Learned Trajectory Embedding for Subspace Clustering

Supplementary Material

A. Experimental Details

As discussed in the main paper, the feature extractor f_θ takes a variable-length input trajectory $\mathbf{x}$ with two input channels representing the x and y values and outputs the feature $\mathbf{f} \in \mathbb{R}^d$, and the dimensionality of the latent space is set to $d = 128$. Specifically, the network consists of the three 1D convolutional layers (with channels changing as follows: $2 \rightarrow 64 \rightarrow 128 \rightarrow 512$), followed by max pooling in the temporal domain and two linear layers ($512 \rightarrow 128 \rightarrow 128$). All convolutional operations in f_θ use kernels of size 3 with stride 1. The subspace estimator consists of the parametric basis functions h^j_ψ with trainable parameters $(\mu_j, \alpha_j, \beta_j, \gamma_j)$ and a multilayer perceptron ω_ζ which infers the subspace basis coefficients from the feature $\mathbf{f}$. In particular, ω_ζ has three linear layers ($128 \rightarrow 512 \rightarrow 1024 \rightarrow 512$), and the resulting 512-dimensional vector is reshaped into a coefficient matrix $\Omega_\zeta(\mathbf{f}) \in \mathbb{R}^{128\times4}$ used in (8). ReLU activation is used after each convolutional and each linear layer. During training, we used Adam optimizer with a learning rate set to 0.001, reduced at each epoch with an exponential decay of 0.999.

B. Ablation Studies

We conduct an ablation study in which we first train only the feature extractor f_θ with InfoNCE loss (and obtain network weights θ_1). Subsequently, we include the subspace estimator g_ϕ and continue training with the entire loss, resulting in weights θ_2. The two sets of weights are compared using the classification error of clustering in the feature space. Table 3 shows that the performance on validation and test data improves with the full architecture and the complete loss, proving the advantage of training the feature extractor f_θ together with the subspace estimator g_ϕ.

		Validation subset		Test subset	
Weights	Arch. + loss	Mean	Median	Mean	Median
θ_1	f_θ + InfoNCE loss	1.80	0.00	2.97	0.00
θ_2	f_θ + g_ϕ + total loss	1.51	0.00	0.85	0.21

Table 3. Classification error (%) of clustering with f_{θ_1} trained using f_θ + InfoNCE, and f_{θ_2} trained using f_θ & g_ϕ + total loss.

C. Time Complexity

As discussed in the main paper, our method is very fast. Its time complexity is analysed below. A single trajectory inference requires $\mathcal{O}(F)$ computations due to the convolutional structure. Passing N full trajectories is therefore $\mathcal{O}(NF)$, and the subsequent clustering requires $\mathcal{O}(N^2)$. The trajectory completion comprises matrix operations of size up to $2F \times 2F$ hence costs $\mathcal{O}(F^3)$. It can also be sped up by employing randomized singular value decomposition.