Ranking Info Noise Contrastive Estimation: Boosting Contrastive Learning via Ranked Positives

David T. Hoffmann[*,1,2], **Nadine Behrmann**[*,1], **Juergen Gall**[3], **Thomas Brox**[2], **Mehdi Noroozi**[1]

[1]Bosch Center for Artificial Intelligence [2]University of Freiburg [3]University of Bonn
{david.hoffmann2, nadine.behrmann}@de.bosch.com

Abstract

This paper introduces Ranking Info Noise Contrastive Estimation (RINCE), a new member in the family of InfoNCE losses that preserves a ranked ordering of positive samples. In contrast to the standard InfoNCE loss, which requires a strict binary separation of the training pairs into similar and dissimilar samples, RINCE can exploit information about a similarity ranking for learning a corresponding embedding space. We show that the proposed loss function learns favorable embeddings compared to the standard InfoNCE whenever at least noisy ranking information can be obtained or when the definition of positives and negatives is blurry. We demonstrate this for a supervised classification task with additional superclass labels and noisy similarity scores. Furthermore, we show that RINCE can also be applied to unsupervised training with experiments on unsupervised representation learning from videos. In particular, the embedding yields higher classification accuracy, retrieval rates and performs better in out-of-distribution detection than the standard InfoNCE loss.

Introduction

Contrastive learning recently triggered progress in self-supervised representation learning. Most existing variants require a strict definition of positive and negative pairs used in the InfoNCE loss or simply ignore samples that can not be clearly classified as either one or the other (Zhao et al. 2021). Contrastive learning forces the network to impose a similar structure in the feature space by pulling the positive pairs closer to each other while keeping the negatives apart.

This binary separation into positives and negatives can be limiting whenever the boundary between those is blurry. For example, different samples from the same classes are used as negatives for *instance recognition*, which prevents the network from exploiting their similarities. One way to address this issue is supervised contrastive learning (SCL) (Khosla et al. 2020), which takes class labels into account when making pairs: samples from the same class are treated as positives, while samples of different classes pose negatives. However, even in this optimal setting with ground truth labels, the problem persists – semantically similar classes share many visual features (Deselaers and Ferrari 2011) with the query – and some samples cannot clearly be categorized as either positive or negative, *e.g.* the dog breeds in Fig. 1. Treating them as positives makes the network invariant towards the distinct attributes of the samples. As a result, the network struggles to distinguish between different dog breeds. If they are treated as negatives, the network cannot exploit their similarities. For transfer learning to other tasks, *e.g.* out-of-distribution detection, a clean structure of the embedding space, s.t. samples sharing certain attributes will be closer, is beneficial.

Another example comes from video representation learning: In addition to spatial crops as for images, videos allow to create temporal crops, *i.e.* creating a sample from different frames of the same video. To date, it is an open point of discussion whether temporally different clips from the same video should be treated as positive (Feichtenhofer et al. 2021) or negative (Dave et al. 2021). Treating them as positives will force the network to be invariant towards changes over time, but treating them as negatives will encourage the network to ignore the features that stay constant. In summary, a binary classification in positive and negative will, for most applications, lead to a sub-optimal solution. To the best of our knowledge, a method that benefits from a fine-grained definition of negatives, positives and various states in between is missing.

As a remedy, we propose Ranking Info Noise Contrastive Estimation (RINCE). RINCE supports a fine-grained definition of negatives and positives. Thus, methods trained with RINCE can take advantage of various kinds of similarity measures. For example similarity measures can be based on class similarities, gradual changes of content within videos, pretrained feature embeddings, or even the camera positions in a multi-view setting etc. In this work, we demonstrate class similarities and gradual changes in videos as examples.

RINCE puts higher emphasis on similarities between related samples than SCL and cross-entropy, resulting in a richer representation. We show that RINCE learns to represent semantic similarities in the embedding space, s.t. more similar samples are closer than less similar samples. Key to this is a new InfoNCE-based loss, which enforces gradually decreasing similarity with increasing rank of the samples.

The representation learned with RINCE on Cifar-100 improves significantly over cross-entropy for classification, retrieval and OOD detection, and outperforms the stronger SCL baseline (Khosla et al. 2020). Here, improvements are

[*]These authors contributed equally.

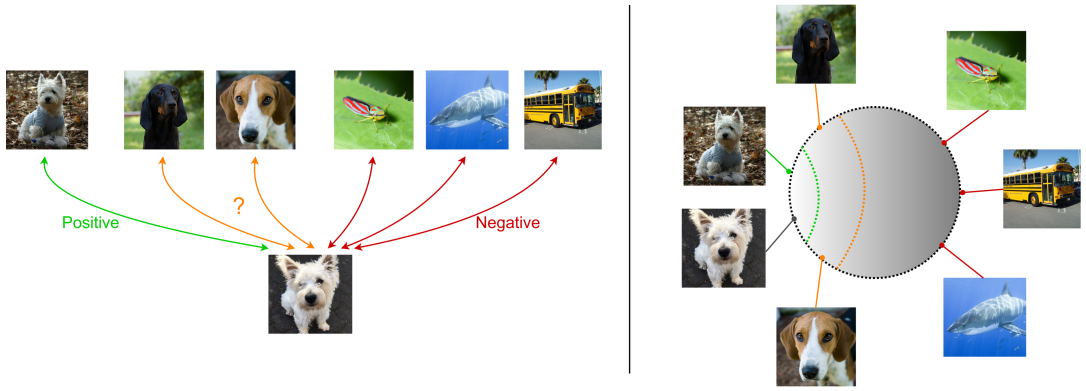


Figure 1: Contrastive Learning should not be binary. In many scenarios a strict separation of samples in "positives" and "negatives" is not possible. So far, this grey zone (left) was neglected, leading to sub-optimal results. We propose a solution to this problem, which embeds same samples very close and similar samples close in the embedding space (right).

particularly large for retrieval and OOD detection. To obtain ranked positives for RINCE, we use the superclasses of Cifar-100. Further, we demonstrate that RINCE works on large scale datasets and in more general applications, where ranking of samples is not initially given and contains noise. To this end, we show that RINCE outperforms our baselines on ImageNet-100 using only noisy ranks provided by an off-the-shelf natural language processing model (Liu et al. 2019). Finally, we showcase that RINCE can be applied to the fully unsupervised setting, by training RINCE unsupervised on videos, treating temporally far clips as weak positives. This results in a higher accuracy on the downstream task of video action classification than our baselines and even outperforms recent video representation learning methods.

In summary, our contributions are: 1) We propose a new InfoNCE-based loss that replaces the binary definition of positives and negatives by a ranked definition of similarity. 2) We study the properties of RINCE in a controlled supervised setting. Here, we show mild improvements on Cifar-100 classification and sensible improvements for OOD detection. 3) We show that RINCE can handle significant noise in the similarity scores and leads to improvements on large scale datasets. 4) We demonstrate the applicability of RINCE to self-supervised learning with noisy similarities in a video representation learning task and show improvements over InfoNCE in all downstream tasks. 5) Code is available at[1]. The Sup. Mat. can be found in (Hoffmann et al. 2022).

Related Works

Contrastive Learning. Contrastive learning has recently advanced the field of self-supervised learning. Current state-of-the-art methods use *instance recognition*, originally proposed by (Dosovitskiy et al. 2016), where the task is to recognize an instance under various transformations. Modern instance recognition methods utilize *InfoNCE* (van den Oord, Li, and Vinyals 2018), which was first proposed as *N-pair* loss in (Sohn 2016). It maximizes the similarity of *positive pairs* – which are obtained from two different views of the same instance – while minimizing the similarity of *negative pairs*, *i.e.* views of different instances. Different views can be generated from multi-modal data (Tian, Krishnan, and Isola 2020), permutations (Misra and van der Maaten 2020), or augmentations (Chen et al. 2020a). The negative pairs play a vital role in contrastive learning as they prevent shortcuts and collapsed solutions. In order to provide challenging negatives, (He et al. 2020) introduce a memorybank with a momentum encoder, which allows to store a large set of negatives. Other approaches explicitly construct *hard negatives* from patches in the same image (van den Oord, Li, and Vinyals 2018) or temporal negatives in videos (Behrmann, Gall, and Noroozi 2021). More recent works omit negative pairs completely (Chen and He 2021; Grill et al. 2020).

In the above cases, positive pairs are obtained from the same instance, and different instances serve as negatives even when they share the same semantics. Previous work addresses this issue by allowing multiple positive samples: (Miech et al. 2020) allows several positive candidates within a video, (Han, Xie, and Zisserman 2020) and (Caron et al. 2020) obtain positives by clustering the feature space, whereas (Khosla et al. 2020) uses class labels to define a set of positives. False negatives are eliminated from the InfoNCE loss by (Huynh et al. 2020), either using labels or a heuristic. Integrating multiple positives in contrastive learning is not straightforward: the set of positives can be noisy and include some samples that are more related than others. In this work, we provide a tool to properly incorporate such samples.

Supervised Contrastive Learning. Labelled training data has been used in many recent works on contrastive learning. (Romijnders et al. 2021) use pseudo labels obtained from a detector, (Tian et al. 2020) use labels to construct better views and (Neill and Bollegala 2021) use similarity of class word embeddings to draw hard negatives. The term *supervised contrastive learning* (SCL) is introduced in (Khosla et al. 2020) showing that SCL outperforms standard cross-entropy.

In the SCL setting ground truth labels are available and can be used to define positives and negatives. Commonly, samples from the same class are treated as positive, while in-

[1]https://github.com/boschresearch/rince

stances from all other classes are treated as negatives. (Khosla et al. 2020) find that the SCL loss function outperforms cross-entropy in the supervised setting. In contrast, (Huynh et al. 2020) aim for an unsupervised detection of false negatives. They propose to only eliminate false negatives from the InfoNCE loss which leads to best results for noisy labels.

Along these lines, (Winkens et al. 2020) show that InfoNCE loss is better suited for out-of-distribution detection than cross-entropy. Here, we introduce a method to deal with non-binary similarity labels and study different versions of it in the SCL setting free from label noise and show that we get similar results in more noisy and even unsupervised settings.

Ranking. *Learning to Rank* has been studied extensively (Burges et al. 2005; Cakir et al. 2019; Cao et al. 2007; Liu 2009). These works aim for downstream applications that require ranking *e.g.* image or document retrieval, Natural Language Processing and Data Mining. In contrast, we are not interested in the ranking per-se, but rather use the ranking task to improve the learned representation.

Some approaches in the field *metric learning* use ranking losses to learn a feature embedding: Contrastive losses such as triplet loss (Weinberger, Blitzer, and Saul 2006) or N-pair loss (Sohn 2016) can be interpreted as ranking the positive higher w.r.t. the anchor than the negative. For instance, (Tschannen et al. 2020) use the triplet loss, to learn representations, but focus on learning invariances. (Ge 2018) learn a hierarchy from data for hard example mining to improve the triplet loss. Further, these approaches only consider two ranks, whereas our method can work with multiple ranks.

Methods

InfoNCE

We start with the most basic form of the *InfoNCE*. In this setting, two different views of the same data – *e.g.* two different augmentations of the same image – are pulled together in feature space, while pushing views of different samples apart. More specifically, for a query q, a single positive p and a set of negatives $\mathcal{N} = \{n_1, \dots n_k\}$ is given. The views are fed to an encoder network f, followed by a projection head g (Chen et al. 2020a). To measure the similarity between a pair of features we use the cosine similarity $\mathrm{cos\_sim}$. Overall the task is to train a critic $h(x, y) = \mathrm{cos\_sim}\left(g(f(x)), g(f(y))\right)$ using the loss:

$$\mathcal{L} = -\log \frac{\exp\left(\frac{h(q,p)}{\tau}\right)}{\exp\left(\frac{h(q,p)}{\tau}\right) + \sum\limits_{n\in\mathcal{N}} \exp\left(\frac{h(q,n)}{\tau}\right)}, \qquad (1)$$

where τ is a temperature parameter (Chen et al. 2020a). The above loss relies on the assumption that a single positive pair is available. One drawback with this approach is that all other samples are treated as negatives, even if they are semantically close to the query. Potential solutions include removing them from the negatives (Zhao et al. 2021) or adding them to the positives (Khosla et al. 2020), which we denote by $\mathcal{P} = \{p_1, \dots, p_l\}$. In other cases, we naturally have access to more than one positive, *e.g.* we can sample several clips from a single video, see Fig. 3. Having multiple positives per query leaves two options, which we discuss in the following.

$\mathrm{Log}_{\mathrm{out}}$ Positives. A straightforward approach to include multiple positives is to compute Eq. (1) for each of them, *i.e.* take the sum over positives outside of the $\log$. This enforces similarity between all positives during training, which suits a clean set of positives well.

$$\mathcal{L}^{\text{out}} = -\sum_{p\in\mathcal{P}} \log \frac{\exp\left(\frac{h(q,p)}{\tau}\right)}{\exp\left(\frac{h(q,p)}{\tau}\right) + \sum\limits_{n\in\mathcal{N}} \exp\left(\frac{h(q,n)}{\tau}\right)}. \qquad (2)$$

However, the set of positives can be noisy, *e.g.* sampling a temporally distant clip may include sub-optimal positives due to drastic changes in the video.

$\mathrm{Log}_{\mathrm{in}}$ Positives. An alternative approach, which is more robust to noise or inaccurate samples (Miech et al. 2020), is to take the sum inside the $\log$, Eq. (3). To minimize this loss, the network is not forced to set a high similarity to all pairs. It can neglect the noisy/false positives, given that a sufficiently large similarity is set for the true positives, see Tab. 4. However, if a discrepancy between positives exists, it results in a degenerate solution of discarding hard positives. For instance, consider supervised learning where both augmentations and class positives are available for a given query: the class positives, which are harder to optimize, can be ignored.

$$\mathcal{L}^{\text{in}} = -\log \frac{\sum\limits_{p\in\mathcal{P}} \exp\left(\frac{h(q,p)}{\tau}\right)}{\sum\limits_{p\in\mathcal{P}} \exp\left(\frac{h(q,p)}{\tau}\right) + \sum\limits_{n\in\mathcal{N}} \exp\left(\frac{h(q,n)}{\tau}\right)}. \qquad (3)$$

The above methods assume a binary set of positives and negatives. Thus, they can not exploit the similarity of positives and negatives. In the following, we discuss the proposed ranking version of InfoNCE that allows us to preserve the order of the positives and benefit from the additional information.

RINCE: Ranking InfoNCE

Let us assume that for a given query sample q, we have access to a set of ranked positives in a form of $\mathcal{P}_1, \dots, \mathcal{P}_r$, where $\mathcal{P}_i$ includes the positives of rank i. Let us also assume $\mathcal{N}$ is a set of negatives. Our objective is to train a critic h such that:

$$h(q, p_1) > \dots > h(q, p_r) > h(q, n) \quad \forall p_i \in \mathcal{P}_i, n \in \mathcal{N}. \qquad (4)$$

Note that $\mathcal{P}_i$ can contain multiple positives. For ease of notation we omit these indices. To impose the desired ranking presented by the positive sets, we use InfoNCE in a recursive manner where we start with the first set of positives, treat the remaining positives as negatives, drop the current positive, and move to the next. We repeat this procedure until there are no positives left. More precisely, the loss function reads $\mathcal{L}_{\text{rank}} = \sum_{i=1}^{r} \ell_i$, where

$$\ell_i = -\log \frac{\sum\limits_{p\in\mathcal{P}_i} \exp\left(\frac{h(q,p)}{\tau_i}\right)}{\sum\limits_{p\in\bigcup_{j\geq i}\mathcal{P}_j} \exp\left(\frac{h(q,p)}{\tau_i}\right) + \sum\limits_{n\in\mathcal{N}} \exp\left(\frac{h(q,n)}{\tau_i}\right)} \qquad (5)$$

Naming	# positives per rank	loss
RINCE-uni	single	Eq. (1)
RINCE-out	multiple	Eq. (2)
RINCE-in	multiple	Eq. (3)
RINCE-out-in	multiple	Eq. (2) (ℓ_1); Eq. (3) ($\ell_i, i > 1$)

Table 1: Different variants of RINCE. For the exact loss functions see the Sup. Mat.

and $\tau_i < \tau_{i+1}$. Eq. (5) denotes the $\mathcal{L}^{\text{in}}$ version of InfoNCE for positives of same rank; other variants are summarized in Tab. 1. The rational behind this loss is simple: The i-th loss is optimized when I) $\exp(h(q, p_i)/\tau_i) \gg 0$, II) $\exp(h(q, p_j)/\tau_i) \to 0$ for $j > i$ and III) $\exp(h(q, n)/\tau_i) \to 0$ for all i, j, n. I) and II) are competing across the losses: ℓ_i entails $\exp(h(q, p_{i+1})/\tau_i) \to 0$ but ℓ_{i+1} requires $\exp(h(q, p_{i+1})/\tau_{i+1}) \gg 0$. This requires the model to trade-off the respective loss terms, resulting in a ranking of positives $h(q, p_i) > h(q, p_{i+1})$.

In the following we explain the intuition behind our choice of τ values based on the analyses of (Wang and Liu 2021); for a more detailed analysis see Sup. Mat. A low temperature in the InfoNCE loss results in a larger relative penalty on the high similarity regions, *i.e.* hard negatives. As the temperature increases, the relative penalty distributes more uniformly, penalizing all negatives equally. A low temperature in ℓ_i allows the network to concentrate on forcing $h(q, p_i) > h(q, p_{i+1})$, ignoring easy negatives. A higher temperature on ℓ_r relaxes the relative penalty of negatives with respect to p_r so that the network can enforce $h(q, p_r) > h(q, n)$.

Experiments

We first study the properties of RINCE in the controlled supervised setting, looking at classification accuracy, retrieval and out-of-distribution (OOD) detection on Cifar-100. Next, we show that RINCE leads to significant improvements on the large scale dataset ImageNet-100 in terms of accuracy and OOD, even with more noisy similarity scores. Last, we showcase exemplary with unsupervised video representation learning that RINCE can be used in an unsupervised setting. For all experiments we follow the MoCo v2 setting (Chen et al. 2020b) with a momentum encoder, a memory bank and a projection head. Throughout the section we compare different versions of RINCE (Tab. 1), to study their behavior in different settings. More ablations in the Sup. Mat.

Learning from Class Hierarchies

The optimal testbed to study the proposed loss functions is the supervised contrastive learning (SCL) setting. The effect of the proposed loss functions can be studied without confounding noise, using ground truth labels and ground truth rankings. In SCL all samples with the same class are considered as positives, thus either Eq. (2), or Eq. (3) is used. However, semantically similar classes share similar visual features (Deselaers and Ferrari 2011). When strictly treated as negatives the model does not mirror the structure available by the labels in its feature space. This, however, is favorable for transferability to other tasks. RINCE allows the model to keep this structure, and learn not only dissimilarities between, but also similarities across classes. We show quantitatively that RINCE learns a higher quality representation than cross-entropy and SCL on Cifar-100 and ImageNet-100 by evaluating on linear classification, image retrieval, and OOD tasks. Unless otherwise stated, we report results for ResNet-50. More implementation details in the Sup. Mat.

Datasets. Cifar-100 (Krizhevsky, Hinton et al. 2009) provides both, class and superclass labels, defining a semantic hierarchy. We use this hierarchy to define first rank positives (same class) and second rank positives (same superclass).

TinyImageNet (Le and Yang 2015) comprises 200 ImageNet (Deng et al. 2009) classes at low resolution. ImageNet-100 (Tian, Krishnan, and Isola 2020) is a 100 class subset of ImageNet. We use the RoBERTa (Liu et al. 2019) model to obtain semantic word embeddings for all class names. Second rank positives are based on the word embedding similarity and a predefined threshold. Details in the Sup. Mat.

Baselines and SOTA. As baselines we use **cross-entropy**, cross-entropy with the same augmentations as RINCE (**cross-entropy s.a.**), **Triplet** loss (Weinberger, Blitzer, and Saul 2006) and **SCL** (Khosla et al. 2020), trained with Eq. (2) (**SCL-out**) or Eq. (3) (**SCL-in**). An advantage of RINCE compared to these baselines is that it benefits from extra information provided by the superclasses. To show that making use of this knowledge is not trivial, we compare to the following baselines: 1) We train SCL on Cifar-100 with 20 superclasses, denoted by **SCL superclass**. 2) **Hierarchical Triplet** (Ge 2018), which uses the superclasses to mine hard examples. 3) **Fast AP** (Cakir et al. 2019), a "learning to rank" approach that directly optimizes Average Precision. 4) **Label smoothing** (Szegedy et al. 2016), which reduces network over-confidence and can improve OOD detection (Lee and Cheon 2020). Here, we assign some probability mass to the classes from the same superclass. 5) A multi-classification baseline, referred to as **two heads**, that jointly predicts class and superclass labels. 6) **SCL two heads**, a variant of two heads, that uses the SCL loss instead of cross-entropy. Details for all baselines are given in the Sup. Mat.

Classification and Retrieval on Cifar. For the classification evaluation we train a linear layer on top of the last layer of the frozen pre-trained networks. The non-parametric retrieval evaluation involves finding the relevant data points in the feature space of the pre-trained network in terms of class labels via a simple similarity metric, *e.g.* cosine similarity. RINCE is superior to the baselines for all experiments, Tab. 2. Note, that all evaluations in Tab. 2 are based on the same pre-trained weights using Cifar-100 fine labels as rank 1 and, if applicable, superclass labels as rank 2.

These experiments indicate that training with RINCE maintains ranking order and results in a more structured feature space in which the samples of the same class are well separated from the other classes. This is further approved by a qualitative comparison between embedding spaces in Fig. 2.

Furthermore, we find that the grouping of classes is learned by the MLP head. The increased difficulty of the ranking task

Method	Cifar100 fine		Cifar100 superclass	AUROC	
	Accuracy	R@1	R@1	$\mathcal{D}_{\text{out}}$: Cifar-10	$\mathcal{D}_{\text{out}}$: TinyImageNet
SCL-out	76.50	N/A	N/A	N/A	N/A
Soft Labels$^\circ$	76.90	N/A	N/A	N/A	67.50
ODIN†	N/A	N/A	N/A	77.20	85.20
Mahalanobis†	N/A	N/A	N/A	77.50	97.40
Contrastive OOD‡	N/A	N/A	N/A	78.30	N/A
Gram Matrices	N/A	N/A	N/A	67.90	98.90
Cross-entropy*	74.52 ± 0.32	74.84 ± 0.21	83.99 ± 0.21	75.32 ± 0.65	77.76 ± 0.77
Cross-entropy s.a.*	75.46 ± 1.09	76.03 ± 1.04	84.68 ± 0.86	75.91 ± 0.10	79.44 ± 0.50
Triplet	68.44 ± 0.18	47.73 ± 0.14	72.29 ± 0.27	70.33 ± 0.54	80.76 ± 0.24
Hierarchical Triplet*	69.27 ± 1.64	65.31 ± 2.69	77.41 ± 1.55	71.97 ± 2.48	76.22 ± 1.27
Fast AP*	66.96 ± 0.88	62.03 ± 0.51	69.56 ± 0.54	69.14 ± 1.02	72.44 ± 0.94
Smooth Labels	75.66 ± 0.27	74.90 ± 0.06	85.59 ± 0.12	74.35 ± 0.65	80.10 ± 0.77
Two heads	74.08 ± 0.40	73.62 ± 0.31	81.92 ± 0.21	$\mathbf{77.99 \pm 0.07}$	78.35 ± 0.39
SCL-in superclass*	74.41 ± 0.15	69.83 ± 0.28	85.35 ± 0.51	74.40 ± 0.72	80.20 ± 1.05
SCL-in*	76.86 ± 0.18	73.20 ± 0.19	82.16 ± 0.24	74.63 ± 0.16	78.96 ± 0.45
SCL-out*	76.70 ± 0.29	74.45 ± 0.39	82.94 ± 0.39	75.32 ± 0.59	79.80 ± 0.70
SCL-in two heads*	77.15 ± 0.14	74.36 ± 0.10	83.31 ± 0.09	75.41 ± 0.16	79.34 ± 0.19
SCL-out two heads*	76.91 ± 0.08	74.87 ± 0.37	83.74 ± 0.16	75.27 ± 0.34	79.64 ± 0.53
Contrastive OOD	N/A	N/A	N/A	74.20 ± 0.40	N/A
RINCE-out	76.94 ± 0.16	76.68 ± 0.09	86.10 ± 0.25	$\underline{77.76 \pm 0.09}$	81.02 ± 0.14
RINCE-out-in	$\mathbf{77.59 \pm 0.21}$	$\underline{77.47 \pm 0.16}$	$\underline{86.20 \pm 0.23}$	76.82 ± 0.44	$\underline{81.40 \pm 0.38}$
RINCE-in	$\underline{77.45 \pm 0.05}$	$\mathbf{77.56 \pm 0.03}$	$\mathbf{86.46 \pm 0.21}$	77.03 ± 0.53	$\mathbf{81.78 \pm 0.05}$

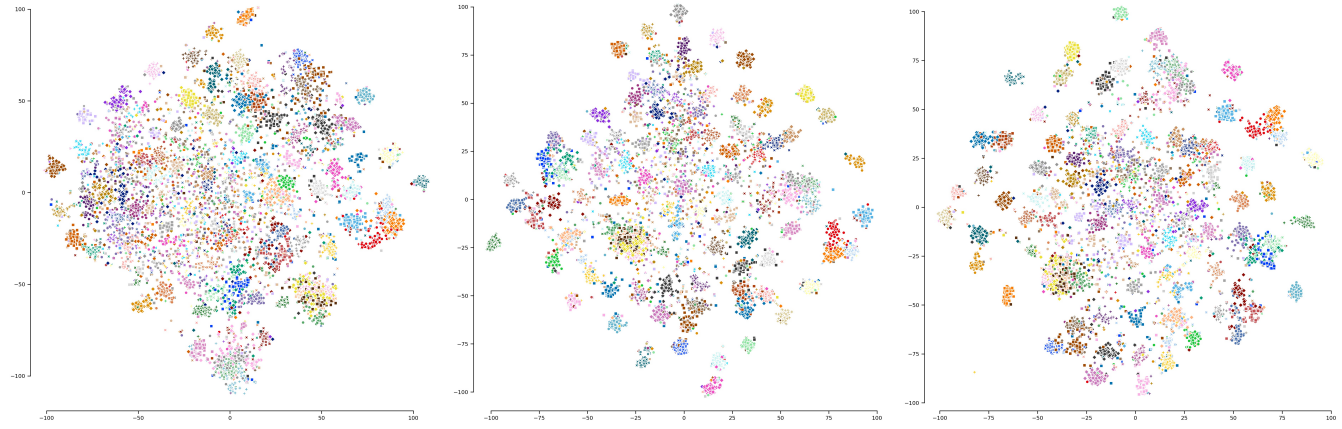
Table 2: Classification, retrieval and OOD results for Cifar-100 pretraining. Left: classification and retrieval; fine-grained task (fine) with 100 classes and superclass task (superclass) with 20 classes. Right: OOD task with inlier dataset $\mathcal{D}_{\text{in}}$: Cifar-100 and outlier dataset $\mathcal{D}_{\text{out}}$: Cifar-10 and TinyImageNet. We report the mean and standard deviation over 3 runs. Contrastive OOD averaged over 5 runs. Best method in bold, second best underlined. Note that, models indicated with † are not directly comparable, since they use data explicitly labeled as OOD samples for tuning. * indicates methods of others trained by us, $^\circ$ uses $2\times$ wider ResNet-40, ‡ $4\times$ wider ResNet-50. The lower part of the table uses ResNet-50. Methods not references in text: Soft Labels (Lee and Cheon 2020), Gram Matrices (Sastry and Oore 2020), Triplet (Weinberger, Blitzer, and Saul 2006).

of RINCE results in a more structured embedding space before the MLP compared with SCL, see Sup. Mat. Fig. 7.

Out-of-distribution Detection. To further investigate the structure of the learned representation of RINCE we evaluate on the task of out-of-distribution detection (OOD). As argued in (Winkens et al. 2020), models trained with cross-entropy only need to distinguish classes and can omit irrelevant features. Contrastive learning differs, by forcing the network to distinguish between each pair of samples, resulting in a more complete representation. Such a representation is beneficial for OOD detection (Hendrycks et al. 2019; Winkens et al. 2020). Therefore, OOD performance can be seen as evaluation of representation quality beyond standard metrics like accuracy and retrieval. RINCE incentivizes the network to learn an even richer representation. Besides that, OOD benefits from good trade-off between alignment and uniformity, which RINCE manages well (Fig. 9 in Sup. Mat.).

We follow common evaluation settings for OOD (Lee et al. 2018; Liang, Li, and Srikant 2018; Winkens et al. 2020). Here Cifar-100 is used as the inlier dataset $\mathcal{D}_{\text{in}}$, Cifar-10 and TinyImageNet as outlier dataset $\mathcal{D}_{\text{out}}$. Note that Cifar-100 and Cifar-10 have disjoint labels and images. For both protocols we only use the test or validation images. Our models are identical to those in the previous section. Inspired by (Winkens et al. 2020), we follow a simple approach, and fit class-conditional multivariate Gaussians to the embedding of the training set. We use the log-likelihood to define the OOD-score. As a result, the likelihood to identify OOD-samples is high, if each in-class follows roughly a Gaussian distribution in the embedding space, compare Fig. 2a and 2c. For evaluation, we compute the area under the receiver operating characteristic curve (AUROC), details in Sup. Mat.

Results and a comparison to the most related previous work is shown in Tab. 2. Note that we aim here to compare the learned representation space via RINCE to its counterparts, *i.e.* cross-entropy and SCL, but show well known methods as reference. Most importantly, RINCE clearly outperforms cross-entropy, all SCL variants, contrastive OOD and our own baselines using the identical OOD approach. Only two-heads outperforms all other methods in the near OOD setting with $\mathcal{D}_{\text{out}}$: Cifar-10. However, performance on all other settings is low, showing weak generalization. This underlines our hypothesis, that training with RINCE yields a more structured and general representation space. Comparing to related works, RINCE not only outperforms Contrastive OOD (Winkens et al. 2020) using the same architecture, but even approaches the $4\times$ wider ResNet on Cifar-10 as $\mathcal{D}_{\text{out}}$. ODIN (Liang, Li, and Srikant 2018) and Mahalanobis (Lee et al. 2018) require samples labelled as OOD to tune parameters of the OOD approach. Here we evaluate in the more



(a) (b) (c)

Figure 2: Qualitative comparison of embedding spaces. T-SNE plot of (a) supervised contrastive learning (*SCL-in*) and (b) *RINCE-in* (c) *RINCE-out-in* on Cifar-100. Best seen in color, on screen and zoomed in. Color and marker type combined indicate class. Labels omitted for clarity. Sup. Mat. contains a version of this plot with color indicating the superclass. RINCE learns a more structured embedding space than SCL, e.g. classes are linearly separable and can be modelled well by a Gaussian.

Method	Accuracy	AUROC $\mathcal{D}_{\text{out}}$: ImageNet-100†	AUROC $\mathcal{D}_{\text{out}}$: AwA2
Cross-entropy s.a.	83.94	79.076 $\pm$ 1.477	79.04
SCL-out	84.18	79.779 $\pm$ 1.274	79.05
RINCE-out-in	**84.90**	**80.473 $\pm$ 1.210**	**80.73**

Table 3: ImageNet-100 classification accuracy and OOD detection for $\mathcal{D}_{\text{in}}$: ImageNet-100, and $\mathcal{D}_{\text{out}}$: ImageNet-100† and AwA2 (Xian et al. 2018). ImageNet-100† denotes three ImageNet-100 datasets with non-overlapping classes.

realistic setting without labelled OOD samples. Despite using significantly less information, RINCE is compatible with them and even outperforms them for $\mathcal{D}_{\text{out}}$: Cifar-10.

Large Scale Data and Noisy Similarities

Additionally, we perform the same evaluations on ImageNet-100, a 100-class subset of ImageNet, see Tab. 3. Here, we use ResNet-18. We obtain the second rank classes for a given class via similarities of the RoBERTa (Liu et al. 2019) class name embeddings. In contrast to the previous experiments, where ground truth hierarchies are known, these similarity scores are noisy and inaccurate – yet it still provides valuable information to the model. We evaluate our model via linear classification on ImageNet-100 and two OOD tasks: AwA2 (Xian et al. 2018) as $\mathcal{D}_{\text{out}}$ and ImageNet-100†, where we use the remaining ImageNet classes to define three non-overlapping splits and report the average OOD.

Result are shown in Tab. 3. Again, RINCE significantly improves over SCL and cross-entropy in linear evaluation as well as on the OOD tasks. This demonstrates 1) that RINCE can handle noisy rankings and 2) that RINCE leads to improvements on large scale datasets. Next, we move to an even less controlled setting and define a ranking based on temporal ordering for unsupervised video representation learning.

Unsupervised RINCE

In this section we demonstrate that RINCE can be used in a fully unsupervised setting with noisy hierarchies by applying it to unsupervised video representation. Inspired by (Tschannen et al. 2020), we construct three ranks for a given query video, same frames, same shot and same video, see Fig. 3.

The first positive x_f is obtained by augmenting the query frames. The second positive x_s is a clip consecutive to the query frames, where small transformations of the objects, illumination changes, etc. occur. The third positive x_v is sampled from a different time interval of the same video, which may show visually distinct but semantically related scenes. Naturally, x_f shows the most similar content to the query frames, followed by x_s and finally x_v. We compare temporal ranking with RINCE to different baselines.

Baselines. We compare to the basic **InfoNCE**, where a single positive is generated via augmentations (Chen et al. 2020a; He et al. 2020), *i.e.* only *frame* positives x_f. When considering multiple clips from the same video such as x_s and x_v, there are several possibilities: We can treat them all as positives (**hard positive**), we can use the distant x_v as a **hard negative** or ignore it (**easy positive**). In both cases $\mathcal{L}^{\text{out}}$, Eq. (2), and $\mathcal{L}^{\text{in}}$, Eq. (3), are possible. Additionally, we compare to two recent methods trained in comparable settings, *i.e.* VIE (Zhuang et al. 2020), LA-IDT (Tokmakov, Hebert, and Schmid 2020).

Ranking Frame-, Shot- and Video-level Positives. We sample short clips of a video, each consisting of 16 frames. We augment each clip with a set of standard video augmentations. For more details we refer to the Sup. Mat. For the anchor clip x, we define positives as in Fig. 3: $p_1 = x_f$ con-

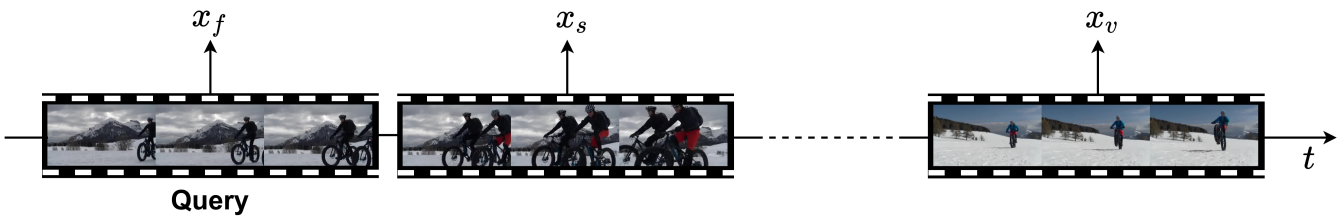


Figure 3: Positives in Videos. For a given query clip we use *frame* positives x_f, *shot* positives x_s and *video* positives x_v.

Method	Loss	Positives	Negatives	Top 1 Accuracy HMDB	Top 1 Accuracy UCF	Retrieval mAP HMDB	Retrieval mAP UCF
VIE	-	-	-	44.8	72.3	-	-
LA-IDT	-	-	-	44.0	72.8	-	-
InfoNCE	$\mathcal{L}$	$\{x_f\}$	$\mathcal{N}$	41.5	71.3	0.0500	0.0688
hard positive	$\mathcal{L}^{\text{in}}$	$\{x_f, x_s, x_v\}$	$\mathcal{N}$	42.6	74.3	0.0685	0.1119
	$\mathcal{L}^{\text{out}}$	$\{x_f, x_s, x_v\}$	$\mathcal{N}$	41.4	73.6	0.0666	0.1204
easy positive	$\mathcal{L}^{\text{in}}$	$\{x_f, x_s\}$	$\mathcal{N}$	42.7	74.5	0.0581	0.1257
	$\mathcal{L}^{\text{out}}$	$\{x_f, x_s\}$	$\mathcal{N}$	40.7	73.5	0.0593	0.1297
hard negative	$\mathcal{L}^{\text{in}}$	$\{x_f, x_s\}$	$\{x_v\} \cup \mathcal{N}$	43.6	74.3	0.0678	0.1141
	$\mathcal{L}^{\text{out}}$	$\{x_f, x_s\}$	$\{x_v\} \cup \mathcal{N}$	43.5	75.2	0.0675	0.1193
RINCE	RINCE-uni	$x_f > x_s > x_v$	$\mathcal{N}$	**44.9**	**75.4**	**0.0719**	**0.1395**

Table 4: Finetuning on UCF and HMDB. $\mathcal{L}$, $\mathcal{L}^{\text{in}}$ and $\mathcal{L}^{\text{out}}$ correspond to Eq. (1), Eq. (3) and Eq. (2), respectively. *Positives* and *Negatives* indicates how x_f, x_s, x_v were incorporated into contrastive learning, where $\mathcal{N}$ denotes the set of negative pairs from random clips. Since we consider only a single positive per rank we use the RINCE-uni loss variant for RINCE.

sists of the same frames as x, $p_2 = x_s$ is a sequence of 16 frames adjacent to x, and $p_3 = x_v$ is sampled from a different time interval than x_f and x_s. Negatives x_n are sampled from different videos. Since each rank i contains only a single positive p_i, Eq. (2) = Eq. (3), we call this variant RINCE-uni. By ranking the positives we ensure that the similarities satisfy $\text{sim}(x, x_f) > \text{sim}(x, x_s) > \text{sim}(x, x_v) > \text{sim}(x, x_n)$, adhering to the temporal structure in videos.

Datasets and Evaluation. For self-supervised learning, we use Kinetics-400 (Kay et al. 2017) and discard the labels. Our version of the dataset consists of 234.584 training videos. We evaluate the learned representation via finetuning on UCF (Soomro, Zamir, and Shah 2012) and HMDB (Kuehne et al. 2011) and report top 1 accuracy. In this evaluation, the pretrained weights are used to initialize a network and train it end-to-end using cross-entropy. Additionally, we evaluate the representation via nearest neighbor retrieval and report mAP. Precision-Recall curves can be found in the Sup. Mat.

Experimental Results. For all experiments we use a 3D-ResNet-18 backbone. Training details can be found in the Sup. Mat. We report the results for RINCE as well as the baselines in Tab. 4. Adding shot- and video-level samples to InfoNCE improves the downstream accuracies. We observe that adding x_v to the set of negatives to provide a hard negative rather than adding it to the set of positives leads to higher performance, suggesting that this should not be a true positive. This is further supported by the second and third row, where all three positives are treated as true positives. Here, $\mathcal{L}^{\text{out}}$, which forces all positives to be similar, leads to inferior performance compared to $\mathcal{L}^{\text{in}}$. $\mathcal{L}^{\text{in}}$ allows more noise in the set of positives by weak influence of false positives x_v. With RINCE we can impose the temporal ordering $x_f > x_s > x_v$ and treat x_v properly, leading to the highest downstream performance. Improvements of RINCE over $\mathcal{L}^{\text{out}}$ is less pronounced on UCF. This is due to the strong static bias (Li, Li, and Vasconcelos 2018) of UCF and $\mathcal{L}^{\text{out}}$ encourages static features. Contrarily, improvements of RINCE over $\mathcal{L}^{\text{out}}$ on HMDB are substantial, due to the weaker bias towards static features. Last, we compare our method to two recent unsupervised video representation learning methods that use the same backbone network in Tab. 4. We outperform these methods on both datasets.

Conclusion

We introduced RINCE, a new member in the family of InfoNCE losses. We show that RINCE can exploit rankings to learn a more structured feature space with desired properties, lacking with standard InfoNCE. Furthermore, representations learned through RINCE can improve accuracy, retrieval and OOD. Most importantly, we show that RINCE works well with noisy similarities, is applicable to large scale datasets and to unsupervised training. We compare the different variants of RINCE. Here lies a limitation: Different variants are optimal for different tasks and must be chosen based on domain knowledge. Future work will explore further applications of obtaining similarity scores, *e.g.* based on distance in a pretrained embedding space, distance between cameras in a multi-view setting or distances between clusters.

Acknowledgments

JG has been supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - GA1927/4-2.

References

Behrmann, N.; Gall, J.; and Noroozi, M. 2021. Unsupervised Video Representation Learning by Bidirectional Feature Prediction. In *WACV*.

Burges, C.; Shaked, T.; Renshaw, E.; Lazier, A.; Deeds, M.; Hamilton, N.; and Hullender, G. 2005. Learning to Rank Using Gradient Descent. In *ICML*.

Cakir, F.; He, K.; Xia, X.; Kulis, B.; and Sclaroff, S. 2019. Deep metric learning to rank. In *CVPR*.

Cao, Z.; Qin, T.; Liu, T.-Y.; Tsai, M.-F.; and Li, H. 2007. Learning to rank: from pairwise approach to listwise approach. In *ICML*.

Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; and Joulin, A. 2020. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *NeurIPS*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020a. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*.

Chen, X.; Fan, H.; Girshick, R.; and He, K. 2020b. Improved baselines with momentum contrastive learning. *arXiv:2003.04297*.

Chen, X.; and He, K. 2021. Exploring simple siamese representation learning. In *CVPR*.

Dave, I.; Gupta, R.; Rizve, M. N.; and Shah, M. 2021. TCLR: Temporal Contrastive Learning for Video Representation. *arXiv preprint arXiv:2101.07974*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

Deselaers, T.; and Ferrari, V. 2011. Visual and semantic similarity in imagenet. In *CVPR*.

Dosovitskiy, A.; Fischer, P.; Springenberg, J. T.; Riedmiller, M.; and Brox, T. 2016. Discriminative Unsupervised Feature Learning with Exemplar Convolutional Neural Networks. In *TPAMI*.

Feichtenhofer, C.; Fan, H.; Xiong, B.; Girshick, R.; and He, K. 2021. A Large-Scale Study on Unsupervised Spatiotemporal Representation Learning. In *CVPR*.

Ge, W. 2018. Deep metric learning with hierarchical triplet loss. In *ECCV*.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; Piot, B.; Kavukcuoglu, K.; Munos, R.; and Valko, M. 2020. Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning. In *NeurIPS*.

Han, T.; Xie, W.; and Zisserman, A. 2020. Self-supervised Co-training for Video Representation Learning. In *NeurIPS*.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR*.

Hendrycks, D.; Mazeika, M.; Kadavath, S.; and Song, D. 2019. Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty. *NeurIPS*.

Hoffmann, D. T.; Behrmann, N.; Gall, J.; Brox, T.; and Noroozi, M. 2022. Ranking Info Noise Contrastive Estimation: Boosting Contrastive Learning via Ranked Positives. *arXiv preprint arXiv:2201.11736*.

Huynh, T.; Kornblith, S.; Walter, M. R.; Maire, M.; and Khademi, M. 2020. Boosting Contrastive Self-Supervised Learning with False Negative Cancellation. *arXiv preprint arXiv:2011.11765*.

Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; Suleyman, M.; and Zisserman, A. 2017. The Kinetics Human Action Video Dataset. *arXiv*, abs/1705.06950.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised Contrastive Learning. *NeurIPS*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. *Tech Report*.

Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; and Serre, T. 2011. HMDB: A large video database for human motion recognition. In *ICCV*.

Le, Y.; and Yang, X. 2015. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7): 3.

Lee, D.; and Cheon, Y. 2020. Soft Labeling Affects Out-of-Distribution Detection of Deep Neural Networks. *arXiv preprint arXiv:2007.03212*.

Lee, K.; Lee, K.; Lee, H.; and Shin, J. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *NeurIPS*.

Li, Y.; Li, Y.; and Vasconcelos, N. 2018. RESOUND: Towards Action Recognition without Representation Bias. In *ECCV*.

Liang, S.; Li, Y.; and Srikant, R. 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*.

Liu, T.-Y. 2009. Learning to Rank for Information Retrieval. *Foundations and Trends® in Information Retrieval*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Miech, A.; Alayrac, J.-B.; Smaira, L.; Laptev, I.; Sivic, J.; and Zisserman, A. 2020. End-to-End Learning of Visual Representations from Uncurated Instructional Videos. In *CVPR*.

Misra, I.; and van der Maaten, L. 2020. Self-Supervised Learning of Pretext-Invariant Representations. In *CVPR*.

Neill, J. O.; and Bollegala, D. 2021. Semantically-Conditioned Negative Samples for Efficient Contrastive Learning. *arXiv preprint arXiv:2102.06603*.

Romijnders, R.; Mahendran, A.; Tschannen, M.; Djolonga, J.; Ritter, M.; Houlsby, N.; and Lucic, M. 2021. Representation Learning From Videos In-the-Wild: An Object-Centric Approach. In *WACV*.

Sastry, C. S.; and Oore, S. 2020. Detecting out-of-distribution examples with gram matrices. In *ICML*.

Sohn, K. 2016. Improved Deep Metric Learning with Multi-class N-pair Loss Objective. In *NIPS*.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv*, abs/1212.0402.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *CVPR*.

Tian, Y.; Krishnan, D.; and Isola, P. 2020. Contrastive Multiview Coding. In *ECCV*.

Tian, Y.; Sun, C.; Poole, B.; Krishnan, D.; Schmid, C.; and Isola, P. 2020. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*.

Tokmakov, P.; Hebert, M.; and Schmid, C. 2020. Unsupervised Learning of Video Representations via Dense Trajectory Clustering. In *ECCV Workshops*.

Tschannen, M.; Djolonga, J.; Ritter, M.; Mahendran, A.; Houlsby, N.; Gelly, S.; and Lucic, M. 2020. Self-Supervised Learning of Video-Induced Visual Invariances. In *CVPR*.

van den Oord, A.; Li, Y.; and Vinyals, O. 2018. Representation Learning with Contrastive Predictive Coding. *arXiv*, abs/1807.03748.

Wang, F.; and Liu, H. 2021. Understanding the Behaviour of Contrastive Loss. In *CVPR*.

Weinberger, K. Q.; Blitzer, J.; and Saul, L. 2006. Distance Metric Learning for Large Margin Nearest Neighbor Classification. In *NIPS*.

Winkens, J.; Bunel, R.; Roy, A. G.; Stanforth, R.; Natarajan, V.; Ledsam, J. R.; MacWilliams, P.; Kohli, P.; Karthikesalingam, A.; Kohl, S.; et al. 2020. Contrastive training for improved out-of-distribution detection. *arXiv preprint arXiv:2007.05566*.

Xian, Y.; Lampert, C. H.; Schiele, B.; and Akata, Z. 2018. Zero-Shot Learning - A Comprehensive Evaluation of the Good, the Bad and the Ugly. In *TPAMI*.

Zhao, N.; Wu, Z.; Lau, R. W. H.; and Lin, S. 2021. What Makes Instance Discrimination Good for Transfer Learning? In *ICLR*.

Zhuang, C.; She, T.; Andonian, A.; Mark, M. S.; and Yamins, D. 2020. Unsupervised Learning From Video With Deep Neural Embeddings. In *CVPR*.