MIX'EM: Unsupervised Image Classification using a Mixture of Embeddings Supplementary Material

Ali Varamesh[1] and Tinne Tuytelaars[1]

ESAT-PSI, KU Leuven
{ali.varamesh,tinne.tuytelaars}@esat.kuleuven.be

1 Experimental setup

1.1 Learning rate, batch size, and schedule

For the experiments reported in the paper, the configuration is as follows:

- We use a learning rate of 1e-4 for STL10 and 5e-4 for CIFAR10/100-20. Changing the learning rate by up to a factor 5 does not significantly change the results, but may require adapting a shorter or longer training schedule to reach the same validation loss level.
- We use cosine decay schedule for the learning rate.
- The base encoder is trained for 600 epochs for each dataset. Next, MIX'EM is trained for 100 epochs on the base encoder. Please note that both stages are self-supervised.
- We use a batch-size of 512 for all datasets. Smaller batch size, as low as 128, degrades the results, though not significantly.

1.2 Augmentations

The data augmentations we follow the same setting as in [1]. This includes the following transformations applied with randomly selected parameters for each image at training:

- Random cropping using PyTorch's 'RandomResizedCrop' with scale in range (0.08,1)
- Horizontal flipping with probability 0.5
- Random color jitter using PyTorch's ColorJitter, where we set $brightness = 0.8$, $contrast = 0.8$, $saturation = 0.8$, and $hue = 0.2$.
- Convert images to gray-scale with probability 0.2
- Apply Gaussian blurring with kernel size 3x3 for CIFAR10/100-20 and 9x9 for STL10.

We also normalize images with dataset specific mean and standard deviation.

Table 1: Loss configuration for models used for comparison to the state-of-the-art

Dataset	λ_1	λ_2	λ_3	λ_4
STL10	1	0.1	0.05	0.1
CIFAR10	1	0.1	0.0	0.05
CIFAR100-20	1	0.01	0.05	0.0

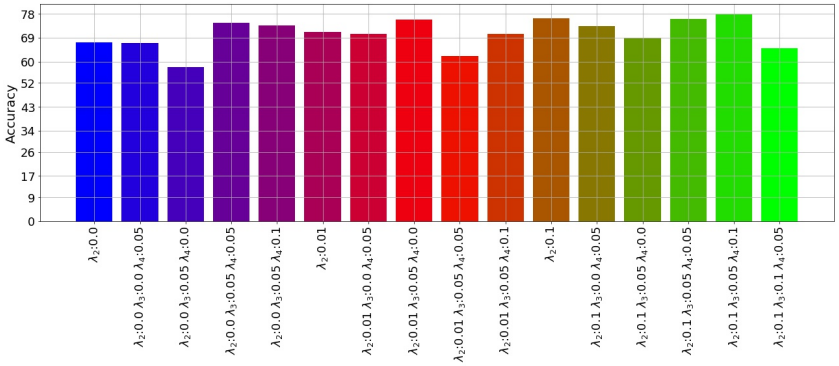


Fig. 1: Clustering accuracy on STL10 test split on models trained using different values of λ_2, λ_3, and λ_4. λ_1 is equal to one for all models.

1.3 Loss weights

Table 1 provides the loss configuration for the results shown in table 2 of the main paper. For further insight, in figures 1-3 we show the clustering accuracy for models trained with different combinations of the λ_2, λ_3, and λ_4 on the three datasets. The best configuration on each dataset can be different depending on the characteristics of each dataset. The validation loss guides us for selecting the best parameter set. Interestingly, for best results on CIFAR100-20, the L_{pull} (with weight λ_4) should be turned off. Intuitively, this makes sense, as the super-classes in CIFAR100-20 are not coherent in many cases. For example. 'bicycle' and 'truck' are in the same super-class in this dataset. Hence, forcing them to have similar representations degrades the results. On the other hand, for both STL10 and CIFAR10, L_{pull} is more important than L_{push} (with weight λ_3).

2 Visualizations

In figures 4-13, we present sample visualizations for each class of the STL10 dataset from its test split, where there are 800 images for each category. For each class, we provide separate visualization of the images based on their dominant mixture component. The model used for generating these images corresponds to the row (6) of table 1 in the main paper (the same model used for STL10 comparison to the state-of-the-art).

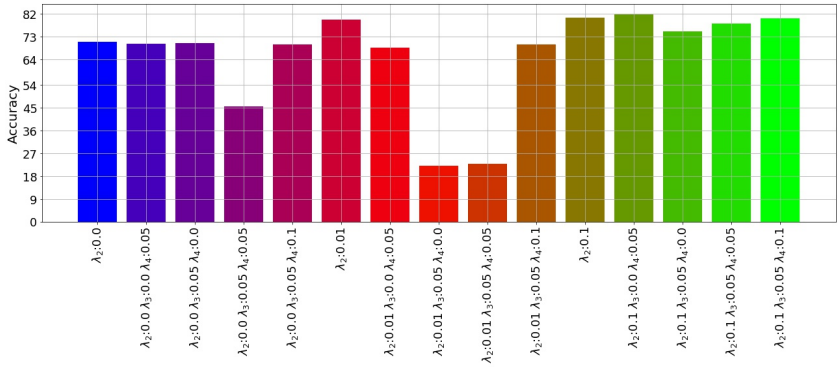


Fig. 2: Clustering accuracy on CIFAR10 test split on models trained using different values of λ_2, λ_3, and λ_4. λ_1 is equal to one for all models.

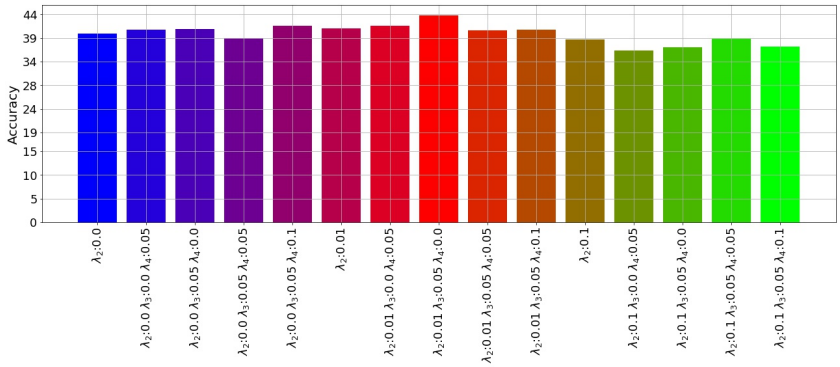
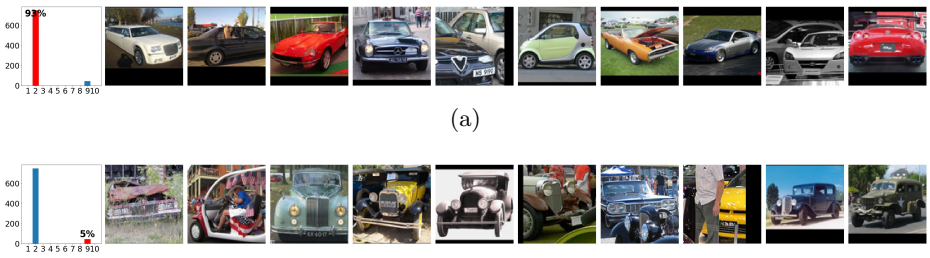


Fig. 3: Clustering accuracy on CIFAR100-20 test split on models trained using different values of λ_2, λ_3, and λ_4. λ_1 is equal to one for all models.

References

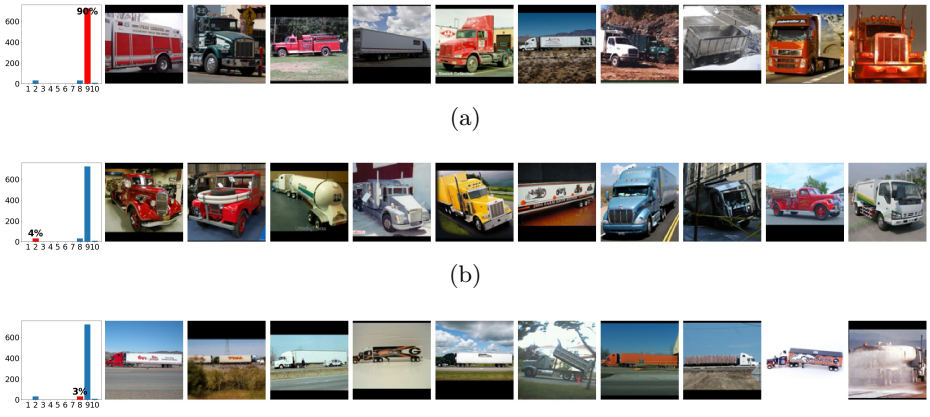
1. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. arXiv preprint arXiv:2002.05709 (2020) 1



(a)

(b)

Fig. 4: Visualization of the 'car' class on STL10 test split. Each row visualizes ten random images with the dominant component equal to a particular component of the MIX'EM. To the left of each row, we show the histogram of component distribution. The bar corresponding to the component visualized in a row is colored in red and annotated with the portion of images assigned to it.



(a)

(b)

(c)

Fig. 5: Visualization of the 'truck' class on STL10 test split. Each row visualizes ten random images with the dominant component equal to a particular component of the MIX'EM. To the left of each row, we show the histogram of component distribution. The bar corresponding to the component visualized in a row is colored in red and annotated with the portion of images assigned to it.

(a)

(b)

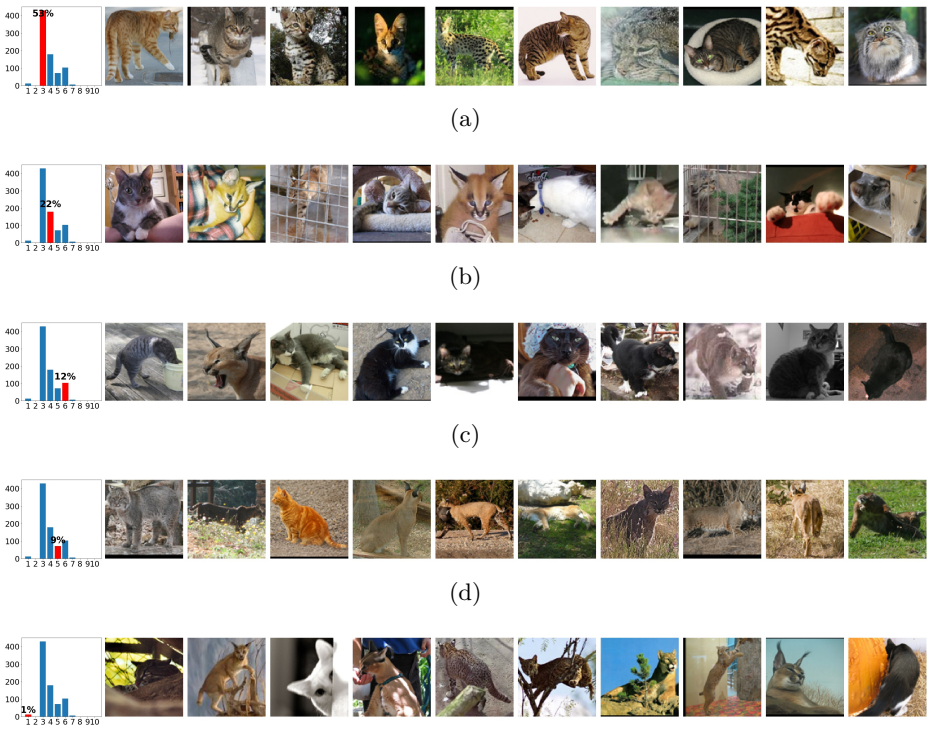
Fig. 6: Visualization of the 'ship' class on STL10 test split. Each row visualizes ten random images with the dominant component equal to a particular component of the MIX'EM. To the left of each row, we show the histogram of component distribution. The bar corresponding to the component visualized in a row is colored in red and annotated with the portion of images assigned to it.

(a)

(b)

Fig. 7: Visualization of the 'airplane' class on STL10 test split. Each row visualizes ten random images with the dominant component equal to a particular component of the MIX'EM. To the left of each row, we show the histogram of component distribution. The bar corresponding to the component visualized in a row is colored in red and annotated with the portion of images assigned to it.

Fig. 8: Visualization of the 'bird' class on STL10 test split. Each row visualizes ten random images with the dominant component equal to a particular component of the MIX'EM. To the left of each row, we show the histogram of component distribution. The bar corresponding to the component visualized in a row is colored in red and annotated with the portion of images assigned to it.



(a)

(b)

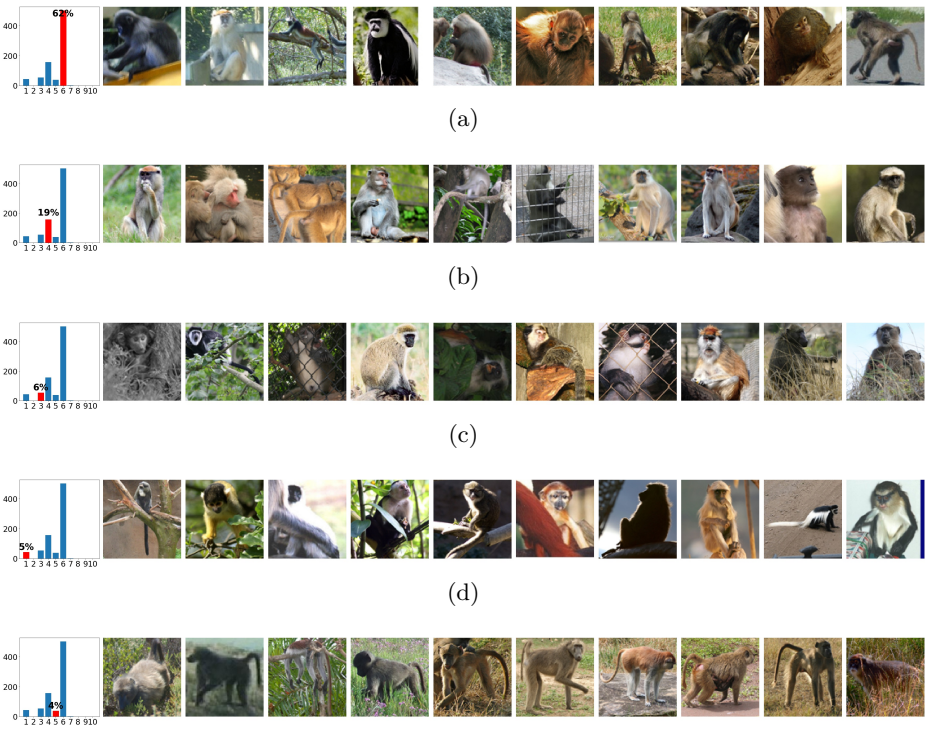
(c)

(d)

(e)

Fig. 9: Visualization of the 'cat' class on STL10 test split. Each row visualizes ten random images with the dominant component equal to a particular component of the MIX'EM. To the left of each row, we show the histogram of component distribution. The bar corresponding to the component visualized in a row is colored in red and annotated with the portion of images assigned to it.

Fig. 10: Visualization of the 'dog' class on STL10 test split. Each row visualizes ten random images with the dominant component equal to a particular component of the MIX'EM. To the left of each row, we show the histogram of component distribution. The bar corresponding to the component visualized in a row is colored in red and annotated with the portion of images assigned to it.



(a)

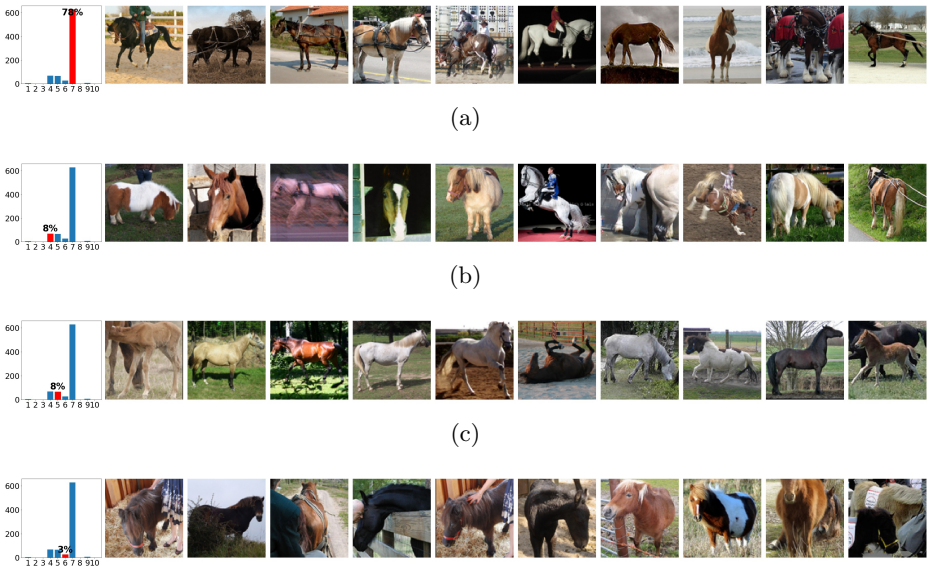
(b)

(c)

(d)

(e)

Fig. 11: Visualization of the 'monkey' class on STL10 test split. Each row visualizes ten random images with the dominant component equal to a particular component of the MIX'EM. To the left of each row, we show the histogram of component distribution. The bar corresponding to the component visualized in a row is colored in red and annotated with the portion of images assigned to it.



(a)

(b)

(c)

(d)

Fig. 12: Visualization of the 'horse' class on STL10 test split. Each row visualizes ten random images with the dominant component equal to a particular component of the MIX'EM. To the left of each row, we show the histogram of component distribution. The bar corresponding to the component visualized in a row is colored in red and annotated with the portion of images assigned to it.

(a)

(b)

(c)

(d)

(e)

(f)

Fig. 13: Visualization of the 'deer' class on STL10 test split. Each row visualizes ten random images with the dominant component equal to a particular component of the MIX'EM. To the left of each row, we show the histogram of component distribution. The bar corresponding to the component visualized in a row is colored in red and annotated with the portion of images assigned to it.