Unsupervised Feature Learning with Emergent Data-Driven Prototypicality

Supplementary Material

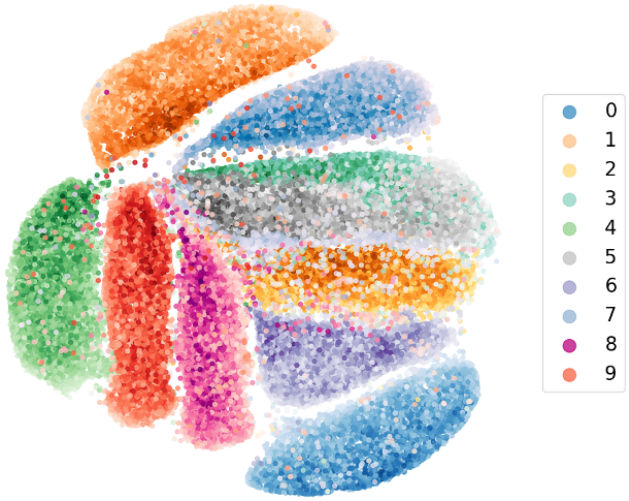

Figure 13. The KNN density estimation on MoCo [18] features of MNIST [25]. The shades of color represent the density value: the darker the color, the higher the density.

7. More Details on K-NN Density Estimation on MNIST

Feature Extraction: We use a LeNet [26] without classifier as the encoder and follow the scheme of MoCo [18] to train the feature extractor. We run the training for 200 epochs and the initial learning rate is 0.06. We use a cosine learning rate scheduler [29].

Visualization: Figure 13 visualize the KNN density estimation on MoCo [18] features of MNIST [25]. The output features have the dimension of 64. To visualize the features, we use t-SNE [42] with the perplexity of 40 and 300 iterations for optimization.

8. More Details on Hyperbolic Instance Assignment

A more detailed description of the hyperbolic instance assignment is given.

Initially, we randomly assign the particles to the images. Given a batch of samples $\{(\mathbf{x}_1, s_1), (\mathbf{x}_2, s_2), ..., (\mathbf{x}_b, s_b)\}$, where $\mathbf{x}_i$ is an image and s_i is the corresponding particle. Given an encoder f_θ, we generate the hyperbolic feature for each image $\mathbf{x}_i$ as $f_\theta(\mathbf{x}_i) \in \mathbb{B}^2$, where $\mathbb{B}^2$ is a two-dimensional Poincaré ball.

we aim to find the minimum cost bipartite matching of the images to the particles. The cost to minimize is the total hyperbolic distance of the hyperbolic features to the particles. We first compute all the pairwise distances between the hyperbolic features and the particles. This is the cost matrix of the bipartite graph. Then we use the Hungarian algorithm to optimize the assignment (Figure 14).

Suppose we train the encoder f_θ for T epochs. We run the hyperbolic instance assignment every other epoch to avoid instability during training. **We optimize the encoder f_θ to minimize the hyperbolic distance of the hyperbolic feature to the assigned particle in each batch**.

9. Details of Adversarial Attacks

For adversarial attacks, we use MNIST and CIFAR 10 as the benchmark and use FGSM [13] to attack the model. For MNIST, we leverage an ϵ of 0.07. For CIFAR10, as the range of the pixel values is from 0 to 255, we leverage an ϵ of 8. For model training, we standardize the pixel values by removing the mean and scaling to unit variance. Thus, the final ϵ on CIFAR10 is 8/(255*std), where std is the standard deviation used for normalization.

10. Details of Baselines

Holdout Retraining: We consider the Holdout Retraining proposed in [5]. The idea is that the distance of features of prototypical examples obtained from models trained on different datasets should be close. In Holdout Retraining, multiple models are trained on the same dataset. The distances of the features of the images obtained from different models are computed and ranked. The prototypical examples are those examples with the closest feature distance.

Model Confidence: Intuitively, the model should be confident on prototypical examples. Thus, it is natural to use the confidence of the model prediction as the criterion for prototypicality. Once we train a model on the dataset, we use the confidence of the model to rank the examples. The prototypical examples are those examples that the model is most

11. Gradually Adding More Congealed Images

We gradually increase the number of original images replaced by congealed images from 100 to 500. Still, as shown in Figure 15, HACK can learn a representation that captures the concept of prototypicality regardless of the number of congealed images. This again confirms the effectiveness of HACK for discovering prototypicality.

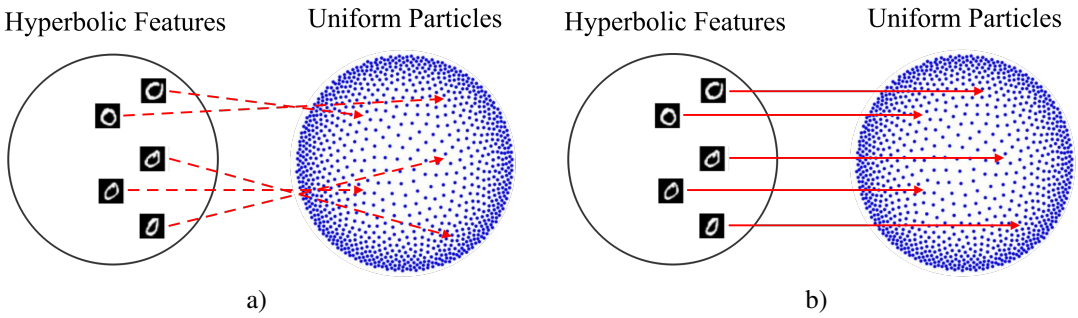

Figure 14. **Hyperbolic Instance Assignment minimizes the total hyperbolic distances between the image features and the particles.** a) Initial assignment. b) Optimized assignment.

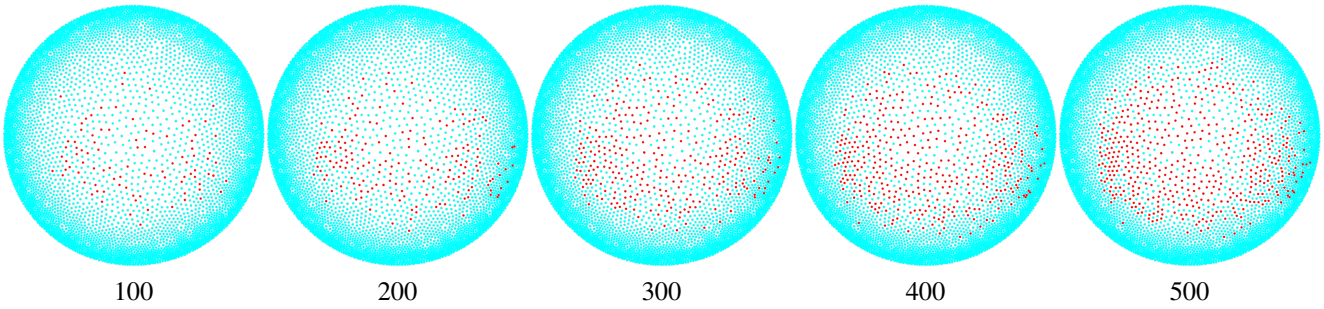

Figure 15. **HACK consistently places congealed images in the center of the Poincaré ball**. We gradually increase the number of original images replaced by congealed images from 100 to 500. The congealed images are marked with red dots and the original images are marked with cyan dots.

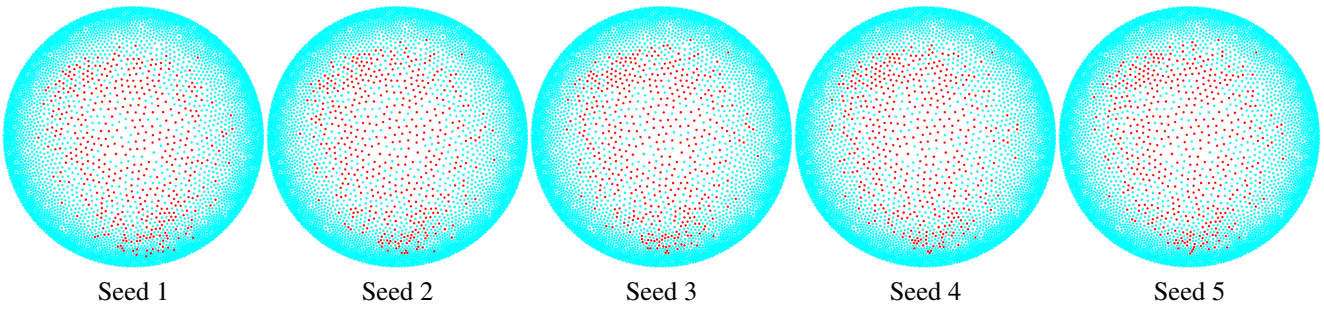

Figure 16. **HACK consistently places congealed images in the center of the Poincaré ball in multiple runs with different random seeds.**. The congealed images are marked with red dots and the original images are marked with cyan dots.

12. Different Random Seeds

We further run the assignment 5 times with different random seeds. The results are shown in Figure 16. We observe that the algorithm does not suffer from high variance and the congealed images are always assigned to the particles in the center of the Poincaré ball. This further confirms the efficacy of the proposed method for discovering prototypicality.

13. Emergence of Prototypicality in the Feature Space

Existing unsupervised learning methods mainly focus on learning features for differentiating different classes or samples [7, 19, 45]. The learned representations are transferred to various downstream tasks such as segmentation and detection. In contrast, the features learned by HACK aim at capturing prototypicality within a single class.

To investigate the effectiveness of HACK in revealing prototypicality, we can include or exclude congealed images

in the training process. When the congealed images are included in the training process, we expect the congealed images to be located in the center of the Poincaré ball while the original images to be located near the boundary of the Poincaré ball. When the congealed images are excluded from the training process, we expect the features of congealed images produced via the trained network to be located in the center of the Poincaré ball.

13.1. Training with congealed images and original images

We follow the same setups as in Section 4.3.1 of the main text. Figure 17 shows the hyperbolic features of the congealed images and original images in different training epochs. The features of the congealed images stay in the center of the Poincaré ball while the features of the original images gradually expand to the boundary.

13.2. Training only with original images

Figure 18 shows the hyperbolic features of the congealed images **when the model is trained only with original images**. As we have shown before, congealed images are naturally more typical than their corresponding original images since they are aligned with the average image. The features of congealed images are all located close to the center of the Poincaré ball. This demonstrates that prototypicality naturally emerges in the feature space.

Without using congealed images during training, we exclude any artifacts and further confirm the effectiveness of HACK for discovering prototypicality. We also observe that the features produced by HACK also capture the fine-grained similarities among the congealing images despite the fact that all the images are aligned with the average image.

14. Discussions on Societal Impact and Limitations.

We address the problem of unsupervised learning in hyperbolic space. We believe the proposed HACK should not raise any ethical considerations. We discuss current limitations below,

Applying to the Whole Dataset Currently, HACK is applied to each class separately. Thus, it would be interesting to apply HACK to all the classes at once without supervision. This is much more challenging since we need to differentiate between examples from different classes as well as the prototypical and semantic structure.

Exploring other Geometrical Structures We consider uniform packing in hyperbolic space to organize the images. It is also possible to extend HACK by specifying other geometrical structures to encourage the corresponding organization to emerge from the dataset.

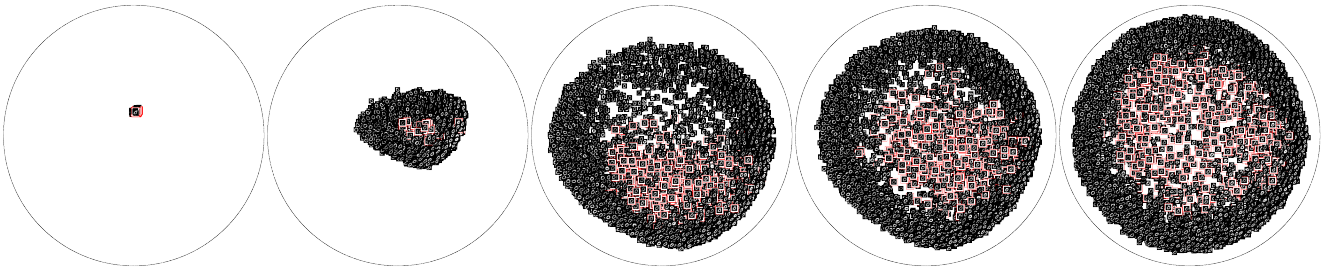

Figure 17. **Atypical images gradually move to the boundary of the Poincaré ball**. This shows that the representations learned by HACK capture prototypicality. Congealed images are in red boxes which are more typical. The network is trained with *both* the congealed images and original images.

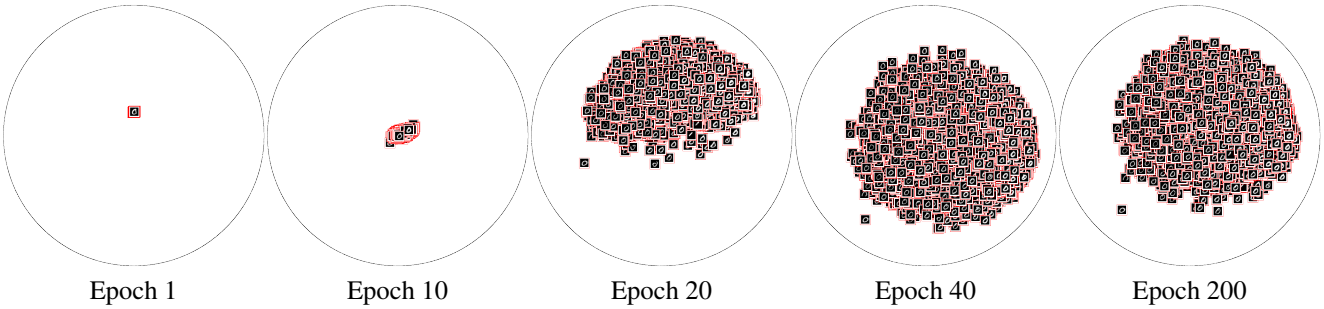

Figure 18. **The representations learned by HACK gradually capture prototypicality during the training process.** Congealed images are in red boxes which are more typical. We produce the features of the congealed images with the trained network in different epochs. The network is *only* trained with original images.