(\mathbf{a}_n;\widehat{\boldsymbol{\theta}}_{\text{enc}})$, which transforms the learning problem from the $|\mathcal{O}|$-dimensional space of $\{\alpha_n\}_{n\in\mathcal{O}}$ to the reduced d-dimensional space of $\boldsymbol{\theta} \in \mathbb{R}^d$. Having the linear function approximant in (6), one can readily *fine-tune* the function with a small labeled data set, even arriving on-the-fly using e.g., online gradient descent. This online processing is especially attractive for time-sensitive applications.

The proposed kernel-driven SeSu learning approach offers several advantages; namely the dimensionality d is a tunable hyperparameter that can even be smaller than $|\mathcal{O}|$, enabling significant computational savings. The encoder, parameterized by $\boldsymbol{\theta}_{\text{enc}}$, effectively leverages unlabeled data to estimate the pre-selected kernel, enriching the model's ability to generalize. Finally, by operating in the embedding space, the computational complexity is reduced from $\mathcal{O}(|\mathcal{O}|^3)$ to $\mathcal{O}(d^3)$, making the approach scalable to large datasets.

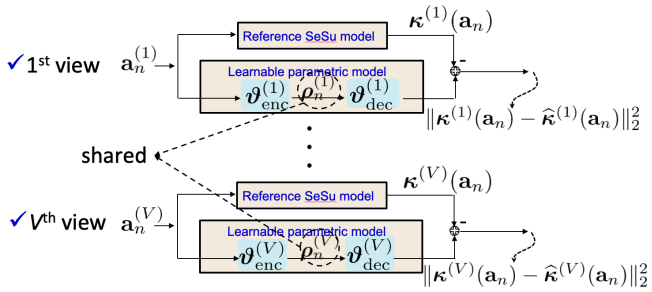

Fig. 2: Multi-view Kernel-driven SeSu.

Upon leveraging the pre-trained encoder $\widehat{\boldsymbol{\theta}}_{\text{enc}}$ and the learned embeddings, the main downstream regressor can be efficiently learned by solving the following optimization problem

$$\boldsymbol{\theta}^* := \arg\min_{\boldsymbol{\theta}} \sum_{n\in\mathcal{O}} \Big(y_n - \boldsymbol{\theta}^\top \boldsymbol{\rho}(\mathbf{a}_n;\widehat{\boldsymbol{\theta}}_{\text{enc}})\Big)^2 + \lambda\,\Omega(\boldsymbol{\theta}), \tag{7}$$

where $\boldsymbol{\theta}$ is the aggregated weight vector – to be learned from sampled data, defined as $\boldsymbol{\theta} = \sum_{n\in\mathcal{O}} \alpha_n \boldsymbol{\rho}(\mathbf{a}_n;\widehat{\boldsymbol{\theta}}_{\text{enc}})$, and $\Omega(\boldsymbol{\theta})$ is a regularization term, such as $\|\boldsymbol{\theta}\|_2^2$ or $\|\boldsymbol{\theta}\|_1$. This step, also referred to as fine-tuning, requires only a few labeled data samples, resulting in a sample-efficient learning framework.

The proposed two-step pre-training and fine-tuning kernel-driven learning approach demonstrates the synergy between self-supervision and kernel methods, offering a computationally efficient yet flexible framework for regression and other learning tasks. By leveraging the learned embeddings, the model achieves both scalability and adaptability when handling graph-structured or high-dimensional data.

III. Multi-view Self-supervised Learning

Multi-view data, often collected from different transformations of a shared signal, are common in various applications. In the problem under consideration, the adjacency matrix $\mathbf{A}$ and the feature matrix $\mathbf{X}$ can naturally be interpreted as multiple views of a shared embedding space. The former captures the adjacency information encoding the relationships among nodes, while the latter represents the features associated with the nodes themselves. To address such multi-view data, one approach is to construct a *second* graph with adjacency matrix $\mathbf{A}^{(2)}$, where the edge between any nodes n and n' is determined based on the partial correlation between their corresponding feature vectors $\mathbf{x}_n$ and $\mathbf{x}_{n'}$. Partial correlation is employed as a similarity measure between nodes due to its intuitive appeal and demonstrated effectiveness across various applications. By leveraging (partial) correlations between features $\mathbf{x}_n$ and $\mathbf{x}_{n'}$, and incorporating hypothesis testing, this method facilitates the construction of a distinct graph to capture relationships among the node features [6].

To formalize learning from such multi-view data, consider a generic setup with $v = 1,\cdots,V$ views of data, each

consisting of distinct features denoted by $\{\mathbf{a}_n^{(\nu)}\}_{n=1}^N, \forall \nu$. The objective is to learn per-view embeddings $\{\boldsymbol{\rho}^{(\nu)}(\mathbf{a}_n^{(\nu)}; \boldsymbol{\theta}_{\text{enc}}^{(\nu)})\}$ for all $n = 1, \cdots, N$ nodes. The embedding for each node is *shared* across all views; that is, for a given node n, it holds that $\boldsymbol{\rho}^{(\nu)}(\mathbf{a}_n^{(\nu)}; \boldsymbol{\theta}_{\text{enc}}^{(\nu)}) = \boldsymbol{\rho}_n, \forall \nu$.

To learn the shared embedding across all views, we employ a dictionary of kernels $\{\boldsymbol{\kappa}^{(\nu)}\}_{\nu=1}^V$, representing a set of reference SeSu models, one for each view. We further rely on separate learnable parametric encoders and decoders per view, denoted by $\{\boldsymbol{\theta}_{\text{enc}}^{(\nu)}\}_{\nu=1}^V$ and $\{\boldsymbol{\theta}_{\text{dec}}^{(\nu)}\}_{\nu=1}^V$, respectively; see Fig. 2 for an illustration.

Once the encoders and decoders are pre-trained using unlabeled data, one can estimate the function using labeled data samples as $\hat{f}(\mathbf{a}; \boldsymbol{\alpha}) = \sum_{n \in \mathcal{O}} \alpha_n \boldsymbol{\rho}^\top(\mathbf{a}) \boldsymbol{\rho}_n := \boldsymbol{\rho}^\top(\mathbf{a}) \boldsymbol{\theta}$, where $\boldsymbol{\rho}(\mathbf{a})$ is the shared embedding for the node with adjacency $\mathbf{a}$.

Remark 2. Instead of using a shared embedding across all the views, one can alternatively assign separate embeddings for each view, denoted as $\{\boldsymbol{\rho}_n^{(\nu)}\}_{\nu=1}^V$ for each node n. These embeddings can then be concatenated to form the embedding of node n as $\boldsymbol{\rho}_n := [\boldsymbol{\rho}_n^{(1)} \cdots \boldsymbol{\rho}_n^{(V)}]^\top$.

IV. Numerical tests

To assess the performance of our proposed approach we used synthetic and real graph datasets. For the former we used a synthetic graph consisting of $N = 60$ nodes, constructed using the stochastic block model (SBM) comprising 10 communities; see [18], [17] for further details. The nodal value of node n is given by the n-th entry of the eigenvector corresponding to the *lowest nonzero eigenvalue* of the graph Laplacian

$$\mathbf{L} := \text{diag}(\mathbf{A}\mathbf{1}_N) - \mathbf{A},$$

with $\mathbf{1}_N$ denoting an $N \times 1$ vector of all ones, the number of observed nodes is $|\mathcal{O}| = 10$, and the number of unobserved (test) nodes is $|\mathcal{U}| = 50$. For the real dataset, we used the Network Delays dataset, where a graph with $N = 70$ nodes is constructed. The nodes represent paths connecting two of the 9 end-nodes on the Internet2 backbone, and the edges represent the shared links between any two paths [2]. The $\{y_n\}_{n=1}^N$ are the measured delays on these paths. The number of observed nodes is $|\mathcal{O}| = 15$, and the number of unobserved ones is $|\mathcal{U}| = 55$.

The last dataset considered is the Temperature Stations dataset, where a graph with $N = 109$ nodes is constructed. The nodes represent weather stations across the US, and the edge weights correspond to the geographic distances between them [1]. Nodal values $\{y_n\}_{n=1}^N$ are the temperature measurements across the stations. Only $|\mathcal{O}| = 15$ measured temperatures are available, while $|\mathcal{U}| = 94$ temperatures are to be predicted.

We begin by evaluating the single-view SeSu learning against a set of benchmarks. Given that our method inherently focuses on dimensionality reduction, we first compare its performance with kernel-based Principal Component Analysis (Kernel-PCA) and an autoencoder (AE) approach. For the main learning task, benchmarks are conducted using linear regression. Additionally, two neural network configurations are considered for encoder-decoder architecture, where the encoder is a 1-layer neural network with $d = 15$ neurons for the SBM dataset, $d = 10$ for the temperature dataset, and $d = 4$ for the network delay dataset, and the decoder, another 1-layer neural network with $d = 60$ neurons for the SBM dataset, $d = 109$ for the temperature dataset, and $d = 70$ for the network delay dataset. The metric used in our evaluation is the normalized mean square error (NMSE), defined as $\text{NMSE} = \frac{\sum_{n \in \mathcal{U}} \|y_n - \hat{y}_n\|_2^2}{\sum_{n \in \mathcal{U}} y_n^2}$, where y_n represents the ground truth value and $\hat{y}_n$ denotes the predicted values. This metric ensures that the error is normalized by the magnitude of the true signal, allowing for meaningful comparisons across datasets. For a fair comparison, we used a radial basis function (RBF) kernel for all (multi-view) SeSu and kernel-PCA approaches, with the width parameter σ set before hand. Table I reports the NMSE of the proposed method compared with alternative approaches across all datasets. Evidently, the single-view kernel-driven SeS algorithm outperforms both Kernel-PCA and autoencoders, delivering at least an order of magnitude improvement over Kernel-PCA.

Dataset	Kernel PCA	Autoencoder	SeSu
Synthetic SBM	0.3595	0.0207	**0.0185**
Temperature	3.0075	0.0781	**0.0472**
Network Delays	0.2232	0.1321	**0.0897**

TABLE I: NMSE comparison across datasets for Kernel PCA, Autoencoder, and single-view SeSu learning.

Dataset	RF	SeSu	Multi-view SeSu
Synthetic SBM	0.0942	0.0185	**0.0164**
Temperature	0.2175	0.0472	**0.0454**
Network Delays	0.0942	0.0897	**0.0484**

TABLE II: NMSE comparison across datasets for RF based method and (multi-view) Kernel-driven SeSu.

To further compare our method, we also considered Random Features (RF)-based learning as another benchmark, as it provides an efficient mechanism to approximate kernel functions, as introduced by [20]. The RF approach carries over all the benefits of our method, including reducing the dimensionality of input features, which helps to deal with computational complexities along with the parametric counterpart of kernel-based learning.

The RF approach capitalizes on $d/2$ (where d is even) random vectors $\mathbf{v}_i$ sampled from the Fourier transform of the kernel's probability distribution, $\pi(\mathbf{v}) = \mathcal{F}(\kappa)$, where $i = 1, \ldots, d/2$, to find the embeddings of the input data. Using these random vectors, an embedding for the input data (also known as a random feature) vector of dimension $d \times 1$ is constructed as

$$\boldsymbol{\rho}_{\mathbf{v}}^{\text{RF}}(\mathbf{a}_n) := \frac{1}{\sqrt{d/2}} \Big[\sin(\mathbf{v}_1^\top \mathbf{a}_n), \cos(\mathbf{v}_1^\top \mathbf{a}_n), \ldots, \sin(\mathbf{v}_{d/2}^\top \mathbf{a}_n), \cos(\mathbf{v}_{d/2}^\top \mathbf{a}_n) \Big]^\top. \quad (8)$$

This RF vector $\boldsymbol{\rho}_{\mathbf{v}}^{\text{RF}}(\mathbf{a}_n)$ can be used as a linear approxima-

tion of the kernel – similar to what we proposed in (5)

$$\hat{\kappa}(\mathbf{a}_n, \mathbf{a}_{n'}) \approx \boldsymbol{\rho}_{\mathbf{v}}^{\mathrm{RF}}(\mathbf{a}_n)^\top \boldsymbol{\rho}_{\mathbf{v}}^{\mathrm{RF}}(\mathbf{a}_{n'}).$$

While the RF method is computationally efficient and scalable, it critically relies on the quality of the drawn random samples $\{\mathbf{v}_i\}_{i=1}^{d/2}$ for kernel approximation. In addition, RF method does not leverage abundant available unlabeled data. In contrast, our proposed approach avoids direct feature mapping and instead leverages a learnable framework that can adaptively infer embeddings, often resulting in improved performance for tasks involving structured data.

The results comparing the RF method with the single-view kernel-driven SeSu are reported in Table II. Clearly, SeSu offers enhanced performance by leveraging unlabeled data samples to learn the embeddings.

Multi-view Data and Its Usage

Multi-view data offers a comprehensive representation by leveraging multiple perspectives of the same dataset. To assess the performance of our proposed multi-view kernel-driven SeSu learning, we utilized two distinct views of the data. The first is derived from the adjacencies $\{\mathbf{a}_n\}_{n=1}^N$, which capture the relational structure among nodes, and the second comes from the features $\{\mathbf{x}_n\}_{n=1}^N$, representing the node-specific attributes. For the features, we employed a separate kernel to process the data, as illustrated in Figure 2. The embedding dimensions used in our experiments are $d = 30$ for the SBM dataset, $d = 20$ for the temperature dataset, and $d = 8$ for the network delay dataset.

The results of this multi-view embedding approach are summarized in the last column of Table II. Notably, leveraging the additional view, derived from the features $\{\mathbf{x}_n\}_{n=1}^N$, consistently reduces the NMSE across all datasets. This highlights the importance of incorporating node features alongside adjacency information to improve predictive performance and representation quality.

V. Conclusion

We propose a novel SeSu learning framework that builds on kernel-based learning to handle multi-view graph-structured data. By treating nodal features and the adjacency matrix as distinct views of a shared embedding, our approach effectively leverages multiple data views. A key advantage of the method is its ability to utilize unlabeled data, enabling improved performance. Numerical tests demonstrate significant improvements in predictive tasks, surpassing competing alternatives.

References

[1] "1981-2010 U.S. climate normals," https://www.ncdc.noaa.gov/data-access/land-based-station-data/land-based-datasets/climate-normals/1981-2010-normals-data, [Online; accessed 29-April-2019].

[2] "One-way ping internet2," http://software.internet2.edu/owamp/, [Online; accessed 29-April-2019].

[3] M. Belkin, I. Matveeva, and P. Niyogi, "Regularization and semi-supervised learning on large graphs," in *Proc. Conf. on Learn. Theory*. Banff, Canada: Springer, Jul. 2004, pp. 624–638.

[4] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Intl. Conf. Mach. Learn.*, Jul. 2020, pp. 1597–1607.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. North Am. Chapter Assoc. Comput. Linguist.: Hum. Lang. Technol., Vol. 1*, 2019, pp. 4171–4186.

[6] G. B. Giannakis, Y. Shen, and G. V. Karanikolas, "Topology identification and learning over graphs: Accounting for nonlinearities and dynamics," *Proc. IEEE*, vol. 106, no. 5, pp. 787–807, 2018.

[7] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Proc. of Adv. Neural Inf. Process.*, vol. 33, pp. 21 271–21 284, 2020.

[8] Z. Hou, X. Liu, Y. Cen, Y. Dong, H. Yang, C. Wang, and J. Tang, "GraphMAE: Self-supervised masked graph autoencoders," in *Proc. ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2022, pp. 594–604.

[9] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4037–4058, 2020.

[10] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Intl. Conf. Learn. Repr.*, 2017.

[11] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang, "Self-supervised learning: Generative or contrastive," *IEEE Trans. Knowledge Data Eng.*, vol. 35, no. 1, pp. 857–876, 2023.

[12] Q. Lu, G. Karanikolas, Y. Shen, and G. B. Giannakis, "Ensemble Gaussian processes with spectral features for online interactive learning with scalability," *Proc. Int. Conf. Artif. Intel. and Stats.*, pp. 1910–1920, 2020.

[13] K. D. Polyzos, Q. Lu, and G. B. Giannakis, "Graph-adaptive incremental learning using an ensemble of Gaussian process experts," *Proc. IEEE Int. Conf. Acoust., Speech, Sig. Process.*, June 2021.

[14] ——, "Online graph-guided inference using ensemble gaussian processes of egonet features," in *Proc. Asilomar Conf. Sig., Syst., Comput.*, 2021, pp. 182–186.

[15] ——, "Ensemble Gaussian processes for online learning over graphs with adaptivity and scalability," *IEEE Trans. Sig. Process.*, vol. 70, pp. 17–30, 2022.

[16] ——, "Weighted ensembles for adaptive active learning," *IEEE Trans. Signal Proc.*, vol. 72, pp. 4178–4190, 2024.

[17] K. D. Polyzos, C. Mavromatis, V. N. Ioannidis, and G. B. Giannakis, "Unveiling anomalous edges and nominal connectivity of attributed networks," *Proc. Asilomar Conf. Sig., Syst., Comput.*, Nov. 2020.

[18] K. D. Polyzos, A. Sadeghi, and G. B. Giannakis, "Bayesian self-supervised learning using local and global graph information," in *IEEE Int. Workshop Comput. Adv. Multi-Sensor Adapt. Process.*, 2023, pp. 256–260.

[19] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *OpenAI Blog*, vol. 1, no. 8, 2018.

[20] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," *Proc. Adv. Neural Inf. Process. Syst.*, pp. 1177–1184, 2008.

[21] D. Romero, M. Ma, and G. B. Giannakis, "Kernel-based reconstruction of graph signals," *IEEE Trans. Sig. Proc.*, vol. 65, no. 3, pp. 764–778, 2017.

[22] B. Schölkopf, A. J. Smola, F. Bach *et al.*, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Cambridge, MA: MIT press, 2002.

[23] Y. Shen, T. Chen, and G. B. Giannakis, "Random feature-based online multi-kernel learning in environments with unknown dynamics," *J. Mach. Learn. Res.*, vol. 20, no. 22, pp. 1–36, 2019.

[24] Y. Shen, G. Leus, and G. B. Giannakis, "Online graph-adaptive learning with scalability and privacy," *IEEE Trans. Sig. Process.*, vol. 67, no. 9, May 2019.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. of Adv. Neural Inf. Process.*, 2017, pp. 5998–6008.

[26] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," in *Proc. Intl. Conf. Mach. Learn.* PMLR, 2019, pp. 6861–6871.