Multi-scale Nyström Method

Woosang Lim[1], Rundong Du[1], Bo Dai[1], Kyomin Jung[2], Le Song[1,3], Haesun Park[1]
Georgia Institute of Technology[1], Seoul National University[2], Ant Financial[3]
woosang.lim@cc.gatech.edu, {rdu, bodai}@gatech.edu, kjung@snu.ac.kr, {lsong, hpark}@cc.gatech.edu

Abstract

Kernel methods are powerful tools for modeling nonlinear data. However, the amount of computation and memory required for kernel methods becomes the bottleneck when dealing with large-scale problems. In this paper, we propose Nested Nyström Method (NNM) which achieves a delicate balance between the approximation accuracy and computational efficiency by exploiting the multilayer structure and multiple compressions. Even when the size of the kernel matrix is very large, NNM consistently decomposes very small matrices to update the eigen-decomposition of the kernel matrix. We theoretically show that NNM implicitly updates the principal subspace through the multiple layers, and also prove that its corresponding errors of rank-k PSD matrix approximation and kernel PCA (KPCA) are decreased by using additional sublayers before the final layer. Finally, we empirically demonstrate the decreasing property of errors of NNM with the additional sublayers through the experiments on the constructed kernel matrices of real data sets, and show that NNM effectively controls the efficiency both for rank-k PSD matrix approximation and KPCA.

1 Introduction

The scalability of kernel methods is the major bottleneck for applying them to large-scale problems due to the computational and memory cost caused by the large dense kernel matrices. Nyström method is one of the effective methods for accelerating the kernel methods by low-rank approximation of the kernel matrix, $\mathbf{K} \in \mathbb{R}^{n\times n}$. There has been a large body of work that further improves the approximation quality and computational efficiency via adopting various sampling methods [5, 15, 4, 8, 13, 3, 11, 23, 10] and refining approximation formula [7, 5, 12, 20, 13, 18]. Especially, for rank-k spectral decomposition of $\mathbf{K}$, there are two basic rank-k Nyström methods which are *rank-k Standard Nyström Method* (SNM) [5] and *orthogonal Nyström method* (ONM) [7]. Recently, their efficient versions which are *SNM using Randomized SVD* (SNM+Rand.SVD) [12] and *Double Nyström Method* (DNM) [13] were proposed. All these four methods implicitly approximate the first k principal directions $\mathbf{U}_{\mathbf{Y},k}$ of n mapped data points $\mathbf{Y}$ in the feature space to compute the rank-k spectral decomposition of $\mathbf{K} = \mathbf{Y}^\top\mathbf{Y}$ with distinct schemes based on different motivations [13]. Rank-k SNM [5] actually computes the first k principal directions $\mathbf{U}_{\mathbf{S},k}$ of s sample mapped points $\mathbf{S}$ in the feature space, and SNM+Rand.SVD [12] uses randomized SVD to improve efficiency for computing the principal directions of sample mapped points. That is, rank-k SNM and SNM+Rand.SVD approximate $\mathbf{U}_{\mathbf{Y},k}$ via $\mathbf{U}_{\mathbf{S},k}$, which is computed by a particular form. However, it is known that both these two approximations lose the orthogonality and are biased to the sample subspace which is range($\mathbf{S}$). On the other hand, the ONM computes the best k approximate principal *orthogonal* direction in the sample subspace range($\mathbf{S}$) in the sense of minimize the KPCA reconstruction error [13]. However, such approximation requires extra computation, resulting higher time complexity $O(s^2n)$ compared to the time complexity of rank-k SNM which is $O(ksn + k^3)$. To further accelerate ONM, DNM [13] uses ONM twice in different scales, so that to compress the sample subspace range($\mathbf{S}$) for reducing the dimension of possible solution space for efficient computing of $\mathbf{U}_{\mathbf{Y},k}$. Although the algorithm performs well in practice, there is no analysis about how its rank-k approximation error varies after compression of sample subspace, and it is not clear whether the double scales are enough in terms of the balance between approximation accuracy and computation efficiency.

To achieve a better trade-off between these two factors,

Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. JMLR: W&CP volume 7X. Copyright 2018 by the author(s).

we extend the DNM to a multi-scale Nyström method. Accelerating the algorithms by exploiting multi-scale structures has been studied for the various methods including FEM [6], Bayesian optimization [21] and neural network [1] to solve the nonlinear problems, and there are also a number of applications such as multi-scale stable kernel construction [17], manifold learning [19], dictionary learning [16], and object detection [2, 14]. Among them, feature pyramid networks [14] successfully achieves both efficient and accurate object detection.

Inspired by the multi-scale approximation, we propose a multi-scale Nyström method, Nested Nyström Method (NNM), for both efficient and accurate eigen-decomposition of PSD matrices. NNM has a multilayer structure which consists of t sublayers and the final layer to efficiently and accurately updates the first k principal direction $\mathbf{U}_{\mathbf{Y},k}$ for computing a rank-k spectral decomposition of $\mathbf{K}$. We note that NNM is a general multi-scale framework which can be combined with any other column sampling, and our contribution is orthogonal to the column samplings. Interestingly, it can be viewed as t fully connected neural networks in the structure of NNM as described in Fig 1. We first briefly introduce the rank-k Nyström algorithms in Section 2. Then, we describe the nested Nyström method and provide an error analysis of NNM accordingly in Section 3. In Section 5, we will demonstrate our theoretical analysis of NNM and show that NNM is efficient for both rank-k PSD matrix approximation and KPCA on several benchmarks.

2 Rank-k Nyström Methods and Their Implicit Equations

In this section, we briefly introduce the notations and discuss the Nyström methods with the implicit equations regarding to approximating principal directions $\mathbf{U}_{\mathbf{Y},k}$ of n mapped data points $\mathbf{Y}$ in the feature space.

By the spectral theorem, for any $n \times n$ PSD matrix $\mathbf{K}$, there exists a matrix $\mathbf{Y} \in \mathbb{R}^{d \times n}$ which can be considered as n mapped data points so that $\mathbf{K} = \mathbf{Y}^\top \mathbf{Y}$ without loss of generality, where d is finite [5]. Even if $\mathbf{K}$ is generated by RBF kernel, the unknown mapped data in feature space can be isomorphically represented as $\mathbf{Y}$ s.t. $\mathbf{K} = \mathbf{Y}^\top \mathbf{Y}$. Then, let $\mathbf{S}$ be the $d \times s$ sample matrix which consists of s sample columns of $d \times n$ matrix $\mathbf{Y}$ corresponding to the column index $\mathcal{J}$, and let $\mathbf{C} = \mathbf{Y}^\top \mathbf{S}$ be the $n \times s$ submatrix of PSD matrix $\mathbf{K}$, which can be regarded as the inner product matrix of the whole data instances and the samples in the feature space. For kernel methods, $\mathbf{C}$ can be computed by using the kernel function κ, *i.e.*, $\mathbf{C}_{(i,j)} = \kappa(\mathbf{x}_i, \mathbf{x}_t)$ where $t \in \mathcal{J}$ is the j-th sample index among s sample indices. Let $\mathbf{K}_\mathbf{S}$ be the $s \times s$ principal submatrix of $\mathbf{K}$ s.t. $\mathbf{K}_\mathbf{S} = \mathbf{S}^\top \mathbf{S}$. For kernel methods using PSD kernel function, without loss of generality, we can also apply these implicit equations $\mathbf{C} = \mathbf{Y}^\top \mathbf{S}$ and $\mathbf{K}_\mathbf{S} = \mathbf{S}^\top \mathbf{S}$. A summary of notations is displayed in Tbl 1.

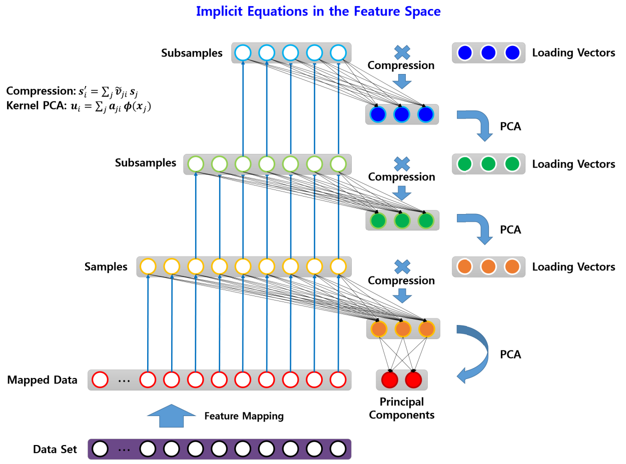

Figure 1: An example of muti-scale structure of NNM with four layers which are three sublayers and the final layer. Implicitly, NNM with four layers consists of three Fully Connected (FC) neural networks to compute the rank-k spectral decomposition of $\mathbf{K}$. The output of each FC neural network can be considered as approximate principal directions of n mapped data points, and NNM uses them to compute loading vectors of the subsamples/samples on the upper layer. By using the computed loading vectors, we can update the next FC neural network.

Table 1: A summary of the notations

Notation	Description
n	the number of instances
s	the number of samples
$\mathbf{K}$	$n \times n$ PSD matrix
$\mathbf{Y}$	$d \times n$ matrix $\mathbf{Y}$ s.t. $\mathbf{K} = \mathbf{Y}^\top \mathbf{Y}$
$\mathbf{S}$	$d \times s$ sample matrix of $\mathbf{Y}$
$\mathbf{C}$	$n \times s$ sample matrix of $\mathbf{K}$, $\mathbf{C} = \mathbf{Y}^\top \mathbf{S}$
$\mathbf{K}_\mathbf{A}$	$\mathbf{K}_\mathbf{A} = \mathbf{A}^\top \mathbf{A}$ for $\forall \mathbf{A}$
$\mathbf{A}'$	compressed matrix for $\forall \mathbf{A}$
$\mathbf{A} = \mathbf{U}_\mathbf{A} \mathbf{\Sigma}_\mathbf{A} \mathbf{V}_\mathbf{A}^\top$	compact SVD for $\forall \mathbf{A}$
$\tilde{\mathbf{A}} = \tilde{\mathbf{U}}_\mathbf{A} \tilde{\mathbf{\Sigma}}_\mathbf{A} \tilde{\mathbf{V}}_\mathbf{A}^\top$	approximate compact SVD for $\forall \mathbf{A}$
$\mathbf{A}_k = \mathbf{U}_{\mathbf{A},k} \mathbf{\Sigma}_{\mathbf{A},k} \mathbf{V}_{\mathbf{A},k}^\top$	rank-k SVD for $\forall \mathbf{A}$
$\tilde{\mathbf{A}}_k = \tilde{\mathbf{U}}_{\mathbf{A},k} \tilde{\mathbf{\Sigma}}_{\mathbf{A},k} \tilde{\mathbf{V}}_{\mathbf{A},k}^\top$	approximate rank-k SVD for $\forall \mathbf{A}$
$\mathbf{A}^\dagger = \mathbf{V}_\mathbf{A} \mathbf{\Sigma}_\mathbf{A}^{-1} \mathbf{U}_\mathbf{A}^\top$	pseudo inverse for $\forall \mathbf{A}$

Now, we are going to discuss rank-k Nyström methods which are rank-k standard Nyström method (SNM) [5], SNM using randomized SVD (SNM+Rand.SVD) [12], orthogonal Nyström method (ONM) [7], and double Nyström method (DNM) [13]. These four methods implicitly approximate the first k principal directions $\tilde{\mathbf{U}}_{\mathbf{Y},k}$ of the subspace spanned by mapped data points $\mathbf{Y}$ in the feature space to compute the rank-k spectral decomposition of $\mathbf{K}$ as

$$\mathbf{K}_k \approx \mathbf{Y}\tilde{\mathbf{U}}_{\mathbf{Y},k}(\tilde{\mathbf{U}}_{\mathbf{Y},k})^\top\mathbf{Y}^\top, \tag{1}$$

where $\mathbf{K} = \mathbf{Y}^\top\mathbf{Y}$ without loss of generality, the rank-k SVD of $\mathbf{Y}$ is $\mathbf{Y}_k = \mathbf{U}_{\mathbf{Y},k}\boldsymbol{\Sigma}_{\mathbf{Y},k}(\mathbf{V}_{\mathbf{Y},k})^\top$, $\tilde{\mathbf{U}}_{\mathbf{Y},k}$ is approximation of $\mathbf{U}_{\mathbf{Y},k}$, and $\mathbf{K}_k = \mathbf{Y}\mathbf{U}_{\mathbf{Y},k}(\mathbf{U}_{\mathbf{Y},k})^\top\mathbf{Y}^\top$ is the best rank-k approximation of $\mathbf{K}$ computed by SVD [13]. Although these methods share Eqn (1) and approximate $\mathbf{U}_{\mathbf{Y},k}$ for $\mathbf{K}_k$, the computed $\tilde{\mathbf{U}}_{\mathbf{Y},k}$ are different since they use distinct approaches based on dissimilar motivations.

Rank-k standard Nyström method (SNM) [5] approximately computes the rank-k spectral decomposition as $(\tilde{\boldsymbol{\Sigma}}^{snm}_{\mathbf{Y},k})^2 = \frac{n}{s}(\boldsymbol{\Sigma}_{\mathbf{S},k})^2$, $\tilde{\mathbf{V}}^{snm}_{\mathbf{Y},k} = \sqrt{\frac{s}{n}}\mathbf{C}\mathbf{V}_{\mathbf{S},k}\boldsymbol{\Sigma}^{-2}_{\mathbf{S},k}$, $\tilde{\mathbf{K}}^{snm}_k = \mathbf{C}\mathbf{K}^\dagger_{\mathbf{S},k}\mathbf{C}^\top$, where $k \le$ rank($\mathbf{S}$), and $\mathbf{K}_{\mathbf{S},k} = \mathbf{V}_{\mathbf{S},k}\boldsymbol{\Sigma}^2_{\mathbf{S},k}\mathbf{V}^\top_{\mathbf{S},k}$. The implicit equations of SNM for rank-k approximation are

$$\tilde{\mathbf{U}}^{snm}_{\mathbf{Y},k} = \mathbf{U}_{\mathbf{S},k}, \quad \tilde{\mathbf{K}}^{snm}_k = \mathbf{Y}(\mathbf{U}_{\mathbf{S},k})(\mathbf{U}_{\mathbf{S},k})^\top\mathbf{Y}^\top, \tag{2}$$

where columns of $\mathbf{U}_{\mathbf{S},k}$ are the first k principal directions of $\mathbf{S}$ s.t. $\mathbf{S}_k = \mathbf{U}_{\mathbf{S},k}\boldsymbol{\Sigma}_{\mathbf{S},k}\mathbf{V}^\top_{\mathbf{S},k}$. We note that $\tilde{\mathbf{K}}^{snm}_k = \mathbf{Y}(\mathbf{U}_{\mathbf{S},k})(\mathbf{U}_{\mathbf{S},k})^\top\mathbf{Y}^\top = \mathbf{C}\mathbf{K}^\dagger_{\mathbf{S},k}\mathbf{C}^\top$. Since SNM approximates $\mathbf{U}_{\mathbf{Y},k}$ as $\mathbf{U}_{\mathbf{S},k}$ Eqn (2) is biased to the sample subspace which is range($\mathbf{S}$). To reduces time complexity $O(ksn + s^3)$ to $O(ksn)$, SNM+Rand.SVD [12] uses randomized SVD to quickly decompose $s \times s$ sample matrix $\mathbf{K}_\mathbf{S}$, and implicitly compute $\tilde{\mathbf{U}}_{\mathbf{S},k}$. Thus, we can consider the implicit equations of SNM+Rand.SVD by replacing $\mathbf{U}_{\mathbf{S},k}$ in Eqn (2) as $\tilde{\mathbf{U}}_{\mathbf{S},k}$.

To efficiently obtain accurate orthonormal eigenvectors in one-shot, orthogonal Nyström method (ONM) was proposed [7]. In fact, rank-k SNM and ONM are identically the same when $k =$ rank($\mathbf{S}$). However, for $k <$ rank($\mathbf{S}$), rank-k SNM and ONM can be distinguished by using the modified approximation formula [13]. The explicit equations of ONM for rank-k approximation are

$$\begin{aligned}\tilde{\boldsymbol{\Sigma}}^{onm}_{\mathbf{Y},k} &= \boldsymbol{\Sigma}_{\mathbf{G},k}, \quad \tilde{\mathbf{V}}^{onm}_{\mathbf{Y},k} = \mathbf{G}\mathbf{V}_{\mathbf{G},k}\boldsymbol{\Sigma}^{-1}_{\mathbf{G},k}, \\ \tilde{\mathbf{K}}^{onm}_k &= \mathbf{G}\mathbf{V}_{\mathbf{G},k}(\mathbf{V}_{\mathbf{G},k})^\top\mathbf{G}^\top,\end{aligned} \tag{3}$$

where $\mathbf{G} = \mathbf{C}\mathbf{V}_\mathbf{S}\boldsymbol{\Sigma}^{-1}_\mathbf{S}$, $\mathbf{K}_\mathbf{S} = \mathbf{V}_\mathbf{S}\boldsymbol{\Sigma}^2_\mathbf{S}\mathbf{V}^\top_\mathbf{S}$ and $\mathbf{G}$ has rank k SVD $\mathbf{G}_k = \mathbf{U}_{\mathbf{G},k}\boldsymbol{\Sigma}_{\mathbf{G},k}\mathbf{V}^{\mathsf{T}}_{\mathbf{G},k}$. The implicit equations of ONM are

$$\tilde{\mathbf{U}}^{onm}_{\mathbf{Y},k} = \mathbf{U}_\mathbf{S}\mathbf{V}_{\mathbf{G},k}, \quad \tilde{\mathbf{K}}^{onm}_k = \mathbf{Y}^\top\tilde{\mathbf{U}}^{onm}_{\mathbf{Y},k}(\tilde{\mathbf{U}}^{onm}_{\mathbf{Y},k})^\top\mathbf{Y}, \tag{4}$$

where $\mathbf{G} = \mathbf{C}\mathbf{V}_\mathbf{S}\boldsymbol{\Sigma}^{-1}_\mathbf{S} = \mathbf{Y}^\top\mathbf{U}_\mathbf{S}$, $\tilde{\mathbf{V}}^{onm}_{\mathbf{Y},k} = \mathbf{G}\mathbf{V}_{\mathbf{G},k}\boldsymbol{\Sigma}^{-1}_{\mathbf{G},k} = \mathbf{Y}^\top\mathbf{U}_\mathbf{S}\mathbf{V}_{\mathbf{G},k}(\boldsymbol{\Sigma}_{\mathbf{G},k})^{-1}$. It is straightforward to verify that $\mathbf{U}_{\mathbf{S},k}$ and $\mathbf{U}_\mathbf{S}\mathbf{V}_{\mathbf{G},k}$ are different when $k < rank(\mathbf{S})$, consequently two approximations of the first k principal directions $\mathbf{U}_{\mathbf{Y},k}$ are different *i.e.*, $\tilde{\mathbf{U}}^{snm}_{\mathbf{Y},k} \neq \tilde{\mathbf{U}}^{onm}_{\mathbf{Y},k}$. Thus, it is again trivial to show that the results of Eqn (2) and Eqn (4) are different when $k < rank(\mathbf{S})$.

The another benefit of ONM is that it solves the sample-based kernel PCA problem (Lem 1) [13]. That is, given sample matrix $\mathbf{S}$, $\tilde{\mathbf{U}}^{onm}_{\mathbf{Y},k}$ minimizes the reconstruction error of kernel PCA with the constraint that approximate principal directions are in the range($\mathbf{S}$). Columns of $\tilde{\mathbf{V}}^{onm}_{\mathbf{Y},k}\tilde{\boldsymbol{\Sigma}}^{onm}_{\mathbf{Y},k}$ are the corresponding principal components.

Lemma 1 *[13] Sample-based kernel PCA is defined as kernel PCA with an additional subspace constraint of* range($\tilde{\mathbf{U}}_{\mathbf{Y},k}$) $\in$ range($\mathbf{S}$)*. Then, ONM is the optimal sample-based kernel PCA.*

Based on Lem 1, double Nyström method (DNM) proposed sample subspace compression, and it uses ONM twice [13]. However, there is no further error analysis regarding a multilayer structure in [13].

We provide Lem 2, a refined version of Lem 1, which states that ONM computes rank-k SVD of $\mathbf{Y}$ such that the computed k principal directions $\tilde{\mathbf{U}}^{onm}_{\mathbf{Y},k}$ minimize reconstruction error of $\mathbf{Y}$ regardless of whether $\mathbf{Y}$ is mean centered or not.

Lemma 2 *Regardless of the condition of mean centering on* $\mathbf{Y}$*, given sample matrix* $\mathbf{S}$*, ONM computes first* k *approximate principal directions* $\tilde{\mathbf{U}}_{\mathbf{Y},k}$ *which minimize reconstruction error of* $\mathbf{Y}$ *with* range($\tilde{\mathbf{U}}_{\mathbf{Y},k}$) $\in$ range($\mathbf{S}$)*, where the reconstruction error of* $\mathbf{Y}$ *is defined as* $\mathrm{RE}(\tilde{\mathbf{U}}_{\mathbf{Y},k}) = \|\mathbf{Y} - \tilde{\mathbf{U}}_{\mathbf{Y},k}\tilde{\mathbf{U}}^\top_{\mathbf{Y},k}\mathbf{Y}\|_\mathrm{F}$.

We will use Lem 2 for the analysis of our method.

3 Nested Nyström Method

If the data size n is large, rank-k SNM and ONM need a relatively larger number of samples s to get accurate spectral decomposition, and the approximation will take longer. We propose NNM which consistently decomposes very small matrices to efficiently update the first k principal directions of $\mathbf{Y}$ and eigendecomposition of $\mathbf{K}$ even though n is large. NNM is described in Alg 1, and its example is displayed in Fig 2, and the detailed description is as follows.

The multilayer architecture of NNM is based on the following three parts: subsampling part, Nyström method part, and compression part. First, we run subsampling part which constructs a nested sequence of subsample matrices and stacks multiple layers with it. Then, we run both Nyström method and compression parts with the nested sequence of subsample matrices on the t sublayers until the final layer. Specifically, NNM updates

Algorithm 1 Nested Nyström Method (NNM)

```
Require: $n \times s$ matrix $\mathbf{C}$ and $s \times s$ matrix $\mathbf{K_S}$, where
    $\mathbf{C} = \mathbf{Y}^\top \mathbf{S}$ and $\mathbf{K_S} = \mathbf{S}^\top \mathbf{S}$, where $s \ll n$
Ensure: rank-$k$ spectral decomposition of $\mathbf{K}$
1: Subsampling part:
    Subsampling indices from the index set $\mathcal{J}$ of $\mathbf{S}$ s.t. $\mathcal{J} \supseteq \mathcal{J}_1 \supseteq ... \supseteq \mathcal{J}_t$, and corresponding $\mathbf{C} \supseteq \mathbf{K_S} \supseteq \mathbf{C}_1 \supseteq \mathbf{K}_{\mathbf{S}_1} \cdots \supseteq \mathbf{C}_t \supseteq \mathbf{K}_{\mathbf{S}_t}$, where $|\mathcal{J}_i| = s_i$, $s \gg s_1 \gg ... \gg s_t$
2: For $i$-th sublayer from the 1st to the $t$-th sublayer:
    Rank-$s_t$ Nyström method: Compute $\tilde{\mathbf{V}}_{\mathbf{S}_{t-i},s_t}$ of $\mathbf{K}_{\mathbf{S}_{t-i}}$ with $\mathbf{C}'_{t-(i-1)}$ and $\mathbf{K}'_{\mathbf{S}_{t-(i-1)}}$ (optional use ONM)
    Compression: Compress sample matrices $\mathbf{C}_{t-i}$ and $\mathbf{K}_{\mathbf{S}_{t-i}}$ as $\mathbf{C}'_{t-i}$ and $\mathbf{K}'_{\mathbf{S}_{t-i}}$ (Eqn (6), Eqn (7))
3: Final layer:
    Run ONM (Eqn (3)) with $\mathbf{C}'$ and $\mathbf{K}'_\mathbf{S}$
```

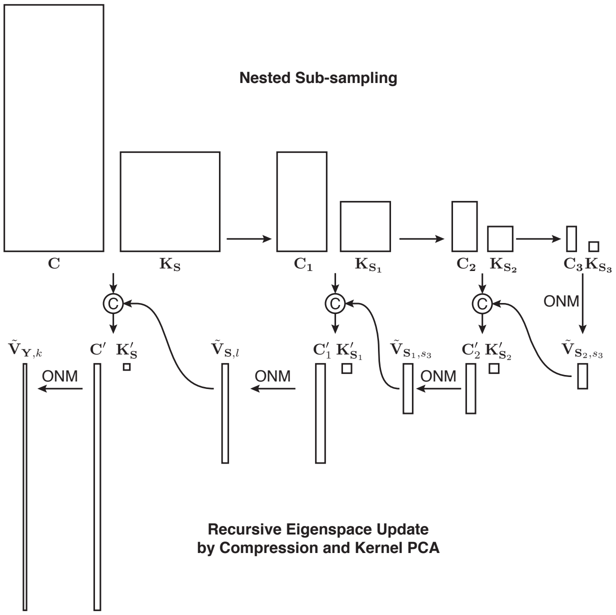

Figure 2: An example of multilayer structure of NNM with 4 layers (3 sublayers and the final layer), which can be described by the explicit equations Eqn (5), Eqn (6) and Eqn (7). Right arrow denotes subsampling. ONM with the arrow denotes that we compute approximate eigenvalues and eigenvectors by using ONM. Circle C means a compression of sample matrices with approximate eigenvectors.

the principal subspace and compresses sample matrices by using the small-dimensional subspace which is compressed and transformed by computed eigenvectors on each sublayer. At the final layer of NNM, we computes the rank-k spectral decomposition by using ONM with the compressed sample matrices, since ONM computes the best principal directions which minimizes reconstruction error given the samples (Lem 1, Lem 2). We note that NNM computes the true rank-k spectral decomposition when the range of compressed samples includes the true rank-k principal subspace, *i.e.*, range($\mathbf{U}_k$) $\subset$ range($\mathbf{S}'$).

Subsampling part: Given indices set $\mathcal{J}$ of s samples and the corresponding sample matrices $\mathbf{S}$ and $\mathbf{K_S}$, we construct a nested index sets $\mathcal{J} \supseteq \mathcal{J}_1 \supseteq ... \supseteq \mathcal{J}_t$ and the corresponding nested sequence of submatrices as Eqn (5).

$$\begin{aligned} &\mathbf{S} \supseteq \mathbf{S}_1 \supseteq \mathbf{S}_2 \supseteq \cdots \supseteq \mathbf{S}_t, \qquad (5) \\ &\mathbf{C} \supseteq \mathbf{K_S} \supseteq \mathbf{C}_1 \supseteq \mathbf{K}_{\mathbf{S}_1} \supseteq \mathbf{C}_2 \supseteq \mathbf{K}_{\mathbf{S}_2} \supseteq \cdots \supseteq \mathbf{C}_t \supseteq \mathbf{K}_{\mathbf{S}_t}, \end{aligned}$$

where $|\mathcal{J}_i| = s_i$, and $s \gg s_1 \gg ... \gg s_t$. Especially, we can understand $(s_{i-1}) \times s_i$ matrix $\mathbf{C}_i$ and $s_i \times s_i$ matrix $\mathbf{K}_{\mathbf{S}_i}$ with implicit equations as $\mathbf{C}_i = \mathbf{S}_{i-1}^\top \mathbf{S}_i$ and $\mathbf{K}_{\mathbf{S}_i} = \mathbf{S}_i^\top \mathbf{S}_i$ for $1 \le i \le t$, where $\mathbf{S}_i$ is $d \times s_i$ and $\mathbf{S}_0 = \mathbf{S}$.

Rank-s_t Nyström method part: In this part, we compute the approximate eigenvectors $\tilde{\mathbf{V}}_{\mathbf{S}_i}$ of $\mathbf{K}_{\mathbf{S}_i}$ by using compressed submatrices $\mathbf{C}'_{\mathbf{S}_{i+1}}$ and $\mathbf{K}'_{\mathbf{S}_{i+1}}$. From the 1st to the $(t-1)$-th sublayer: We compute the first s_t approximate eigenvectors $\tilde{\mathbf{V}}_{\mathbf{S}_i,s_t}$ of $\mathbf{K}_{\mathbf{S}_i}$ by using compressed submatrices $\mathbf{C}'_{i+1}$ and $\mathbf{K}'_{\mathbf{S}_{i+1}}$ on the $(t-i)$-th layer, where $i \in \{1, 2, ..., (t-1)\}$ and $\mathbf{C}'_t = \mathbf{C}_t$ and $\mathbf{K}'_{\mathbf{S}_t} = \mathbf{K}_{\mathbf{S}_t}$. On the t-th sublayer: We compute the first s_t approximate eigenvectors $\tilde{\mathbf{V}}_{\mathbf{S},s_t}$ of $\mathbf{K_S}$ by using $\mathbf{C}'_1$ and $\mathbf{K}'_{\mathbf{S}_1}$, and select $\tilde{\mathbf{V}}_{\mathbf{S},\ell}$ from $\tilde{\mathbf{V}}_{\mathbf{S},s_t}$, where $s_t \ge \ell \ge k$. We recommend using the ONM for rank-s_t Nystrom method at each layer due to both its efficiency and accuracy, especially Lem 1 and Lem 2. Then, the time complexity of rank-s_t Nystrom method part with ONM at i-th layer is $O(s_t^2 s_{t-i})$ for $i \in \{1, 2, ..., (t-1)\}$, and the time complexity of rank-s_t Nystrom method part with ONM at t-th layer is $O(s_t^2 s)$. These time complexities are very small, since $n \gg s \gg s_1 \gg ... \gg s_t \ge \ell \ge k$.

Compression part: In this part, we compress sample matrices by using the approximate eigenvectors. From the 1st to the $(t-1)$-th sublayer: we compress sample matrices $\mathbf{C}_i$ and $\mathbf{K}_{\mathbf{S}_i}$ by using $\tilde{\mathbf{V}}_{\mathbf{S}_i,s_t}$ as

$$\mathbf{C}'_i = \mathbf{C}_i \tilde{\mathbf{V}}_{\mathbf{S}_i,s_t}, \;\; \mathbf{K}'_{\mathbf{S}_i} = (\tilde{\mathbf{V}}_{\mathbf{S}_i,s_t})^\top \mathbf{K}_{\mathbf{S}_i} \tilde{\mathbf{V}}_{\mathbf{S}_i,s_t}, \qquad (6)$$

where $\tilde{\mathbf{V}}_{\mathbf{S}_i,s_t}$ is computed at $(t-i)$-th layer, and $i \in \{1, 2, ..., (t-1)\}$. On the t-th sublayer: we compress sample matrices $\mathbf{C}$ and $\mathbf{K_S}$ by using $\tilde{\mathbf{V}}_{\mathbf{S},\ell}$ with $k \le \ell \le s_t$

$$\mathbf{C}' = \mathbf{C} \tilde{\mathbf{V}}_{\mathbf{S},\ell}, \;\; \mathbf{K}'_\mathbf{S} = (\tilde{\mathbf{V}}_{\mathbf{S},\ell})^\top \mathbf{K_S} \tilde{\mathbf{V}}_{\mathbf{S},\ell}. \qquad (7)$$

We can connect the compression of sample matrices to the compression of sample subspace with implicit equations

$$\mathbf{C}'_i = \mathbf{S}_{i-1}^\top \mathbf{S}'_i,\ \ \mathbf{C}' = \mathbf{Y}^\top \mathbf{S}' \qquad (8)$$
$$\mathbf{K}'_{\mathbf{S}_i} = \mathbf{S}_i'^\top \mathbf{S}'_i,\ \ \mathbf{K}'_{\mathbf{S}} = \mathbf{S}'^\top \mathbf{S}',$$

where $\mathbf{S}'_i = \mathbf{S}_i \tilde{\mathbf{V}}_{\mathbf{S}_i,s_t}, \mathbf{S}' = \mathbf{S}\tilde{\mathbf{V}}_{\mathbf{S},\ell}$, $i \in \{1, 2, ..., (t-1)\}$, and $\mathbf{S}_0 = \mathbf{S}$.

Based on Eqn (8), we can think that the sample subspace range($\mathbf{S}_i$) is compressed into a smaller dimensional subspace range($\mathbf{S}'_i$), where $i \in 0, 1, ..., (t-1)$ and $\mathbf{S}_0 = \mathbf{S}$. From the 1st to the $(t-1)$-th sublayer, we efficiently and accurately update the compressed sample subspace range($\mathbf{S}'_i$) by using the eigenvectors of sample matrices. Since the compressed sample subspace range($\mathbf{S}'_i$) is biased to the principal subspace of s_i subsamples, we preserve the s_t dimension of compressed sample subspace until the t-th sublayer. At the t-th layer, we compress the sample subspace range($\mathbf{S}$) with a smaller dimension ℓ for a shorter running time instead of using s_t, since the principal subspace of s samples is more closer to the rank-k principal subspace of n nodes. That is, we can use $\tilde{\mathbf{V}}_{\mathbf{S},\ell}$ for compression of $\mathbf{S}$ instead of using $\tilde{\mathbf{V}}_{\mathbf{S},s_t}$, where $k \leq \ell \leq s_t$.

The time complexity of compression part at $(t-i)$-th layer is $O(s_t s_i s_{i-1})$ for $i \in \{1, 2, ..., (t-1)\}$, and the time complexity of compression part at t-th layer is $O(\ell sn)$, where $s_0 = s$. We note that all performances in the compression parts are only matrix multiplications.

Time complexity analysis: Suppose that we construct the nested sequence of subsamples with $\sum_{j=1}^t s_j = O(s)$, *e.g.*, $\sum_{j=1}^t s_j \leq s$, where $s \gg s_1 \gg ... \gg s_t \geq \ell \geq k$. Then, we provide Proposition 1 which states the time and space complexities of NNM.

Proposition 1 *Suppose that we use ONM for rank-s_t Nyström method parts in NNM, and set the nested sequence of subsamples with $\sum_{j=1}^t s_j = O(s)$, where $s \gg s_1 \gg ... \gg s_t \geq \ell \geq k$. Then, the total time and space complexities of NNM are $O(\ell sn + s_t s_1 s)$ and $O(sn)$, respectively.*

The detailed proof of Proposition 1 is in the Appendix. A large portion of the total time complexity $O(\ell sn + s_t s_1 s)$ is $O(\ell sn)$ corresponding to the simple matrix multiplications in the compression parts. In Section 5, we will show that the running time of NNM is linear for s.

Selecting number of subsamples and sublayers: We can select the number of subsamples based on the condition $\sum_{j=1}^t s_j = O(s)$ of Proposition 1. We first set M for $M \cdot s \geq \sum_{j=1}^t s_j$. The simple choice of M is 1 or 2, then we have $s \geq \sum_{j=1}^t s_j$ or $2s \geq \sum_{j=1}^t s_j$. We then define a relation among the numbers of subsamples. One of the simplest way is to set $s_i = as_{i-1}$ for $i \in \{1, 2, ..., t\}$, where $s_0 = s$ and $0 < a \leq 1$. A smaller a leads to a shorter running time, but a larger a is better to obtain a small approximation error. To attain both efficiency and accuracy, we can set $a = \frac{1}{2}$, then $\sum_{j=1}^t s_j = \sum_{j=1}^t \frac{1}{2^j} s \leq s$. For example, we can set $s_1 = 1000$, $s_2 = 500$, and $s_3 = 250$ when $s = 2000$ and $t = 3$. We note that s_t and ℓ are tuning parameter, where $s \gg s_1 \gg ... \gg s_t \geq \ell \geq k$. Finally, we note that the proper number of sublayer t should satisfy $M \cdot s \geq \sum_{j=1}^t s_j$ and $s \geq s_j$. For example, suppose that we set $a = \frac{1}{2}$ and $M = 2$, then we have $2 \cdot s \geq \sum_{j=1}^t s_j$ and $s_j = 2^{t-j} s_t$. Then, for $s_t = 250$ and $s \in [2000, 5000]$, the proper number of sublayer t is between 1 and 4, since we have $2 \cdot s \geq \sum_{j=1}^t 2^{t-j} s_t$ and $s \geq 2^{t-j} s_t$ when $t \in \{1, 2, 3, 4\}$.

Several properties of NNM: We note that NNM is a generalized multilayer architecture, not a simple approximation version. For example, NNM with no sublayer is equivalent to ONM, and NNM with one sublayer is equivalent to the DNM. But the main difference is that the upper error bound of NNM further decreases when we decompose the same sized sample matrix with additional layers. That is, we can compute more accurate rank-k decomposition within the same short time. We will show it in Section 5.

We can use any sampling method along with NNM, since NNM does not need any assumption for properties of sample matrices $\mathbf{C}$ and $\mathbf{K}_\mathbf{S}$. Thus, for kernel methods, we can apply any sampling method both for constructing sample matrices and a nested set of subsamples.

We note that it is possible to replace ONM in the rank-s_t Nyström method part with other eigendecomposition methods. However, if we use the ONM in the rank-s_t Nyström method part, then the benefit will be small time complexity, low errors, and easy implementation. Furthermore, it guarantees that the upper error bound of NNM decreases when we use an additional sublayer. We will prove it as Thm 1 in Section 4.

We do not consider rank-k SNM at any layers instead of ONM. Since, if we use the SNM at any layers, we can not guarantee that the error decreases even we use additional sublayers or increase ℓ. We provide a formal statement as Proposition 3 in Section 4.

3.1 Extension of NNM

In this section, we discuss the extension of NNM which is described in Alg 2. Suppose that, given the s samples and NNM with t sublayers, we want to compute s_b additional samples to update the spectral decomposition of $n \times n$ PSD matrix $\mathbf{K}$ by using the s_a extended

Algorithm 2 Extension of NNM with Additional Samples

```
Require: the number of additional samples $s_b$, NNM
    with $t$ sublayers and its inputs and outputs
Ensure: rank-$k$ spectral decomposition of $\mathbf{K}$
    NNM with $(t+1)$ sublayers, additional $s_b$ samples
    (total $s_a = (s + s_b)$ samples),
    appended sample matrices $\mathbf{C}_a$ and $\mathbf{K}_{\mathbf{S}_a}$
1: Additional sampling:
    Sampling additional $s_b$ points
    Constructing appended sample matrices: $n \times s_a$
    matrix $\mathbf{C}_a$ and $s_a \times s_a$ matrix $\mathbf{K}_{\mathbf{S}_a}$
2: $(t+1)$-th sublayer:
    Rank-$s_t$ Nyström method: Compute $\tilde{\mathbf{V}}_{\mathbf{S}_a,\ell}$ of $\mathbf{K}_{\mathbf{S}_a}$
    by using the sample matrices of $\mathbf{K}_{\mathbf{S}_a}$ compressed
    by $\tilde{\mathbf{V}}_{\mathbf{S},s_t}$
    Compression: Compress sample matrices $\mathbf{C}_a$ and
    $\mathbf{K}_{\mathbf{S}_a}$ as $\mathbf{C}'_a$ and $\mathbf{K}'_{\mathbf{S}_a}$ by using $\tilde{\mathbf{V}}_{\mathbf{S}_a,\ell}$ (Eqn (7))
3: Final layer:
    Run ONM (Eqn (3)) with $\mathbf{C}'_a$ and $\mathbf{K}'_{\mathbf{S}_a}$
```

samples, where $s_a = (s + s_b)$. Then, by extending the multilayer structure of NNM, we can efficiently update the spectral decomposition. The extension of NNM which consists of three components: additional sampling, $(t+1)$-th sublayer, and the final layer. The followings are description of NNM.

Additional sampling: We can use either uniform or non-uniform sampling. For non-uniform sampling, we can efficiently compute s_b additional samples by using the rank-k spectral decomposition obtained from NNM with t layers, *e.g.*, approximate column norm sampling [5], approximate leverage score sampling [13], and adaptive partial sampling [11]. The implicit equation of s_a samples is $\mathbf{S}_a = \begin{bmatrix}\mathbf{S} & \mathbf{S}_b\end{bmatrix}$, and the implicit equations of appended $n \times s_a$ sample matrix $\mathbf{C}_a$ and $s_a \times s_a$ sample matrix $\mathbf{K}_{\mathbf{S}_a}$ are $\mathbf{C}_a = \mathbf{Y}^\top \mathbf{S}_a$ and $\mathbf{K}_{\mathbf{S}_a} = \mathbf{S}_a^\top \mathbf{S}_a$, respectively.

$(t+1)$**-th sublayer:** In the rank-s_t Nyström method part, we efficiently compute $\tilde{\mathbf{V}}_{\mathbf{S}_a,\ell}$ of $\mathbf{K}_{\mathbf{S}_a}$. Let $\mathbf{C}_0$ and $\mathbf{K}_\mathbf{S}$ be the $s_a \times s$ and $s \times s$ sample matrices of $\mathbf{K}_{\mathbf{S}_a}$, respectively. Then, we compress $\mathbf{C}_0$ and $\mathbf{K}_\mathbf{S}$ as $\mathbf{C}'_0$ and $\mathbf{K}'_\mathbf{S}$ by using $\tilde{\mathbf{V}}_{\mathbf{S},s_t}$ which was computed at the t sublayer of NNM, and compute $\tilde{\mathbf{V}}_{\mathbf{S}_a,\ell}$ of $\mathbf{K}_{\mathbf{S}_a}$ by using ONM and the compressed sample matrices $\mathbf{C}'_0$ and $\mathbf{K}'_\mathbf{S}$. In the compression part, we compress sample matrices $\mathbf{C}_a$ and $\mathbf{K}_{\mathbf{S}_a}$ as $\mathbf{C}'_a$ and $\mathbf{K}'_{\mathbf{S}_a}$ by using $\tilde{\mathbf{V}}_{\mathbf{S}_a,\ell}$

Final layer: We compute $\tilde{\mathbf{V}}_k$ by using $\mathbf{C}'_a$, $\mathbf{K}'_{\mathbf{S}_a}$ and ONM.

The time complexity of rank-s_t Nyström method part using ONM in $(t+1)$-th sublayer is $O(s_t^2 s_a)$, and the time complexity of compression part is $O(\ell s_a n)$. At the final layer, the time complexity of ONM with $\mathbf{C}'_a$ and $\mathbf{K}'_{\mathbf{S}_a}$ is $O(\ell^2 n)$.

We can combine different sampling strategies for computing spectral decomposition of PSD matrices by using the extension of NNM. For example, we can easily combine uniform and approximate leverage score sampling, since the time complexity of computing approximate leverage scores using $\tilde{\mathbf{V}}_k$ computed by NNM with t sublayers is just $O(kn)$. Then, the total time complexity of NNM with (t+1) sublayers is $O(\ell s_a n + s_t s_a s)$ which is linear for s_a when $(s + \sum_{j=1}^{t} s_j) = O(s_a)$. In Section 5, we will compare the experimental results of NNM by using uniform sampling and uniform + approximate leverage score sampling. Finally, we note that we can use multi-sublayers between t-th sublayer and the final layer, if s_a/s is too large.

4 Error Analysis of NNM

In this section, we provide an error analysis of NNM. First, we provide the implicit representations of compressed sample subspaces to analyze the error of NNM. Next, we show the upper error bounds of NNM, and prove that the upper error bounds decreases when we use additional sublayers.

4.1 Representations of Compressed Sample Subspaces

NNM efficiently and accurately updates the compressed sample matrix $\mathbf{S}'_i$ so that range($\mathbf{S}'_i$) closely approximates the true principal subspace based on Eqn (8) until the final layer. That is, we want to compute $\mathbf{S}'$ s.t. range($\mathbf{U}_k$) $\subset$ range($\mathbf{S}'$). Consequently, we need to analyze how compressed sample subspace varies range($\mathbf{S}'_i$) through the multilayer structure to analyze NNM.

First, we provide the implicit representation of the principal subspace as range($\mathbf{U}_k$) = range($\mathbf{U}_k \mathbf{\Sigma}_{\mathbf{Y},k}$) = range($\mathbf{Y}\mathbf{V}_{\mathbf{Y},k}$). Similarly, we can give the implicit representations of compressed sample subspaces range($\mathbf{S}'_i$) and range($\mathbf{S}'$) as range($\mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},s_t}$) and range($\mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},\ell}$), respectively. Lem 3 formally states their implicit representations.

Lemma 3 *Given the mutilayer Nyström structure of NNM with t sublayers, let $\mathbf{S} = \mathbf{Y}\mathbf{P}$ and $\mathbf{S}_i = \mathbf{S}_{i-1}\mathbf{P}_i$, where $\mathbf{P}_i^\top \mathbf{P}_i = \mathbf{I}$ for column sampling, $\mathbf{P}_0 = \mathbf{P}$, and $\mathbf{S}_0 = \mathbf{S}$. Then, we have $\mathbf{S}_i = \mathbf{Y}\mathbf{Z}_i$ with $\mathbf{Z}_i = \mathbf{P}\mathbf{P}_1 \cdots \mathbf{P}_i$ and $\mathbf{S}'_i = \mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},s_t}$ on the $(t-i)$-th layer, and $\mathbf{S}' = \mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},\ell}$ on the t-th layer, where $\tilde{\mathbf{V}}_{\mathbf{Y},s_t} = \mathbf{Z}_i \tilde{\mathbf{V}}_{\mathbf{S}_i,s_t}$ and $\tilde{\mathbf{V}}_{\mathbf{Y},\ell} = \mathbf{P}\tilde{\mathbf{V}}_{\mathbf{S},\ell}$.*

By Lem 3, we note that if range($\mathbf{Y}\mathbf{V}_{\mathbf{Y},k}$) $\subset$ range($\mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},\ell}$) = range($\mathbf{S}'$), then NNM computes the rank-k spectral decomposition with the optimal error.

4.2 Decrease of Upper Error Bounds of NNM

For the case of using linear combination input $\mathbf{S} = \mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},s}$ with $\tilde{\mathbf{V}}_{\mathbf{Y},s}^\top \tilde{\mathbf{V}}_{\mathbf{Y},s} = \mathbf{I}$, the generalized upper error bounds of ONM have been proven [13]. Since the input of the final layer of NNM can be considered as $\mathbf{S}' = \mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},\ell}$ by Lem 3, we provide Lem 4 which states the upper bounds of the final error of NNM.

Lemma 4 *Suppose that $\mathbf{S}' = \mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},\ell}$ is the compressed samples as an input of the final layer of NNM, where $\tilde{\mathbf{V}}_{\mathbf{Y},\ell}^\top \tilde{\mathbf{V}}_{\mathbf{Y},\ell} = \mathbf{I}$ and $k \leq \ell \leq s_t$. Then, upper error bounds of NNM for kernel PCA and rank-k matrix approximation are*

$$\mathrm{NRE}(\tilde{\mathbf{U}}_{\mathbf{Y},k}) \leq \mathrm{NRE}(\mathbf{U}_{\mathbf{Y},k}) + \sqrt{\frac{2\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})}{\gamma_k}} \qquad (9)$$

$$\mathrm{MRE}(\tilde{\mathbf{K}}_k) \leq \mathrm{MRE}(\mathbf{K}_k) + \sqrt{\frac{2\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})}{\gamma_k}}\,\mathrm{tr}(\mathbf{K}),$$

where $\tilde{\mathbf{V}}_{\mathbf{Y},k}$ is any submatrix consisting of k columns of $\tilde{\mathbf{V}}_{\mathbf{Y},\ell}$, $\tilde{\mathbf{U}}_{\mathbf{Y},k}$ consists of the first k approximate principal directions which are implicitly generated by kernel PCA, $\mathrm{NRE}(\tilde{\mathbf{U}}_{\mathbf{Y},k}) = \|\mathbf{Y} - \tilde{\mathbf{U}}_{\mathbf{Y},k}\tilde{\mathbf{U}}_{\mathbf{Y},k}^\top \mathbf{Y}\|_{\mathrm{F}}/\|\mathbf{Y}\|_{\mathrm{F}}$ is the normalized reconstruction error (NRE) of kernel PCA, $\mathrm{MRE}(\tilde{\mathbf{K}}_k) = \|\mathbf{K} - \tilde{\mathbf{K}}_k\|_{\mathrm{F}}$ is the reconstruction error of rank-k PSD matrix approximation, γ_k is the k-th eigengap, $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$ is the sum of errors of eigenvalues from $\tilde{\mathbf{V}}_{\mathbf{Y},k}$ with $\tilde{\mathbf{V}}_{\mathbf{Y},k}^\top \tilde{\mathbf{V}}_{\mathbf{Y},k} = \mathbf{I}$ s.t. $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k}) = \mathrm{tr}(\mathbf{V}_{\mathbf{Y},k}^\top \mathbf{Y}^\top \mathbf{Y}\mathbf{V}_{\mathbf{Y},k}) - \mathrm{tr}(\tilde{\mathbf{V}}_{\mathbf{Y},k}^\top \mathbf{Y}^\top \mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},k})$.

These upper error bounds in Lem 4 only depend on $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$, since γ_k and $\mathrm{tr}(\mathbf{K})$ are constant given the $\mathbf{K}$. The approximation errors in Eqn (9) go to the optimal errors which are $\mathrm{NRE}(\mathbf{U}_{\mathbf{Y},k})$ and $\mathrm{MRE}(\mathbf{K}_k)$ as $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$ goes to 0, and $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k}) = 0$ when $\mathrm{range}(\mathbf{V}_{\mathbf{Y},k}) \subset \mathrm{range}(\tilde{\mathbf{V}}_{\mathbf{Y},\ell})$. Since reducing $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$ is important, we need to show how $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$ varies through the sublayers.

Suppose that, given the mutilayer structure of NNM with t sublayers, we use ONM for kernel PCA parts and only i sublayers which are from the first sublayer to i-th sublayer until the final layer, where $i \in \{1, 2, ..., (t-1)\}$. Then, the i-th sublayer becomes the last sublayer before the final layer, and we have $\mathbf{S}'_j = \mathbf{Y}(\mathbf{Z}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,\ell})$ as the input of the final layer with $j = (t-i)$ by Lem 3 and Lem 4. Thus, we can select $\tilde{\mathbf{V}}_{\mathbf{Y},k} = \mathbf{Z}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k}$ for $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$, *i.e.*, $\epsilon_2(\mathbf{Z}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k})$ since $\tilde{\mathbf{V}}_{\mathbf{Y},\ell} = \mathbf{Z}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,\ell}$.

Proposition 2 states that the sum of eigenvalue error $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$ decreases as we use additional sublayers.

Table 2: The summary of 4 real data sets. n is the number of instances and m is the dimension of the original data.

data set	Letter	MNIST	MiniBooNE	Covertype
n	20000	60000	130064	581012
m	16	784	50	54

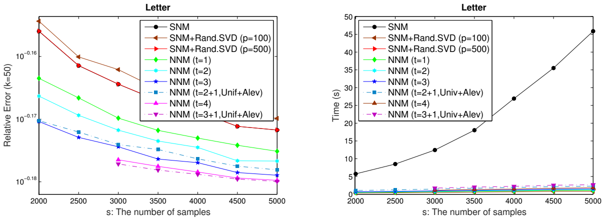

Figure 3: Performance comparison of SNM, SNM+Rand.SVD, and NNM. The upper left figure displays the results for rank-k kernel matrix approximation, and the upper right figure displays their running time. The results show that the error of NNM decreases as we use additional sublayers within the short time.

Proposition 2 *Suppose that we use ONM for kernel PCA parts in NNM. Then, we have $\epsilon_2(\mathbf{Z}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k})$ and $\epsilon_2(\mathbf{Z}_{j-1} \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_{j-1},k})$ for NNM using i and $(i+1)$ sublayers respectively, and $\epsilon_2(\mathbf{Z}_{j-1} \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_{j-1},k}) \leq \epsilon_2(\mathbf{Z}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k})$.*

Due to the page limit, the proofs are in the Appendix. By Lem 4 and Proposition 2, we provide our main theoretical result Thm 1 which states that the quality of compressed input at the final layer is important, and we can increase accuracy by using more sublayers.

Theorem 1 *Suppose that we use ONM for the kernel PCA parts in the sublayers. Then, the upper error bound of NNM in Lem 4 decreases when we use additional sublayers.*

Finally, Proposition 3 states that rank-k SNM does not have the benefit for using additional sublayers or increasing ℓ parameter.

Proposition 3 *Suppose that we compress $\mathbf{C}$ and $\mathbf{K}_{\mathbf{S}}$ as $\mathbf{C}'$ and $\mathbf{K}'_{\mathbf{S}}$ with $\tilde{\mathbf{V}}_{\mathbf{S},\ell} = \mathbf{V}_{\mathbf{S},\ell}$ at the t-th layer, and we run the rank-k SNM with $\mathbf{C}'$ and $\mathbf{K}'_{\mathbf{S}}$ at the final layer. Then $\tilde{\mathbf{K}}^{snm}_k$ are the same regardless of values of ℓ, where $\ell \geq k$. That is, the rank-k spectral decomposition using rank-k SNM is biased by s samples.*

5 Experiments

In this section, we present experimental results that demonstrate our theoretical work. We compare rank-k Nyström methods to the rank-k kernel matrix approximation and KPCA. The three error measures which we used are *matrix reconstruction error* $(\mathrm{MRE}(\tilde{\mathbf{K}}_k) =$

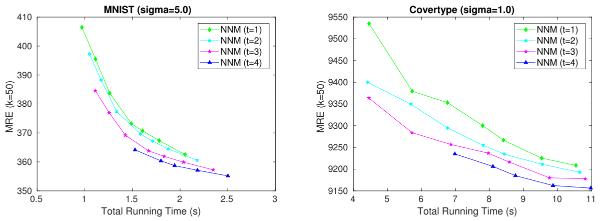

Figure 4: Performance comparison for rank-k kernel matrix approximation among the NNM with 1, 2, 3, 4 sublayers. NNM ($t = 4$) is more efficient than NNM ($t = 1, 2, 3$).

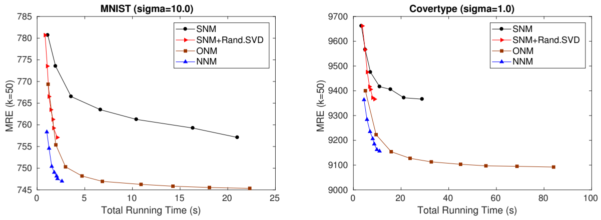

Figure 5: Comparison of MRE($\tilde{\mathbf{K}}_k$) for rank-k kernel matrix approximation among the four representative methods with: SNM, SNM+Rand.SVD, ONM, NNM (ours). NNM is more efficient than other state-of-the art Nyström methods given the short time.

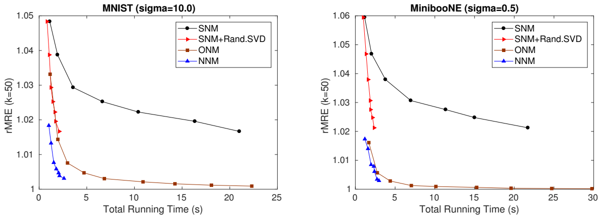

Figure 6: Comparison of convergence to the optimal error with rMRE($\tilde{\mathbf{K}}_k$) for rank-k kernel matrix approximation. It shows that error of NNM rapidly decreases compared to other Nyström methods.

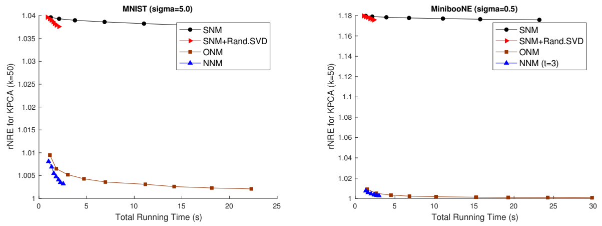

Figure 7: Comparison of convergence to the optimal error with rNRE($\tilde{\mathbf{K}}_k$) for KPCA. It shows that reconstruction error of KPCA of NNM rapidly decreases compared to other Nyström methods.

$\|\mathbf{K} - \tilde{\mathbf{K}}_k\|_{\mathrm{F}}$), *relative matrix reconstruction error* (rMRE($\tilde{\mathbf{K}}_k$) $= \frac{\|\mathbf{K}-\tilde{\mathbf{K}}_k\|_{\mathrm{F}}}{\|\mathbf{K}-\mathbf{K}_k\|_{\mathrm{F}}} \in [1,\infty)$), and *relative KPCA reconstruction error* (rNRE($\tilde{\mathbf{U}}_k$) $= \frac{\mathrm{NRE}(\tilde{\mathbf{U}}_k)}{\mathrm{NRE}(\mathbf{U}_k)} \in [1,\infty)$), where $\|\mathbf{K}-\mathbf{K}_k\|_{\mathrm{F}}$ and NRE($\mathbf{U}_k$) are the optimal error which comes from SVD. The optimum of rMRE and rNRE is 1. To construct PSD matrix $\mathbf{K}$, we use RBF kernel which is defined as $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|_2^2}{2\sigma^2}\right)$, where σ is a kernel parameter. We select 4 real data sets for evaluating performances, and summarize them in Tbl 2.

We empirically compare the NNM described in Alg 1 with four representative Nyström methods: SNM [5], SNM+Rand.SVD [12], ONM [7], and DNM [13]. We abbreviate NNM with i sublayers to NNM ($t = i$) for convenience, and DNM is the same with NNM ($t = 1$). For all methods except NNM ($t = 4$), we set $s = 500j$ with $j = 4, 5, ..., 10$ in Fig 3. We set $s = 3000, 3500, ..., 5000$ for NNM ($t = 4$). We use the same amount of sn kernel matrix elements for all methods.

We set p parameter as 100, 500 for SNM+Rand.SVD, since the error of SNM+Rand.SVD decreases to the error of SNM when $p = 500$ regardless of data sets and experiment settings. We report the additional experimental results with $p = 500, 1000, 2000$ in the Appendix. To compare NNM with SNM+Rand.SVD, we use the following parameters: $s_1 = (1000 + 50j)$, $s_2 = (500+25j)$, $s_3 = (250+25j)$, $\ell = (k+150+5j)$ for $s = 2000, 2500$, and $s_1 = (2000+50j)$, $s_2 = (1000+50j)$, $s_3 = (500+25j)$, $s_4 = (250+25j)$, $\ell = (k+150+5j)$ for $s \geq 3000$. With these parameters, we have $s_j \approx 2^{t-j}s_t$, and set the maximum number of sublayers t as 4 to satisfy $2 \cdot s \geq \sum_{j=1}^{t} 2^{t-j}s_t$ and $s \geq 2^{t-j}s_t$ (see the paragraph of selecting t in Section 3).

In Fig 3, we can see that the errors of NNM are smaller than SNM and SNM+Rand.SVD, and the errors of NNM further decrease as we use additional sublayers within the short time. We also run NNM by combining uniform sampling (Unif) and approximate leverage score sampling (ALev) based on the extension of NNM. For example, ($t = 2 + 1$,Unif+Alev) means that we run NNM with 2 sublayers by using Unif, and extend NNM with 1 sublayers by using ALev. In Fig 3, although the error of NNM ($t = 2 + 1$,Unif+Alev) is higher than NNM ($t = 3$), the error of NNM ($t = 3 + 1$,Unif+Alev) is smaller than the error of NNM ($t = 4$), since the accuracy of approximate leverage scores computed by NNM increases as we use more sublayers. Fig 4 shows that the error of NNM decreases as we use additional sublayers regardless of data sets. We can see that NNM ($t = 3, 4$) sublayers are more accurate than NNM ($t = 1, 2$) within the same short time. Fig 5 shows that the errors of NNM are smaller than errors of other state-of-the art Nyström methods within the same short time. Fig 6 and Fig 7 show that the errors of NNM both for rank-k kernel matrix approximation and KPCA rapidly decrease compared to other rank-k Nyström methods.

6 Conclusion

In this paper, we presented a multi-scale Nyström architecture, called nested Nyström Method (NNM), which efficiently and accurately updates the rank-k spectral decomposition of PSD matrix on the multilayer structure with the nested sequence of subsamples and subspace compression. Both theoretically and empirically, we demonstrated that the error of NNM decreases as we use additional layers. Finally, we showed that NNM is more efficient than other rank-k Nyström methods.

7 Acknowledgments

W. Lim acknowledges support from 2016-2017 Google Ph.D. Fellowship in machine learning. W. Lim and R. Du acknowledge Qingqing Liu who is a Ph.D. student in School of Mathematics at Georgia Tech and verified the proofs of theorem in our paper. K. Jung was supported by the National Research Foundation of Korea(NRF)(No.2016R1A2B2009759). L. Song and B. Dai were supported in part by NSF CAREER IIS-1350983, NSF IIS-1639792 EAGER, NSF CNS-1704701, ONR N00014-15-1-2340, Intel ISTC, NVIDIA and Amazon AWS.

References

[1] Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[2] Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *Proceedings of ECCV*, pages 354–370. Springer, 2016.

[3] Amit Deshpande, Luis Rademacher, Santosh Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. *Theory of Computing*, 2:225–247, 2006.

[4] Petros Drineas, Malik Magdon-Ismail, Michael W. Mahoney, and David P Woodruff. Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, 13:3475–3506, 2012.

[5] Petros Drineas and Michael W. Mahoney. On the Nyström method for approximating a gram matrix for improved kernel-based learning. *The Journal of Machine Learning Research*, 6:2153–2175, 2005.

[6] Yalchin Efendiev, Thomas Y Hou, Victor Ginting, et al. Multiscale finite element methods for nonlinear problems and their applications. *Communications in Mathematical Sciences*, 2(4):553–589, 2004.

[7] Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the Nyström method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):214–225, 2004.

[8] Alex Gittens and Michael W. Mahoney. Revisiting the Nyström method for improved large-scale machine learning. In *Proceedings of ICML*, 2013.

[9] Roger A Horn and Charles R Johnson. *Topics in matrix analysis.* Cambridge Univ. Press, 1991.

[10] Cho-Jui Hsieh, Si Si, and Inderjit S Dhillon. Fast prediction for large-scale kernel machines. In *Proceeding of NIPS*, pages 3689–3697, 2014.

[11] Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Sampling methods for the Nyström method. *The Journal of Machine Learning Research*, 98888:981–1006, 2012.

[12] Mu Li, James Tin-Yau Kwok, and Baoliang Lü. Making large-scale nyström approximation possible. In *ICML 2010-Proceedings, 27th International Conference on Machine Learning*, page 631, 2010.

[13] Woosang Lim, Minhwan Kim, Haesun Park, and Kyomin Jung. Double Nyström method: An efficient and accurate Nyström scheme for large-scale data sets. In *Proceedings of ICML*, pages 1367–1375, 2015.

[14] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of CVPR*, 2017.

[15] Michael W. Mahoney and Petros Drineas. Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009.

[16] Francesca Petralia, Joshua T Vogelstein, and David B Dunson. Multiscale dictionary learning for estimating conditional distributions. In *Proceedings of NIPS*, pages 1797–1805, 2013.

[17] Jan Reininghaus, Stefan Huber, Ulrich Bauer, and Roland Kwitt. A stable multi-scale kernel for topological machine learning. In *Proceedings of CVPR*, pages 4741–4748, 2015.

[18] Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco. Less is more: Nyström computational regularization. In *Proceedings of NIPS*, pages 1657–1665, 2015.

[19] Chang Wang and Sridhar Mahadevan. Multiscale manifold learning. In *Proceedings of AAAI*, 2013.

[20] Shusen Wang and Zhihua Zhang. Improving CUR matrix decomposition and the Nyström approximation via adaptive sampling. *The Journal of Machine Learning Research*, 14(1):2729–2769, 2013.

[21] Ziyu Wang, Babak Shakibi, Lin Jin, and Nando Freitas. Bayesian multi-scale optimistic optimization. In *Proceedings of AISTATS*, pages 1005–1014, 2014.

[22] Christopher Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In *Proceedings of NIPS*, 2001.

[23] Kai Zhang and James T. Kwok. Clustered Nyström method for large scale manifold learning and dimension reduction. *IEEE Transactions on Neural Networks*, pages 1576–1587, 2010.

A Additional Experimental Results

In this section, we provide the experimental results with various setting: different sigma, different rank-k, different ℓ, and etc. We empirically compare the NNM described in Alg 1 with four representative Nyström methods: standard Nyström method (SNM) [22], standard Nyström method using randomized SVD (SNM+Rand.SVD) [12], one-shot Nyström method (ONM) [7], and double Nyström method (DNM) [13]. We note that DNM is equivalent to NNM with 1 sublayer. For convenience, we abbreviate NNM with i sublayers to NNM ($t = i$). If we consider the final layer, then NNM ($t = i$) has $(i + 1)$ total layers. For all methods, we set $s = 500j$, where $j = 4, 5, ..., 10$. Thus, there are 7 episodes for each test and the corresponding 7 points on the each line in the figures. We use the same amount of sn kernel matrix elements for all methods. We report only the decomposition time in this section, since we report the running time including kernel construction time in the main section (Section 5).

We tested the value of p parameter of SNM+Rand.SVD from 5 to 2000, since the errors of SNM+Rand.SVD decrease to the errors of SNM as we increase the value p parameter. We report the experimental results of SNM+Rand.SVD with $p = 500$, since the errors of SNM+Rand.SVD decrease close to the errors of SNM regardless of data sets and experiment settings when we increase p as 500. We observed that the errors of SNM+Rand.SVD did not further decrease even if we increased the value of p parameter over 500 in the experiments, *e.g.*, $p = 1000, 2000$, since the errors of SNM+Rand.SVD converge to the errors of SNM as we increase p parameter. To compare NNM with SNM+Rand.SVD, we use the following parameters for Fig 8 Fig 9 and Fig 10: For NNM with $t = 3$, we set $s_1 = (1000 + 50j)$, $s_2 = (500 + 25j)$, $s_3 = (250 + 25j)$, $\ell = (k + 80 + 5j)$, where $j = 4, 5, ..., 10$. For NNM with $t = 2$, we set $s_1 = (500 + 25j)$, $s_2 = (250 + 25j)$, $\ell = (k + 80 + 5j)$. For NNM with $t = 1$, we set $s_1 = (250 + 25j)$, $\ell = (k + 80 + 5j)$. In Fig 8 Fig 9 and Fig 10, we exclude the kernel construction time for report the running time. Meanwhile, in Fig 3 and Fig 4 in Section 5, we reported the running time including kernel construction time.

Fig 8 displays the experimental results on 4 different real data sets. In this experiment, we set $k = 20$ instead of $k = 50$. We set σ for 4 data sets as follows: $\sigma = 1.0$ for Letter, $\sigma = 5.0$ for MNIST, $\sigma = 1.0$ for MiniBooNE, and $\sigma = 1.0$ for Covertype. For our method, we display the experimental results of NNM with 3 sublayers. Fig 8 shows that NNM computes more accurate rank-k matrix approximation compared to the other Nyström methods within the same time. Especially, for Letter and MNIST data sets, we can notice that the error of NNM rapidly decreases to the optimal error. Although SNM+Rand.SVD is slightly

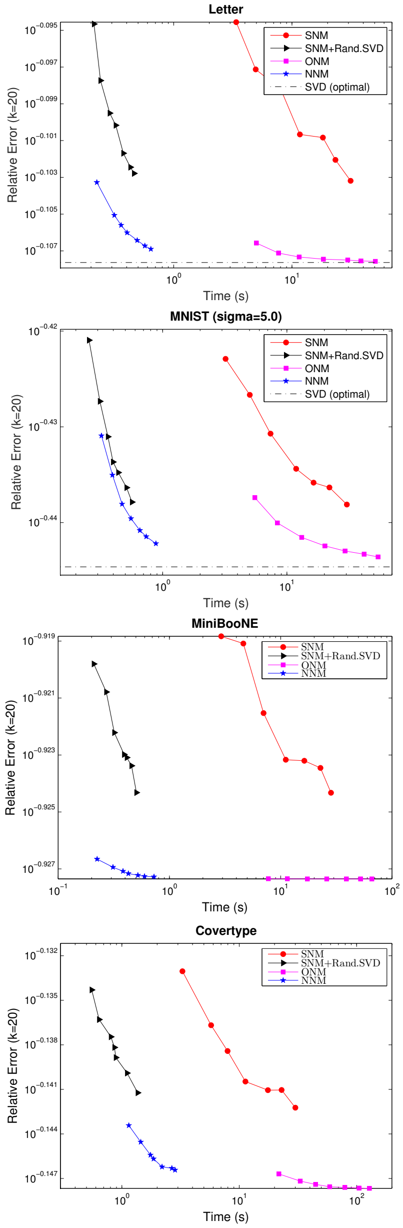

Figure 8: Performance comparison with $k = 20$ among the four representative methods: SNM [22], ONM [7], SNM + Rand.SVD [12], and NNM (ours). There are 7 episodes for each test, and there are 7 points on the each line in the figures. That is, we gradually increase the number of samples s as 2000, 2500, 3000, ... , 5000, and there are corresponding 7 points on the each line. We perform SVD algorithm only on the Letter and MNIST data sets due to the memory limit. The results show that NNM is more accurate than other Nyström methods within the same short time.

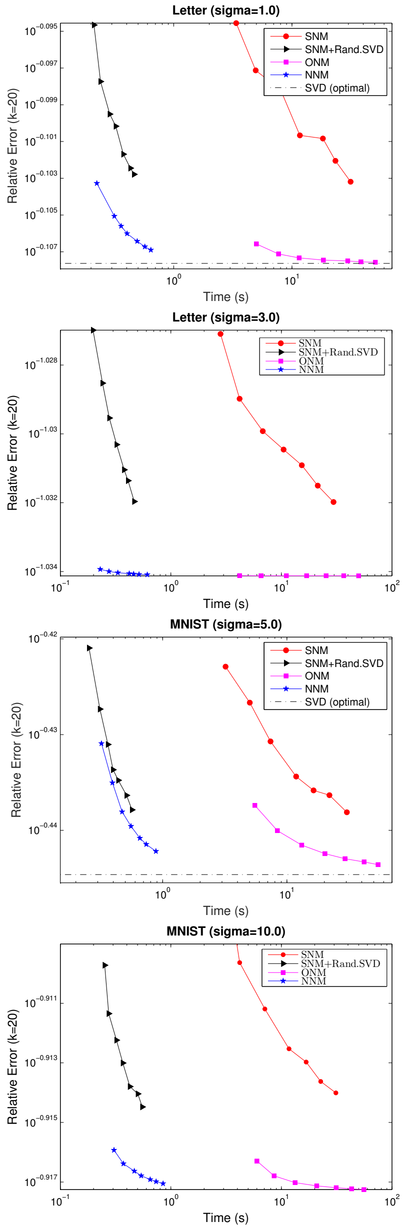

Figure 9: Performance comparison with the different sigma values of kernel function. In this experiment, we display the experimental results of NNM with 3 sublayers. The results show that the errors of NNM is smaller than other Nyström methods within the same short time regardless of sigma values.

faster, the errors of NNM are much smaller than the errors of SNM+Rand.SVD. ONM is the accurate, but it takes much longer. Fig 9 also shows that NNM is both accurate and efficient regardless of the sigma values of kernel function.

Fig 10 shows that the errors of SNM+Rand.SVD do not further decrease even if we increase the value of p parameter over 500 in the experiments, *e.g.*, $p = 1000, 2000$. We can also notice that the errors of SNM+Rand.SVD converge to the errors of SNM.

B Proof of Proposition 1

Proof 1 *For the proof of Proposition 1, we use $\sum_{j=1}^{t} s_j = O(s)$ and properties of big O notation. The time complexity of kernel PCA part with ONM at i-th layer is $O(s_t^2 s_{t-i})$ for $i \in \{1, 2, ..., (t-1)\}$, and the time complexity of kernel PCA part with ONM at t-th layer is $O(s_t^2 s)$. Then, the sum of time complexities in kernel PCA parts is bounded as*

$$s_t^2 s + \sum_{i=1}^{t-1} s_t^2 s_{t-i} \leq (M+1) s_t^2 s,$$

where $\sum_{j=1}^{t} s_j \leq M \cdot s$ and we usually set $M \leq 2$ in the experiments. Thus, by the properties of big O notation, the total time complexity of kernel PCA parts using ONM is $O(s_t^2 s)$.

The time complexity of compression part at $(t-i)$-th layer is $O(s_t s_i s_{i-1})$ for $i \in \{1, 2, ..., (t-1)\}$, and the time complexity of compression part at t-th layer is $O(\ell s n)$, where $s_0 = s$. Then, the sum of time complexities in compression parts is bounded as

$$\begin{aligned} \ell s n + \sum_{i=1}^{t-1} s_t s_i s_{i-1} &= \ell s n + s_t s_1 s + \sum_{i=1}^{t-2} s_t s_{i+1} s_i \\ &\leq \ell s n + (M+1) s_t s_1 s, \end{aligned}$$

where $s_0 = s$ and $\sum_{j=1}^{t} s_j \leq M \cdot s$. Thus, the total time complexity of kernel PCA parts is $O(\ell s n + s_t s_1 s)$.

Finally, at the final layer, the time complexity of ONM with $\mathbf{C}'$ and $\mathbf{K}'_{\mathbf{S}}$ is $O(\ell^2 n)$, since $\mathbf{C}'$ is $n \times \ell$ and $\mathbf{K}'_{\mathbf{S}}$ is $\ell \times \ell$. Therefore, the total time complexity of NNM is $O(\ell s n + s_t s_1 s)$.

C Proof of Lem 3

Lem 3 formally states implicit representations of compressed subspaces.

Lemma 3: Given the mutilayer Nyström structure of NNM with t sublayers, let $\mathbf{S} = \mathbf{YP}$ and $\mathbf{S}_i = \mathbf{S}_{i-1}\mathbf{P}_i$, where $\mathbf{P}_i^\top \mathbf{P}_i = \mathbf{I}$ for column sampling, $\mathbf{P}_0 = \mathbf{P}$, and $\mathbf{S}_0 = \mathbf{S}$. Then, we have $\mathbf{S}_i = \mathbf{YZ}_i$ with $\mathbf{Z}_i = \mathbf{PP}_1 \cdots \mathbf{P}_i$ and $\mathbf{S}'_i = \mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},s_t}$ on the $(t-i)$-th layer, and $\mathbf{S}' = \mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},\ell}$ on the t-th layer, where $\tilde{\mathbf{V}}_{\mathbf{Y},s_t} = \mathbf{Z}_i \tilde{\mathbf{V}}_{\mathbf{S}_i,s_t}$ and $\tilde{\mathbf{V}}_{\mathbf{Y},\ell} = \mathbf{P}\tilde{\mathbf{V}}_{\mathbf{S},\ell}$.

Proof 2 *For a general case, we assume that a mutilayer Nyström structure of NNM has t sublayers with the nested sequence of samples in Eqn (5). Without loss of generality, we can define $\mathbf{P}_j$ for column sampling s.t. $\mathbf{S} = \mathbf{YP}$ and $\mathbf{S}_i = \mathbf{S}_{i-1}\mathbf{P}_i$ with $(\mathbf{P}_i)^\top \mathbf{P}_i = \mathbf{I}$, where*

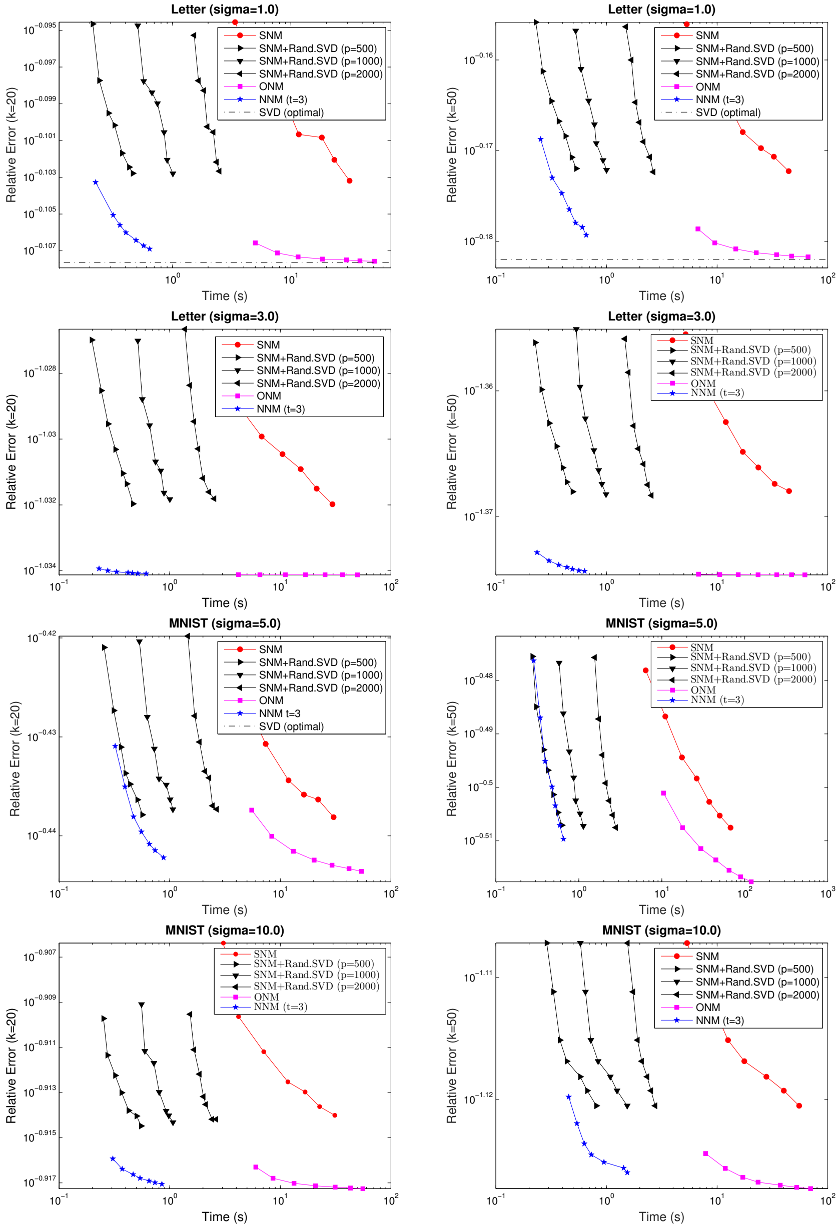

Figure 10: Performance comparison for $k = 20, 50$ with different sigma values among the four representative methods: SNM [22], ONM [7], SNM + Rand.SVD [12], and NNM (ours). There are 7 episodes for each test, and there are 7 points on the each line in the figures. We gradually increase the number of samples s as 2000, 2500, 3000,..., 5000, and there are corresponding 7 points on the each line. Regardless of rank-k and sigma values, experimental results show that the NNM is more accurate than other Nyström methods within the same short time.

$\mathbf{P}_0 = \mathbf{P}$, *and* $\mathbf{S}_0 = \mathbf{S}$. *Then, we have* $\mathbf{S}_i = \mathbf{Y}\mathbf{Z}_i$, *where* $\mathbf{Z}_j = \mathbf{P}\mathbf{P}_1 \cdots \mathbf{P}_j$ *and* $\mathbf{Z}_j = \mathbf{P}$.

By Eqn (6), Eqn (7) and Eqn (8), we have $\mathbf{S}'_i = \mathbf{S}_i\tilde{\mathbf{V}}_{\mathbf{S}_i,s_t}$ *on the* $(t-i)$*-th sublayer for* $i \in \{1, ..., (t-1)\}$ *and* $\mathbf{S}' = \mathbf{S}\tilde{\mathbf{V}}_{\mathbf{S},\ell}$ *on the* t*-th sublayer. Thus, if we apply* $\mathbf{S}_i = \mathbf{Y}\mathbf{Z}_i$ *to the* $\mathbf{S}'_i = \mathbf{S}_i\tilde{\mathbf{V}}_{\mathbf{S}_i,s_t}$ *and* $\mathbf{S}' = \mathbf{S}\tilde{\mathbf{V}}_{\mathbf{S},\ell}$, *then we have* $\mathbf{S}'_i = \mathbf{Y}\mathbf{Z}_i\tilde{\mathbf{V}}_{\mathbf{S}_i,s_t}$ *on the* $(t-i)$*-th layer, and* $\mathbf{S}' = \mathbf{Y}\mathbf{P}\tilde{\mathbf{V}}_{\mathbf{S},\ell}$ *on the* t*-th layer. Since* $(\mathbf{Z}_i\tilde{\mathbf{V}}_{\mathbf{S}_i,s_t})^\top\mathbf{Z}_i\tilde{\mathbf{V}}_{\mathbf{S}_i,s_t} = \mathbf{I}$ *and* $(\mathbf{P}\tilde{\mathbf{V}}_{\mathbf{S},\ell})^\top\mathbf{P}\tilde{\mathbf{V}}_{\mathbf{S},\ell} = \mathbf{I}$, *we can think that* $\mathbf{S}'_i = \mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},s_t}$ *on the* $(t-i)$*-th layer with* $\tilde{\mathbf{V}}_{\mathbf{Y},s_t} = \mathbf{Z}_i\tilde{\mathbf{V}}_{\mathbf{S}_i,s_t}$, *and* $\mathbf{S}' = \mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},\ell}$ *on the* t*-th layer with* $\tilde{\mathbf{V}}_{\mathbf{Y},\ell} = \mathbf{P}\tilde{\mathbf{V}}_{\mathbf{S},\ell}$.

By Lem 3, we note that if $\text{range}(\mathbf{Y}\mathbf{V}_{\mathbf{Y},k}) \subset \text{range}(\mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},\ell}) = \text{range}(\mathbf{S}')$, then NNM computes the rank-k spectral decomposition with the optimal error.

D Proof of Lem 4

For the case of using linear combination input $\mathbf{S} = \mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},s}$ with $\tilde{\mathbf{V}}_{\mathbf{Y},s}^{\top}\tilde{\mathbf{V}}_{\mathbf{Y},s} = \mathbf{I}$, the generalized upper error bounds of ONM have been proven as Proposition 4 [13].

Proposition 4 *[13] If we set* $\mathbf{S} = \mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},s}$ *with* $\tilde{\mathbf{V}}_{\mathbf{Y},s}^{\top}\tilde{\mathbf{V}}_{\mathbf{Y},s} = \mathbf{I}$*, then errors of kernel PCA and rank-k matrix approximation using ONM are bounded as follows:*

$$\mathrm{NRE}(\tilde{\mathbf{U}}_{\mathbf{Y},k}) \leq \mathrm{NRE}(\mathbf{U}_{\mathbf{Y},k}) + \sqrt{\frac{2\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})}{\gamma_k}}$$

$$\mathrm{MRE}(\tilde{\mathbf{K}}_k) \leq \mathrm{MRE}(\mathbf{K}_k) + \sqrt{\frac{2\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})}{\gamma_k}}\,\mathrm{tr}(\mathbf{K}),$$

where $\tilde{\mathbf{U}}_{\mathbf{Y},k}$ *consists of the first k approximate principal directions which are implicitly generated by kernel PCA,* $\mathrm{NRE}(\tilde{\mathbf{U}}_{\mathbf{Y},k}) = \|\mathbf{Y} - \tilde{\mathbf{U}}_{\mathbf{Y},k}\tilde{\mathbf{U}}_{\mathbf{Y},k}^{\top}\mathbf{Y}\|_{\mathrm{F}}/\|\mathbf{Y}\|_{\mathrm{F}}$ *is the normalized error (NRE) of kernel PCA,* $\mathrm{MRE}(\tilde{\mathbf{K}}_k) = \|\mathbf{K} - \tilde{\mathbf{K}}_k\|_{\mathrm{F}}$ *is the error of rank-k PSD matrix approximation,* γ_k *is the k-th eigengap,* $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$ *is the sum of errors of eigenvalues from* $\tilde{\mathbf{V}}_{\mathbf{Y},k}$ *s.t.* $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k}) = \mathrm{tr}(\mathbf{V}_{\mathbf{Y},k}^{\top}\mathbf{Y}^{\top}\mathbf{Y}\mathbf{V}_{\mathbf{Y},k}) - \mathrm{tr}(\tilde{\mathbf{V}}_{\mathbf{Y},k}^{\top}\mathbf{Y}^{\top}\mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},k})$*, and* $\tilde{\mathbf{V}}_{\mathbf{Y},k}$ *is any submatrix consisting of k columns of* $\tilde{\mathbf{V}}_{\mathbf{Y},s}$*.*

The input of the final layer of NNM can be considered as $\mathbf{S}' = \mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},\ell}$ by Lem 3, then we can consider s samples, $\tilde{\mathbf{V}}_{\mathbf{Y},s}$, and $\mathbf{S}$ in Proposition 4 as compressed ℓ samples, $\tilde{\mathbf{V}}_{\mathbf{Y},\ell}$, and $\mathbf{S}'$, respectively. Since the input of the final layer of NNM satisfy the condition of input in Proposition 4, we can provide Lem 4 which states the upper bounds of the final errors of NNM.

Lemma 4: Suppose that $\mathbf{S}' = \mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},\ell}$ is the compressed samples as an input of the final layer of NNM, where $\tilde{\mathbf{V}}_{\mathbf{Y},\ell}^{\top}\tilde{\mathbf{V}}_{\mathbf{Y},\ell} = \mathbf{I}$ and $k \leq \ell \leq s_t$. Then, upper error bounds of NNM for kernel PCA and rank-k matrix approximation are

$$\mathrm{NRE}(\tilde{\mathbf{U}}_{\mathbf{Y},k}) \leq \mathrm{NRE}(\mathbf{U}_{\mathbf{Y},k}) + \sqrt{\frac{2\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})}{\gamma_k}} \qquad (10)$$

$$\mathrm{MRE}(\tilde{\mathbf{K}}_k) \leq \mathrm{MRE}(\mathbf{K}_k) + \sqrt{\frac{2\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})}{\gamma_k}}\,\mathrm{tr}(\mathbf{K}),$$

where $\tilde{\mathbf{V}}_{\mathbf{Y},k}$ is any submatrix consisting of k columns of $\tilde{\mathbf{V}}_{\mathbf{Y},\ell}$, $\tilde{\mathbf{U}}_{\mathbf{Y},k}$ consists of the first k approximate principal directions which are implicitly generated by kernel PCA, $\mathrm{NRE}(\tilde{\mathbf{U}}_{\mathbf{Y},k}) = \|\mathbf{Y} - \tilde{\mathbf{U}}_{\mathbf{Y},k}\tilde{\mathbf{U}}_{\mathbf{Y},k}^{\top}\mathbf{Y}\|_{\mathrm{F}}/\|\mathbf{Y}\|_{\mathrm{F}}$ is the normalized reconstruction error (NRE) of kernel PCA, $\mathrm{MRE}(\tilde{\mathbf{K}}_k) = \|\mathbf{K} - \tilde{\mathbf{K}}_k\|_{\mathrm{F}}$ is the reconstruction error of rank-k PSD matrix approximation, γ_k is the k-th eigengap, $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$ is the sum of errors of eigenvalues from $\tilde{\mathbf{V}}_{\mathbf{Y},k}$ with $\tilde{\mathbf{V}}_{\mathbf{Y},k}^{\top}\tilde{\mathbf{V}}_{\mathbf{Y},k} = \mathbf{I}$ s.t. $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k}) = \mathrm{tr}(\mathbf{V}_{\mathbf{Y},k}^{\top}\mathbf{Y}^{\top}\mathbf{Y}\mathbf{V}_{\mathbf{Y},k}) - \mathrm{tr}(\tilde{\mathbf{V}}_{\mathbf{Y},k}^{\top}\mathbf{Y}^{\top}\mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},k})$.

E Proof of Proposition 2

We provide Proposition 2 which states that the eigenvalue error of $\mathbf{K}$ from $\tilde{\mathbf{V}}_{\mathbf{Y},k}$ decreases as we use additional sublayers.

Proposition 2 Suppose that we use ONM for kernel PCA parts in NNM. Then, we have $\epsilon_2(\mathbf{Z}_j\tilde{\mathbf{V}}_{\mathbf{S}_j,k}^{onm})$ and $\epsilon_2(\mathbf{Z}_{j-1}\tilde{\mathbf{V}}_{\mathbf{S}_{j-1},k}^{onm})$ for NNM using i and $(i+1)$ sublayers respectively, and $\epsilon_2(\mathbf{Z}_{j-1}\tilde{\mathbf{V}}_{\mathbf{S}_{j-1},k}^{onm}) \leq \epsilon_2(\mathbf{Z}_j\tilde{\mathbf{V}}_{\mathbf{S}_j,k}^{onm})$.

Proof 3 *To prove Proposition 2, we need Cor 1, Lem 7, Lem 6 and Lem 5, and their statements and proofs are in the following subsections.*

Suppose that, given the mutilayer structure of NNM with t sublayers, we use ONM for kernel PCA parts and only i sublayers which are from the first sublayer to i-th sublayer until the final layer, where $i \in \{1, 2, ..., (t-1)\}$*. Then, the i-th sublayer becomes the last sublayer before the final layer, and we have* $\mathbf{S}'_j = \mathbf{Y}(\mathbf{Z}_j\tilde{\mathbf{V}}_{\mathbf{S}_j,\ell}^{onm})$ *as the input of the final layer with* $j = (t-i)$ *by Lem 3 and Lem 4. Since* $\tilde{\mathbf{V}}_{\mathbf{Y},\ell} = \mathbf{Z}_j\tilde{\mathbf{V}}_{\mathbf{S}_j,\ell}^{onm}$*, we can select* $\tilde{\mathbf{V}}_{\mathbf{Y},k} = \mathbf{Z}_j\tilde{\mathbf{V}}_{\mathbf{S}_j,k}^{onm}$ *which is the first k columns of* $\mathbf{Z}_j\tilde{\mathbf{V}}_{\mathbf{S}_j,\ell}^{onm}$ *for* $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$*.*

Similarly, suppose that we use only $(i+1)$ *sublayers from the first sublayer to* $(i+1)$*-th sublayer until the final layer, where* $i \in \{1, 2, ..., (t-1)\}$*. Then, we have* $\mathbf{S}'_{j-1} = \mathbf{Y}\mathbf{Z}_{j-1}\tilde{\mathbf{V}}_{\mathbf{S}_{j-1},\ell}^{onm}$ *as the input of the final layer with* $j = (t-i)$*, and we can select* $\tilde{\mathbf{V}}_{\mathbf{Y},k} = \mathbf{Z}_{j-1}\tilde{\mathbf{V}}_{\mathbf{S}_{j-1},k}^{onm}$ *for* $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$ *since* $\tilde{\mathbf{V}}_{\mathbf{Y},\ell} = \mathbf{Z}_{j-1}\tilde{\mathbf{V}}_{\mathbf{S}_{j-1},\ell}^{onm}$*.*

Thus, if we apply $\tilde{\mathbf{V}}_{\mathbf{Y},k} = \mathbf{Z}_j\tilde{\mathbf{V}}_{\mathbf{S}_j,k}^{onm}$ *to* $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$ *for using i sublayer and* $\tilde{\mathbf{V}}_{\mathbf{Y},k} = \mathbf{Z}_{j-1}\tilde{\mathbf{V}}_{\mathbf{S}_{j-1},k}^{onm}$ *to* $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$ *for using* $(i+1)$ *sublayers, then we have* $\epsilon_2(\mathbf{Z}_j\tilde{\mathbf{V}}_{\mathbf{S}_j,k}^{onm})$ *and* $\epsilon_2(\mathbf{Z}_{j-1}\tilde{\mathbf{V}}_{\mathbf{S}_{j-1},k}^{onm})$*, respectively.*

Next, we need to prove Eqn (11)

$$\epsilon_2(\mathbf{Z}_{j-1}\tilde{\mathbf{V}}_{\mathbf{S}_{j-1},k}^{onm}) \leq \epsilon_2(\mathbf{Z}_j\tilde{\mathbf{V}}_{\mathbf{S}_j,k}^{onm}), \qquad (11)$$

where $j = (t-i)$ *and* $i \in \{1, 2, ..., (t-1)\}$*. Eqn (11) is equivalent to Eqn (12) by Lem 5.*

$$\mathrm{tr}((\tilde{\mathbf{V}}_{\mathbf{S}_j,k}^{onm})^{\top}\mathbf{S}_j^{\top}\mathbf{S}_j\tilde{\mathbf{V}}_{\mathbf{S}_j,k}^{onm}) \qquad (12)$$
$$\leq \mathrm{tr}((\tilde{\mathbf{V}}_{\mathbf{S}_{j-1},k}^{onm})^{\top}\mathbf{S}_{j-1}^{\top}\mathbf{S}_{j-1}\tilde{\mathbf{V}}_{\mathbf{S}_{j-1},k}^{onm}),$$

where $j = (t-i)$ *and* $i \in \{1, 2, ..., (t-1)\}$*.*

By Lem 6, we have

$$\begin{aligned}\mathrm{tr}((\tilde{\mathbf{V}}_{\mathbf{S}_j,k}^{onm})^{\top}\mathbf{S}_j^{\top}\mathbf{S}_j\tilde{\mathbf{V}}_{\mathbf{S}_j,k}^{onm}) &\\ \leq \mathrm{tr}(((\tilde{\mathbf{V}}_{\mathbf{S}_j,\ell}^{onm})^{\top}\mathbf{S}_j^{\top}\mathbf{S}_j\tilde{\mathbf{V}}_{\mathbf{S}_j,\ell}^{onm})_k) &\\ = \mathrm{tr}(((\mathbf{S}'_j)^{\top}\mathbf{S}'_j)_k) = \mathrm{tr}(\mathbf{K}_{\mathbf{S}'_j,k}),&\end{aligned}$$

where $(\cdot)_k$ stands for the best rank k approximation, and $\tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k}$ consists of the first k columns of $\tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,\ell}$.

Without loss of generalization, we can define $\mathbf{P}_j$ to satisfy $\mathbf{S}_j = \mathbf{S}_{j-1}\mathbf{P}_j$, then $(\mathbf{P}_j)^\top \mathbf{P}_j = \mathbf{I}$. Now, we have

$$\mathbf{S}'_j = \mathbf{S}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,\ell} = \mathbf{S}_{j-1}\mathbf{P}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,\ell} = \mathbf{S}_{j-1}\mathbf{P}'_j,$$

where $\mathbf{P}'_j = \mathbf{P}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,\ell}$ for the cases in Thm 1. By using compact SVD of $\mathbf{S}'_j$ s.t. $\mathbf{S}'_j = \mathbf{U}_{\mathbf{S}'_j}\mathbf{\Sigma}_{\mathbf{S}'_j}\mathbf{V}^\top_{\mathbf{S}'_j}$, then we have

$$\begin{aligned} &\operatorname{tr}(((\mathbf{S}'_j)^\top \mathbf{S}'_j)_k) \\ &\qquad = \operatorname{tr}(\mathbf{U}^\top_{\mathbf{S}'_j,k}\mathbf{S}'_j(\mathbf{S}'_j)^\top \mathbf{U}_{\mathbf{S}'_j,k}) \\ &\qquad = \operatorname{tr}(\mathbf{U}^\top_{\mathbf{S}'_j,k}\mathbf{S}_{j-1}\mathbf{P}'_j(\mathbf{P}'_j)^\top \mathbf{S}^\top_{j-1}\mathbf{U}_{\mathbf{S}'_j,k}) \\ &\qquad \le \operatorname{tr}(\mathbf{U}^\top_{\mathbf{S}'_j,k}\mathbf{S}_{j-1}\mathbf{S}^\top_{j-1}\mathbf{U}_{\mathbf{S}'_j,k}). \end{aligned}$$

The last inequality holds because of Lem 6 and the fact that $(\mathbf{P}'_j)^\top \mathbf{P}'_j = \mathbf{I}$.

So far, we have shown

$$\begin{aligned} &\operatorname{tr}((\tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k})^\top \mathbf{S}^\top_j \mathbf{S}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k}) \\ &\qquad \le \operatorname{tr}(\mathbf{U}^\top_{\mathbf{S}'_j,k}\mathbf{S}_{j-1}\mathbf{S}^\top_{j-1}\mathbf{U}_{\mathbf{S}'_j,k}). \end{aligned}$$

Now, we want to argue that

$$\begin{aligned} &\operatorname{tr}(\mathbf{U}^\top_{\mathbf{S}'_j,k}\mathbf{S}_{j-1}\mathbf{S}^\top_{j-1}\mathbf{U}_{\mathbf{S}'_j,k}) \\ &\qquad \le \operatorname{tr}((\tilde{\mathbf{U}}^{onm}_{\mathbf{S}_{j-1},k})^\top \mathbf{S}_{j-1}\mathbf{S}^\top_{j-1}\tilde{\mathbf{U}}^{onm}_{\mathbf{S}_{j-1},k}). \end{aligned}$$

We note that $\tilde{\mathbf{U}}^{onm}_{\mathbf{S}_{j-1},k}$ and $\mathbf{U}'_{\mathbf{S}_j,k}$ are in the range of $\mathbf{S}'_j$. Then, by Cor 1,

$$\begin{aligned} &\operatorname{tr}((\tilde{\mathbf{U}}^{onm}_{\mathbf{S}_{j-1},k})^\top \mathbf{S}_{j-1}\mathbf{S}^\top_{j-1}\tilde{\mathbf{U}}^{onm}_{\mathbf{S}_{j-1},k}) \\ &\qquad = \max_{\tilde{\mathbf{U}}_{\mathbf{S}_{j-1},k}} \operatorname{tr}\left(\tilde{\mathbf{U}}^\top_{\mathbf{S}_{j-1},k}\mathbf{S}_{j-1}\mathbf{S}^\top_{j-1}\tilde{\mathbf{U}}_{\mathbf{S}_{j-1},k}\right) \\ &\qquad\qquad \textit{subject to} \quad \tilde{\mathbf{U}}^\top_{\mathbf{S}_{j-1},k}\tilde{\mathbf{U}}_{\mathbf{S}_{j-1},k} = \mathbf{I}, \\ &\qquad\qquad \operatorname{range}(\tilde{\mathbf{U}}_{\mathbf{S}_{j-1},k}) \subset \operatorname{range}(\mathbf{S}'_j). \end{aligned}$$

Thus, we have

$$\begin{aligned} &\operatorname{tr}((\tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k})^\top \mathbf{S}^\top_j \mathbf{S}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k}) \\ &\qquad \le \operatorname{tr}(\mathbf{U}^\top_{\mathbf{S}'_j,k}\mathbf{S}_{j-1}\mathbf{S}^\top_{j-1}\mathbf{U}_{\mathbf{S}'_j,k}) \\ &\qquad \le \operatorname{tr}((\tilde{\mathbf{U}}^{onm}_{\mathbf{S}_{j-1},k})^\top \mathbf{S}_{j-1}\mathbf{S}^\top_{j-1}\tilde{\mathbf{U}}^{onm}_{\mathbf{S}_{j-1},k}). \end{aligned}$$

Next, by Lem 7, we have

$$\begin{aligned} &\operatorname{tr}((\tilde{\mathbf{U}}^{onm}_{\mathbf{S}_{j-1},k})^\top \mathbf{S}_{j-1}\mathbf{S}^\top_{j-1}\tilde{\mathbf{U}}^{onm}_{\mathbf{S}_{j-1},k}) \\ &\qquad \le \operatorname{tr}((\tilde{\mathbf{V}}^{onm}_{\mathbf{S}_{j-1},k})^\top \mathbf{S}^\top_{j-1}\mathbf{S}_{j-1}\tilde{\mathbf{V}}^{onm}_{\mathbf{S}_{j-1},k}). \end{aligned}$$

Consequently, we have

$$\begin{aligned} &\operatorname{tr}((\tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k})^\top \mathbf{S}^\top_j \mathbf{S}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k}) \\ &\qquad \le \operatorname{tr}((\tilde{\mathbf{V}}^{onm}_{\mathbf{S}_{j-1},k})^\top \mathbf{S}^\top_{j-1}\mathbf{S}_{j-1}\tilde{\mathbf{V}}^{onm}_{\mathbf{S}_{j-1},k}). \end{aligned}$$

E.1 Lem 5

Lemma 5 *Given the $\mathbf{S}_j = \mathbf{Y}\mathbf{Z}_j$, if we consider $\tilde{\mathbf{V}}_{\mathbf{Y},k} = \mathbf{Z}_j\tilde{\mathbf{V}}_{\mathbf{S}_j,k}$, then the followings are equivalent*

$$\begin{aligned} &\underset{\tilde{\mathbf{V}}_{\mathbf{S}_j,k}}{\operatorname{maximize}} \operatorname{tr}((\tilde{\mathbf{V}}_{\mathbf{S}_j,k})^\top \mathbf{S}_j\mathbf{S}_j\tilde{\mathbf{V}}_{\mathbf{S}_j,k}) \\ \Longleftrightarrow\ &\underset{\tilde{\mathbf{V}}_{\mathbf{S}_j,k}}{\operatorname{minimize}} \epsilon_2(\mathbf{Z}_j\tilde{\mathbf{V}}_{\mathbf{S}_j,k}). \end{aligned}$$

Proof 4 *Let us remind that the sum of eigenvalue errors from $\tilde{\mathbf{V}}_{\mathbf{Y},k}$ is defined as*

$$\begin{aligned} \epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k}) &= \operatorname{tr}(\mathbf{V}^\top_{\mathbf{Y},k}\mathbf{Y}^\top\mathbf{Y}\mathbf{V}_{\mathbf{Y},k}) - \operatorname{tr}(\tilde{\mathbf{V}}^\top_{\mathbf{Y},k}\mathbf{Y}^\top\mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},k}) \\ &= \operatorname{tr}(\mathbf{\Sigma}^2_{\mathbf{Y},k}) - \operatorname{tr}(\tilde{\mathbf{V}}^\top_{\mathbf{Y},k}\mathbf{Y}^\top\mathbf{Y}\tilde{\mathbf{V}}_{\mathbf{Y},k}), \end{aligned}$$

where $\mathbf{K} = \mathbf{Y}^\top\mathbf{Y}$, $\mathbf{Y} = \mathbf{U}_\mathbf{Y}\mathbf{\Sigma}_\mathbf{Y}\mathbf{V}^\top_\mathbf{Y}$, $\mathbf{Y}_k = \mathbf{U}_{\mathbf{Y},k}\mathbf{\Sigma}_{\mathbf{Y},k}\mathbf{V}^\top_{\mathbf{Y},k}$, and $(\tilde{\mathbf{V}}_{\mathbf{Y},k})^\top\tilde{\mathbf{V}}_{\mathbf{Y},k} = \mathbf{I}$.

If we consider $\tilde{\mathbf{V}}_{\mathbf{Y},k} = \mathbf{Z}_j\tilde{\mathbf{V}}_{\mathbf{S}_j,k}$, then we have

$$\begin{aligned} &\epsilon_2(\mathbf{Z}_j\tilde{\mathbf{V}}_{\mathbf{S}_j,k}) \\ &\qquad = \operatorname{tr}(\mathbf{\Sigma}^2_{\mathbf{Y},k}) - \operatorname{tr}((\mathbf{Z}_j\tilde{\mathbf{V}}_{\mathbf{S}_j,k})^\top\mathbf{Y}^\top\mathbf{Y}\mathbf{Z}_j\tilde{\mathbf{V}}_{\mathbf{S}_j,k}) \\ &\qquad = \operatorname{tr}(\mathbf{\Sigma}^2_{\mathbf{Y},k}) - \operatorname{tr}((\tilde{\mathbf{V}}_{\mathbf{S}_j,k})^\top\mathbf{S}^\top_j\mathbf{S}_j\tilde{\mathbf{V}}_{\mathbf{S}_j,k}). \end{aligned}$$

Since $\operatorname{tr}(\mathbf{\Sigma}^2_{\mathbf{Y},k})$ is constant given the rank-k and $\mathbf{K}$, we complete the proof.

E.2 Lem 6

Lemma 6 *[9] Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ have singular values $\sigma_1(\mathbf{A}) \ge \cdots \ge \sigma_q(\mathbf{A}) \ge 0$, where $q = \min\{m, n\}$. For each $k = 1, \ldots, q$ we have*

$$\begin{aligned} \sum_{i=1}^k \sigma_i(\mathbf{A}) = \max\{|\operatorname{tr}(\mathbf{X}^\top\mathbf{A}\mathbf{Y})| : \mathbf{X} \in \mathbb{R}^{m\times k}, \mathbf{Y} \in \mathbb{R}^{n\times k}, \\ \mathbf{X}^\top\mathbf{X} = \mathbf{Y}^\top\mathbf{Y} = \mathbf{I}\}. \end{aligned}$$

E.3 Cor 1

Corollary 1 *For a mutilayer Nyström architecture of NNM, we have*

$$\begin{aligned} \tilde{\mathbf{U}}^{onm}_{\mathbf{S}_{j-1},k} = \underset{\tilde{\mathbf{U}}_{\mathbf{S}_{j-1},k}}{\operatorname{argmax}} \operatorname{tr}(\tilde{\mathbf{U}}^\top_{\mathbf{S}_{j-1},k}\mathbf{S}_{j-1}\mathbf{S}^\top_{j-1}\tilde{\mathbf{U}}_{\mathbf{S}_{j-1},k}) \quad (13) \\ \textit{subject to} \quad \tilde{\mathbf{U}}^\top_{\mathbf{S}_{j-1},k}\tilde{\mathbf{U}}_{\mathbf{S}_{j-1},k} = \mathbf{I}, \\ \operatorname{range}(\tilde{\mathbf{U}}_{\mathbf{S}_{j-1},k}) \subset \operatorname{range}(\mathbf{S}'_j). \end{aligned}$$

That is, given the subsample matrices $\mathbf{S}_j$ and $\mathbf{S}_{j-1}$, ONM minimizes the sum of eigenvalue errors of $\mathbf{K}_{\mathbf{S}_{j-1}}$ under the formula of Eqn (13).

Proof 5 *Since, $\mathbf{S}'_j = \mathbf{S}_j\tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,\ell} = \mathbf{S}_{j-1}\mathbf{P}_j\tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,\ell} = \mathbf{S}_{j-1}\mathbf{P}'_j$, we note that $\tilde{\mathbf{U}}^{onm}_{\mathbf{S}_{j-1},k}$ and $\mathbf{U}'_{\mathbf{S}_j,k}$ are in the range of $\mathbf{S}'_j$. Then, we can easily derive Cor 1 from Lem 1 and Lem 2.*

E.4 Lem 7

Lemma 7 *We have*

$$\mathrm{tr}((\tilde{\mathbf{V}}^{onm}_{S_j,k})^\top \mathbf{S}_j^\top \mathbf{S}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k}) \geq \mathrm{tr}((\tilde{\mathbf{U}}^{onm}_{\mathbf{S}_j,k})^\top \mathbf{S}_j \mathbf{S}_j^\top \tilde{\mathbf{U}}^{onm}_{\mathbf{S}_j,k}),$$

where $\tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k} = \mathbf{S}_j^\top \tilde{\mathbf{U}}^{onm}_{\mathbf{S}_j,k} (\tilde{\mathbf{\Sigma}}^{onm}_{\mathbf{S}_j,k})^{-1}$.

Proof 6 *First we have* $(\tilde{\mathbf{U}}^{onm}_{\mathbf{S}_j,k})^\top \mathbf{S}_j \mathbf{S}_j^\top \tilde{\mathbf{U}}^{onm}_{\mathbf{S}_j,k} = (\tilde{\mathbf{\Sigma}}^{onm}_{\mathbf{S}_j,k})^2$. *Next, suppose that we consider full SVD of* $\mathbf{S}_j$ *s.t.* $\mathbf{S}_j = \mathbf{U}_{\mathbf{S}_j} \mathbf{\Sigma}_{\mathbf{S}_j} \mathbf{V}_{\mathbf{S}_j}^\top$, *then we have*

$$\begin{aligned}
&\mathrm{tr}((\tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k})^\top \mathbf{S}_j^\top \mathbf{S}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k}) \\
&\qquad = \mathrm{tr}((\tilde{\mathbf{U}}^{onm}_{\mathbf{S}_j,k})^\top \mathbf{U}_{\mathbf{S}_j} \mathbf{\Sigma}^4_{\mathbf{S}_j} \mathbf{U}_{\mathbf{S}_j}^\top \tilde{\mathbf{U}}^{onm}_{\mathbf{S}_j,k} (\tilde{\mathbf{\Sigma}}^{onm}_{\mathbf{S}_j,k})^{-2}) \\
&\mathrm{tr}((\tilde{\mathbf{U}}^{onm}_{\mathbf{S}_j,k})^\top \mathbf{S}_j \mathbf{S}_j^\top \tilde{\mathbf{U}}^{onm}_{\mathbf{S}_j,k}) \\
&\qquad = \mathrm{tr}((\tilde{\mathbf{U}}^{onm}_{\mathbf{S}_j,k})^\top \mathbf{U}_{\mathbf{S}_j} \mathbf{\Sigma}^2_{\mathbf{S}_j} \mathbf{U}_{\mathbf{S}_j}^\top \tilde{\mathbf{U}}^{onm}_{\mathbf{S}_j,k}).
\end{aligned}$$

If we consider $\mathbf{B} = \mathbf{U}_{\mathbf{S}_j}^\top \tilde{\mathbf{U}}^{onm}_{\mathbf{S}_j,k}$, *then the proof is completed by Lem 8.*

E.5 Lem 8

Lemma 8 *Suppose matrix* $\mathbf{B} = (b_{ij}) \in \mathbb{R}^{m \times n}$ *has orthonormal columns, i.e.* $\mathbf{B}^\top \mathbf{B} = \mathbf{I}_n$, $\mathbf{\Sigma}^2 = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_m^2)$. *Then, we have*

$$\mathrm{tr}(\mathbf{B}^\top \mathbf{\Sigma}^4 \mathbf{B} \mathbf{D}^{-1}) \geq \mathrm{tr}(\mathbf{B}^\top \mathbf{\Sigma}^2 \mathbf{B}),$$

where $\mathbf{D} = \mathrm{diag}(d_1, \ldots, d_n)$ *is the the diagonal matrix consisting of diagonal elements of* $\mathbf{B}^\top \mathbf{\Sigma}^2 \mathbf{B}$.

Proof 7 *By matrix multiplication we have* $d_i = \sum_{j=1}^m \sigma_j^2 b_{ji}^2$ *and thus the ith diagonal element of* $\mathbf{B}^\top \mathbf{\Sigma}^4 \mathbf{B} \mathbf{D}^{-1}$ *is* $\left(\sum_{j=1}^m \sigma_j^4 b_{ji}^2\right) \Big/ d_i$. *By Cauchy-Schwarz inequality,*

$$\begin{aligned}
\sum_{j=1}^m \sigma_j^4 b_{ji}^2 &= \left(\sum_{j=1}^m (\sigma_j^2 b_{ji})^2\right) \left(\sum_{j=1}^m b_{ji}^2\right) \\
&\geq \left(\sum_{j=1}^m \sigma_j^2 b_{ji}^2\right)^2 = d_i^2.
\end{aligned}$$

Therefore, $\left(\sum_{j=1}^m \sigma_j^4 b_{ji}^2\right) \Big/ d_i \geq d_i$. *Thus, we have*

$$\begin{aligned}
\mathrm{tr}(\mathbf{B}^\top \mathbf{\Sigma}^4 \mathbf{B} \mathbf{D}^{-1}) &= \sum_{i=1}^n \left(\sum_{j=1}^m \sigma_j^4 b_{ji}^2\right) \Big/ d_i \\
&\geq \sum_{i=1}^n d_i = \mathrm{tr}(\mathbf{B}^\top \mathbf{\Sigma}^2 \mathbf{B}).
\end{aligned}$$

F Proof of Thm 1

To prove Thm 1, we need Lem 3, Lem 5, Proposition 2, and Lem 4, whose statements and proofs are in the Appendix.

Theorem 1 Suppose that we use ONM for the kernel PCA parts in the sublayers. Then, the upper error bound of NNM in Lem 4 decreases when we use additional sublayers. That is, if we add an additional sublayer to the NNM structure with i sublayers as $(i+1)$-th sublayer, then upper error bounds further decrease as $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$ decreases.

Proof 8 *Suppose that, given the mutilayer structure of NNM with t sublayers, we use ONM for kernel PCA parts and only i sublayers which are from the first sublayer to i-th sublayer until the final layer, where* $i \in \{1, 2, ..., (t-1)\}$. *Then, the i-th sublayer becomes the last sublayer before the final layer, and we have* $\mathbf{S}'_j = \mathbf{Y}(\mathbf{Z}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,\ell})$ *as the input of the final layer with* $j = (t-i)$ *by Lem 3 and Lem 4. Since* $\tilde{\mathbf{V}}_{\mathbf{Y},\ell} = \mathbf{Z}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,\ell}$, *we can select* $\tilde{\mathbf{V}}_{\mathbf{Y},k} = \mathbf{Z}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k}$ *which is the first k columns of* $\mathbf{Z}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,\ell}$ *for* $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$.

Similarly, suppose that we use only $(i+1)$ *sublayers from the first sublayer to* $(i+1)$*-th sublayer until the final layer, where* $i \in \{1, 2, ..., (t-1)\}$. *Then, we have* $\mathbf{S}'_{j-1} = \mathbf{Y}\mathbf{Z}_{j-1} \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_{j-1},\ell}$ *as the input of the final layer with* $j = (t-i)$, *and we can select* $\tilde{\mathbf{V}}_{\mathbf{Y},k} = \mathbf{Z}_{j-1} \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_{j-1},k}$ *for* $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$ *since* $\tilde{\mathbf{V}}_{\mathbf{Y},\ell} = \mathbf{Z}_{j-1} \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_{j-1},\ell}$.

Thus, if we apply $\tilde{\mathbf{V}}_{\mathbf{Y},k} = \mathbf{Z}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k}$ *to* $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$ *for using i sublayer and* $\tilde{\mathbf{V}}_{\mathbf{Y},k} = \mathbf{Z}_{j-1} \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_{j-1},k}$ *to* $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$ *for using* $(i+1)$ *sublayers, then we have* $\epsilon_2(\mathbf{Z}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k})$ *and* $\epsilon_2(\mathbf{Z}_{j-1} \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_{j-1},k})$, *respectively. Then, by Lem 2, we have* $\epsilon_2(\mathbf{Z}_{j-1} \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_{j-1},k}) \leq \epsilon_2(\mathbf{Z}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k})$.

We also note that, given the ℓ *column vectors of* $\mathbf{Z}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,\ell}$, $\epsilon_2(\mathbf{Z}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k})$ *is the minimum among* $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$, *where* $\tilde{\mathbf{V}}_{\mathbf{Y},k}$ *is any submatrix consisting of k column vectors of* $\mathbf{Z}_j \tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,\ell}$ *and* $j = (t-i)$ *for using i sublayer. We can easily prove it by considering Lem 1, Lem 2 and the definition of* $\tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k}$ *which is the fist k column vectors of* $\tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,\ell}$ *s.t.*

$$\begin{aligned}
\tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k} = &\underset{\tilde{\mathbf{V}}_{\mathbf{S}_j,k}}{\mathrm{argmax}}\, \mathrm{tr}((\tilde{\mathbf{V}}_{\mathbf{S}_j,k})^\top \mathbf{S}_j \mathbf{S}_j \tilde{\mathbf{V}}_{\mathbf{S}_j,k}) \\
&\quad \textit{subject to } \mathrm{range}(\tilde{\mathbf{V}}_{\mathbf{S}_j,k}) \subset \mathrm{range}(\mathbf{S}'_{j+1}), \\
&\qquad (\tilde{\mathbf{V}}_{\mathbf{S}_j,k})^\top \tilde{\mathbf{V}}_{\mathbf{S}_j,k} = \mathbf{I},
\end{aligned}$$

where we compute $\tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k}$ *by using ONM and the compressed sample matrices* $\mathbf{C}'_{j+1} = \mathbf{S}_j^\top \mathbf{S}'_{j+1}$ *and* $\mathbf{K}'_{\mathbf{S}_{j+1}} = (\mathbf{S}_{j+1})^\top \mathbf{S}'_{j+1}$. *Since maximizing* $\mathrm{tr}((\tilde{\mathbf{V}}_{\mathbf{S}_j,k}^\top \mathbf{S}_j \mathbf{S}_j \tilde{\mathbf{V}}_{\mathbf{S}_j,k})$

is equivalent to minimizing $\epsilon_2(\mathbf{Z}_j\tilde{\mathbf{V}}_{\mathbf{S}_j,k}$ *by Lem* 5, $\epsilon_2(\mathbf{Z}_j\tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,k})$ *is the minimum among* $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$*, where* $\tilde{\mathbf{V}}_{\mathbf{Y},k}$ *is any submatrix consisting of* k *column vectors of* $\mathbf{Z}_j\tilde{\mathbf{V}}^{onm}_{\mathbf{S}_j,\ell}$*.*

Thus, we complete the proof, since the minimum of sum of eigenvalue errors $\epsilon_2(\tilde{\mathbf{V}}_{\mathbf{Y},k})$ *in Lem 4 decreases as we use additional sublayers until the final layer.*

G Proof of Proposition 3

We provide Proposition 3 which states why the SNM should not be utilized at the final layer of NNM.

Proposition 3 Suppose that we compress $\mathbf{C}$ and $\mathbf{K}_\mathbf{S}$ as $\mathbf{C}'$ and $\mathbf{K}'_\mathbf{S}$ with $\tilde{\mathbf{V}}_{\mathbf{S},\ell} = \mathbf{V}_{\mathbf{S},\ell}$ at the t-th layer, and we run the SNM with $\mathbf{C}'$ and $\mathbf{K}'_\mathbf{S}$ at the final layer. Then $\tilde{\mathbf{K}}^{snm}_k$ are the same regardless of values of ℓ, where $\ell \geq k$. That is, the rank-k spectral decomposition using rank-k SNM is biased by s samples.

Proof 9 *Suppose that* $\tilde{\mathbf{V}}_{\mathbf{Y},\ell} = \mathbf{V}_{\mathbf{S},\ell}$ *for some* $\ell \geq k$*, and* $\sigma_i(\mathbf{S}) \neq 0$ *for* $i = 1, ..., \ell$*. Then, we have* $\mathbf{C}' = \mathbf{C}\mathbf{V}_{\mathbf{S},\ell} = \mathbf{Y}^\top\mathbf{S}\mathbf{V}_{\mathbf{S},\ell}$ *and* $\mathbf{K}'_\mathbf{S} = \mathbf{S}'^\top\mathbf{S}' = \mathbf{V}_{\mathbf{S},\ell}\mathbf{S}^\top\mathbf{S}\mathbf{V}_{\mathbf{S},\ell} = \mathbf{\Sigma}^2_{\mathbf{S},\ell}$*. Consequently,* $\mathbf{K}^\dagger_{\mathbf{S}',k} = \mathbf{\Sigma}^{-2}_{\mathbf{S},k}$*. If we run the standard Nyström method at the final layer with* $\mathbf{C}'$ *and* $\mathbf{K}'_\mathbf{S}$*, then we have*

$$\begin{aligned}\tilde{\mathbf{K}}^{snm}_k = \mathbf{C}'\mathbf{K}^\dagger_{\mathbf{S}',k}\mathbf{C}'^\top &= \mathbf{Y}^\top\mathbf{S}\mathbf{V}_{\mathbf{S},\ell}\mathbf{\Sigma}^{-2}_{\mathbf{S},k}\mathbf{V}^\top_{\mathbf{S},\ell}\mathbf{S}^\top\mathbf{Y} \\ &= \mathbf{Y}^\top\mathbf{U}_{\mathbf{S},\ell}\mathbf{\Sigma}_{\mathbf{S},\ell}\mathbf{\Sigma}^{-2}_{\mathbf{S},k}\mathbf{\Sigma}_{\mathbf{S},\ell}\mathbf{U}^\top_{\mathbf{S},\ell}\mathbf{Y} \\ &= \mathbf{Y}^\top\mathbf{U}_{\mathbf{S},k}\mathbf{U}^\top_{\mathbf{S},k}\mathbf{Y}.\end{aligned}$$