I. ADDITIONAL TOY EXAMPLES

The main purpose of this supplementary paper is to provide some additional toy examples to highlight a few more aspects of the methods derived using the Similarity Embedding Framework (SEF). As in the main paper, the toy dataset is composed of the same 500 samples of the MNIST dataset (digits 0, 1 and 2). Also, some videos that illustrate the low dimensional space during the optimization process are provided.

A. PCA-like Embeddings

Fig. 1. Comparing (K-)PCA to (K)S-PCA (Toy Dataset)
First, the (K-)PCA method is compared to the (K)S-PCA in Figure 1. The S-PCA method learns an embedding similar to the PCA embedding. However, the K(ernel)-PCA method disproportionately spreads only one class of the data. The KS-PCA method restricts this phenomenon, due to the intrinsic outlier-resistance of the SEF, and manages to better spread the rest of the classes leading to a more balanced embedding.

B. LDA-like Embeddings

![Fig. 2. Comparing LDA to S-LDA (Toy Dataset)](image)

The LDA method is also compared to the S-LDA method in Figure 2. The LDA method manages to perfectly overfit this toy dataset. The S-LDA method is more resistant to overfitting as shown in Figures 2b-c. This also explains the better behavior of the S-LDA method, when compared to the LDA method (experiments of Section IV-C of the main paper). Only when a larger learning rate and no regularization are used, the S-LDA (Figure 2c) manages to overfit the data in a way similar to the LDA method. The experiment is repeated using 1000 training samples from the MNIST dataset (Figure 3). The overfitting is significantly reduced and the S-LDA method (Figure 3c) leads to significantly better separation of the classes in the low dimensional space.

![Fig. 3. Comparing LDA to S-LDA (MNIST Dataset, 1000 training samples)](image)
C. Laplacian Eigenmaps-like Embeddings

It is reminded that to learn embeddings similar to those of the Laplacian Eigenmaps (LE) method the target similarity matrix is set to:

$$[T]_{ij} = \begin{cases} 
1, & \text{if } i \text{ is among the } k \text{-nearest neighbors of } j \\
\exp(-\frac{|x_i - x_j|^2}{\sigma_{LE}}), & \text{otherwise} 
\end{cases}$$

while the optimization mask to:

$$[M]_{ij} = \begin{cases} 
1, & \text{if } i \text{ is among the } k \text{-nearest neighbors of } j \\
\alpha_{LE}, & \text{otherwise } (0 \leq \alpha_{LE} \leq 1) 
\end{cases}$$

The parameter \(\alpha_{LE}\) defines the importance of maintaining the original similarity for non-neighboring points and acts as a regularizer to prevent collapsing all the points into one.

The S-LE and the KS-LE methods are compared to the LE method in Figure 4. The value of \(\alpha_{LE}\) was set to 0.00005 and \(k = 30\) neighbors were used. The obtained embeddings for the S-LE and the KS-LE methods are not identical to the embedding of the LE method. This is mainly attributed to the different regularization technique that is used. The effect of the regularization parameter is illustrated in Figures 5a-c using the KS-LE method. Using
larger values allow for better separation of the data (but leads to larger intra-neighbor scatter as well), while using no regularization tends to collapse all the data into one point.

D. Cloning Existing Techniques
In this Subsection, more toy examples of cloning various techniques are provided. In Figures 6a-c the cloning of the LE method is illustrated, while in Figures 7a-c the cloning embeddings of the ISOMAP method are provided. The linear projection provides very similar embeddings to the original ones, while the kernel projection manages to almost perfectly recreate the original space. Recall that the SEF did not learn an embedding identical to the LE when the (K)S-LE method was used (Figure 4). However, when the (distribution of the) LE method is used as a target (c(K)S-LE method instead of (K)S-LE method), it is able to accurately clone it (Figure 6).

Finally, the t-SNE algorithm is used to visualize 1000 training samples of the MNIST dataset in Figure 8. The cKS-tSNE method (Figure 8b) almost perfectly clones the original space provided by the t-SNE.

E. Videos of the optimization process

To better illustrate the optimization process, 10 videos have been created. Each one visualizes the low dimensional space during the optimization for a different target similarity matrix. The videos can be found in the following links:

<table>
<thead>
<tr>
<th>Method</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-PCA</td>
<td><a href="https://youtu.be/6s7RIY1vY9o">https://youtu.be/6s7RIY1vY9o</a></td>
</tr>
<tr>
<td>KS-PCA</td>
<td><a href="https://youtu.be/plKEzcCacJA">https://youtu.be/plKEzcCacJA</a></td>
</tr>
<tr>
<td>S-LDA</td>
<td><a href="https://youtu.be/fUsMZgGZg4o">https://youtu.be/fUsMZgGZg4o</a></td>
</tr>
<tr>
<td>KS-LDA</td>
<td><a href="https://youtu.be/2Io2Jt0gt04">https://youtu.be/2Io2Jt0gt04</a></td>
</tr>
<tr>
<td>S-LapPCA</td>
<td><a href="https://youtu.be/09FJ7v5yKfk">https://youtu.be/09FJ7v5yKfk</a></td>
</tr>
<tr>
<td>KS-LapPCA</td>
<td><a href="https://youtu.be/MdgoDadu5PE">https://youtu.be/MdgoDadu5PE</a></td>
</tr>
<tr>
<td>S-SVM-A</td>
<td><a href="https://youtu.be/4R9QNwkSF_c">https://youtu.be/4R9QNwkSF_c</a></td>
</tr>
<tr>
<td>KS-SVM-A</td>
<td><a href="https://youtu.be/EAPSdT5rhA">https://youtu.be/EAPSdT5rhA</a></td>
</tr>
<tr>
<td>cS-tSNE</td>
<td><a href="https://youtu.be/607_ZOVOAYs">https://youtu.be/607_ZOVOAYs</a></td>
</tr>
<tr>
<td>cKS-tSNE</td>
<td><a href="https://youtu.be/WRzuKuKDeeU">https://youtu.be/WRzuKuKDeeU</a></td>
</tr>
</tbody>
</table>

(or alternatively, all the videos can be accessed using the following playlist: https://www.youtube.com/playlist?list=PLRPmPOizkG3Lm3pQsaVrbjVHxFNBFG2F)

II. TRAINING USING LIMITED TRAINING DATA

In this Section the effect of varying the number of the training data on the quality of the learned projections is evaluated. First, the classification accuracy for projections learned with the proposed S-LDA method and the standard LDA method are compared in Figure 9. Note that the LDA can be numerically unstable when the number of training samples is less than the number of dimensions of the data (the default implementation provided by the well known scikit-learn library was used). The methods were evaluated using 5, 7, 10, 15 and 20 training samples per class and the experiments were repeated 10 times (the mean and the standard deviation are reported). It is demonstrated that the proposed S-LDA method is able to learn significantly better projections (leading to higher classification accuracy), especially when a small number of training samples per class is used.
Fig. 9. Comparing the LDA and the proposed S-LDA methods using varying number of training samples

Also, in Figure 10 the PCA and the cS-PCA methods are compared using 5, 7, 10, 15 and 20 training samples per class. Again, in most cases using the proposed cS-PCA method allows for learning better projections, avoiding possible over-fitting phenomena (even when small training datasets are used). The only exception to this is for the
MNIST dataset, where the PCA performs slightly better. However, as demonstrated in the main paper, the cS-PCA method performs better when more training samples are used.

Fig. 10. Comparing the PCA and the proposed cS-PCA methods using varying number of training samples
REFERENCES