

7. Supplementary

7.1. Dataset Details

We evaluate the performance of HEX across different domains and levels of granularity. To assess the robustness and versatility of HEX, we choose 15 diverse datasets that span a wide range of scenarios that vary based on size, diversity, granularity, and difficulty of the data setting. These datasets include popular ones for benchmarking such as CIFAR-100 [35], Cifar-10 [35], TinyImageNet-200 [52], STL-10 [12], iNaturalist 2021 [49], and ImageNet [15]. Additionally, our intent is to also conduct experiments under conditions with significant class similarity, which makes them more susceptible to local collapse in their representations compared to standard datasets. Therefore, we select nine fine-grained datasets including FGVC Aircraft [39], Oxford 102 Flowers [41], Stanford Cars [14], Stanford Dogs [30], Oxford Pets [43], NABirds [48], Caltech 101 [19], Food 101 [5], and CUB-200-2011 [50]. A description of each dataset is given in Table 12.

In addition to ImageNet, we include its subset, ImageNet-100 [15], for comparative experiments. ImageNet-100 introduces a smaller number of classes than ImageNet and naturally has less class overlap. By comparing datasets with similar data distribution conditions, but differing only in the degree of class overlap, we aim to observe how our method improves performance by preventing local collapse.

7.2. Limitations of Algorithm

Despite the advantages of HEX, there are some potential limitations. For example, in certain cases manual choosing the thresholding parameter performs better than the adaptive strategy. Consequently, some type of hyperparameter search would have to occur in these situations. Additionally, our method seems to perform better in situations with a higher degree of class overlap and complexity. Therefore, performance benefits may not be as great within lower complexity data settings. Furthermore, our algorithm relies on the emergence of hierarchical structures during training. However, it isn't always clear from a human perspective what the nature of a hierarchy can look like for certain application domains. Further research into the nature of hierarchies in certain domains may be needed to fully understand the meaning of hierarchical emergence in general.

7.3. Additional Training Details

7.3.1 Code Acknowledgement

We make use of the solo-learn codebase for all experiments [13]. The link to their code can be found at this [repository](#). Our specific implementations will be released upon acceptance of the paper.

7.3.2 Specific SSL Method Details

In Table 6, we show all the hyperparameters associated with each of the SSL algorithms highlighted in the results section of our paper. The hyperparameters in the table are the basic hyperparameters that were associated with the solo-learn codebase. However, these parameters change slightly depending on the applied dataset. These changes were minor and very method-specific. Examples of this include slight variations in the queue size to reflect the size of the dataset that the SSL method was trained on. Otherwise, this table reflects the basic training hyperparameters for all SSL methods in the paper.

Additionally, there were instances within the paper where integration of the HEX loss required structural changes to the optimization process of the associated SSL method. For example, the losses for SimCLR and NNCLR were directly replaced by their HEX version of the loss. In Section 4, we detail the version of the loss that directly replaces the SimCLR loss. The NNCLR version of HEX is equivalent to the SimCLR version of HEX except the augmented sample in the positive set is that of its nearest neighbor, rather than derived from the same sample as the anchor image. The All4One methodology uses the NNCLR loss as part of their methodology. However, the HEX version of All4One replaces its NNCLR loss with the NNCLR + HEX version. In the case of dimensional contrastive approaches like VicReg and Barlow Twins, we add the SimCLR version of HEX to each of these losses with an α parameter to weight the contribution of each loss. For VicReg specifically, we also had to consider the hyperparameters that went into weighting each of its other loss functions. We empirically found that scaling the HEX loss by a factor of 5 allowed stable training alongside the other loss functions in the VicReg formulation. Barlow Twins did not require any additional tuning to adapt to additional HEX regularization.

All methods received the default augmentation policy described within the solo-learn github. The only parameter that was changed in a dataset specific manner was the random resized crop parameter. If the images were smaller than 224x224, they received a random crop size that is equal to the size of the images within the dataset. However, if the images were larger, then the parameter was defaulted to 224x224.

7.3.3 Transformer Experiments

We used the standard vision transformer and the Swin transformer for all experiments in our study. We kept all hyperparameters and optimizers the same as in the case of the ResNet-50 experiments with the exception of transformer specific changes such as a decoder embed dimension of 512, depth of 8, patch size of 4, and 16 decoder heads.

7.3.4 Hyperparameter Variation

We show in Figure 7 the impact of varying batch size on the performance of our method. We see that the performance improvements of HEX are maintained even when this hyperparameter is varied.

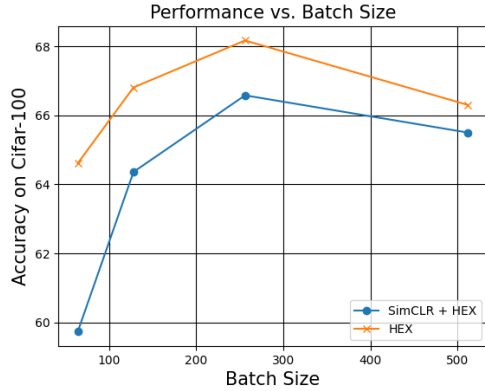


Figure 7. This shows performance variation as batch size is varied. The HEX approach is able to maintain performance improvements.

We also show an analysis of varying the threshold hyperparameter when using manual strategies in Table 7. We see that the performance on the CIFAR-100 dataset is sensitive to the chosen thresholding parameters. This demonstrates the superiority of the adaptive threshold setting method over the manual threshold setting method as this does not require potentially expensive tuning of hyperparameters. In Table 7, we also detail how these hyperparameters were manually set. This, alongside improved results, illustrates that manual hyperparameter tuning is potentially a sub-optimal strategy to the adaptive method when considering the results across all experiments on the CIFAR-100 dataset. In the Table, “cos” refers to an additional thresholding strategy that lowers the threshold value in a continuous fashion from its starting value to a minimal value along a cosine curve that is a function of the epoch of training.

7.4. Additional Analysis

7.4.1 Threshold Analysis

The adaptive version of the HEX loss uses the cosine similarity distribution to assign a threshold at different points in training. We show an example of the exact value of this threshold at different points in training in Figure 8. This was computed as the average threshold value of all 10000 images in the CIFAR-100 test set. However, in practice this threshold is computed on a batch-wise basis rather than across the entire dataset as a whole. We see that the adaptive threshold gradually decreases during training to reflect the shifting cosine similarity distribution of the dataset. Later

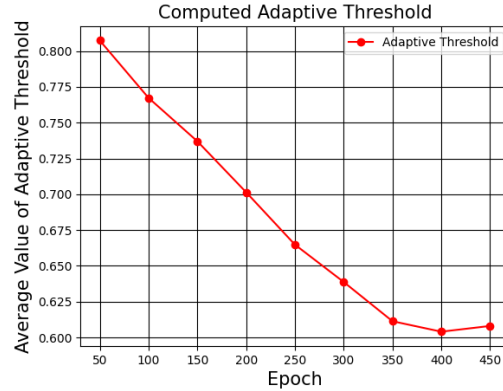


Figure 8. This shows the value of the threshold parameter in the adaptive setting across different epochs of training on the CIFAR-100 test set.

in training, this decrease starts to level out as the hierarchical structure of the dataset emerges with persistent larger cosine similarity values at samples within the same hierarchical grouping.

Additionally, in Figure 9 we show the average number of samples in the Cifar-100 test set are above the threshold parameter at different epochs of training. This value gradually increases as the threshold parameter is lowered as a result of growing confidence in the hierarchical structure of the representation space at later points of training.

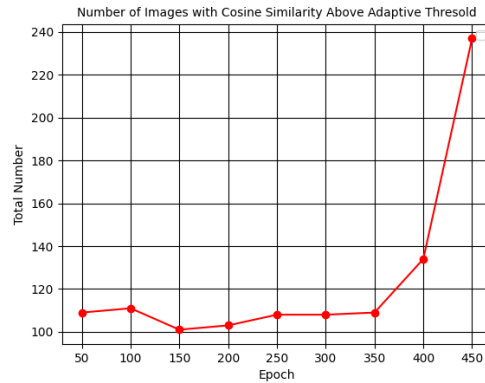


Figure 9. This shows the value of the threshold parameter in the adaptive setting across different epochs of training on the Cifar-100 test set.

7.4.2 Re-weighting Ablation

We also show an ablation study comparing our method against a different approach [45] that also introduces a re-weighting of negative terms. The difference is that our method explicitly identifies samples at different points of

Method	Projection	Method Specific Parameters	Temperature	Optimizer	Batch Size	LR	Decay
SimCLR	2048-128	N/A	0.1	LARS	256	0.4	1e-5
NNCLR	2048-4096-256	Queue = 98304	0.2	LARS	256	1.0	1e-5
All4One	2048-4096-256	Queue = 98304 Momentum = 0.99	0.2	LARS	256	1.0	1e-5
Barlow	2048 - 2048	Scale Loss = 0.1	N/A	LARS	256	0.3	1e-4
BYOL	4096 - 256 - 4096	Momentum = 0.99	N/A	LARS	256	1.0	1e-5
Moco v2	2048-256	Queue Size = 65536 Momentum = 0.99	0.2	SGD	256	0.3	1e-4
VicReg	2048 - 2048	Sim_weight = 25 Var_weight = 25 Cov_weight = 1	N/A	LARS	256	0.3	1e-4
SimSiam	2048-2048-512	N/A	0.2	SGD	256	0.5	1e-5

Table 6. This table shows the baseline training parameters for each SSL method. Variations from these parameters are discussed in the text.

Hyperparameter Tuning of Manual Strategy				Accuracy	
dist_thres	dist_min	step_down	step_type	Top-1	Top-5
0.95	0.65	-	cos	66.63	88.96
0.85	0.45	-	cos	60.57	86.37
0.95	0.65	0.1	step	66.20	89.04
0.85	0.45	0.1	step	65.63	88.18
0.95	0.65	0.05	step	66.23	88.70
0.85	0.45	0.05	step	65.08	88.29
Proposed Adaptive Strategies				67.56	90.02

Table 7. This demonstrates 6 different manual hyperparameter tuning approaches. In CIFAR-100 datasets, we can see none of them were superior to the adaptive strategy. “step” refers to the stepwise manual strategy while “cos” represents the cosine thresholding strategy.

Method	Cifar-100 Linear Evaluation Accuracy
HCL	64.64
SimCLR	64.46
HEX	67.56

Table 8. We show a comparison with a different re-weighting strategy that does not have a mechanism to target hierarchical groupings. HCL is trained in the manner described in its paper, [45] while we train SimCLR and HEX in the manner described in this paper.

training based on the intuition of identifying hierarchical groupings while the other approach generally re-weights the entire batch as a whole. We show performance improvements in Table 8 within a fine-tuning setting on Cifar-100 with the training parameters described in this paper while using a stepwise strategy for HEX. This shows that our performance improvements are not just a result of reweighting, but also due to the importance of identifying the right hierarchically aligned samples.

7.4.3 Fine-grained pretraining Ablation

We show an ablation study to evaluate the efficacy of our proposed HEX method on a less diverse representation space. Fine-grained datasets are less diverse due to all their classes sharing features in common with each other. In the main paper, we showed the result of fine-tuning and transfer learning within the context of a pre-trained Imagenet model. Instead of pretraining on ImageNet datasets, which has greater feature diversity, we pre-train a NNCLR model with fine-grained datasets as depicted in Table 9. We then compare against the same NNCLR setup with our additional HEX regularization. As shown, applying HEX regularization during SSL pretraining consistently results in better classification performance across most datasets compared to not applying HEX. This demonstrates that our HEX technique effectively disentangles the local-collapse phenomena.

	NNCLR		NNCLR + HEX	
	Top-1	Top-5	Top-1	Top-5
Cars [14]	42.92	71.99	46.60	75.66
Flowers [41]	32.94	59.22	28.63	55.88
NABirds [48]	21.83	45.19	22.26	45.67
Caltech-101 [19]	72.60	93.79	73.73	94.35
Dogs [30]	42.06	73.75	47.40	77.95
Pets [43]	47.78	81.90	53.23	85.88
Aircraft [39]	30.03	57.82	29.07	56.68
Caltech-Birds [50]	19.49	43.22	20.76	47.36

Table 9. By comparing with and without HEX methods on pre-trained models with fine-grained datasets, we show that HEX can still enhance classification performance in a less diverse representation space.

	iNat21		
	Type	Top-1	Top-5
SimCLR	None	20.67	38.64
SimCLR + HEX	Ada	22.01	40.92
SimCLR + HEX	Sup	23.35	42.94

Table 10. This shows the performance of HEX using one of the superclasses within iNat21.

	Imagenet		
	Type	Top-1	Top-5
All4One	None	52.83	78.29
All4One + HEX	Ada	54.30	79.09

Table 11. This shows the performance of HEX on All4One on the Imagenet dataset with 50 epochs of pre-training within the linear evaluation setting.

7.4.4 Hierarchical Emergence

To get an intuitive sense of the emergence of hierarchies during training, we take each model m_e described in Section 3 and pass in the test set to get an associated representation matrix R with each row corresponding to a test image representation r_i . We associate each r_i with its label y_i and superclass label y_{si} . Together, this information is used to produce Figures 10 and 11. In Figure 10 we show a UMAP [40] visualization of the representation space at the beginning and end of training under different conditions and subsets of the test data space. In the top row of this figure, we see all points in the test set labeled by their superclass. It is visually evident that all superclass clusters become more separable by the end of training. This is shown more clearly in the middle row of Figure 10 where we randomly choose samples from 5 random superclasses and show the organization of their clusters at the beginning and end of training. We also analyze the organization of classes within a single superclass in the bottom row of Figure 10. These plots show that classes within a superclass also become more separable with respect to each other. All of these plots together indicate that representations naturally cluster in terms of superclasses as well as regular ground truth classes over the course of SSL training. This is further evidenced in Figure 11 where we observe that the KNN accuracy of the representations with respect to both superclass and ground truth labels both increase over the course of training for the dimension-based Barlow Twins [54] and sample-based NNCLR strategies.

We also show an additional study of using hierarchies in the iNaturalist21 dataset. This dataset is composed in a hierarchical fashion that corresponds to the taxonomies that come with animal and plant species. We show in Table 10 that appropriately using the hierarchy information for HEX

in a supervised fashion of the associated superclasses leads to performance improvements in the same way as the Cifar-100 datasets in the main paper.

7.4.5 Additional ImageNet Experiments

We also added an additional experiment with 50 epochs of pre-training with the All4One methodology. This is shown in Table 11. We see that HEX is able to achieve performance improvements on the Imagenet dataset for this algorithm as well.

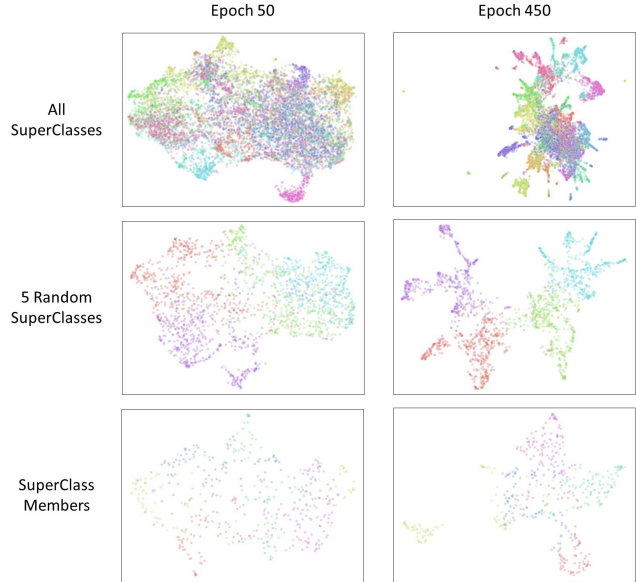


Figure 10. This plot shows the umap embeddings of the samples of Cifar-100 labeled by their superclass label. We show the organization of the representation space under different data access settings.

7.4.6 Embedding Space Analysis

We analyze a variety of trends with respect to cosine similarity distributions in Figures 12 and 13. In Figure 12, we show the average cosine similarity value for each anchor image with all other images in the test set of Cifar-100 at different epochs of training with the NNCLR method within both the projection space and representation space of the model. On average, for both spaces the average cosine similarity of superclass samples is higher than that of regular samples. Additionally, we see that the average cosine similarity of the representation space is higher than that of the projection space possibly due to retaining a greater number of redundant features between samples. From this analysis, it is unclear whether to compute the cosine similarity with respect to the representation space or projection space when estimating the presence of hierarchical groupings. To

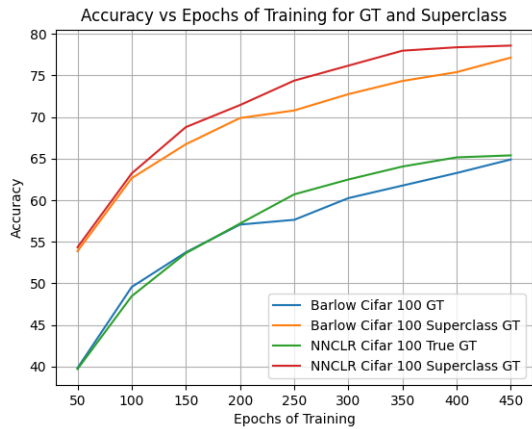


Figure 11. This shows the KNN accuracy using the cosine similarity distance metric for the NNCLR [17] and Barlow Twins [54] SSL methods across 450 epochs of training for both ground truth Cifar-100 labels as well as labels denoting the superclass of each sample.

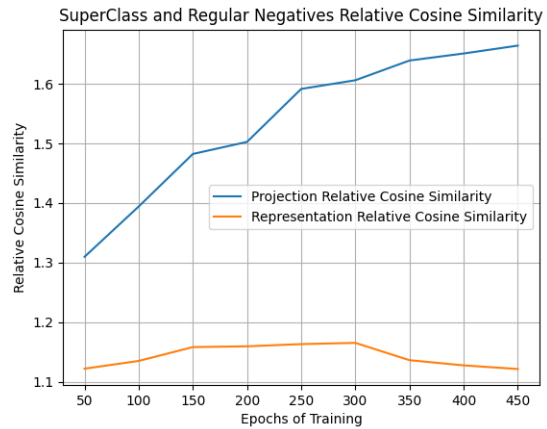


Figure 13. This shows the relative cosine similarity between superclass and regular negatives over the course of training on Cifar-100 with the NNCLR method.

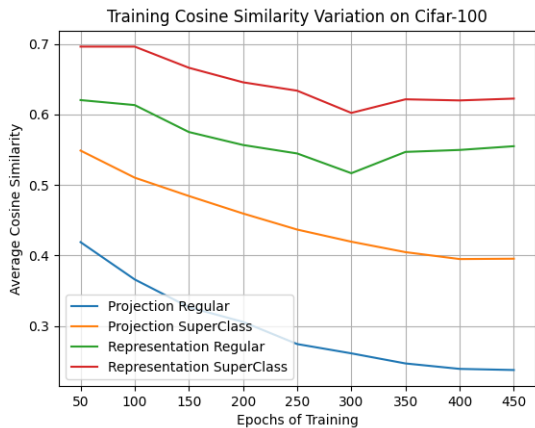


Figure 12. This shows the variation in average pairwise cosine similarity between superclass negatives and regular negatives for each instance in the Cifar-100 test set throughout different epochs of training. This plot is from the NNCLR method.

analyze these trends further, we take the ratio of superclass cosine similarity to regular class cosine similarity for both spaces across all epochs of training. We observe in Figure 13 that this ratio increases over training for the projection space, but not the representation space. This indicates that the cosine similarity metric of the embedding space becomes more hierarchically aligned as training progresses due to a greater relative difference in the cosine similarity between the two subsets. It is interesting to note that this does not hold for the cosine similarity of the representation space which shows that the space in which this metric is computed matters in terms of serving as a useful indicator

of hierarchical separability.

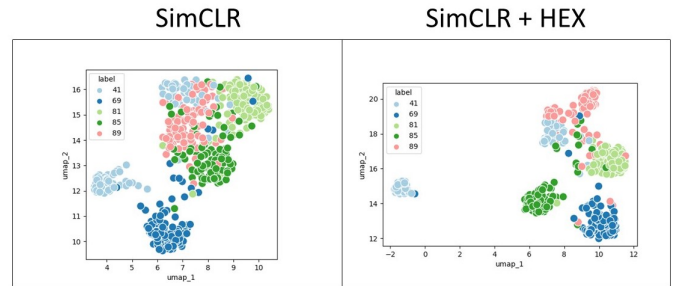


Figure 14. This shows the embedding space of the vehicle superclass before and after the application of HEX to the SimCLR algorithm.

We also show in Figure 14, the embedding space of the Cifar-100 test set for samples from the vehicle superclass for both SimCLR and SimCLR + HEX. This visualization was created using the UMAP algorithm on the representation space of each model. We see that the application of HEX causes additional spread for these locally clustered samples within the same superclass. This results in greater separability between classes that are prone to collapsing with respect to each other compared to the embedding space produced by SimCLR alone.

We also visually show the samples with the highest cosine similarity in randomly generated batches of 128 as each anchor image in Figure 15. This plot was produced using a model trained with the SimCLR methodology for 400 epochs of training. Every image is labeled with the superclass that it belongs to. Note that in most cases, the retrieved samples are members of the same superclass as each anchor image.











































Anchor Image	Retrieved Samples				
Scenes 	Scenes 	Furniture 	Scenes 	Furniture 	Scenes 
Food Bins 	Food Bins 	Food Bins 	Food Bins 	Food Bins 	Fruits 
Furniture 	Furniture 	Furniture 	Furniture 	Vehicles 1 	People 
People 	People 	People 	People 	People 	Fruits 
Invertebrae 	Furniture 	Vehicles 2 	Invertebrae 	Invertebrae 	Med Mammal 
Flowers 	Flowers 	Flowers 	Tiny Mammal 	Food Bins 	Flowers 
Furniture 	Furniture 	Herbivores 	Electrical 	Furniture 	Vehicles 1 

Figure 15. This shows the images with the highest cosine similarity in each batch with the given anchor image. The label above each image identifies the superclass that it belongs to. These images were drawn from the Cifar-100 test set.

Dataset	Abbreviation & Link	Description	# of classes
CIFAR-100 [35]	cifar100	100 classes of 32x32 color images, including animals, vehicles, and various objects commonly found in the world.	100
CIFAR-10 [35]	cifar10	10 classes of 32x32 color images featuring everyday objects and scenes such as airplanes, cars, and animals.	10
Tiny ImageNet [52]	tinyimagenet200	200 classes of 64x64 images, a smaller version of the ImageNet dataset, used for object recognition and classification tasks.	200
STL-10 [12]	stl10	10 classes of 96x96 images, designed for developing unsupervised feature learning, deep learning, and self-taught learning algorithms.	10
iNaturalist 2021 [49]	inat21	Large-scale dataset with over 10,000 species, collected from photographs of plants and animals in their natural environments for fine-grained classification.	10,000
ImageNet [15]	imagenet	Large dataset with over 1,000 classes, used for image classification and object detection, containing millions of images across a wide variety of categories.	1,000
ImageNet-100	imagenet100	Subset of ImageNet with 100 classes, providing a more manageable dataset for specific research and development purposes.	100
FGVC Aircraft [39]	aircraft	Aircraft categorization dataset with 100 classes, featuring various aircraft models including different variants and manufacturers.	100
Oxford 102 Flowers [41]	flowers	102 category flower dataset, containing images of flowers commonly found in the United Kingdom, used for fine-grained visual classification.	102
Stanford Cars [14]	cars	Car categorization dataset with 196 classes, covering a wide range of car models from various manufacturers, including different years and trims.	196
Stanford Dogs [30]	dogs	Dog breed classification dataset with 120 classes, containing images of different dog breeds, used for fine-grained classification tasks.	120
Oxford Pets [43]	pets	Cat and dog breeds dataset with 37 classes, each breed has roughly 200 images, used for pet recognition and classification.	37
NABirds [48]	NABirds	North American birds dataset with 1011 species, featuring images of birds in various poses and environments, used for fine-grained bird species identification.	1011
Caltech 101 [19]	caltech-101	101 object categories dataset and background, containing images of various objects including animals, buildings, and tools, used for object recognition and classification.	102
Food 101 [5]	food	101 food categories with 101,000 images, featuring various dishes and cuisines from around the world, used for food recognition tasks.	101
CUB-200-2011 [50]	Caltech-Birds	200 bird species dataset with annotated bounding boxes and part locations, used for fine-grained bird species classification and localization.	200

Table 12. Overview of various image datasets