Imperial College
London

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

Self-supervision and contrastive learning

Author:
Bohua Peng (CID: 01830818)

Date: May 4, 2021

Contents

Abstract

Self-supervised learning has gained much attention recently because it does not require extra manual annotations. In this paper, we attempt to have a thorough study on literature reviews of three types of the most popular self-supervised learning over these years. The paper would in turn introduce hand-crafted pretext training, generative methods, and contrastive learning methods. In particular, we will focus on modern contrastive learning methods that combine metric learning, clustering and mutual information maximization into a unified framework. In these methods, useful embeddings can be learned by drawing closer two different augmented embeddings of the same sample, while repelling embeddings of different samples. Then we will discuss some successful inductive biases introduced by recent methods and a probabilistic interpretation of InfoNCE loss. Next, we will discuss our experiments results of contrastive learning methods trained on CIFAR10 and an additional pathological classification experiment. Finally, we will discuss some promising future directions.

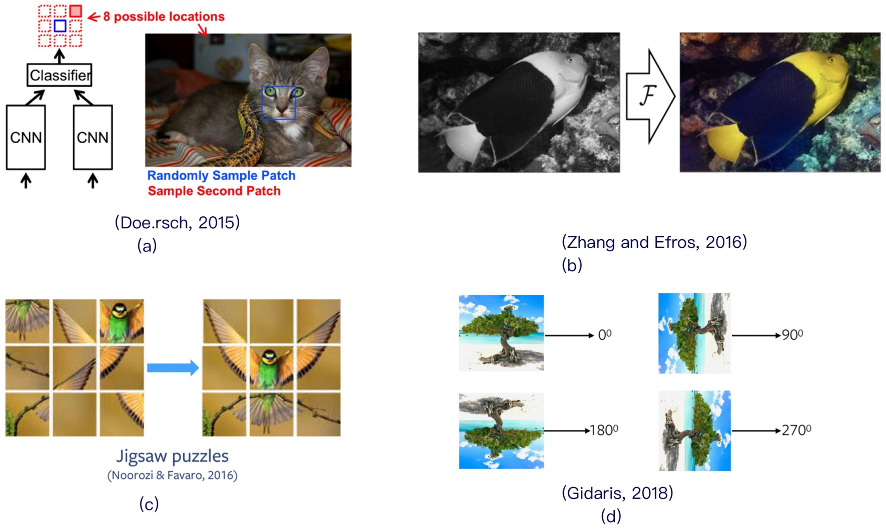
1 Introduction

The recent success of supervised discriminative models relies heavily on manually labelled data. With the help of the internet, a large amount of available data such as CIFAR10 and ImageNet boost these methodologies. However, data annotated by specialists can be hard to acquire for certain applications and mislabelled data can be poisonous to machine learning models. These problems push researchers to think out alternative approaches to leverage a large amount of unlabelled data.

Representation learning, originally applied in natural language processing, aims to extract useful features from the input domain to improve the performance of unknown downstream tasks. From information bottleneck theory, a good representation should capture large enough information for downstream tasks inputs(Hjelm et al., 2019; Linsker, 1988). This principal, known as InfoMax, builds the backbone for most of the representation learning methods. Meanwhile, since the input data often reside in a high dimensional space, to achieve aforementioned goal, representation learning methods often use dimensionality reduction techniques such as kernel function or deep learning to compress the inputs into lower-dimensional latent spaces. Due to structural contraints in these mappings, information loss is inevitable. However, representation learning remains a difficult challenge because the mutual information is intractable in the high dimensional space and the downstream tasks are uncertain. In this report, we opt to discuss a recently surging genre called self-supervised learning.

Self-supervised representation learning can be categorised as a subset of unsupervised learning where self-defined signals are adopted as supervision and transformation-invariant representation is learnedDoersch et al. (2016). In the self-supervised framework, researchers hypothesize that deep learning models learn some general knowledge (representation) by solving self-supervised tasks, also known as pretext tasks. Performing self-supervised learning requires good pretexts and task-specific losses. These necessary ingredients distinguish different methods from each other.

2 Handcrafted pretext training



(Doe.rsch, 2015)
(a)

(Zhang and Efros, 2016)
(b)

Jigsaw puzzles
(Noorozi & Favaro, 2016)
(c)

(Gidaris, 2018)
(d)

Figure 1: Handcraft pretext:(a) Predicting relative position between patches (b) Coloration (c) Predicting rotations (d) Jigsaw puzzle

In the early stage of exploration, some handcrafted pretext tasks are proposed for representation learning. Effective methods include predicting relative positions between patches(Doersch et al., 2016), predicting rotation(Gidaris et al., 2018), colourization, jigsaw puzzle(Noroozi and Favaro, 2017), superresolution, and shuffle & learn. A common idea for these methods is that these pretext tasks should derive pseudo-labels automatically and use them as supervised signals for training. For example, the jigsaw puzzle records the positions of patches after shuffling and forces the network to perform a classification task by predicting the permutation of patches. In general, these pretext tasks should be made difficult enough to encourage the model to learn meaningful representations that generalize to downstream tasks. A universal encoder, e.g., ResNet50, can be used to train on one of these tasks or a sequence of them hoping to capture some transformation invariant features. Yet, these heuristics are quite fragile and require domain-specific knowledge to really contribute to downstream tasks. These handcrafted pretraining methods are soon outperformed by their contrastive learning rivals.

3 Autoencoder based methods

Another line of research resides in generative approaches and their common pretext task is modelling the input distributions with reconstruction. They maximize the mutual information either with autoencoding or adversarial training. Although mutual information is often intractable in high dimensional settings, variational inference approaches this by separating the tractable KL divergence terms out and maximizing the evidence lower bound. The vanilla variational inference learns inefficiently because each data point has a set of variational parameters. Amortised variational inference (Kingma and Welling, 2019), also well known as VAE, solves this with a weight sharing network for all data points. A bottleneck of these methods is the amortisation gap often results in underfitting and blurry reconstructions.

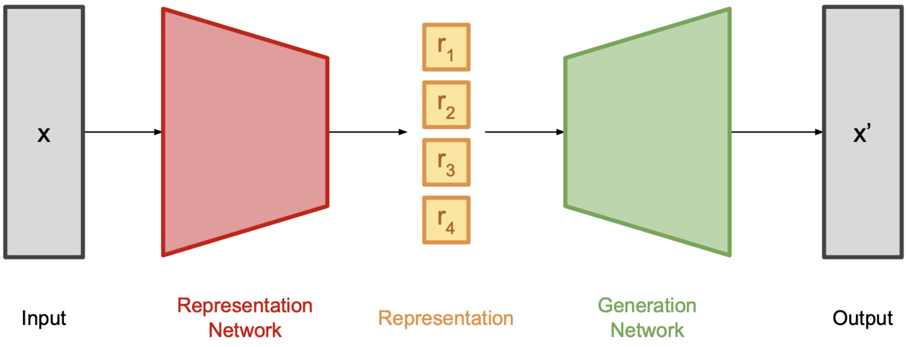


Figure 2: Disentangled VAE model

Another benefit of reconstruction pre-training with VAE is disentanglement of representations, shown in Figure 2. This is because we pose a fully factorised prior on the latent codes. This allows the approximated posteriors (the representation distribution) to be a factorised Gaussian as well. Therefore the representation distribution is designed to be smooth, disentangled and spatially coherent. Interpolations on these smooth disentangled latent spaces gives us controllable generative samples. Further regularization on these latent codes allows the embedding space to learn high-level semantics from co-occurent multimodalities. It also worth mentioning that these methods often need a trade off between generality and controllability.

Although the interpretability of VAE is fascinating, the fidelity of generative samples is often capped by posterior collapse. One the other hand, the generator and discriminator of GAN learn the data distribution through playing a minmax game. The fidelity of samples generated by large scale GAN (Brock et al., 2019) is of the top level these days against other generative methods. Both these two methods try to learn representations through modelling the generative process of data. However, it is often hard to model the distribution of data from a high dimensional space, e.g., large images. And measuring the quality of representations by Inception Score or interpretability might not be the only way to go. A general way to learn useful representation needs to be found.

4 Contrastive learning

The most straight forward intuition of contrastive learning is to encourage the representations that are semantically similar to be closer while pushing away the representations of diverse samples. The key is learning by comparing. In the next few subsections, we will first introduce the different architectures and their connection with metric learning. Particularly, we will discuss the Siamese network, similarity functions, and modern contrastive learning methods. Then we will analyse loss functions and the probabilistic interpretation of InfoNCE loss. Finally, we will discuss hard negative sampling and other future directions.

4.1 Architecture

Contrastive learning has a strong connection with deep metric learning in terms of model architectures and similarity measurement. Metric learning aims to learn an optimal kernel (score) function with the binary labels of inputs pairs. A positive pair contains two different images with the same object, and the method belongs to supervised learning. For modern unsupervised contrastive learning, as shown in Figure 3, a positive pair contains two augmented versions (views) of the same image and the goal is to extract good representations through comparing. Hence, we need architectures dedicating for comparison and a score function.

A popular architecture designed for verification is the Siamese network (Chopra et al., 2005). It is composed of two twin encoders and their outputs are jointly trained on top with a similarity function to learn the relationship between a pair of inputs. The training process is as follows. First, the twin encoders take in two input images simultaneously and produce two representations. Then a score reflecting the similarity of these representations is computed with a distance function (either predefined or learned). Next, a contrastive loss compares the distance with a margin. And if the distance of a positive pair is outside the margin, the twin encoders will be updated to pull the representations of these two images towards each other. The repelling process for the negative pair works similarly when the distance is smaller than a margin. Here, the training labels only indicate whether two images are a positive pair or a negative pair. That is, we only need weak supervised signal that tells us if two inputs are semantically the same or not. Original Siamese network share weights between the two encoders, but recent modifications show this is mainly a design choice(Chen and He, 2020).

Through comparing, the Siamese network learns a score function, and the twin encoders learn embedding spaces that can be used for downstream tasks. Since neural network is very flexible, this architecture can theoretically model any kernel function(Widjaja, 2003). We can perform classification with the k nearest neighbours algorithm with additional class labels from a support set. Siamese network is therefore widely applied in verification tasks and few shots learning. For instance, CNN encoders can be used for signature verification, face verification, and pedestrian reidentification. LSTM encoders can be trained in this way to improve speech verification (Chopra et al., 2005) and automatic speech recognition (van den Oord et al., 2018).

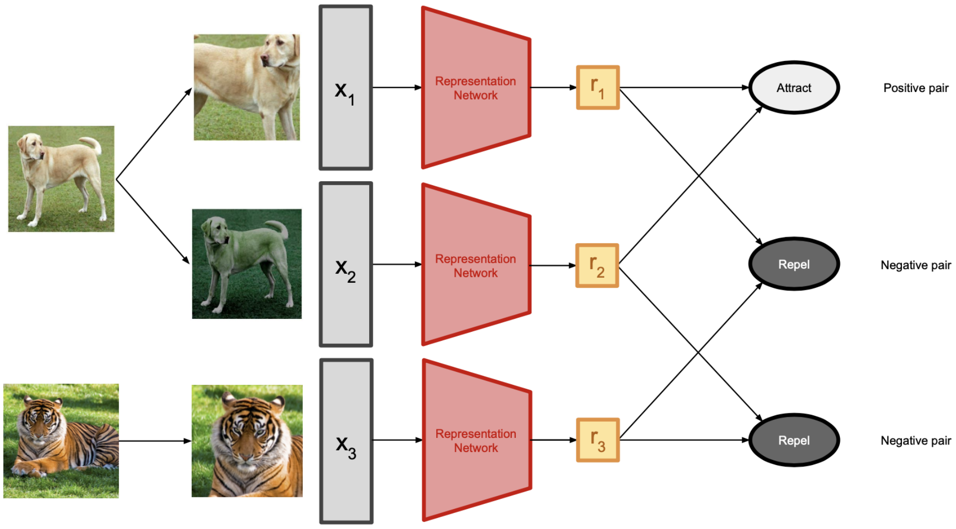


Figure 3: Triplet contrastive learning framework

A common issue of these early Siamese network-based methods is that reducing the distance of a positive pair will not automatically increase the distance of negative pairs, leaving the training inefficient. As shown in Figure 3, one solution is to train a triplet network () with a set of triplets. Each triplet includes an anchor, a positive sample and a negative sample. Here, the distance between the anchor and the positive sample and the distance between the anchor and the negative sample form a normalizer (denominator in the distance function). Given a triplet, the distance of the negative pair now increases simultaneously as the distance of the positive pair increases. This boost optimization during training.

4.1.1 Similarity functions

Another essential component for both contrastive learning and metric learning is the metric function which criticizes the similarity or dissimilarity of two representations. Here, metric functions are also referred to as distance, score function or critic functions in different works. To measure the distance between two embeddings, early contrastive learning methods adopt energy-based metrics such as Euclidean or Manhattan distance. Recently, the inner-product similarity family gains their popularity after the introduction of attention mechanism, e.g., bilinear model $q^T W k$, separable model $\phi(q)^T \phi(k)$ and cosine similarity (here are typically MLPs). Particularly, cosine similarity, also known as cosine embedding loss, reduces the magnitude discrepancy and encourages the model to focus on angular distance. That is, embeddings are normalized onto a super sphere and rotated to be as aligned as possible for positive pairs, and repelled to be orthogonal for negative pairs. This similarity can be computed swiftly after layer normalization.

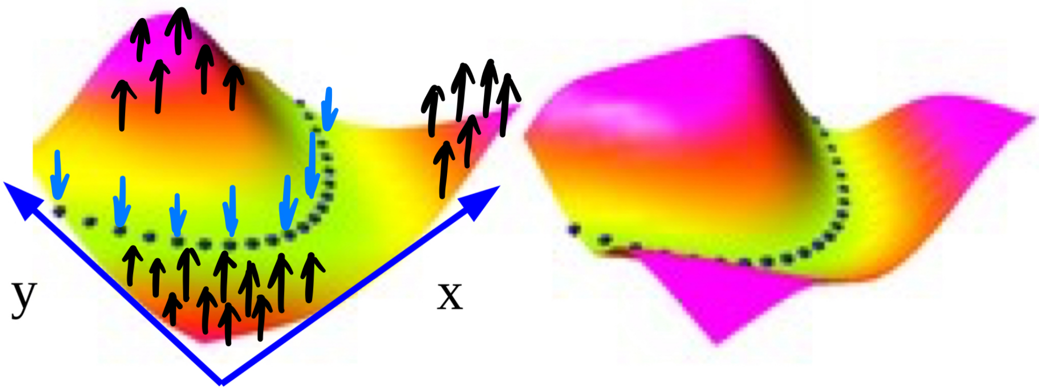


Figure 4: A illustration of energy surface where blue dots represent positive sample and black arrows represent negative samples

The intermediate network outputs give us representations. Energy manifold is one of the most popular theories that models these representations. According to energy manifold theories, the positive and negative samples will help our model learn a low dimensional manifold where positive samples are in the manifold with low energy, whereas negatives samples are outside the manifold with high energy (LeCun et al., 2006). To enforce the smoothness of the manifold, we need a large number of negative samples to push up the energy surface around the data manifold as shown in Figure4.

However, a bottleneck in implementing this idea is how to compute the representations of a large number of negative samples efficiently. Furthermore, another problem appears when applying the Siamese network to unsupervised representation learning. Although we expect the twin networks to extract abundant transformation-invariant representations from two augmented views, they often have the mode collapse issue where trivial constant representations are returned. Recent contrastive learning methods proposed some intriguing solutions to these problems. As shown in Figure, we categorize recent approaches into online methods and memory bank-based methods. Each category will be explained below.

4.1.2 Memory bank-based method

Rather than discarding the representations after computation, (Wu et al., 2018) adopts a memory bank to store them. This is similar to the experience replay buffer widely used in reinforcement learning where the agent learns efficiently from experience sampled from a buffer. During training, computed representations are uniformly sampled as negative samples before computing the contrastive loss with the representation of the current image. However, this brings up an issue of feature inconsistency. That is, representations only get updated when last sampled and some outdated representations computed long ago harm the training. PIRL alleviates this issue with a moving average over representations. Another contribution of PIRL is to prove jigsaw puzzle pretraining (strong data augmentation) is effective for extracting invariant features.

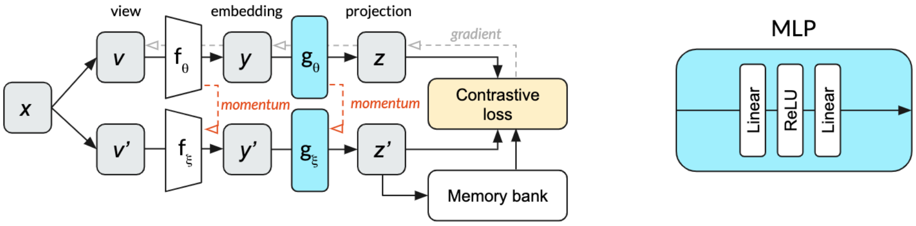


Figure 5: A memory bank based contrastive learning framework (MoCo)

MoCo replaces the memory bank with a queue where newly computed representations are enqueued and oldest representations are automatically dequeued. Since it is hard to maintain keys and values in a queue, sampling is removed and the inner product between the whole queue and the current representation is computed on the fly. Another popular design is the momentum encoder whose parameters gets updated by the parameters of the query encoder in a moving-averaged way. It is recently reported that MoCo needs a large queue size to get good performance.

4.1.3 Online methods

With comprehensive empirical studies, SimCLR introduces a simple online contrastive learning framework and for the first time achieves comparable performance with its supervised counterparts. One major finding is that combinations of augmentations are critical for extracting invariant features from different augmentations of the same images. Although contrastive learning with single augmentation encompasses many pretexts implicitly, a combination of transformations reduces redundant information of two views and rules out trivial solutions, e.g., image entropy which result in constant representations. Another contribution is to add a multiple-layer perceptron as a nonlinear projection head between the contrastive loss and the representations. This nonlinear projection head also helps the model to identify the transformation-invariant representations.

On top of that, abundant empirical studies show that properly scaling up cosine similarity with temperature improves performance on downstream tasks. Besides, a symmetric contrastive loss, NT-Xent, is introduced to avoid mode collapse. As shown in Figure 7, for a positive pair, when the positions of the query and the key are

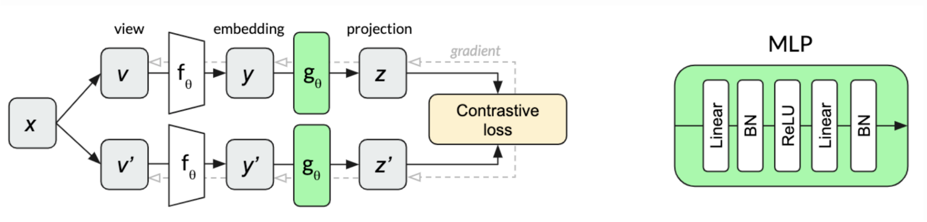


Figure 6: A illustration of SimCLR framework

swapped, the network should predict a comparable similarity score with the original setup. Combining these two similarity scores provides a better spatial layout for the positive and negative pairs on the hyperplane. This technique is widely adopted in later works. Finally, SimCLR discards the memory bank and compares an image with the rest of the images in its batch. This is essential for models that learn from a server when the amount of data is huge and most images are only learned once. Another popular work, Contrastive Predictive Coding (CPC) also falls into this category. Although their pretraining is to predict if next frame audio is a positive sample, such contrastive learning framework can be generalized to images and videos and sentences.

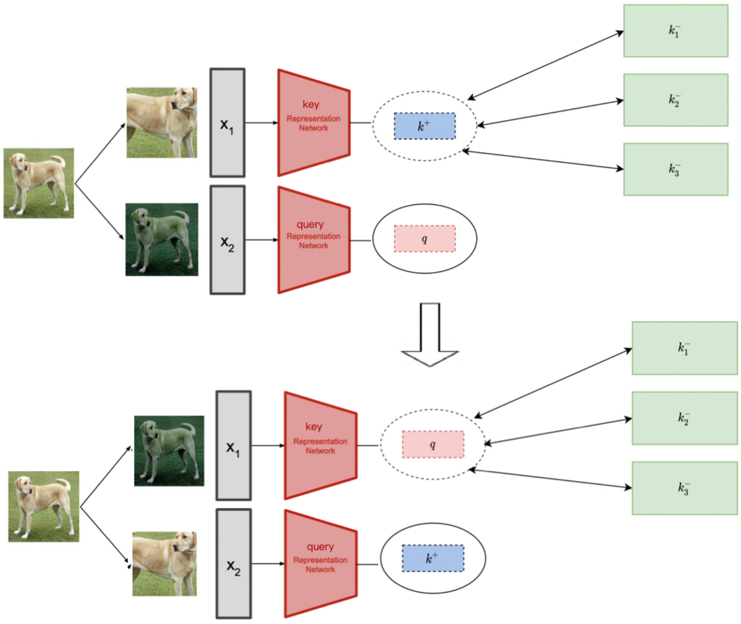

k_1^-
k_2^-
k_3^-
x_1
x_2
key Representation Network
query Representation Network
k^+
q
k_1^-
k_2^-
k_3^-
x_1
x_2
key Representation Network
query Representation Network
q
k^+

Figure 7: Swapping to compute a contrastive loss

A problem for online contrastive learning is that negative samples in the same batch only represent a subset of negative samples. This may probably be the reason for the collapsing representations frequently reported in the online methods. A direct solution is to increase batch size. The batch size in the official configuration of SimCLR is 4096, which is impossible to train on a single GPU. Therefore, the model is trained with multiple GPUs in parallel or TPUs. The commonly used SGD optimizer is replaced with LARS to tackle large mini-batch optimization where the learning rate scaling is proportional to the square root of the batch size. It is reported that the Batch Normalization module will leak information to replicated models deployed on other GPUs, so MoCov2 introduces a Shuffle Batch Normalization module which computes mini-batch information separately on a single GPU.

4.1.4 Clustering-based methods

All these aforementioned contrastive learning methods can be categorised as instance discrimination methods and their commonly shared drawback is the representations are not encouraged to capture global structural information. This issue arises because samples are forced to be negative pairs as long as they are from different instances. This assumption ignores semantic connections between different instances. Specifically, two images from the same batch may contain objects from the same class or even the same object in extreme cases. Pushing these instances apart in the latent space is detrimental for learning higher semantic information from data. To address this problem, clustering is combined with instance discrimination which allows samples to be compared on a cluster level. Similar to instance-wise methods, these clustering-based methods also have memory bank based versions and online version.

Prototypical contrastive learning (PCL) assumes each data point is assigned to a cluster centroid (prototype) in the latent space. The prototypical network is updated with Expectation-Maximization (EM) algorithm. In the E step, k cluster centroids are estimated by the GPU-based K-means algorithm(Johnson et al., 2017). In the M step, the code of each sample is compared with the codes of all the cluster centroids; the contrastive cross-entropy (NCE) is computed with assignments and similarity scores; the parameters of the score function are updated through backpropagation. To prevent representations from assigning to a single cluster, the temperature is replaced with cluster concentration which explicitly downscales the similarity scores of samples within a loose cluster and pulls the representations towards the cluster centroids. However, updating the cluster representations with k-means requires recomputing all representations of the entire training set after every epoch. Besides, the algorithm is trained offline because it takes a few epochs for the k-means algorithm to converge from randomness(Johnson et al., 2017).

To make clustering-based contrastive learning online, swapping assignment between multiple views (SwAV) introduces a quantisation head that directly predicts cluster embeddings from the embeddings of the current batch. First, SwAV considers a soft clustering assignment problem that shares the same closed-form solution as a constrained optimal transport problem. Under a batch learning setting, each data point has a soft label that is updated with a formula supported by current embeddings using the Sinkhorn-Knopp algorithm(Cuturi, 2013). To address the mode collapse issue, SwAV aggregates strong data augmentation and swapped prediction. Specifically, the representation of one augmented view is compared with the soft label

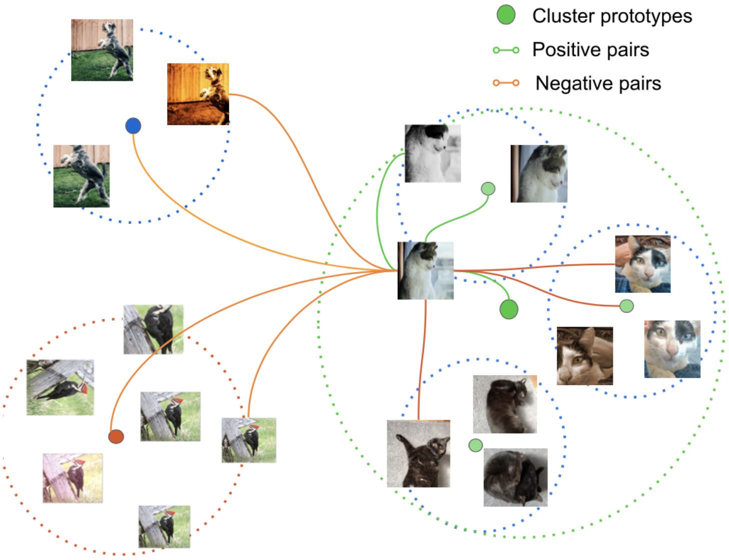


Figure 8: Clustering-based methods introduce cluster-level comparision to contrastive loss and capture fine-grained semantics(Kotar et al., 2021)

of another augmented view. Another online method called InterCLR also uses a swapped prediction but more focus on aggregating positive groups and negative groups, i.e., samples with the same pseudo-labels are considered as positive pairs and vice versa. Like SimCLR, SwAV needs a large batch size to predict cluster embeddings and compares in a batch-wise fashion. When the batch size is smaller than the designed size of prototypes, the prototypical embeddings are kept in a buffer until enough, which makes the model trained like RNNs.

Clustering-based methods improve the performance of representations on classification because they impose proper inductive bias with quantisation heads and pseudo-labels. A common point shared by these methods and semi-supervised leaning is that data points have global aggregated structures and decision boundaries should not pass through high-density manifolds. Unlike semi-supervised pseudo labels methods, the number of prototypes can be larger than the number of real classes. Therefore, prototypes can be learned to capture fine-grained semantic features. For example, as shown in Figure 8, the separate prototypes in the cat class represent features of their separate manifolds, namely black cats, sleepy cats and cats with a big nose as vectors.

Interestingly, there is a natural clustering phenomenon for representations learned by instance-level discrimination methods, manifested by their accuracy on downstream classification task performed by k nearest neighbours and latent code visualisations. Assuming prototypes follow isotropic Gaussian distributions, clustering-based methods take one step further: images within the same distribution but not within the same prototype should have a larger distance than images from the same prototype. This is often true for Inception image crops of ImageNet. As a result, their learned representations show stronger clustering in the latent space.

4.2 Loss functions

Early works use energy-based contrastive loss for comparison. Modern contrastive learning methods apply probabilistic contrastive loss which distincts themselves from previous methods. We will analyse how NCE loss adopt mutual information maximization principal and why comparison is crucially important. Furthermore, we will discuss how NCE based methods are linked with metric learning.

Recently, several self-supervised learning approaches have reached state-of-the art performance in the representation learning with InfoMax principle. A good method extracts invariant representation g(x) that maximizes the mutual information I(X; g(x)) from different transformation of the same scene under some structural constraints. However, the mutual information is notoriously hard to optimize in high-dimensional spaces, and in practice we use a surrogate function to lower bound it.

Aaron van den Oord (van den Oord et al., 2018) first proved that the equation can be optimized by maximizing InfoNCE loss. To get deeper understanding of InfoNCE, we elaborate how it evolve from Noise Contrastive Estimation (NCE) and its different formats. For simplicity, we use the same notations as in MoCo, representing two different views of the same image P(x) as query (q) and positive key (k^+) and potential views from other images $P_n(x)$ as negative key (k^-). The NCE function that maximizes the log likelihood of $p(k^+|q)$–$p(k^-|q)$ was originally proposed for instance-level discrimination between real data and generated noise. With the idea of learning by comparison, the loglikelihood of $P(y=1|q,k^+)$ of the query (q) and its positive key (k^+) , is maximized while the loglikelihood of $P(y=0|q,k^-)$ is minimized through sampling. It has been shown that InfoNCE approximates this maximum log likelihood estimation and the optimal solution is equivalent to maximizing mutual information $I(q,k^+)$. Hence, by modelling the probability $P(y=1|q,k^+)$ directly with normalized similarity, recent InfoNCE based contrastive learning is a discriminative method.

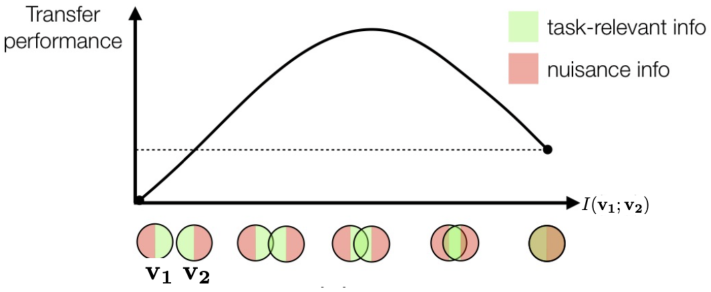


Figure 9: As the redundant information increases, transfer performance drops(Tian et al., 2020)

Following the InfoMax principal, several modifications such as NT-Xent, ProtoNCE made a progress on the LSVRC benchmark. However, although InfoMax provides principled guidance for contrastive learning, recent studies **??** show that maximising the mutual information of positive pairs alone does not guarantee good representation learning. In this work, a invertible neural network (RealNVP) is used for contrastive learning. Ideally, invertible neural networks such as RealNVP and Glow provide smooth and invertible mappings for distributions and therefore mutual information is tractable in their experiments. Their experiments show a tight mutual information lower bound does not guarantee good downstream performance. Another recent work (Tian et al., 2020) finds a parabola shaped relationship between mutual information $I(v1, v2)$ and transfer performance. The interesting conjecture is that good views should maximize relevant information for downstream tasks while reducing as much irrelevant information as possible, shown in Figure 9. This work pushes us to think what defines similarity. For instance, if we feed our model with orange cats crops continuously without random color jittering or grayscaling, the extracted representations are very likely to have feline features code and orange color code mixed up. For a downstream cat or dog classification, our model will probably assume cats have to be orange and thus performing poorly.

Decoupling task-relevant information from irrelevant may sound frightening especially when downstream tasks are assumed unknown. Recently, however, Barlow Twins also justified redundancy reduction via signal decorrelation (Zbontar et al., 2021). Barlow Twins directly models the covariance of representations and penalize the off-diagonal terms. Furthermore, this method does use negative pairs for comparison directly and there fore belongs to general self-supervised learning.

Recent success of contrastive learning cannot be attributed to mutual information maximization alone, and we emphasizes the importance of data augmentation and the inductive bias introduced by network architectures.

4.3 Evaluation

The benchmark for unsupervised contrastive learning is the ImageNet LSVRC 2012. We will discuss three evaluation methods for unsupervised contrastive learning pretraining on this benchmark. The most widely used one is the linear evaluation protocol ((Bachman et al., 2019))where the feature extraction encoder is frozen and a linear layer is trained on top with the labelled training dataset. It worth mentioning data augmentation and regularization are not allowed. Different methods can then directly compare their accuracy on the test dataset.

Another evaluation method is the performance of the learned representations on semi-supervised classification tasks. Semi-supervised classification assumes the decision boundary will not cross through high-density data manifolds. Hence, it implicitly examines whether the embedding space has captured some global structural information of data. Different models are finetuned on a small fraction of labelled data without data augmentation and regularization. Finally, they can compare their

classification rate on the test dataset.
The last method is evaluating the representations' performance on transfer learning tasks. There are more than 20 diverse downstream tasks varying from image segmentation (Cityscapes) to optical flow estimation (KITTI Optical Flow). For image classification, VOC2007 and Caltech101 are commonly used for fair comparision**??**.

5 Future direction

5.1 Hard negative mining

Part with the recent probabilistic interpretation based on InfoMax, a recent study shows finding "hard" negative samples with important signals matters for contrastive learning.In contrastive learning, "hard" negative samples belong to negative sample pairs that are frequently predicted as positive pairs or predicted as negative pairs with a very high loss(Sun et al., 2019). With the intuition of the human learns from their mistakes, we hope the model to improve by correcting these misclassified hard negative pairs. Recent hard mining methods make good progress on classification task with some help from semi-supervised learning. As shown in Figure 10, instead of using Mixup techniques in the input space, **??** mixes up the embeddings of hard negative samples in the lower dimensional embedding space. These synthetic data points increase the density of hard negative samples in the embedding space and help the model to learn invariant transformation in the representations.

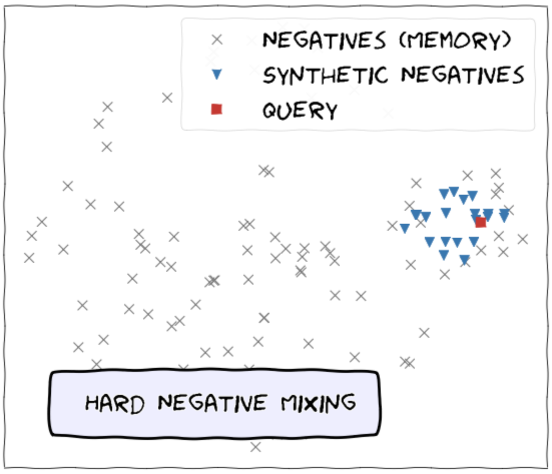


Figure 10: Hard negative mining (Kalantidis et al., 2020)

From a probabilistic perspective, we suggest more investigation be put into combining "hard" samples mining with InfoMin. As shown in Figure **??** and Figure **??**, a hard "negative" sample can be a negative key image that has a lot of mutual information with the query image but a different semantic meaning. A "hard" positive sample can be a positive key that has little mutual information with the query image but the same semantic meaning.

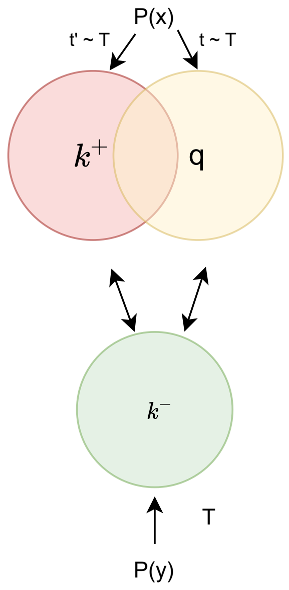


Figure 11: easy samples

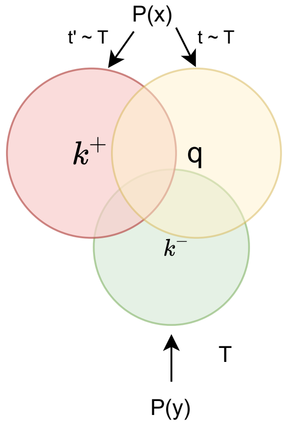


Figure 12: hard samples

6 Experiments

6.1 Part1 Method justification on CIFAR10

We carried out a set of empirical studies on CIFAR10 to verify the effectiveness of several recent contrastive learning methods. Our goal is to learn a representation encoder that embeds input images to a lower-dimensional latent space via contrastive learning. Our downstream task is an image classification task. The linear evaluation protocol will be used to show the effectiveness of different methods. As one of the most well-known benchmarks for image classification, CIFAR-10 consists of 60,000 colour images from 10 classes. Since contrastive learning requires no labels, we use the training dataset and the testing dataset provided by torchvision. To ensure reproducibility, we fix our random seeds at the beginning.

In part1, we evaluate the performance of 4 contrastive methods, namely naïve memory bank, MoCo, SimCLR and SwAV. Here, naïve memory bank (NMB) refers to the PIRL without jigsaw pretraining and multi-head projection. These four methods follow a general Siamese network. The training process can be described as follows. To ensure a fair comparison, we fix our data augmentations as resizedcrop (32x32) followed by colour jitter. This is the best augmentation combination reported for the ImageNet benchmark. We use two ResNet18 as twin encoders and the size of its output embedding is (512,). For simplicity, we use a linear projection that converts the dimension of embedding to 128. These projected embeddings are all normalized before computing cosine similarity. After the similarity is scaled by the temperature, cross-entropy is computed with the labels of the negative pairs set as 0. Finally, we optimize the parameters with a momentum SGD optimizer. The K nearest neighbours algorithm is used for finetuning the hyperparameters of our models because we are not allowed to train a classification head with label data in the middle of training. The KNN monitor also allows us to do early stopping when necessary. The support set of our KNN monitor is the training dataset.

First, we compare the performance of two memory bank-based methods, naïve memory bank (NMB) and MoCo. The NMB's memory bank has the same size as the training dataset as 50000 and the index of each sample can be tracked easily. At the start of training, there is a warmup epoch to fill the memory bank with all the representations of the training dataset, which can be seen from the large magnitude of the training loss curve. Since both methods are memory-based contrastive learning methods, they can be trained seamlessly on a single Tesla GPU. Fig13 and Fig14 compare the training loss of NMB and MoCo. MoCo have a much smoother training curve than NMB. Fig17 and Fig18 compare the validation accuracy, monitored by KNN, of NMB and MoCo. NMB's validation curve is also spikier than MoCo's. This is because the embeddings in MoCo queue are all recently computed. By contrast the embeddings in the naive memory bank are only updated when last seen, and they can be computed hundreds of iterations ago. The momentum encoder also stablizes training by following the target encoder with a moving average. From these observation, we conclude a larger memory bank does not necessarily mean better downstream performance as the model is more likely to encounter the embedding inconsistency problem.

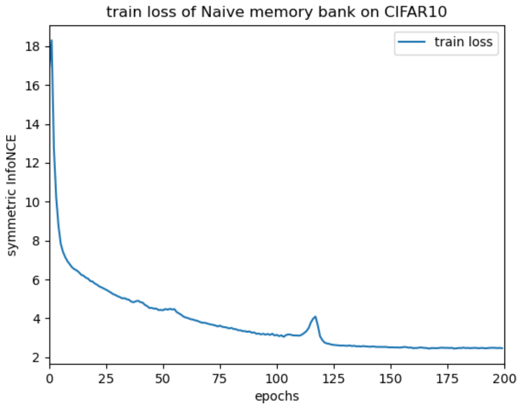


Figure 13: training loss of NMB

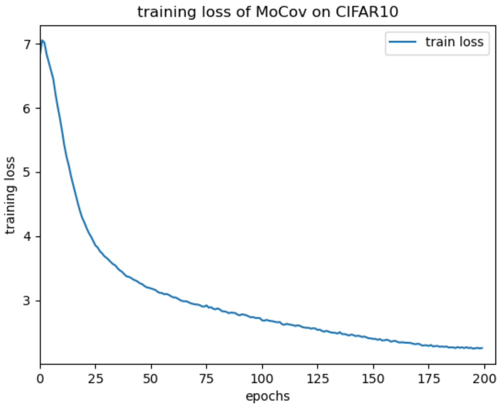


Figure 14: training loss of MoCo

After pretraining, we follow linear evaluation protocol to test our model. First, we freeze our encoder and train a linear classification head on top without any augmentation. Then we test our model on the test dataset.

To evaluate both methods in semi-supervised learning, we reload the checkpoint of our pre-trained encoder and finetune it on 1% of our training data with labels. Still, data augmentation is not allowed. The results are shown in Table 1. MoCo outperforms NMB totally. More importantly, MoCo outcompetes supervised learning on the semi-supervised learning task by 30% which proves the effectiveness of both methods. KNN-based contrastive classifiers also show promising results on the semi-

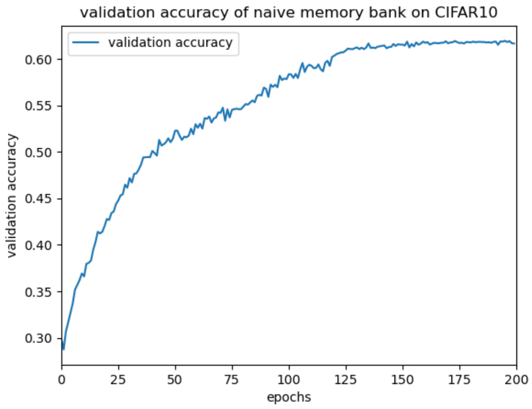


Figure 15: validation accuracy of NMB

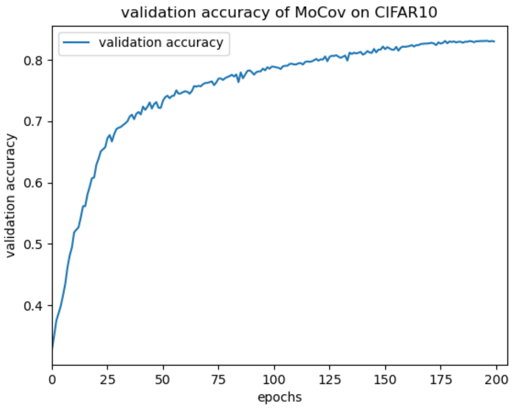


Figure 16: validation accuracy of MoCo

supervised learning task which demonstrates their utility on few shots learning.

Table 1: Performance of different methods in image classification

	label fraction 1%		label fraction 100%		
Methods	KNN	Semi-supervised	KNN	Top1	Top5
NMB	38.10	46.33	61.65	52.10	97.8
MoCo	69.57	73.36	85.20	81.54	99.19
SimCLR	57.02	59.20	67.92	62.50	88.73
SwAV	64.37	70.10		80.96	99.61
Supervised		40.00		85.02	

Here, we verify the effectiveness of two online contrastive learning methods, SimCLR and SwAV. Since SimCLR requires a large batch size to work, we train our model on an 8-core TPU v2 that is publicly available in Colab. We distribute each core with a batch of 256 samples and that makes 2048 in total! To avoid mode collapse, we use normalized temperature-scaled cross-entropy loss (NT-Xent) which is a symmetric (swapped) contrastive loss. We adopt the LARS optimizer from the official implementation(Chen et al., 2020). It is reported that when the number of training epochs is large, a larger temperature (0.5) works better than the smaller temperature (0.1). We believe decreasing the temperature enforces local smoothness. As the number of epochs increases, the variance of the representations increases and the similarity functions can be overconfident which harms representation learning.

Both SimCLR and SwAV are trained end-to-end, so there is neither memory bank nor momentum encoder in their architectures. Both pretrainings are cut off after 100 epochs. Under the linear evaluation protocol, the accuracy of SimCLR is 62.50%, whereas the accuracy of SwAV is 80.96%.

It worth mentioning that SwAV runs much faster than SimCLR. It only takes about

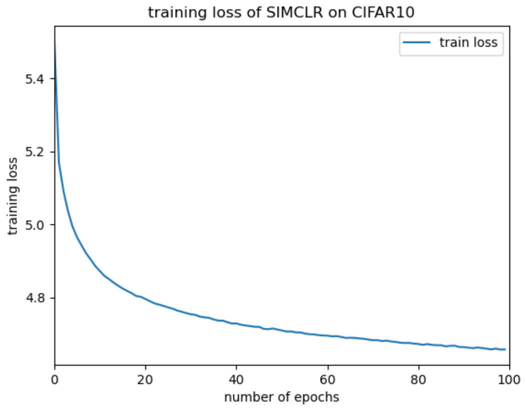


Figure 17: training loss of SimCLR

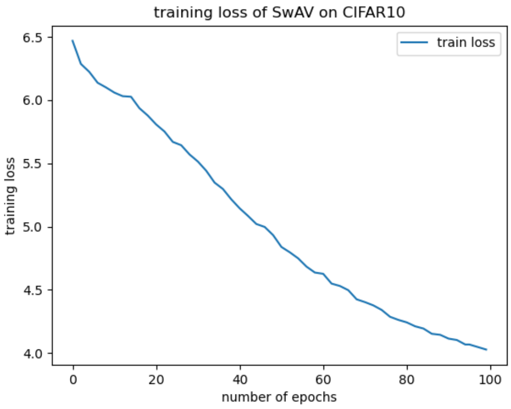


Figure 18: training loss of SwAV

5 hours to run 100 epochs on a single Tesla GPU. This benefits from the direct estimation of cluster embeddings and soft pseudo labels. These cluster embeddings are simply the weights of a linear layer that takes image representations as inputs. Even though SwAV is not fully trained, it achieves a high accuracy. The results of accuracy on semi-supervised learning demonstrate the effectiveness both methods.

We visualize the learned representations of MoCo with t-SNE (perplexity=200), shown in Figure19. The samples from the same class are grouped together naturally. This demonstrates that there are underlying structures in the embedding space and our representations successfully capture some high-level semantics via contrastive unsupervised representation learning.

6.2 Part2 medical image classification

In the second part, we explore how contrastive learning methods perform on a medical image classification task. In particular, we attempt to apply unsupervised contrastive representation learning on a pathological dataset called PatchCamelyon (PCam). Here, our downstream task is a binary classification task where the tissues are predicted as benevolent or malignant. As shown in Figure**??**, we build a unsupervised contrastive learning framework where transformation-invariant representations are extracted from bottom to top. More precisely, the embeddings of two augmented versions of the same input image are brought closer and the embeddings of different images are pushed away.

6.2.1 Data Analysis

A quick introduction to our dataset (PCam): the PatchCamelyon benchmark is an image classification dataset that contains 327,680 images (96 x 96) extracted from

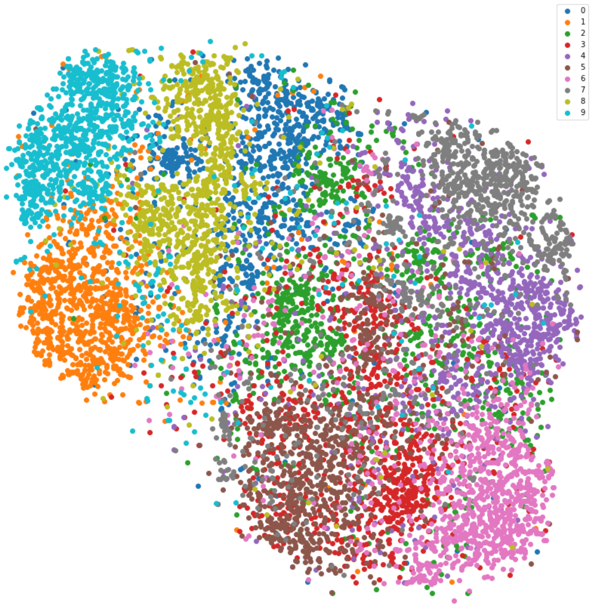


Figure 19: T-SNE visualization of MoCo learned representations for CIFAR10

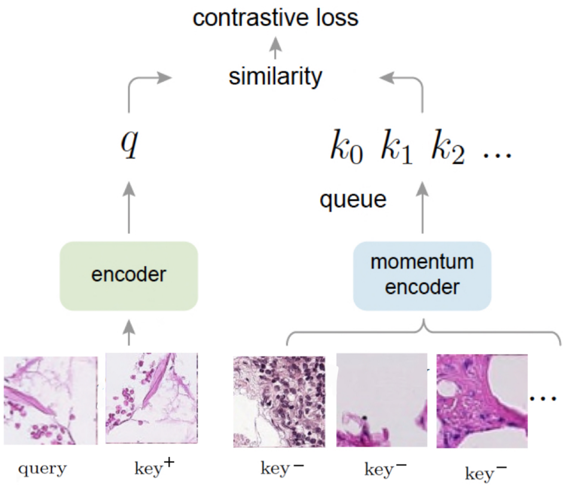


Figure 20: The contrastive representation learning framework for PCam

histopathologic scans of lymph node sections. Each image is annotated with a binary label indicating the presence of a tumour.
First, we do a simple data analysis on our dataset. There are 262,144 images in our training set and 32,768 images in our test set. 50.25% of our training data is malignant and 50.27% of our testing data is malignant. This is a balanced benchmark. Then we compute the histograms of the first 100 tumour patches and first 100 non-tumour patches shown in Figure21 and Figure22. The entropy of the tumour patches is 7.83, which is comparable with the entropy of non-tumour patches (7.67). The histogram of the tumour patches is slightly different from the histogram of the non-tumour patches, but this is probably because different acquisition devices have different illumination and contrast. It is hard for non-specialists to tell the difference from first-order information.

6.2.2 Implementation Details

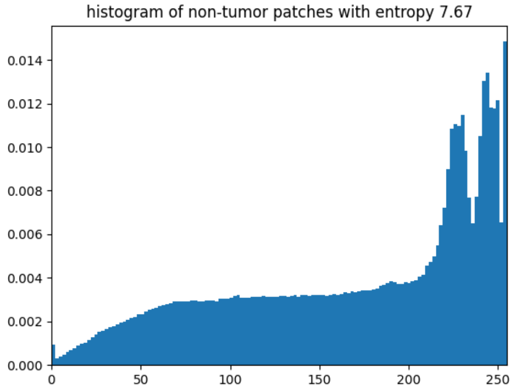


Figure 21: Histogram of benevolent

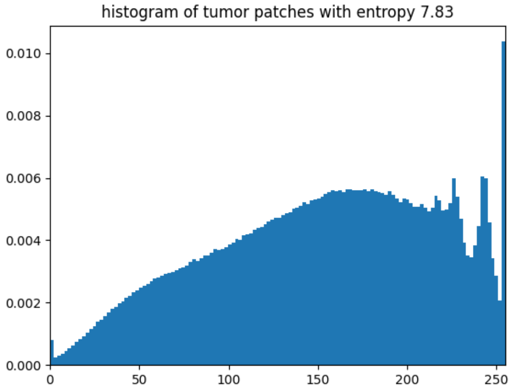


Figure 22: Histogram of malignant

We use random resized crop (96x96) and colour jittering as our first augmentation setup and jigsaw puzzle as our second augmentation setup. We use ResNet18 as encoders and use SGD as our optimizer with a weight decay of 0.1 with milestones at 60 and 120 epochs. In terms of hyper-parameters, we set our batch size as 512, and the size of memory bank as 4096, and the temperature for InfoNCE as 0.07. We train these models with four V100 (16GB) in parallel for 300 epochs which takes approximately 3 days. Our train loss curve is shown in Figure23. The linear evaluation results are presented in Table 2.
Our demo code – Google Colab Jupyter Notebook

6.2.3 Analysis and discussion

In the Figure23, there are several spikes that might be caused by setting learning rate decaying milestones too early. However, the loss is decreasing which demonstrates the convergence of our method. As shown in Table 2, the top1 accuracy of

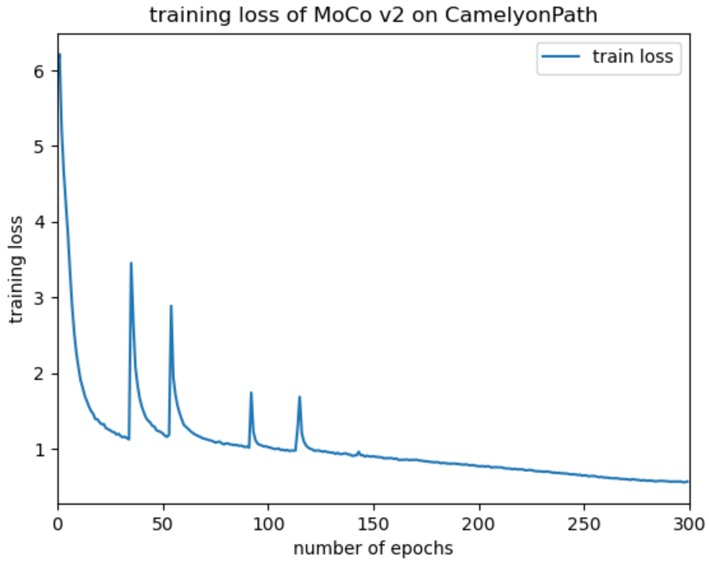


Figure 23: The loss curve of the first model trained on PCam

Table 2: Linear evaluation results of MoCov2 trained on PatchCamelyon

Method	Resized Crop	Color Jittering	Jigsaw	Top1 Accuracy
MoCo	✓	✓	×	61.70
MoCo	×	×	✓	65.22

MoCo with random resized crop and colour jittering is 61.70%, and top1 accuracy of MoCo with jigsaw puzzle augmentation is 65.22%. We argue jigsaw augmentation is more proper for this dataset. Since the linear evaluation protocol only allows data normalization as augmentation, the linear classification head on top is not well trained. If we use a nonlinear classification head and finetune the whole model with more complex augmentated labelled data, we can have much better results. The visualization of learned latent space is shown in Figure24. Although there is some overlap between the two classes in the embedding space, positive samples are mainly mapped to the right part whereas negative samples are mapped to the right part. This demonstrates the effectiveness of the MoCo contrastive learning method.

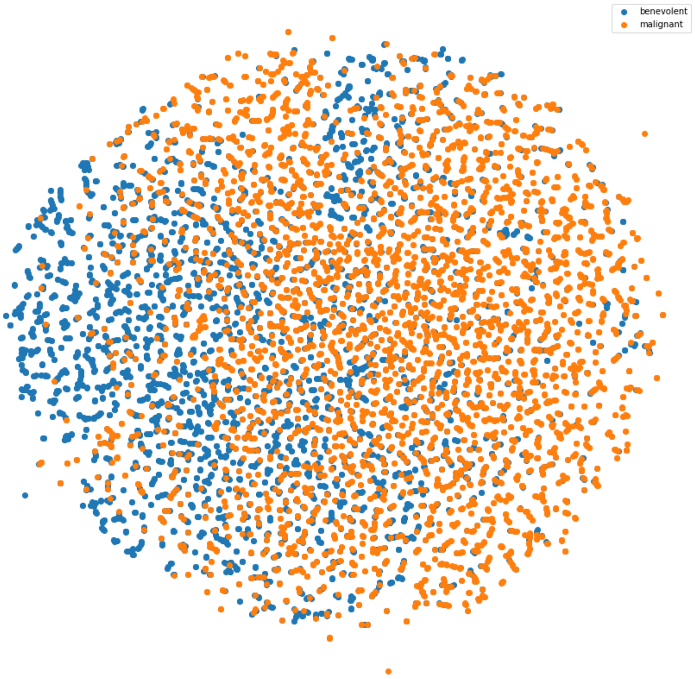


Figure 24: T-SNE visualization of learned representations for PCam

References

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. arXiv preprint arXiv:1906.00910, 2019. pages 14

Andrew Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. **ArXiv**, abs/1809.11096, 2019. pages 5

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020. pages 18

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. **ArXiv**, abs/2011.10566, 2020. pages 6

Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In **2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)**, volume 1, pages 539–546. IEEE, 2005. pages 6, 7

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. **Advances in neural information processing systems**, 26:2292–2300, 2013. pages 11

Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction, 2016. pages 3, 4

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations, 2018. pages 4

Devon Hjelm, A. Fedorov, Samuel Lavoie-Marchildon, and Karan Grewal. L earning deep representations by mutual information estimation and maximization r. 2019. pages 3

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus, 2017. pages 11

Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning, 2020. pages 15

Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. **Foundations and Trends® in Machine Learning**, 12(4):307–392, 2019. ISSN 1935-8245. doi: 10.1561/2200000056. URL `http://dx.doi.org/10.1561/2200000056`. pages 5

Klemen Kotar, Gabriel Ilharco, Ludwig Schmidt, Kiana Ehsani, and R. Mottaghi. Contrasting contrastive self-supervised representation learning models. **ArXiv**, abs/2103.14005, 2021. pages 12

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. **Predicting structured data**, 1(0), 2006. pages 8

R. Linsker. Self-organization in a perceptual network. **Computer**, 21:105–117, 1988. pages 3

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles, 2017. pages 4

Zhuojin Sun, Y. Wang, and R. Laganière. Hard negative mining for correlation filters in visual tracking. **Machine Vision and Applications**, 30:487–506, 2019. pages 15

Yonglong Tian, C. Sun, Ben Poole, Dilip Krishnan, C. Schmid, and Phillip Isola. What makes for good views for contrastive learning. **ArXiv**, abs/2005.10243, 2020. pages 13, 14

Aäron van den Oord, Y. Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. **ArXiv**, abs/1807.03748, 2018. pages 7, 13

A. Widjaja. Learning with kernels: Support vector machines, regularization, optimization, and beyond. **IEEE Transactions on Neural Networks**, 16:781–781, 2003. pages 6

Zhirong Wu, Yuanjun Xiong, S. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. **2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition**, pages 3733–3742, 2018. pages 8

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction, 2021. pages 14