CONSISTENT ASSIGNMENT FOR REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce Consistent Assignment for Representation Learning (CARL). An unsupervised learning method to learn visual representations by combining contrastive learning with deep clustering. By viewing contrastive learning from a clustering perspective, CARL learns unsupervised representations by learning a set of general prototypes that serve as energy anchors to enforce different views of a given image to be assigned to the same prototype. Unlike contemporary work on contrastive learning with deep clustering, CARL proposes to learn the set of general prototypes in an online fashion, using gradient descent without the necessity of performing offline clustering or using side algorithms to solve the cluster assignment problem. CARL achieves comparable results with current state-of-the-art methods in the CIFAR-10, -100, and STL10 datasets.

1 INTRODUCTION

Unsupervised visual representation learning focuses on creating meaningful representations from data and inductive biases. Lately, methods based on Siamese neural networks (Bromley et al., 1994) and contrastive loss functions (He et al., 2020; Chen et al., 2020a) have significantly reduced the accuracy gap between supervised and unsupervised based representations. Indeed, for some downstream tasks, unsupervised-based representations already surpass their supervised counterparts (Caron et al., 2020). In computer vision, approaches to representation learning can be categorized into three groups: (1) contrastive learning methods using instance discrimination, (2) clustering-based methods, and (3) a mixture of the two.

Recent state-of-the-art unsupervised representation learning rely on contrastive learning (Tian et al., 2020; He et al., 2020; Chen et al., 2020c;a; Chen & He, 2020; Oord et al., 2018; Chen et al., 2020b). These methods optimize an instance discrimination pretext task where each image and its transformations are treated as individual classes. They compare feature vectors of individual images with the goal of organizing the feature space such that similar concepts are placed closer while moving different ones farther.

On the other hand, traditional clustering methods aim to learn the data manifold by comparing groups of features that share semantic structure based on a distance metric. When combined with deep learning, clustering methods are often designed as two-step algorithms: first, a sizable portion of the dataset is clustered, and then the meta clustering information, e.g., prototypes and pseudo-labels, are used as supervised signals in a posterior optimization task (Caron et al., 2018; 2019; Asano et al., 2019; Yan et al., 2020).

Recent work has also attempted to combine the benefits of contrastive learning and clustering (Li et al., 2020; Caron et al., 2020). In particular, Expectation-Maximization approaches alternate between finding the clusters and maximizing the mutual information between the embeddings and the cluster centroids (Li et al., 2020). Inspired by them, our work merges the benefits of both approaches by bridging the gap between clustering and contrastive learning. On the one hand, we use unsupervised clustering dynamics to generate robust prototypes that organize the feature space. On the other, we use contrastive learning to compare the distributions of the views' assignments w.r.t. the clusters. Our experiments show that by mixing both approaches, we can learn useful visual representations in an unsupervised way that performs on par with existing methods in downstream tasks.

From a clustering perspective, we can think of contrastive learning as learning clustered representations at the image-level. However, given the nature of the task, these clusters fail to capture semantic information from the heterogeneous unknown classes because the learned clusters only comprise representations from synthetic views of an image. Moreover, since contrastive learning methods handle different images as negatives in the training process, even if two given samples share the same class information, their representations will be pushed farther apart from each other. In the end, each image will have its own cluster structure.

We propose an alternative method to learn high-level features by clustering views based on consistent assignments. Unlike concurrent work that uses K-Nearest Neighbors (Altman, 1992) or K-Means (Lloyd, 1982) as priors to enforce (learn) a cluster mapping, our method learns the prototypes online. Instead of directly maximizing similarities between image embeddings, we force the distribution of positive views' assignments to be consistent among a set of finite learnable prototypes. If the number of prototypes equals the number of observations in the dataset, we would be forcing each cluster only to contain synthetic views of a given observation. This is equivalent to contrastive learning with instance discrimination. However, if we set the number of prototypes to be smaller than the number of observations in the dataset, by the pigeonhole principle, the learned prototypes will not only cluster different views of an image together, but it will also contain representations of different images that are similar enough to be assigned to the same cluster.

Regarding our contributions, (i) we propose a learning framework that leverages current contrastive learning methods with clustering-based algorithms to improve the learned representations. Unlike contemporary work, our method proposes to learn the clusters' assignments in an online fashion using gradient descent with no need for pre-clustering steps e.g., K-Means or offline procedures to solve the clustering assignment problem, such as the Sinkhorn-Knopp algorithm (Cuturi, 2013). (ii) We contrast high-level structures (the distributions of the views over the cluster assignments) instead of low-level ones (such as the representations). And, (iii) Our learned prototypes do not need to hold the semantics of the data but rather become energy anchors that self-organize the space to learn better representations. Moreover, our proposed loss function does not require a more extensive set of negative representations, which avoids the common problem of treating representations for the same class as negatives.

1.1 Related Work

This work builds on top of two main approaches to unsupervised visual representations learning: deep clustering and un/self-supervised contrastive learning.

Self-supervised learning concerns the idea of devising pretext tasks that extract supervised signals from the data (Doersch et al., 2015; Noroozi & Favaro, 2016; Gidaris et al., 2018; Zhang et al., 2016). Most of these methods work with the same principle. They corrupt the input with stochastic random transformations and challenge the network to predict some property of the corrupted input.

One such pretext task is instance discrimination (Dosovitskiy et al., 2015; Wu et al., 2018). It describes a classification task in which each image is treated as a unique class and, therefore, stochastic transformations of the same image, often called views, should belong to the same class. Dosovitskiy et al. (2015) proposed to optimize this task by learning a linear classifier where the number of output classes matches the number of observations in the dataset. Following, Wu et al. (2018) proposed to use a Noise-Contrastive Estimation (NCE) approximation of the non-parametric softmax classifier that could scale to large datasets gracefully. Currently, contrastive learning (Hadsell et al., 2006) methods rely on an NCE based loss function called InfoNCE (Oord et al., 2018; Tian et al., 2019). Recent work describes optimizing the InfoNCE loss through the lens of maximizing the mutual information between representations of the same image (Hjelm et al., 2018; Henaff, 2020; Bachman et al., 2019). In practice, the success of InfoNCE requires a high number of negative embeddings. Nonetheless, since negatives are usually randomly sampled from the dataset, it often leads to a false-negative problem where representations from images of the same class are treated as negatives (Saunshi et al., 2019).

He et al. (2020) presented MoCo, a contrastive learning framework that employs an additional momentum encoder to provide consistent instance representations for the InfoNCE loss. Chen et al. (2020a) presented SimCLR, a Siamese-based (Bromley et al., 1994) contrastive learning method trained with InfoNCE that relies on large bath sizes to draw a high number of negative samples.

BYOL (Grill et al., 2020) proposes a framework that does not require negative samples and learns visual representations by approximating augmented views of the same data point using an ℓ_2 loss in the latent space. Unlike contrastive learning, we seek to learn prototype vectors that act as anchors on our embedding space. We use these anchors as energy beacons for the images. Our goal is to use the energy distributions induced by the similarity of the images w.r.t. the prototypes to find representations that share similarities in the embedding space.

Recent work proposed clustering-based methods for deep unsupervised representation learning (Asano et al., 2019; Yan et al., 2020; Caron et al., 2019; Li et al., 2020; Caron et al., 2020). DeepCluster (Caron et al., 2018) learns representations by predicting cluster assignments. One of the limitations of this approach is that the classification layer needs to be reinitialized once per clusterization. DeeperCluster (Caron et al., 2019) builds on top of (Caron et al., 2018) and presents an algorithm to combine hierarchical clustering with unsupervised feature learning using the rotation prediction pretext task (Gidaris et al., 2018). Similarly, Prototypical Contrastive Learning (PCL) Li et al. (2020), formulates a self-supervised visual representation learning framework as an Expectation-Maximization (EM) algorithm.

Our method utilizes a conceptually distinct methodology. Unlike (Caron et al., 2018; Li et al., 2020), our method does not require a pre-clusterization step of the entire corpus, which vastly reduces memory and computing power requirements. Moreover, since we do not use K-Means clustering as a proxy to learn an additional task, we do not need to reinitialize any layers during optimization, nor is our method susceptible to limitations and assumptions implied by the K-Means algorithm. Instead, we propose to learn the prototypes end-to-end by consistently enforcing different views of the same image to be assigned to the same prototypes. Lastly, our method does not rely on handcrafted pretext tasks.

Similar to our work, Caron et al. (2020) proposes an online clustering method to learn visual representations by contrasting cluster assignments on a second set of latent variables. To avoid collapsing modes where all observations are assigned to a few classes, they use the Sinkhorn-Knopp algorithm (Cuturi, 2013) to solve the cluster assignment problem over the latent variables to guide the encoder during learning. Unlike (Caron et al., 2020), our method learns cluster assignments end-to-end via gradient descent and avoids trivial solutions by enforcing a regularization term over the cluster assignments of views in a given batch. Moreover, we use a simpler representation space that does not need a second set of latent variables to stabilize the clusters.

Van Gansbeke et al. (2020) presented a two-step algorithm for unsupervised classification and proposed the SCAN-Loss (Semantic Clustering by Adopting Nearest neighbors) as part of the learning pipeline. Their algorithm extracts a set of nearest neighbors from each observation and uses them as priors to learn a second network for semantic clustering. Our implementation builds on top of the SCAN-Loss, but unlike Van Gansbeke et al. (2020), we employ a Siamese network architecture to learn representations via cluster assignments, end-to-end, without the necessity of optimizing for a second self-supervised task or mining of nearest neighbors.

2 Proposed Method

Our proposal, Consistent Assignment for Representation Learning (CARL), re-frames contrastive learning through a clustering perspective to learn robust representations (Section 2.1). CARL builds distributions of the similarities between the prototypes and the image's views (Section 2.2). To avoid collapsing the representation in a subset of clusters, we impose an uninformative prior to CARL prototypes (Section 2.3). CARL then optimizes both the similarity to the learned prototypes and the uninformative prior (with a decay schedule for learning). Fig. 1 illustrates the learning pipeline.

2.1 Contrastive Learning from a Clustering Perspective

Let $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ be a dataset containing N unlabeled images. And, let a view $v_i = T(x_i)$ of an observation x_i as the application of a stochastic function T that is designed to change the content of x_i subjected to preserving the task-relevant information encoded in it. In practice, we can create as many views as needed by applying the stochastic function T. Contrastive learning methods propose to learn visual embeddings by solving an instance discrimination pretext task that is usually

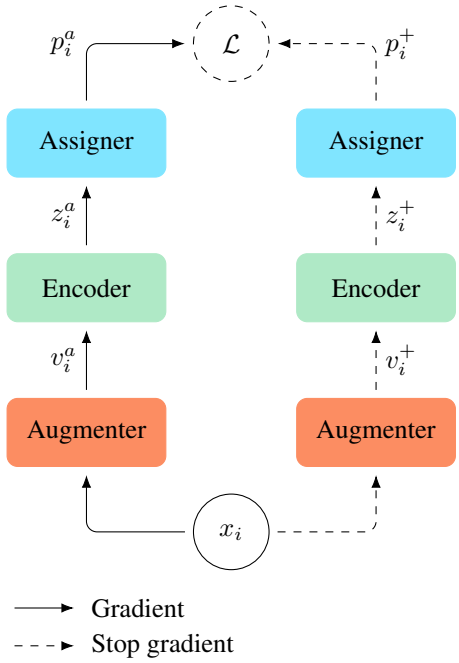

Figure 1: An observation x_i gets transformed into two stochastic views, $v_i^{(\cdot)}$, that are further encoded into a representation space, $z_i^{(\cdot)}$. We obtain a distribution, $p_i^{(\cdot)}$, of the representations' assignments to a set of clusters. Our objective $\mathcal{L}$ is to compare these distributions and minimize their difference. We use one view as an anchor (superscripted by a) that is trained, while another positive view (superscripted by $+$) that is not.

optimized using the InfoNCE loss (Oord et al., 2018) defined as

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{\exp\left(\text{sim}\left(z_i^a, z_i^+\right)/\tau\right)}{\sum_j^M \exp\left(\text{sim}\left(z_i^a, z_j\right)/\tau\right)}, \tag{1}$$

where z_i^a and z_i^+ are anchor and positive representations taken from an encoder function f such that $z_i^{(\cdot)} = f\left(v_i^{(\cdot)}\right)$, τ is the temperature parameter, and $\text{sim}(\cdot)$ is a similarity function, e.g., the cosine similarity. If we view contrastive learning from a clustering perspective, the InfoNCE loss (1) is minimized when all possible variants $\left\{v_i^j\right\}_j$ of an image x_i are clustered into the same prototype while representations from within a cluster are far apart from the M negative representations in the denominator of (1).

We propose an approach where, instead of comparing against other instances (He et al., 2020) or prototypes of the classes (Li et al., 2020), we learn a set of K general prototypes $\mathcal{C} = \{c_1, c_2, \ldots, c_K\}$, $K \ll N$, against which we compare the views to determine their similarity and to promote consistency and confidence when assigning views to clusters. I.e., views must agree with high confidence in their cluster assignments.

2.2 Consistent Assignment for Representation Learning

As with previous methods, we treat augmented versions of a given image as views and use them as positive examples for optimization. Our objective is to transform two positive samples into a distribution of their likelihood to belong to a set of K clusters. To do so, we encode each view through an encoder function $z_i = f(v_i) \in \mathbb{R}^d$. The encoder f comprises a backbone convolutional neural network, such as a ResNet (He et al., 2016), followed by a non-linear multilayer perceptron (MLP) head.

Our objective is not to cluster the data in an unsupervised manner but rather to learn a set of prototypes that will serve as anchors to differentiate views. Our hypothesis is that similar views should

have similar assignments w.r.t. the prototypes. Hence, to convert the representations into these assignments, we first compare the representation z_i against all the prototypes to obtain an energy distribution

$$q_i[j] = \langle z_i, c_j \rangle, \tag{2}$$

where $q_i[j]$ is the j-th element of the energy distribution q_i for the i-th view. We learn the set of prototypes through a linear layer. Thus, to get a distribution of a given view over all prototypes, we normalize the energy using the softmax function and obtain the posterior probability distribution, i.e., the probability of assigning the view v_i to the cluster k. Hence, our normalized probability for the k-th class given our view v_i is

$$p_i[k] = P(i \text{ assigned to } k \mid v_i) = \frac{\exp(q_i[k])}{\sum_{j=1}^{K} \exp(q_i[j])}, \tag{3}$$

where $q_i[j]$ denotes the j-th element of the i-th un-normalized vector output of the classifier for the respective view.

As we mentioned before, our main objective is to contrast the distributions of two views' likelihood w.r.t. the clusters. To do so, the encoder and assigner operate in a Siamese setup where a pair of views from a given sample is independently transformed in its corresponding distribution, see Fig. 1. To ensure the similarity between the views, we optimize the views' distributions p_i^a and p_i^+ over the clusters in $\mathcal{C}$, so that the two distributions are consistent with one another. In other words, by learning a consistent assignment of views over the clusters, a given prototype will be invariant to augmented versions of an input sample. Moreover, because the number of prototypes is smaller than the number of observations in the dataset, the clusters will also contain different observations that share similarities in the embedding space.

We compute the similarity between the views' distributions as their dot product

$$\mathcal{L}_c = -\frac{1}{B} \sum_{i}^{B} \log \langle p_i^a, p_i^+ \rangle, \tag{4}$$

where B is the size of a minibatch over which we are aggregating the samples. In the ideal case of two one-hot vectors signaling the same perfect assignment, the dot product above yields its maximum value of one, and the negative $\log$ is minimized.

2.3 Preventing Trivial Solutions

Only forcing different views v_i^a and v_i^+ to have the same cluster assignment using our consistency loss (4) leads the networks to find a trivial solution where all representations z_i are assigned to the same cluster—cf. Fig. 2. To prevent such triviality, we force the distribution over the classes, P, to be uninformative by minimizing the Kullback-Leibler divergence w.r.t. a uniform distribution, U. Our regularization is

$$\mathcal{L}_{\text{KL}} = \text{KL}(P \parallel U) = \log_2(n) + \sum_{c \in C} \hat{p}_i^c \log\left(\hat{p}_i^c\right), \tag{5}$$

where $\hat{p}_i^c$ is the expected distribution over a minibatch of size B,

$$\hat{p}_i^c = \frac{1}{B} \sum_{i}^{B} p_i^c. \tag{6}$$

In other words, we maximize the Shannon entropy of the average distribution of the predictions. We can interpret the KL-divergence (5) as regularizing the encoder f to encourage the approximate posterior (3) to be closer to the uniform distribution.

Minimizing the KL-divergence (5) will force the predictions within a batch to be spread across all clusters. Since we do not know the underlying class distribution in advance, the KL-divergence (5) acts as a non-informative prior where we assume that the observations $\mathcal{X}$ are uniformly assigned among all K prototypes.

By combining the consistency assignment loss (4) with the KL-divergence regularization (5) we obtain our final learning objective

$$\mathcal{L} = \mathcal{L}_c + \lambda_e \mathcal{L}_{\text{KL}}, \tag{7}$$

Table 1: Top-1 accuracy averaged over 3 runs on the linear evaluation protocol (He et al., 2020).

Epochs	50	100	150	200	50	100	150	200	50	100	150	200
	CIFAR-10				CIFAR-100				STL-10			
BYOL	59.03	68.90	73.25	76.46	29.33	37.26	42.31	45.68	70.48	76.39	79.53	80.74
SimCLR	67.68	72.29	74.69	76.33	37.99	43.09	45.74	47.29	74.26	77.36	78.79	80.57
MoCo v2	59.00	65.57	69.42	71.62	33.07	39.12	42.22	44.35	66.78	71.50	74.19	75.46
CARL	66.81	72.32	75.27	77.41	32.87	38.27	42.19	44.38	73.06	76.50	78.78	80.27

where λ_e is an epoch-dependent function that returns a scalar that prevents mode collapse at the beginning of training. We observed that training is very susceptible to such collapsing to a single assignment if not regularized. However, in practice, we noticed that keeping a large fixed value of λ_e during training also prevents the encoder from learning more complex representations. Thus, we recommend a function λ_e that decreases as training progresses. In theory, any decay schedule, such as an exponential or cosine, could be used. We propose a linear decay schedule

$$\lambda_e = \begin{cases} b - \frac{b-a}{E} e & \text{if } e \leq E, \\ a & \text{otherwise,} \end{cases} \tag{8}$$

where b and a denote the start and ending values of the decay, E represents the number of epochs in which the decay will happen, and e is the epoch counter.

3 Unsupervised Feature Evaluation

We evaluate CARL representations extracted from a ResNet-18 backbone encoder and compare the performance with different state-of-the-art methods using the linear evaluation protocol proposed by He et al. (2020). All the models were trained for 200 epochs with the same cosine learning rate decay (Loshchilov & Hutter, 2016) schedule and equal batch sizes. For more information, refer to Appendix A. Table 1 compares our results with previous approaches on CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009), and STL-10 (Coates et al., 2011). We use MoCo v2 as our primary baseline. Note that MoCo v2 alone is a solid baseline since it has been shown to outperform SimCLR on ImageNet linear evaluation. Our method was able to outperform our primary baseline (MoCo) on most of the datasets and performs on par (if not superior) with other implementations, including BYOL and SimCLR. Moreover, unlike SimCLR and MoCo v2, our method does not use negative samples, nor does it require a momentum-based target encoder to prevent collapsing (Grill et al., 2020). The results on Table 1 also show a different side from the state-of-the-art methods since, usually, they are only trained on large-scale datasets like the ImageNet (Deng et al., 2009), which raises questions regarding how they perform of smaller datasets. Additionally, we show an ablation study in the Appendix B.

4 Conclusion

In this work, we presented Consistent Assignment for Representation Learning (CARL). An unsupervised method that learns visual representations by forcing augmented versions of an observation to be consistently assigned over a finite set of learnable prototypes. Unlike contrastive learning methods, we propose a higher-level pretext task that operates over the distributions of views instead of directly optimizing the view's embeddings. Our method also differs from recent work that merges clustering with contrastive learning since it does not require pre-clusterization steps or algorithms to solve the cluster assignment problem. Instead, we learn a set of general prototypes that act as energy anchors for the views' representations, entirely online using gradient descent. We studied some of the main components of CARL and the effects of different configurations of hyperparameters. Lastly, our results show that representations learned by CARL rival or even surpass current state-of-the-art in contrastive learning without resorting to a large number of negative samples or extra encoders.

REFERENCES

Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*, 2019.

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

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, pp. 737–737, 1994.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 132–149, 2018.

Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2959–2968, 2019.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020a.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020b.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning, 2020c.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

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

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pp. 1422–1430, 2015.

Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1734–1747, 2015.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pp. 1735–1742. IEEE, 2006.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pp. 4182–4192. PMLR, 2020.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020.

Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2): 129–137, 1982.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pp. 69–84. Springer, 2016.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. A theoretical analysis of contrastive unsupervised representation learning. In *International Conference on Machine Learning*, pp. 5628–5637. PMLR, 2019.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Learning to classify images without labels. *arXiv preprint arXiv:2005.12320*, 2020.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3733–3742, 2018.

Xueting Yan, Ishan Misra, Abhinav Gupta, Deepti Ghadiyaram, and Dhruv Mahajan. Clusterfit: Improving generalization of visual representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6509–6518, 2020.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pp. 649–666. Springer, 2016.

A IMPLEMENTATION DETAILS

For all experiments, the encoder function f comprises a ResNet-18 backbone followed by a non-linear 2-hidden layer fully-connected network as projection head as defined by (Grill et al., 2020). For all methods, the function f encodes an image x_i into a 128-dim representation and we closely follow the hyperparameters proposed by (He et al., 2020). Namely, learning rate (0.03) and decay schedule (cosine with no restart), as well as the choice of the optimizer (SGD with momentum) and the batch sizes (256), are consistent among all methods. The set of data augmentations used to create the views follow (Chen et al., 2020a).

Regarding our implementations of MoCo v2, BYOL, and SimCLR, except for MoCo, in which we used the official code repository, the other two methods are our implementations (closely following the official releases in terms of code design and additional hyperparameters). Lastly, for feature evaluation, we strictly follow the linear evaluation protocol proposed by (He et al., 2020).

B ABLATIONS

In this section, we evaluate the effects of the main hyperparameters of our method. When not specified otherwise, all experiments follow a similar protocol. We learn representations using a ResNet-18 backbone trained for 150 epochs, and the KL weight penalty λ_e is linearly decayed over the first $E = 100$ epochs. To evaluate the multiple experimental setups, we train linear classifiers on top of the encoder's frozen features following the linear evaluation protocol proposed by He et al. (2020) and report average Top-1 accuracy over three independent runs.

B.1 DOES DECREASING THE KL WEIGHT PENALTY IMPROVES REPRESENTATION LEARNING?

The hyperparameter λ_e controls the contribution of the KL regularization (5) to the consistency loss (4). Especially at the beginning of training, a higher contribution for the KL term avoids mode collapsing, where the network optimizes the consistency loss (4) by assigning all observations to the same prototype. Van Gansbeke et al. (2020) make similar claims for the entropy regularization in their SCAN-loss (Van Gansbeke et al., 2020), and suggest a high (constant) value for the scalar hyperparameter λ_e to avoid such trivialities.

We hypothesize that keeping a high value of λ_e over the course of training also prevents the network from learning complex features. To verify this hypothesis, we trained CARL on the STL-10 (Coates et al., 2011) unsupervised dataset for 200 epochs. We measure the performance by training a linear classifier on top of the frozen features of the ResNet-18 backbone. We linearly decay the magnitude of the λ_e hyperparameter, following (8), from $b = 2.0$ to $a = 1.0$ over the first $E = 100$ epochs instead of keeping it constant for one of the experiments. As shown in Fig. 2, we observe that the quality of the representations learned by CARL benefits from decreasing the contribution of the KL regularization. Also, a smaller value of λ_e may guide the encoder to a non-optimal solution at the beginning of training.

B.2 DOES THE NUMBER OF GENERAL PROTOTYPES INFLUENCE THE QUALITY OF THE REPRESENTATIONS?

To evaluate the effect of learning a different number of prototypes, we trained CARL with ResNet-18 backbones on the CIFAR-10 and CIFAR-100 datasets (Krizhevsky et al., 2009) for 150 epochs. For all experiments, the KL penalty function λ_e starts as $b = 4$ and is linearly decreased to $a = 1.5$ over the first $E = 100$ epochs. Fig. 3 suggests that over clustering benefits the quality of the learned representations. Moreover, the optimal number of general-prototypes depends on the number of actual classes of the dataset, where the best results are obtained when the number of general prototypes is nearly one order of magnitude larger than the number of actual classes of the data.

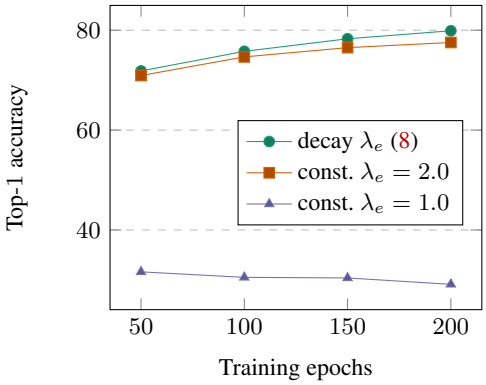

Figure 2: Effect of the uninformative prior's scheduling λ_e on the overall performance in STL-10. Notice that a linear decay scheduling outperforms its constant counterparts and that a lower value of λ_e produces non-optimal solutions due to mode collapsing.

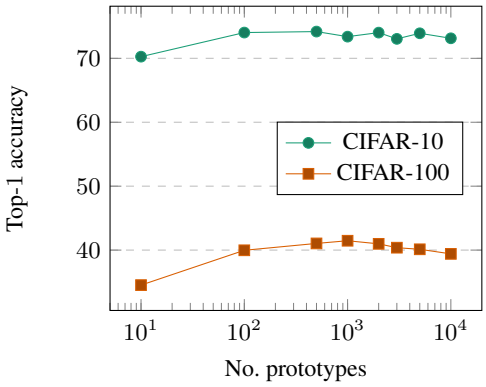

Figure 3: Effect of learning a different number of prototypes on the quality of the representations. Empirical tests suggest an inverse U-shape curve where the optimal number of prototypes lies near one order of magnitude w.r.t. the actual number of classes of the dataset.