Approaches for finding sample pairs in contrastive learning

Klara M. Gutekunst

University of Kassel, Germany
klara.gutekunst@student.uni-kassel.de

Abstract. Unsupervised learning techniques are of interest to many researchers, as they allow training models on data without any labels. Self-Supervised Learning (SSL) is a subset of unsupervised learning, where labels are generated from unlabelled training data. Contrastive Learning (CL) is a SSL technique that is frequently used in representation learning. The representations of similar samples are supposed to be encoded within close range of each other, while the representations of dissimilar samples are pushed apart. The selection of these dis-/similar pairs is subject to research. This paper reviews different approaches for finding sample pairs in CL, with a focus on hard sample mining.

Keywords: Contrastive Learning · Self-Supervised Learning · hard sample mining.

1 Introduction

SSL is an unsupervised learning technique that allows training models on data without any labels. The idea is to generate labels from the unlabelled training data contemplating a pre-text task.

Researchers usually select self-supervised pre-text tasks such that the targets can be generated without human annotations Cao et al. [2020]. The pre-text task instance discrimination considers each instance from the dataset its class. A sample and its augmentations are considered positive pairs, while all other samples are considered negative pairs. Hence, the model acquires an understanding of distinguishing between the instances and learns invariance to (image) transformations Cao et al. [2020], Caron et al. [2020], Zhuang et al. [2019], Hao et al. [2024], Wang and Liu [2021].

In order to explain why the embedding space proximity of generated samples to the anchor x is relevant to the efficiency during training, one can consider a simple example in Euclidean space. Imagine images as input to a Neural Network (NN), which projects them onto $f_\theta(x) \in \mathbb{R}^d$, where θ are the parameters of the NN. The effect of the distance between the anchor x and the positive x^+ (negative x^-) sample on the loss is visualized in Figure 1. Since difficult samples hold more (gradient) information, they have a higher loss value. Hence, close negative pairs (x, x^-) are considered hard Robinson et al. [2021], while distant positive pairs (x, x^+) are considered hard.

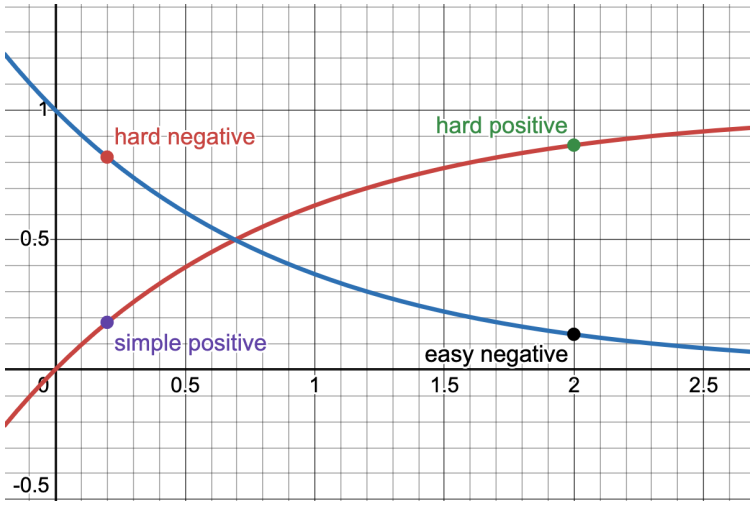


Fig. 1: The impact of the distance between a generated sample and its anchor (x-axis) on the loss function (y-axis). The red curve represents the loss function for positive samples, whereas the blue curve denotes the loss function for negative samples. Hard samples convey more (gradient) information than easy samples and thus, have a higher loss value. While distant positive pairs are considered hard, for negative samples, small proximity ones are considered hard.

The selection of positive pairs (x, x^+) is subject to multiple papers, which propose different strategies. In the case of unsupervised learning, generally, the positive sample x^+ is generated by applying a transformation to the anchor x. Popular image augmentation techniques include random cropping, colour jittering, Gaussian blur, rotation and scaling Ho and Nvasconcelos [2020], Robinson et al. [2021], Zhuang et al. [2024]. Graph augmentations involve adding or removing edges and nodes, whereas words or sentences are added or masked in the context of natural language processing Zhuang et al. [2024].

Another approach is to use so-called Positive-Unlabeled (PU) learning, where learning is carried out on a set of positive and a set of unlabeled samples Chuang et al. [2020].

Simple selection techniques of negative samples include uniform sampling from the dataset or batch. However, this approach is prone to two issues Robinson et al. [2021], Dong et al. [2024]:

1. Selected samples are not necessarily hard negatives since the approach does not consider the embedding space proximity.
2. Selected samples might actually belong to the same class as the anchor and thus, can be denoted False Negative (FN).

Some approaches use a so-called class oracle to boost the performance of the model. If FNs Hao et al. [2024], Zhuang et al. [2024], Xia et al. [2022], i.e.

samples that belong to the latent class of the anchor but are considered negative, are sampled during hard negative mining, samples of the same class are pushed further apart in the embedding space. To avoid performance deterioration, this approach removes the FNs from the set of negative samples and thus, increases the performance of the model Kalantidis et al. [2020].

2 Related Work

Naturally, when using descriptive statistics to describe data, (empirical estimated) distributions play a crucial role. Some researchers obtain their positive and negative samples from distributions. Unfortunately, since CL is a form of unsupervised learning, the true data distribution of the different classes is often not available. Therefore, some scientists formulate assumptions or simplify the problem. Possible assumptions include that the data distribution is uniform, or approximating the positive sample distribution by sampling from a set of transformations or using the overall data distribution as a proxy for the negative sample distribution Chuang et al. [2020], Robinson et al. [2021]. In other cases, the class distributions are approximated using beta mixture models (BMMs) Xia et al. [2022].

Given the assumption that the data distribution is sufficiently well approximated, it is possible to consider probabilities of samples being FNs during the selection process of samples. Needless to say, the goal is to avoid sampling FNs as negative samples. To this end, scientists have proposed different strategies, which mostly boil down to incorporating the possibility of a potential negative sample being a FN or True Negative (TN) Chuang et al. [2020], Robinson et al. [2021], Xia et al. [2022].

Irrespective of its usage in the context of estimation of distributions, data augmentation is a common technique to create positive samples. Often, an augmentation strategy is randomly sampled from a set of possible augmentations. The motivation behind this is to increase the diversity of the positive samples in order to drive the model to learn features invariant to translations Cao et al. [2020], Caron et al. [2020], Zhuang et al. [2019], Hao et al. [2024], Wang and Liu [2021].

Since CL objectives are often formulated in terms of distances or similarities between pairs of samples, the idea of using clustering techniques is a natural choice. Intuitively, clusters of similar samples should be considered as positive samples and thus, should be encoded close to each other. Conversely, samples from different clusters should be encoded far apart.

Multiple methods have been proposed to generate samples via clustering. Some ideas focus on high intra-cluster similarities to improve the alignment of the embeddings Zhong et al. [2020]. Other ideas define different neighbour regions to condense representation within an inner radius while repelling samples from an outer radius Zhuang et al. [2019]. Another approach is to consider both Euclidean distance and semantic similarity to generate hard samples Iscen et al. [2018a]. Moreover, the Prototypical Contrastive Learning (PCL) technique de-

fines positive samples as cluster centroids from one of the multiple clusterings for different numbers of clusters to encode the hierarchical structure of the data Li et al. [2021].

Another prominent concept is the usage of memory banks to store embeddings of the data. Kalantidis et al. fill these memory banks with embeddings of negative samples and propose two approaches for generating new hard negatives: Two of the most difficult samples currently stored in the memory bank are randomly selected and mixed. The second approach is to use only one of the existing negative samples and mix it with the anchor to create a new sample.

Xia et al. [2022] propose a method that extends the idea of Kalantidis et al. [2020] by weighting randomly selected negative samples with their relative similarity to the anchor when mixing them to create more difficult negative samples.

It is also possible to use the memory bank to store the embeddings of the positive samples. Similarly, either randomly chosen samples can be used individually or the samples can be weighted by their hardness during loss calculation Dong et al. [2024].

Multiple approaches, including Deep Robust Clustering (DRC), PCL and ProGCL, use the Expectation-Maximization (EM) algorithm to reduce computational costs or to find solutions via approximations. Both DRC Zhong et al. [2020] and ProGCL Li et al. [2021] are clustering-based methods while ProGCL Xia et al. [2022] is a distribution-based method.

Since most approaches use a temperature parameter to control the hardness of the negative samples, Wang and Liu [2021] and Hao et al. [2024] investigate the impact of the temperature on the performance of the model. They find that the CL loss function optimizes hard samples by penalizing them according to their hardness. If the temperature is small, only the closest points are penalized and others are not. This can result in a uniformly distributed embedding space.

Zhuang et al. [2024] propose a curriculum learning approach to generate hard negative samples. They outline why curriculum learning is beneficial for CL and how it can be implemented.

3 Sampling techniques

The following sections present different sampling techniques for positive and negative samples in CL. Firstly, sampling strategies based on distributions are discussed. Secondly, clustering-based sampling techniques are presented and the impact of temperature on the sampling process is discussed. Then, a technique relying on a memory bank is introduced. Finally, curricular weighting is discussed. The latter is a technique that is not directly of interest in terms of sample generation but in terms of sample selection during training.

3.1 Sample from distributions

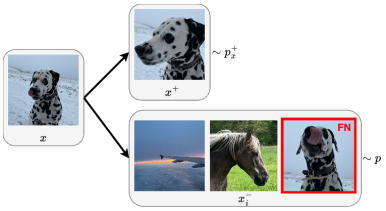
Assuming the distribution p of a variable x is known, it is possible to sample from it. Sampling can either be done directly or via approximation schemes such

as rejection sampling and Monte Carlo sampling Robinson et al. [2021]. However, in the context of CL, the true data distribution of the different classes is often not available due to the nature of unsupervised learning scenarios. Therefore, some scientists formulate assumptions or simplify the problem.

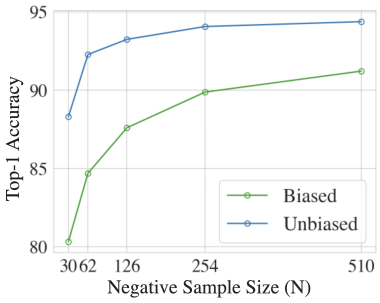
Let $\rho(c), c \in \mathcal{C}$ be the distribution over the latent classes and let $h : \mathcal{X} \to \mathcal{C}$ be the ground truth assigning class labels $c \in \mathcal{C}$ to inputs $x \in \mathcal{X}$. Hence, $x \sim x'$ if $h(x) = h(x')$ Robinson et al. [2021], Chuang et al. [2020]. Presuming that $x^- \sim q$ and the anchor x is drawn from the data distribution p, i.e. $x \sim p$.

PU approximation Chuang et al. [2020] assume a PU learning scenario, where positive samples and an unlabeled image dataset $p(x)$ are available. Since positive samples may not be available in reality, the positive distribution p^+ is mimicked by data augmentations.

Chuang et al.'s goal is to sample TNs. They denote sampling bias the phenomenon of sampling FNs as illustrated in Figure 2a. When randomly sampling negative samples from the data distribution $p(x)$ a negative sample can inherently belong to the same latent class as the anchor. The negative effect of sampling bias on the model's performance is illustrated in Figure 2b. Consequently, they propose a debiased contrastive objective that corrects for sampling FNs in an unsupervised scenario. The idea is to generate positive samples using augmentations, to sample negative samples x^- from the data distribution $p(x)$ and to add a correction term for FNs in the loss function.



(a) Visualization of sampling bias similar to Chuang et al. [2020]. Sampling x_i^- from p can result in FN.



(b) Negative influence of sampling bias on accuracy from Chuang et al. [2020].

Fig. 2: Visualization of sampling bias and its effect on the model's performance.

Choosing the hardness of negative samples Robinson et al. [2021] extend Chuang et al. [2020]'s approach to sampling from distributions of image data.

Again, sampling from the positive sample distribution is approximated by sampling from a set of transformations while negative samples x^- are obtained by sampling from the data distribution $p(x)$.

Robinson et al. [2021] introduce the concentration parameter β to enable user-specific hardness of the negative samples. β is a multiplicative factor for the similarity, i.e. dot product $\beta f(x)^{\mathrm{T}} f(x^-)$, of two sample feature space representations. Hence, large values of β lead to sampling very hard negative samples and thus, increase the risk of sampling FNs.

Graphs Another prominent data structure is graphs, which can model social networks, citation networks, or knowledge graphs. Xia et al. [2022] propose an approach to mine negative samples for CL on graphs called ProGCL. As a motivation, the authors compare the similarity between the anchor and TNs to the similarity between the anchor and FNs across multiple datasets for both conventional CL and Graph Contrastive Learning (GCL). The resulting distributions shown in Figure 3 indicate that for GCL the majority of highly similar negative samples are FNs, while for conventional CL there seems to be no clear trend. Moreover, the authors found that both the TN and FN distributions are best modeled by a BMM since it is able to fit the skewed empirical distribution. The distribution is fitted using the EM algorithm on a subset of samples for a reduction of computational costs.

(a) CIFAR-10 (Image)

(b) Coauthor-CS (Graph). Most similar negative samples are FNs.

Fig. 3: Histogram of similarity between anchor and negative samples for CL and GCL from Xia et al. [2022]. Blue denotes the empirical distribution of the TNs, while orange denotes FNs.

Based on this observation, the authors propose considering both the probability of a sample being a TN and the sample's similarity to the anchor, i.e. its hardness, during the selection process of negative samples.

3.2 Sample via clustering

The choice of clustering methods to optimize CL objectives arises naturally from the fact that positive pairs ought to be close to each other in the embedding space while samples of negative pairs ought to repel each other. Intuitively, clusters of similar samples are considered positive samples. Conversely, samples from different clusters are denoted negative samples.

Simple approaches, including DRC Zhong et al. [2020], aim to create low intra-class diversity clusterings by considering both the Assignment Feature (AF), obtained from a Convolutional Neural Network (CNN) with a fully connected layer, and the Assignment Probability (AP), calculated via the softmax function of the AF, during clustering using a dedicated loss function. This approach is motivated by their claim that existing methods cluster dissimilar AF together, due to the usage of the maximum sensitivity of the softmax function used during cluster assignment.

Opposed to methods such as DRC, Swapping Assignments between multiple Views of the same image (SwAV) does not directly encourage similar embeddings but similar cluster assignments for positive pairs Caron et al. [2020]. The idea is to swap the assignments between two positive samples of the same image to encourage similar cluster assignments for similar instances. A positive sample is obtained by applying a random augmentation to the anchor.

Local Aggregation Zhuang et al. [2019] optimizes a low-dimensional feature space mapping by iteratively identifying close neighbours and updating the embedding function. This soft clustering technique is called Local Aggregation (LA).

At each step during training of the embedding function $f_\theta : \mathcal{X} \to \mathcal{Z}$, two sets of neighbours are identified for each datapoint's embedding z_i which are illustrated in Figure 4. The first set C_i contains z_i's close neighbours in the feature space, while the second set B_i contains z_i's background neighbours. B_i is used as a means to judge distance and similarity, while C_i's members should be embedded closer to z_i. In other words, C_i can be considered the set of positive samples while B_i denotes the set of negative samples. The level of LA $L(C_i, B_i|\theta, x_i)$ characterizes the relative level of closeness within C_i compared to B_i. $L(C_i, B_i|\theta, x_i)$ should be maximized.

The set B_i consists of the k nearest neighbours of z_i in terms of cosine distance in the feature space. k is a hyperparameter and Zhuang et al. [2019] set $k = 4096$. In order to construct C_i, first H k-means clusterings are performed with sligthly different conditions. Then, all of z_i's clusters are united to form C_i. H and k are hyperparameters. Zhuang et al. [2019] find that more clusterings, i.e. higher H, lead to isotropic clusters since outliers which arise from random processes are averaged out. If H is too high compared to the number of clusters k, the performance decreases. They state that $H = 3, k = 10000$ and $H = 10, k = 30000$ are better values than $H = 10, k = 10000$ in terms of ResNet-18 nearest neighbour validation performance.

Finally, the level of LA $L(C_i, B_i|\theta, x_i)$ is defined as the negative log-likelihood of the feature space representation z_i of x_i being in C_i given B_i, i.e., being recognized as a close neighbour given being recognized as a background neighbour. The loss to minimize is $\mathcal{L} = L(C_i, B_i|\theta, x_i) + \lambda \|\theta\|^2$. Zhuang et al. [2019] choose to rely on hyperparameter settings from another work rather than conducting a hyperparameter search.

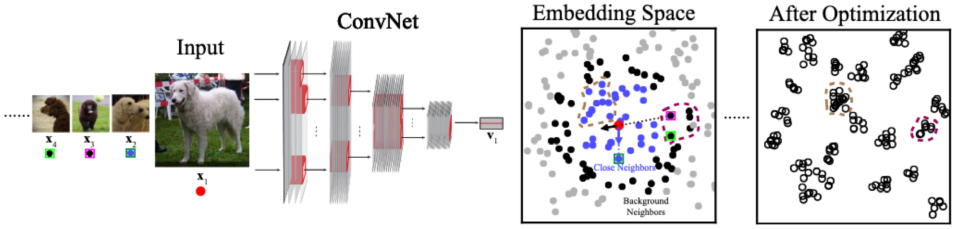


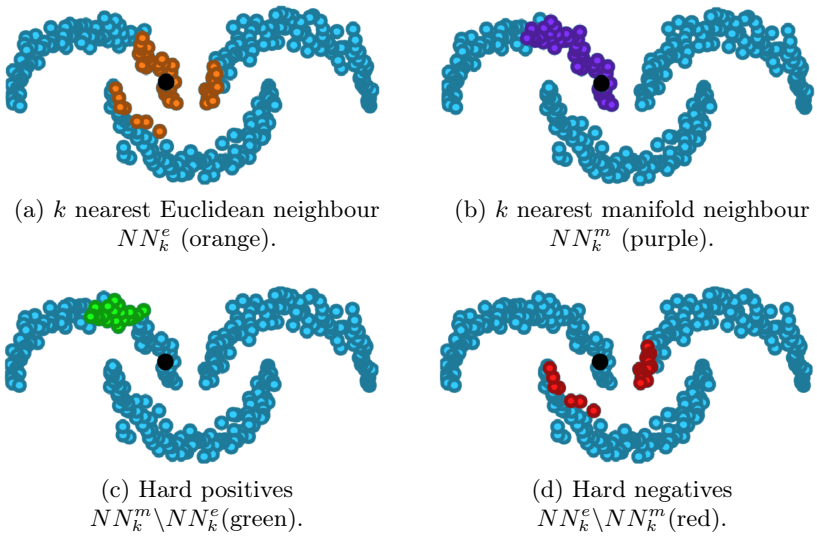
Fig. 4: Illustration from Zhuang et al. [2019]. A CNN produces the feature space embedding $z_i = f_\theta(x_i)$. The embeddings are displayed as points in the feature space. The red point is the anchor z_i, whereas blue points are close neighbours C_i and black points are background neighbours B_i. The arrows denote influences between the neighbours.

Mining on manifolds According to Iscen et al. [2018a], the initial representation of the data is obtained by e.g. a pre-trained CNN. Hard pair mining is performed in order to re-train the network.

For re-training, a combination of different definitions of proximity induces mining for hard positives and negatives as displayed in Figure 5. Given an anchor, the neighbours on the same manifold which are not neighbours in terms of Euclidean proximity are considered hard positive samples. These positive samples should be embedded closer to the anchor in the Euclidean space. Hard negative samples, on the other hand, are neighbours in Euclidean space, but on different manifolds. These samples should be embedded further away from the anchor in the Euclidean space.

The authors define the k nearest Euclidean neighbour NN_k^e and the k nearest manifold neighbour NN_k^m. The hard positives are defined as $NN_k^m \backslash NN_k^e$. The pool NN_k^m is ordered by descending manifold similarity to the anchor to ensure that high-confidence samples are chosen first. k controls the diversity of the hard positives. The larger k is, the more diverse, i.e. hard, the hard positives are. The pool of hard negatives is defined as $NN_k^e \backslash NN_k^m$. NN_k^e is ordered by descending Euclidean distance to the anchor to keep the hardest samples.

Iscen et al. [2018a] propose a method where the manifold is estimated by mode-seeking, i.e. a random walk process, on the Euclidean nearest neighbour graph induced by the Euclidean similarity function s_e. The graph is undirected,



(a) k nearest Euclidean neighbour NN_k^e (orange).

(b) k nearest manifold neighbour NN_k^m (purple).

(c) Hard positives $NN_k^m \backslash NN_k^e$(green).

(d) Hard negatives $NN_k^e \backslash NN_k^m$(red).

Fig. 5: Visualization of different proximity definitions and the hard negatives/positives from Iscen et al. [2018a]. The anchor is the black point.

weighted and represented by a sparse symmetric adjacency matrix. The adjacency matrix consists of the reciprocal k nearest neighbours of each sample. To reduce the negative impact of outliers, only reciprocal nearest neighbours are incorporated Iscen et al. [2017, 2018a,b]. The weighted adjacency matrix entries a_{ij} defined in (1) from Iscen et al. [2018a] are calculated via the Euclidean distance between the samples if both nodes are in each other's nearest neighbourhood. Diagonal entries are set to zero Iscen et al. [2018a,b]. The manifold is computed once at the beginning Iscen et al. [2018a].

$$a_{ij} = \begin{cases} s_e(y_i, y_j), & \text{if } y_i \in NN_k^e(y_j) \wedge y_j \in NN_k^e(y_i) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

While traditional mode-seeking relies on metric features, such as distances, mode-seeking on graphs uses the concept of random walks instead Minsu Cho and Kyoung Mu Lee [2012]. A random walk, i.e. a linear combination of the identity matrix and a scaled version of the adjacency matrix, is simulated multiple times on the graph. Minsu Cho and Kyoung Mu Lee [2012] define the so-called authority modes on a graph as the most frequently visited nodes by random walks among their local neighbours. They correspond to the local maxima of the underlying probability distribution of random walks over the graph. Since the anchors should be diverse and relevant, they are chosen to be the authority modes Iscen et al. [2018a].

Inspired by PageRank, the possibility of random jumps is included in order to ensure convergence to a stationary distribution. The probability of visiting a

node is denoted by the authority score π which is defined in (2) from Minsu Cho and Kyoung Mu Lee [2012] where $p(i,j)$ are the entries of the Markov transition matrix P. $\pi(j)$ takes into account the probability of visiting a node j from a node i and the probability of a random jump from one of the nodes chosen uniformly at random. Minsu Cho and Kyoung Mu Lee [2012] set α to 0.9. The authority score π can be computed by, for instance, the power method Minsu Cho and Kyoung Mu Lee [2012], Langville and Meyer [2004].

$$\pi(j) = \alpha \sum_{i \in \mathcal{V}} \pi(i) p(i,j) + (1-\alpha)\frac{1}{N}, \text{ where } p(i,j) = \frac{a_{ij}}{\sum_{k \in \mathcal{V}} a_{ik}} \tag{2}$$

Minsu Cho and Kyoung Mu Lee [2012] define node relevancy $\Psi(s,t)$ via (3) where $d(s)$ denotes the out-degree of a node. To incorporate the reachability of a node, the probability of reaching node t from node s in k steps $p_k(s,t)$ is defined via the k_{th} power of the Markov transition matrix P. To support similar authority scores between neighbouring nodes, the exponential term including a weighting factor γ is included. $\Psi(s,t)$ is not symmetric and depends on the random walk step k. The authors propose to use the node relevancy $\Psi(s,t)$ for a node s to determine the manifold neighbours $N_\varepsilon^m(s)$ for s via the usage of a threshold ε: $N_\varepsilon^m(s) = \{t \in \mathcal{V} | \Psi(s,t) > \varepsilon\} \cup \{s\}$ Minsu Cho and Kyoung Mu Lee [2012].

$$\Psi(s,t) = d(s) p_k(s,t) \exp(-\gamma \left\{\pi(t) - \pi(s)\right\}^2), \text{ where } d(s) = \sum_{j \in \mathcal{V}} a_{sj} \tag{3}$$

They propose the Authority-Ascent Shift (AAS) as a nonparametric estimator of the authority modes. A node s is shifted to node $\mathcal{A}(s)$ calculated in (4). This formula chooses the local neighbour t of s that maximizes the difference of the authority scores π. The authors argue that the AAS is finite and converges since the graph is finite. When AAS is completed, manifolds are represented as clusters.

$$\mathcal{A}(s) = \underset{t \in \mathcal{N}_\varepsilon(s)}{\operatorname{argmax}} \left\{p_k(s,t)\left[\pi(t) - \pi(s)\right]\right\} \tag{4}$$

The authors propose multiple loss functions to train the model. They, for instance, apply the contrastive loss $l_c(x^r, x^+, x^-) = \|x^r - x^+\|^2 + [m - \|x^r - x^-\|]^2$, the triplet loss $l_t(x^r, x^+, x^-) = \left[m + \|x^r - x^+\|^2 - \|x^r - x^-\|\right]^2$, and weighted versions of both contrastive and triplet loss, where the loss is multiplied by the manifold similarity of anchor and positive sample Iscen et al. [2018a]. m is a margin parameter and x^r is the representation of the anchor.

Resulting hard positives and negatives are displayed in Figure 6 and Figure 7.

Prototypical Contrastive Learning Li et al. [2021] use clustering in the feature space to optimize the sample's representation. They assign multiple prototypes of different granularity to each sample. The prototypes are found by

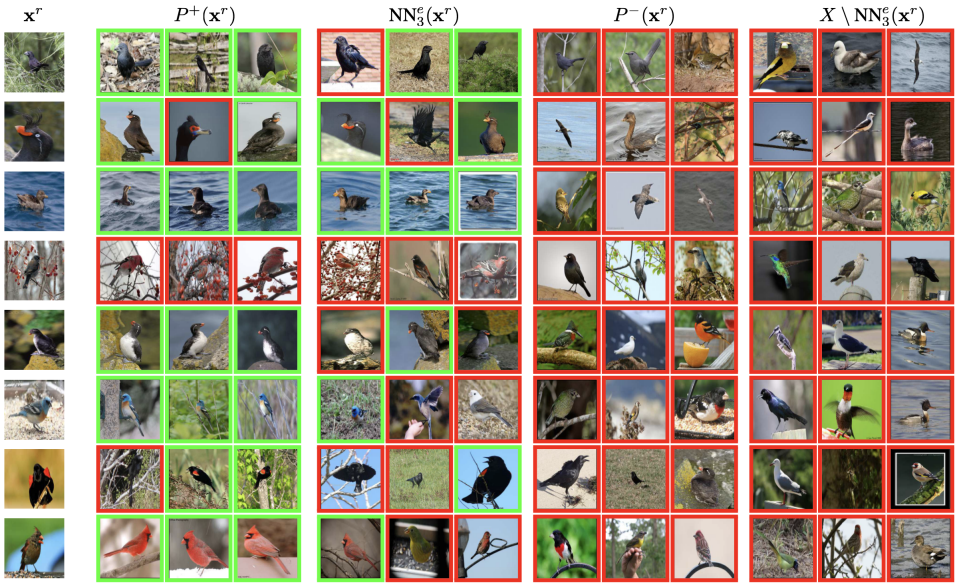


Fig. 6: Illustration of CUB200-2011 from Iscen et al. [2018a]. The anchor is denoted x^r. A selection of hard positives from $P^+(x^r)$ is compared to the baseline approach by sampling from the closest neighbours to x^r in terms of Euclidean distance. Analogously, a selection of hard negatives from $P^-(x^r)$ is compared to the baseline $X \backslash NN_3^e$, i.e. sampling from the set that contains all samples but the three closest ones in terms of Euclidean distance. The borders of the images denote the ground-truth class, i.e. if bird species is the same. Green borders mean that the image belongs to the same class as the anchor, whereas red denotes different class images. It becomes apparent that the sampled hard positives consist of fewer false positives as the baseline.

clustering and they are considered positive samples. The sample's embedding is enforced to be similar to those of its prototypes by a contrastive loss. The so-called prototypical contrastive loss ProtoNCE is optimized using an EM algorithm as displayed in Figure 8. The prototypes are latent variables.

k-means clustering is used to find the prototypes in the E-step. The clustering is performed on the samples' embeddings obtained from the momentum encoder, whose parameters are a moving average of the main encoder's parameters and thus, smoother Li et al. [2021]. The prototypes are the cluster centroids.

The M-step updates the network parameters θ by optimizing the ProtoNCE loss. The minimization of the ProtoNCE loss is equivalent to maximizing the estimated log-likelihood. The optimal parameters are those that map a sample close to its prototypes. The result is obtained under the assumptions of a uniform prior over the cluster centroids, i.e. prototypes, and an isotropic Gaussian distribution of the sample's embeddings around the prototypes.

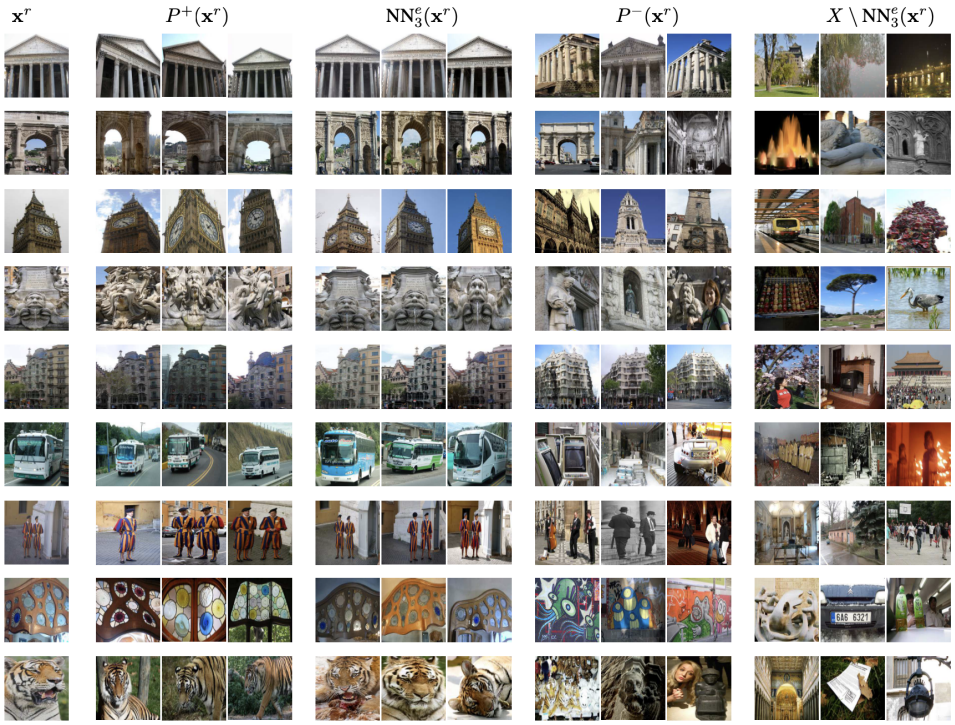


Fig. 7: Illustration Oxford5k and Paris6k images crawled from Flickr from Iscen et al. [2018a]. The dataset contains multiple images for each landmark building, i.e. class Zhao et al. [2019]. The anchor is denoted x^r. A selection of hard positives from $P^+(x^r)$ is compared to the baseline approach by sampling from the closest neighbours to x^r in terms of Euclidean distance. Analogously, a selection of hard negatives from $P^-(x^r)$ is compared to the baseline $X \backslash NN_3^e$, i.e. sampling from the set that contains all samples but the three closest ones in terms of Euclidean distance. It becomes apparent that the hard negatives display visually similar but semantically different images to the anchor.

The ProtoNCE loss from (6) extends the InfoNCE from (5) by not only enforcing similarity between the sample and one positive sample while retaining dissimilarity to r negative samples, but also considering its prototypes c_s^m. To improve the stability of the results, M clusterings with different numbers of clusters k_m are executed. Different numbers of clusters lead to different granularity of the prototypes and thus, a hierarchical structure is encoded into the loss function.

$$\mathcal{L}_{InfoNCE} = -\sum_{i=1}^{N} \log \frac{\exp \frac{z_i \cdot z_i'}{\tau}}{\sum_{j=0}^{r} \exp \frac{z_i \cdot z_j'}{\tau}} \tag{5}$$

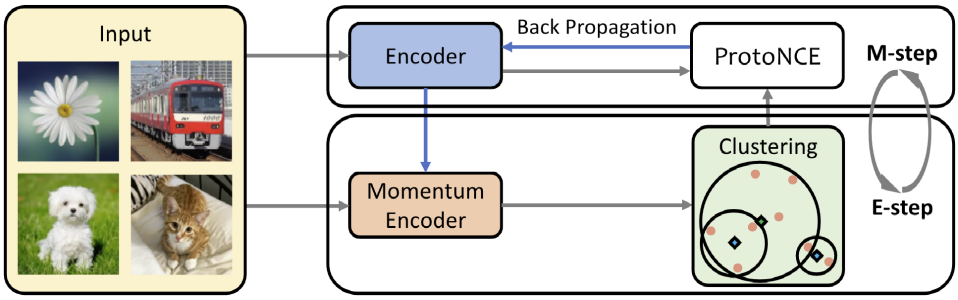


Fig. 8: Illustration from Li et al. [2021]. The training of the PCL algorithm is shown. M k_m-means clusterings based on the feature space defined by the momentum encoder are performed in the E-step. The prototypes $c^m_{\tilde{k}}$, $\tilde{k} \in [1, k_m]$, illustrated as green/ blue rectangles, are the cluster centroids. The M-step updates the network parameters θ by optimizing the ProtoNCE loss.

$$\mathcal{L}_{ProtoNCE} = \mathcal{L}_{InfoNCE} - \sum_{i=1}^{N} \frac{1}{M} \sum_{m=1}^{M} \log \frac{\exp \frac{z_i \cdot c^m_s}{\phi^m_s}}{\sum_{j=0}^{k_m} \exp \frac{z_i \cdot c^m_j}{\phi^m_j}} \tag{6}$$

The parameter τ is called temperature. Wang and Liu [2021] have carried out an exhaustive study on the impact of temperature on the loss function of CL. They found that the contrastive loss function optimizes hard samples by penalizing them according to their hardness. Small temperatures τ penalize very hard negative samples, i.e. similar to the anchor. Hence, the samples' representations are pushed further apart and thus, the embedding space becomes more uniformly distributed Wang and Liu [2021], Hao et al. [2024]. However, when τ is too small, the embedding spaces' semantic structure deteriorates. For $\tau \to \infty$, the loss function's hardness-aware property disappears and hard negatives are not penalized anymore.

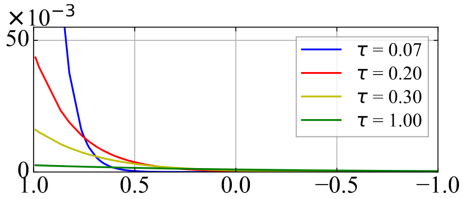
3.3 Mixing of Contrastive Hard negatIves

Kalantidis et al. [2020] claim that most approaches to sampling negatives rely on time-consuming updates of a memory bank or big batches to achieve good performance. They propose Mixing of Contrastive Hard negatIves (MoCHi), a method that computes samples online claiming without any computational overhead.

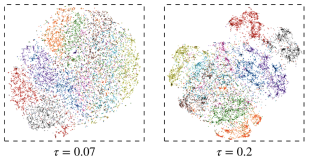
They use a memory bank Q of size $|Q| = K$ to store negative samples. There are different ways to choose K by either saving the whole dataset, a queue of the last batches, or all images in the current batch.

The definition of MoCHi(N, s, s') is as follows:

1. N: Number of negative samples from the memory bank to consider during negative mining.



(a) For small values of τ, similar samples, i.e. x-axis values close to 1.0, receive high penalties. If τ is too small, only the closest one or two points are penalized and thus, the performance deteriorates. As $\tau \to 1$, the magnitude of the penalty is similar for all negative samples.



(b) Visualization of embedding space for different τ. Small values of τ lead to a more uniform distribution of the embedding space, because similar samples are repelled.

Fig. 9: Illustrations from Wang and Liu [2021].

2. s: Number of samples to generate with approach 1.
3. s': Number of samples to generate with approach 2.

For both approaches to generating negative samples, the authors use the same memory bank Q. Given a fixed anchor, firstly, Q is ordered in descending similarity to the anchor representation in the feature space producing $\tilde{Q}$. The similarity between $n_j \in \tilde{Q}$ and the query q is calculated via a scaled dot product of both samples. Secondly, all but the first, i.e. most similar, N samples are discarded from $\tilde{Q}$. The first approach proposed by Kalantidis et al., chooses two random samples $n_i, n_j \in \tilde{Q}$ and a mixing coefficient $\alpha_k \in (0, 1)$, to generate the new sample as a linear combination of the existing hard negative samples as described in (7).

$$h_k = \frac{\tilde{h_k}}{\left\|\tilde{h_k}\right\|}; \tilde{h_k} = \alpha_k n_i + (1 - \alpha_k) n_j \tag{7}$$

The second approach generates new samples by using the anchor representation q and a random sample $n_j \in \tilde{Q}$ to generate the new sample as described in (8). The mixing coefficient $\beta_k \in (0, 0.5)$ is used to balance the influence of the anchor and the hard negative sample. By limiting the influence of the anchor, the authors enforce a stronger influence of the negative sample. This approach is considered to produce even harder negative samples than the first approach.

$$h'_k = \frac{\tilde{h'_k}}{\left\|\tilde{h'_k}\right\|}; \tilde{h'_k} = \beta q + (1 - \beta_k) n_j \tag{8}$$

Using (7) and (8), s and s' hard negative samples are generated respectively. Afterward, the similarity of each generated sample h_k (or h'_k respectively) to the anchor q is calculated via $\frac{q^T h_k}{\tau}$. This similarity is used to update the memory bank $\tilde{Q}$ by inserting the new sample.

There are two loss functions proposed by Kalantidis et al. [2020] to train the model. Firstly, the alignment loss is used to determine the absolute distance between representations with the same class label. Secondly, the uniformity loss is used to determine the distribution of representations on the hyperphere. It is calculated via the logarithm of the average pairwise Gaussian potential between all embeddings.

Kalantidis et al. [2020] conducted experiments to investigate the effect of the parameters s, s'. They present metrics such as average precision and accuracy for different ratios of s and s' for object detection and image segmentation tasks on COCO as well as linear classification on ImageNet-1K and object detection on PASCAL VOC. Generally, the scores differ only in magnitudes of 1%. Additionally, the authors state that for $s > 0, s' = 0$ the model learns faster but achieves similar performance to the baseline, i.e. the model without MoCHi.

The authors listed modifications to MoCHi that lead to inferior performance. They tried to define the mixing coefficient via the similarity of the hard negative samples to the anchor. Moreover, they introduced non-uniformity when sampling n_i, n_j from $\tilde{Q}$ by defining a probability distribution over similarities to the query. Alternatively, they proposed omitting the parameters s, s' and sampling hard negatives using alternately approaches 1 and 2 until $N\%$ of the top samples in $\tilde{Q}$ correspond to synthesized samples.

$$\tilde{h_k} = \alpha_k \cdot n_i + (1 - \alpha_k) \cdot n_j, \text{ where } \alpha_k = \frac{p(n_i \text{ is } TP)}{p(n_i \text{ is } TP) + p(n_j \text{ is } TP)} \tag{9}$$

The researches that introduced ProGCL (section 3.1) in Xia et al. [2022] present an extension of the hard mining strategy MoCHi.They claim, that a high portion of the negative samples generated by MoCHi are FNs. They therefore propose to weigh the selected negative samples for mixing according to their relative probability of being a TN as displayed in (9).

3.4 Curricular weighting

Zhuang et al. [2024] propose a weighting scheme for the negative samples in the context of CL, which emulates human learning. Initially, the model is trained on easy negative samples and gradually prioritizes harder negative samples as displayed in Figure 10. Therefore, the weight of hard negative samples is increased over time according to the model performance. Furthermore, the authors introduce a $\mathcal{L}_2$ regularization term to mitigate the influence of FNs.

Firstly, the embeddings are $\mathcal{L}_2$ normalized. Secondly, the similarity between the anchor and the other samples in the batch is calculated via the dot product. The similarity of the positive pair is denoted $s_{ii'}$ while the similarity of the negative pair is designated s_{ij}. If the similarity of a negative pair is bigger than the similarity of the positive pair, i.e. $s_{ij} > s_{ii'}$, the negative pair is considered a hard negative and the weight of the negative sample is multiplied by a factor w_{ij} as illustrated in (10).

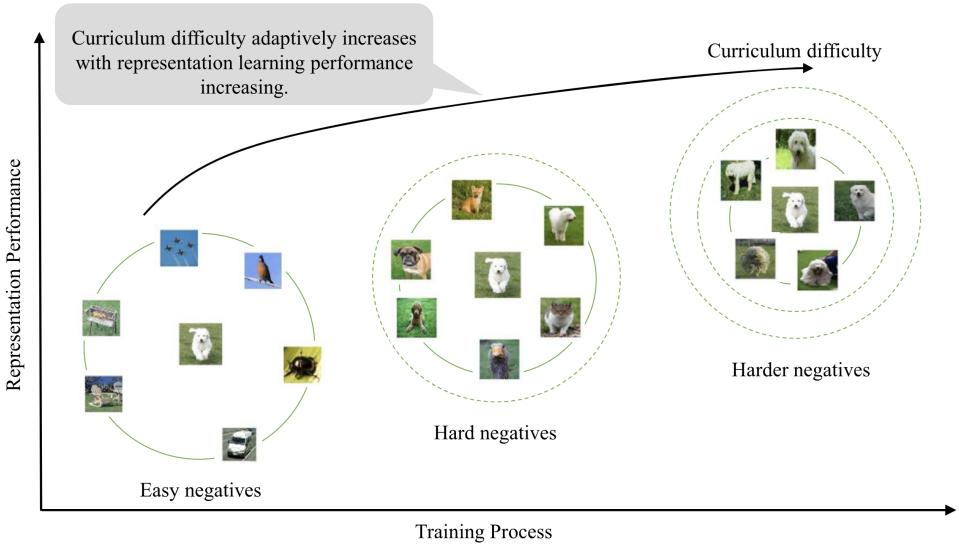


Fig. 10: Selection of negative samples for different stages of the training process from Zhuang et al. [2024]. Initially, the model is trained on easy negative samples. The difficulty of the negative samples is gradually increased over time. Solid circles represent the samples that are prioritized for training.

$$N(i,j,t) = \begin{cases} s_{ij}, & s_{ij} \leq s_{ii'} \\ s_{ij} \cdot w_{ij}, & s_{ij} > s_{ii'} \end{cases}, \text{ where } w_{ij} = t + s_{ij} \tag{10}$$

w_{ij} is governed by the negative pair's similarity s_{ij} emphasising hard negatives and the parameter t which is adapted during training. At the beginning of the training process, t is set close to zero and thus, $w_{ij} < 1$ to assign smaller relative weights to hard negatives. As the training progresses, t is increased to assign higher weights $w_{ij} > 1$ to hard negatives.

$$r_i(j) = \frac{\left|\frac{\partial \mathcal{L}_i}{\partial s_{ij}}\right|}{\left|\frac{\partial \mathcal{L}_i}{\partial s_{ii'}}\right|} \tag{11}$$

By calculating the ratio $r_i(j)$ of the hard negative gradient and positive gradient with respect to the loss for different values of t by (11), the authors show that the model gradually focuses on hard negatives. The resulting plot is depicted in Figure 11.

The parameter t is updated according to (12), where m is a momentum encoder. Intuitively, t is a moving average of the average similarity of the positive pairs which is implicitly aligned with the model's performance. Since this average similarity is expected to increase over time, so will t Zhuang et al. [2024].

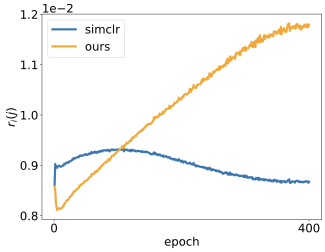


Fig. 11: Display of the gradient ratio of hard negative and positive samples (orange) throughout training from Zhuang et al. [2024]. As the training progresses, the model focuses more on hard negative samples, i.e. the ratio increases.

$$t^{(k)} = m \cdot t^{(k-1)} + (1-m) \cdot \frac{1}{N} \sum_{i=1}^{N} s_{ii'}^{k} \tag{12}$$

Zhuang et al. stress that in order to avoid losing the semantic structure of the embedding space due to highly weighted FNs, weights $w_{ij} > 1$ are $\mathcal{L}_2$ regularized. The comparison of the model's accuracy with and without the proposed regularization approach on both the CIFAR-10 and the STL-10 datasets support the usage of regularization Zhuang et al. [2024].

Moreover, the authors conduct an ablation study to investigate the impact of the proposed weighting scheme by comparing it to fixed small and high t values, as well as fixed transitions of t values during training according to cosine functions. The proposed adaptive weighting scheme outperforms the fixed weighting schemes.

4 Critic

Even though the presented sampling techniques are very promising, after gathering and evaluating the information, some critical points arise which will be discussed in the following.

Many scientists have led their paper with the statement that the generation of positive samples has already been subject to many studies. However, the majority of the papers subject to this work used random transformations to generate positive samples Robinson et al. [2021], Ho and Nvasconcelos [2020], Caron et al. [2020]. The effort spent on initially identifying the best augmentation strategies for positive samples is undeniable, but the thought spent on positive sample generation in these papers seems to be rather shallow.

In terms of explainability, the usage of cluster-based sample generation techniques seems to be a more promising approach than stating that an augmentation sampled randomly from a set of transformations was used without further motivation or detail.

Iscen et al. [2018a]'s manifold mining approach, on the other hand, poses an interesting and more graspable approach to sample generation. However, according to the paper, the manifold is calculated only once at the beginning of the training process. This is possible either due to the assumption that the manifold does not change during the training process or due to computational costs. Since the manifold is calculated based on the Euclidean nearest neighbour graph which is dependent on the embedding, the manifold structure could change during the training process of the embedding function. Therefore, recalculation of the manifold could be of interest.

Some clustering-based approaches such as PCL Li et al. [2021] seem to need M clusterings per iteration which seems computationally expensive. However, since a batch of samples changes the embedding, recalculation of the clusters is necessary. Nevertheless, the computational costs of this approach should be considered when evaluating PCL's overall usability.

Even though most papers use the dot product to calculate the similarity between samples, Dong et al. [2024] and LA Zhuang et al. [2019] use cosine similarity. The cosine similarity is a measure of the angle between two vectors and does not consider the magnitude. Hence, using cosine similarity to calculate the similarity between samples seems questionable.

The curse of dimensionality states that in high-dimensional spaces, distances become meaningless. Consequently, cluster-based approaches in high-dimensional spaces might not be as effective as in lower-dimensional spaces. Since dimensionality reduction always leads to information loss, one has to consider this drawback when using clustering-based approaches such as DRC Zhong et al. [2020], SwAV Caron et al. [2020], PCL Li et al. [2021], LA Zhuang et al. [2019] or mining manifolds Iscen et al. [2018a]. To avoid information loss, i.e. working on the original high-dimensional data, the usage of approaches such as MoCHi Kalantidis et al. [2020] seem beneficial.

The clustering-based approach DRC Zhong et al. [2020] was briefly discussed in subsection 3.2. DRC aims to address the issue of existing methods enforcing the representations of samples and their augmentations to be assigned to the same cluster regardless of their AF. Hence, they consider both the AF and the AP during clustering. Since the AP is calculated via the softmax function of the AF, it is debatable whether the usage of AP in the loss is necessary.

5 Own ideas

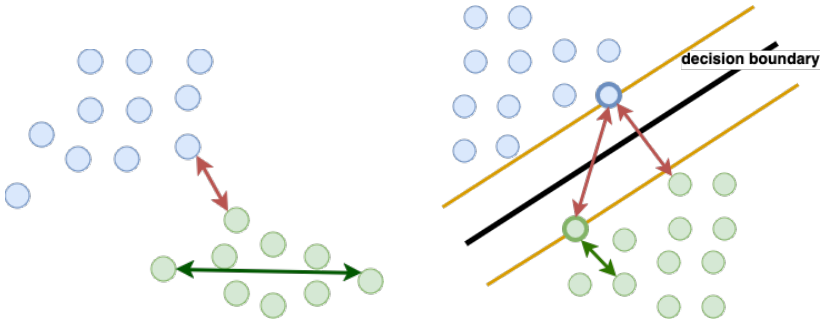
The majority of clustering-based techniques use algorithms such as k-means to generate clusters. k-means' implicit assumptions include that clusters are spherical, isotropic and have roughly the same numbers of samples. To loosen these assumptions, other clustering algorithms could be used. For instance, Density Based Spatial Clustering of Applications with Noise (DBSCAN) could be a promising alternative depending on the underlying data. According to Zhuang et al. [2019], DBSCAN scales well to large datasets. Moreover, it is not necessary to specify the number of clusters k beforehand and it is able to detect clusters of

arbitrary shapes. However, Zhuang et al. claim that DBSCAN performs poorly on high-dimensional data or highly variable density functions.

A combination of different clustering algorithms for methods such as LA Zhuang et al. [2019] or PCL Li et al. [2021] could be beneficial. Since LA uses m different k_m values for k-means clusterings to generate hierarchical clusters of different granularity, this diversity could be increased by using different clustering algorithms to introduce different cluster shapes.

Since most techniques discussed in this work have image input data CNNs seem to be a natural choice for generating embeddings. However, using Autoencoders (AEs) to generate embeddings could be a promising alternative. Since AEs are trained using the reconstruction error of the input and output, no additional data is necessary. This idea is purely speculative and has not been tested yet.

If the data is clustered as displayed in Figure 12a, the sample pairs within a cluster that have the longest shortest path could be considered as hard positive samples. Conversely, sample pairs in different clusters with the shortest shortest path could be considered as hard negative samples.



(a) Longest shortest path pairs within a cluster are considered hard positive pairs (green) while shortest path pairs of different clusters are considered hard negative (red).

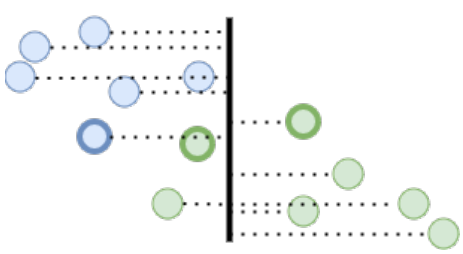
(b) Visualization of the Support Vector Machine (SVM) approach. Support Vectors (SVs) are displayed as bold points. Red arrows denote (hard) negative pairs while green arrows denote (hard) positive pairs.

Fig. 12: Samples are displayed as points and their colour denotes their cluster membership.

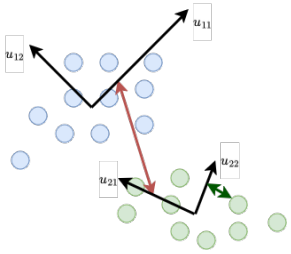
Even if the input data is not initially a graph, one could transform it into a graph via, for instance, the Euclidean nearest neighbour graph discussed in section 3.2. In computer science, Social Network Analysis (SNA) is a well-known field. SNA scientists have developed many measures to analyze graphs. Therefore, another idea is to use the concept of cliques. A clique is a subset of vertices of an undirected graph such that every two distinct vertices in the clique are

adjacent. Vertices in a clique are easy positives and reachable vertices outside the clique are hard positives. This idea would consider the clique vertices to be interchangeable.

Another idea is to use SVMs to generate (hard) samples as visualized in Figure 12b. Initially, the data is split into two classes. Since true class labels are not available in CL, clustering results are used as a proxy. Then, a SVM is trained on the data. After training, the decision boundary is determined such that the margin between the clusters is maximal. The samples that are closest to the decision boundary are considered as SVs. Each pair of SVs from different sides of the decision boundary is considered a hard negative pair. Similarly, distant samples from the decision boundary and their SVs are considered (hard) positive pairs.



(a) Visualization of the Fisher's linear discriminant approach. To facilitate visualization, the projection of the samples is only implied by dotted lines but not displayed explicitly. Overlapping projected samples (bold circles) of different clusters are considered hard negative pairs.



(b) Display of the Principal Component Analysis (PCA) idea. The first two PCA axes of each cluster are displayed as lines. Green arrows denote positive pairs while red arrows denote negative pairs.

Fig. 13: Samples are displayed as points and their colour denotes their cluster membership.

It might be possible to use Fisher's linear discriminant to generate hard negative samples. Fisher's linear discriminant is a linear projection that maximizes the distance between the means of the classes. Again, two classes are determined via clustering. As visualized in Figure 13a, the projected samples of different classes that are closest to each other are considered negative samples.

Since PCA is a linear transformation, that maximizes the variance of projected data, it is the best linear dimension reduction technique in terms of minimization of information loss. Again, multiple classes are determined via clustering. The first n PCA axes are determined for each cluster. Since these axes explain the most variance, equidistant samples lying on these axes could be used as generated samples. Pairs of projections and their generated axes samples are

considered positive samples while pairs of projections and samples from different axes are considered negative samples. This idea is visualized in Figure 13b. An alternative approach could be to use cosine similarity between vectors spanning PCA axes of different clusters rather than the dot product of two samples, because the magnitude of the axes is not of interest but only their direction.

Moreover, the concept of curricular weighting proposed in subsection 3.4 could be extended. The notion of hard negative samples being samples in the batch that are more similar to the anchor than positive samples proposed in Zhuang et al. [2024] is only one version to generate hard negative samples. Using other sample generation techniques discussed in this work in combination with the idea of gradually increasing the level of hardness during training could be beneficial.

The main drawback of the ideas presented in this section is the initial clustering. If this clustering produces poor results, i.e. the cluster's samples have different inherent labels, all proposed ideas will produce FNs and thus, the training process will be negatively affected.

Another issue is working on $k > 2$ clusters when using either the SVM or the Fisher's linear discriminant approach. Both techniques are originally designed for binary classification problems. In order to extend the ideas to multi-class problems, one-vs-one or one-vs-all approaches have to be used. These extensions are prone to ambiguities during inference and come with additional computational effort.

Usage of generative AI and AI-assisted technologies

I hereby declare that I used ChatGPT and Grammarly for rewording and spelling correction purposes. These tools assisted me in enhancing the clarity and accuracy of the text.

Bibliography

Yue Cao, Zhenda Xie, Bin Liu, Yutong Lin, Zheng Zhang, and Han Hu. Parametric Instance Classification for Unsupervised Visual Feature learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 15614–15624. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/b427426b8acd2c2e53827970f2c2f526-Paper.pdf.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33: 9912–9924, 2020.

Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6002–6012, 2019.

Zhezheng Hao, Haonan Xin, Long Wei, Liaoyuan Tang, Rong Wang, and Feiping Nie. Towards Expansive and Adaptive Hard Negative Mining: Graph Contrastive Learning via Subspace Preserving. In *Proceedings of the ACM on Web Conference 2024*, WWW '24, pages 322–333, New York, NY, USA, May 2024. Association for Computing Machinery. ISBN 9798400701719. https://doi.org/10.1145/3589334.3645327. URL https://doi.org/10.1145/3589334.3645327.

Feng Wang and Huaping Liu. Understanding the Behaviour of Contrastive Loss. pages 2495–2504, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Wang_Understanding_the_Behaviour_of_Contrastive_Loss_CVPR_2021_paper.html.

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive Learning with Hard Negative Samples, January 2021. URL http://arxiv.org/abs/2010.04592. arXiv:2010.04592.

Chih-Hui Ho and Nuno Nvasconcelos. Contrastive Learning with Adversarial Examples. In *Advances in Neural Information Processing Systems*, volume 33, pages 17081–17093. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/c68c9c8258ea7d85472dd6fd0015f047-Paper.pdf.

Jin Zhuang, Xiao-Yuan Jing, and Xiaodong Jia. Mining negative samples on contrastive learning via curricular weighting strategy. *Information Sciences*, 668:120534, May 2024. ISSN 0020-0255. https://doi.org/10.1016/j.ins.2024.120534. URL https://www.sciencedirect.com/science/article/pii/S002002552400447X.

Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased Contrastive Learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 8765–8775. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/63c3ddcc7b23daa1e42dc41f9a44a873-Paper.pdf.

Hengkui Dong, Xianzhong Long, and Yun Li. Rethinking samples selection for contrastive learning: Mining of potential samples. *Knowledge-Based Systems*, 299:111979, September 2024. ISSN 0950-7051. https://doi.org/10.1016/j.knosys.2024.111979. URL https://www.sciencedirect.com/science/article/pii/S0950705124006130.

Jun Xia, Lirong Wu, Ge Wang, Jintao Chen, and Stan Z. Li. ProGCL: Rethinking Hard Negative Mining in Graph Contrastive Learning, June 2022. URL http://arxiv.org/abs/2110.02027. arXiv:2110.02027.

Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard Negative Mixing for Contrastive Learning, December 2020. URL http://arxiv.org/abs/2010.01028. arXiv:2010.01028.

Huasong Zhong, Chong Chen, Zhongming Jin, and Xian-Sheng Hua. Deep robust clustering by contrastive learning. *arXiv preprint arXiv:2008.03030*, 2020.

Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Mining on manifolds: Metric learning without labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7642–7651, 2018a.

Junnan Li, Pan Zhou, Caiming Xiong, and Steven C. H. Hoi. Prototypical Contrastive Learning of Unsupervised Representations, March 2021. URL http://arxiv.org/abs/2005.04966. arXiv:2005.04966.

Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, Teddy Furon, and Ondrej Chum. Efficient Diffusion on Region Manifolds: Recovering Small Objects With Compact CNN Representations. pages 2077–2086, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Iscen_Efficient_Diffusion_on_CVPR_2017_paper.html.

Ahmet Iscen, Yannis Avrithis, Giorgos Tolias, Teddy Furon, and Ondřej Chum. Fast Spectral Ranking for Similarity Search. pages 7632–7641, 2018b. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Iscen_Fast_Spectral_Ranking_CVPR_2018_paper.html.

Minsu Cho and Kyoung Mu Lee. Mode-seeking on graphs via random walks. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 606–613. IEEE, June 2012. ISBN 978-1-4673-1228-8 978-1-4673-1226-4 978-1-4673-1227-1. https://doi.org/10.1109/CVPR.2012.6247727. URL http://ieeexplore.ieee.org/document/6247727/.

Amy Langville and Carl Meyer. Deeper Inside PageRank. *Internet Mathematics*, 1(3):335–380, January 2004. ISSN 1542-7951. https://doi.org/10.1080/15427951.2004.10129091. URL http://www.internetmathematicsjournal.com/article/1388.

Guoping Zhao, Mingyu Zhang, Jiajun Liu, and Ji-Rong Wen. Unsupervised adversarial attacks on deep feature-based retrieval with GAN. *CoRR*, abs/1907.05793, 2019. URL http://arxiv.org/abs/1907.05793.

General acronyms

AE Autoencoder
AF Assignment Feature
AP Assignment Probability
BMM beta mixture model
CL Contrastive Learning
CNN Convolutional Neural Network
DBSCAN Density Based Spatial Clustering of Applications with Noise
EM Expectation-Maximization
FN False Negative
GCL Graph Contrastive Learning
NN Neural Network
PCA Principal Component Analysis
PU Positive-Unlabeled
SNA Social Network Analysis
SSL Self-Supervised Learning
SVM Support Vector Machine
SV Support Vector
TN True Negative

CL approach-specific acronyms

AAS Authority-Ascent Shift
DRC Deep Robust Clustering
LA Local Aggregation
MoCHi Mixing of Contrastive Hard negatIves
PCL Prototypical Contrastive Learning
SwAV Swapping Assignments between multiple Views of the same image