Exploit Clues from Views:
Self-Supervised and Regularized Learning for Multiview Object Recognition

Chih-Hui Ho   Bo Liu   Tz-Ying Wu   Nuno Vasconcelos

University of California, San Diego

{chh279,boliu,tzw001,nvasconcelos}@ucsd.edu

Abstract

Multiview recognition has been well studied in the literature and achieves decent performance in object recognition and retrieval task. However, most previous works rely on supervised learning and some impractical underlying assumptions, such as the availability of all views in training and inference time. In this work, the problem of multiview self-supervised learning (MV-SSL) is investigated, where only image to object association is given. Given this setup, a novel surrogate task for self-supervised learning is proposed by pursuing "object invariant" representation. This is solved by randomly selecting an image feature of an object as object prototype, accompanied with multiview consistency regularization, which results in view invariant stochastic prototype embedding (VISPE). Experiments shows that the categorization and retrieval results using VISPE outperform that of other self-supervised learning methods on seen and unseen data. VISPE can also be applied to semi-supervised scenario and demonstrates robust performance with limited data available. Code is available at https://github.com/chihhuiho/VISPE

1. Introduction

3D recognition has received increasing attention in computer vision in recent years. A popular approach, which we pursue in this work, is to rely on the multiview object representation. Several multiview recognition approaches have been proposed in the literature, including the use of recurrent neural networks [9, 19, 39], feature aggregation from different views [14, 15, 58], graph modeling [13] and integration with other modalities [22, 45, 50, 68]. While achieving good recognition performance, two strong assumptions are made. The first is that a dense set of views, covering the entire range of view angles, is available per object [58]. While some methods support missing views during inference [14, 23, 30], a complete view set is always assumed for training. The second is that all these images are la-
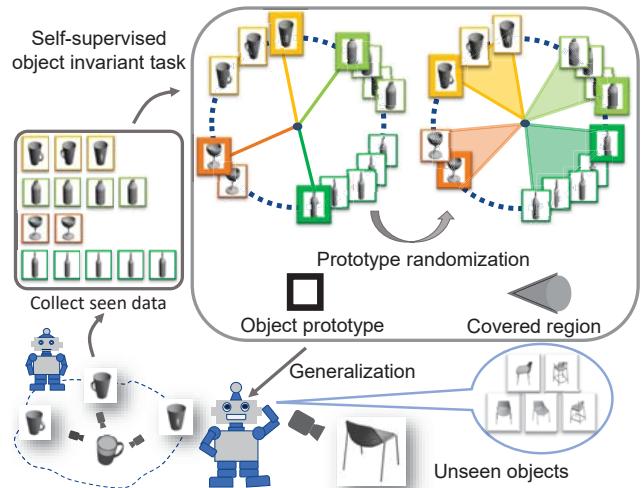
Figure 1: Lightweight unsupervised multiview object recognition. A household robot collects multiple object views by moving around, aggregating a multiview object database without view labels. A self-supervised learning algorithm is applied to this database to create an embedding that maps images from same object into an object invariant. At inference time, this embedding generalizes to new views, objects, and object classes.

beled, for both object classes and view angle. The two assumptions make multiview techniques difficult to implement and limit their generalization. For example, while previous works [14, 30, 58] show strong performance on training classes, recognition on classes unseen during training is usually not considered.

These limitations prevent many applications of interest. Consider, for example, the setting of Fig. 1, where a household robot of limited memory is tasked with picking scattered objects and returning them to their locations. In this setting, it is impractical to pre-train the robot with a dense set of labelled views for each object class in the world. Instead, the robot must be able to efficiently learn objects from unseen classes after deployment. This is similar to problems like image retrieval [10, 29] or face verification [52, 60],

which are usually solved by metric learning. An embedding is learned from a large dataset of annotated objects, unseen object classes are modelled by projecting example images onto the embedding, and classification is performed with a nearest neighbor classifier. However, a multi-view embedding is challenging to learn in this manner, due to the need for complete and labeled sets of views. In the setting of Fig. 1 this means that, after the home robot is deployed, view angle labels must be collected by manually controlling the pose of the training objects, which is impractical.

This problem can be avoided by the introduction of *multiview self-supervised learning* (MV-SSL) methods. SSL is now well established for problems where annotation is difficult [27,28]. The idea is to use "free labels," i.e. annotations that can be obtained without effort, to define a surrogate learning task. However, the many surrogate tasks proposed in the literature [2, 34, 36, 49, 67, 69] are poorly suited for multiview recognition. This is because multiview embeddings must enforce an *invariance* constraint, namely that all views of an object map into (or cluster around) a single point in the embedding, which is denoted the *object invariant*. For embeddings with this property, views of objects unseen during training will naturally cluster around object invariants, without requiring view labels, consistency of view angles across objects, or even the same number of views per object. In this case, it suffices for the home robot to collect a set of views per object, e.g. by moving around it, as illustrated in Fig. 1. To emphasize the low-complexity of object acquisition under this set-up, we refer to it as *lightweight unsupervised multiview object recognition* (LWUMOR).

In this work, we seek embeddings with good LWUMOR performance. We consider proxy embeddings [43], which have been shown to perform well for multiview recognition when dense views and class labels are available [23]. To derive an SSL extension, we propose a new surrogate task, where object instances are used as training "classes," i.e. object identities serve as free labels for learning. We hypothesize, however, that due to the concentration of supervision on class prototypes, these embeddings *only* capture the metric structure of images in the neighborhood of these prototypes, thus overfitting to the training classes. We address this problem with a randomizing procedure, where the parameters of the softmax layer are sampled stochastically from the embeddings of different object views, during training. This has two interesting consequences. First, it forces the learning algorithm to produce an embedding that supports many classifiers, spreading class supervision throughout a much larger region of feature space, and enhancing generalization beyond the training classes. Second, because this supervision is derived from randomized object views, it encourages a stable multiview representation, even when only different view subsets are available per object.

To further enhance multiview recognition performance,

this randomization is complemented by an explicit invariance constraint, which encourages the classifier parameters to remain stable under changes of view-point. We denote the resulting MV-SSL embeddings as *view invariant stochastic prototype embeddings (VISPE)*. Experimental results on popular 3D recognition datasets show that self-supervised VISPE embeddings combine 1) better performance outside the training set than standard classification embeddings, and 2) faster convergence than metric learning embeddings. Furthermore, for multiview recognition, VISPE embeddings outperform previous SSL methods.

Overall, this work makes three main contributions. The first is the LWUMOR formulation of MV-SSL. This enables multiview recognition without object class or pose labels, and generalizes well to objects unseen at training time. The second is a new surrogate task that relies on randomization of object views to encourage stable multiview embeddings, and outperforms previous SSL surrogates for multitview recognition. The third is the combination of randomization and invariance constraints implemented by VISPE to learn embeddings of good LWUMOR performance. Extensive experiments validate the ability of these embeddings to learn good invariants for multiview recognition.

2. Related work

This work is related to multiview recognition, SSL, and regularization by network randomization.

2.1. Multiview recognition

Multiview recognition is a 2D image-based approach to 3D object recognition. One of the earliest methods is the multiview CNN (MVCNN) [58], which takes multiple images of an object as input and performs view aggregation in the feature space to obtain a shape embedding. Representing 3D objects by 2D images has been shown effective for classification [14, 30] and retrieval [21, 38]. Subsequent research extended the idea by performing hierarchical view aggregation [14, 15]. However, because view aggregation disregards the available supervision for neighboring relationships between views [19], recurrent neural networks [9, 19, 39] and graph convolutional neural networks [13] have been proposed to model multiview sequences. Aside from multiview modeling, [30] treats viewpoint as a latent variable during optimization and achieves better classification accuracy and pose estimation. In the retrieval setting, [21, 38] combine the center loss [64] with a triplet loss [52] to form compact clusters for features from the same object class.

All these methods share several assumptions that make them impractical for LWUMOR. The MVCNN [58] assumes that all object views are presented at both training and inference. Methods that model view sequences [9, 19] require even more detailed viewpoint supervision. Previ-

ous works [14, 23, 30] found that these methods experience a significant performance drop when only partial views are available for inference. [30] minimized this drop by treating viewpoint as an intermediate variable, while [23] proposed to overcome it with hierarchical multiview embeddings. All these methods assume a full set of training views.

In this work, we relax this constraint, investigating the LWUMOR setting, where only partial object views are available for both training and inference, and no view or image class labels are given. This forces the use of SSL techniques to learn the "implicit" shape information present in a set of object views, and encourages embeddings that generalize better to unseen classes.

2.2. Self-supervised Learning

SSL leverages free labels for a surrogate task to train a deep network. Many surrogate tasks have been proposed in the literature. While we provide a brief review of many of these in what follows, most do not seek object invariants and are unsuitable for MV-SSL.

Context based approaches [12, 44, 49] seek to reconstruct images. Autoencoders [17] map images to a low dimensional latent space, from which they can be reconstructed. Similarly, a context encoder [49] reconstructs missing patches from an image conditioned on their surroundings. [12] further leverages spatial image context by predicting the relative positions of randomly cropped patches. Image coloring techniques, which recover the colors of grey scale images [69] or predict pixelwise hue and chromatic distributions [34, 35] leverage color as a form of image context.

Motion based approaches [2, 26, 48, 63] exploit the spatiotemporal coherence of images captured by a moving agent, in terms of relative position [2], optical flow [48] or temporal video structure [25, 63]. The surrogate task becomes to predict camera transformations [2] or segmenting objects [48].

Sequence sorting is another popular task, where sequences can consist of randomly cropped image patches [46] and video clips [3, 42]. Similarly, there have been proposals to remove some color channels from image patchs [31] or adding various types of jitter to video clips [36, 61, 62].

Data augmentation type of tasks transform images, leveraging the difference after transformation to define surrogate tasks. [16] predicts the rotation angle of the transformed image, while [67] learns a transformation invariant feature.

View based tasks have been proposed for multiple applications, such as object recognition [24], hand [59] and human [32] pose estimation. Our work is similar to [24] as both consider object recognition. However, [24] requires image sequences while our approach has no such constraint and tackles the problem in an entirely different manner .

Cluster based methods group data with visual similarities into clusters and discriminate different clusters. While [5,6]

group multiple images into the same cluster, [1, 65] treat each image as a cluster. Our work shares the high level idea of the latter, by treating each object as a cluster, but differs in terms of memory usage and efficiency. While [1] is known to be computational demanding, [65] is both memory expensive and inefficient, by requiring storage of features from *all* dataset instances. Furthermore, each feature is updated only once per epoch, which leads to noisy optimization. Our method leverages multiple object views to avoid these problems and is more suitable for the LWU-MOR setting.

2.3. Network randomization

Randomization has been shown to improve network performance [57] and robustness [66]. It can be explained as a form or model ensembling [33, 37], by combining models trained under different conditions. One of the simplest yet most practical randomization procedures is dropout [4, 57], which removes units in the network during training. Dropmax [37] proposed to instead remove classes, training a stochastic variant of the softmax for better classification. The proposed method explores an orthogonal randomization direction, where feature vectors from different object views are chosen as object prototypes during training.

3. Multiview Self Supervised Learning

In this section, we discuss the proposed MV-SSL approach.

3.1. Light Weight Unsupervised Multiview Object Recognition

We start by defining the LWUMOR problem and introducing a surrogate task for its solution. Consider a set of objects $\mathcal{O} = \{o_i\}_{i=1}^N$, where $o_i \in \mathcal{O}$ is the ith object instance. This consists of a set $o_i = \{x_i^j\}_{j=1}^{V_i}$ of variable V_i image views, captured from *unspecified* viewpoints. $x_i^j \in \mathcal{X}$ denotes the jth view of object o_i. The goal of LWUMOR is to learn an embedding that supports recognition of new views, objects, and object classes from $\mathcal{O}$. In this work, this is addressed with SSL, defining the surrogate task as object instance classification. Each object instance is treated as a different class, establishing a labelled image dataset $\mathcal{D} = \{(x_i^j, y_i^j) \mid y_i^j = i, \forall j \in V_i\}$. The surrogate task is solved by a classifier based on an embedding $f_\theta : \mathcal{X} \to \mathbb{R}^k$ of parameters θ, which maps image x into k-dimensional feature vector $f_\theta(x)$. This is implemented by a convolutional neural network (CNN). It should be emphasized that this surrogate task requires *no view alignment or labels.* This is unlike previous multiview SSL approaches, which require either view [24] or camera transformation labels [2].
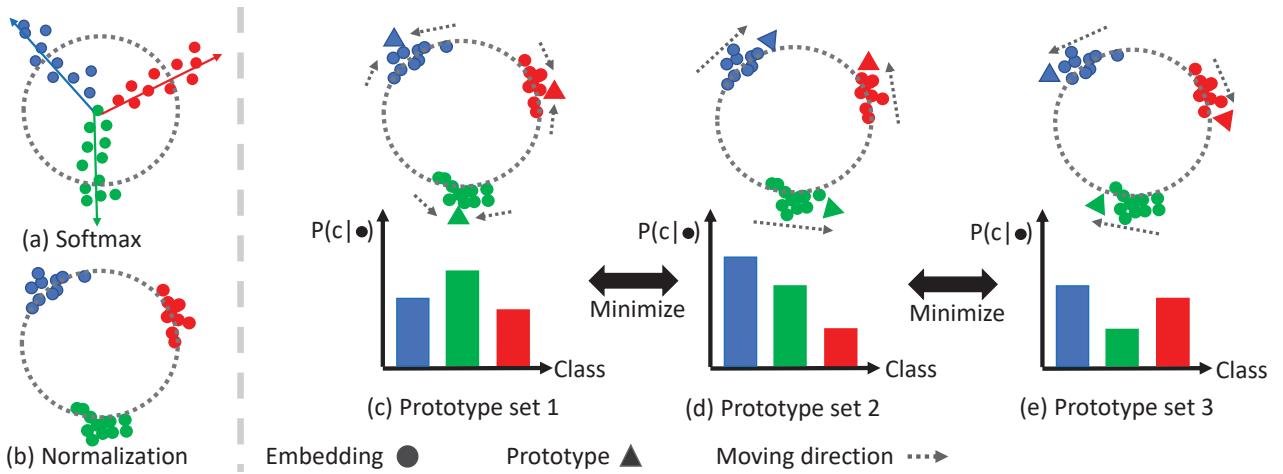
Figure 2: Regularization of a self-supervised embedding by prototype randomization. In all figures, each color represents a single object and views of the same object are marked with same color. (a) softmax embedding, unnormalized features. Solid arrows represent the weight vectors w_i learned per instance i. (b) Normalized embedding. (c-e) randomization: 3 different sets of prototypes are used for training. Dashed lines show how view embeddings are encouraged to move towards the corresponding object prototype. Bar plots illustrate how the posterior class probabilities of a given image change when the prototyopes are switched. The proposed multiview consistency regularization seeks to further improve the generalization power of the embedding by minimizing these variations.

3.2. Modeling

As is common for CNNs, a classifier can be implemented with a softmax layer

$$P_{Y|X}(i|x) = \frac{\exp(w_i^T f_\theta(x))}{\sum_{k=1}^{N} \exp(w_k^T f_\theta(x))}, \quad (1)$$

where w_i is the parameter vector of instance i. This is referred as the "instance classifier" in what follows and is trained by cross-entropy minimization, using the free instance labels for supervision. The optimal embedding and classifier parameters are learned by minimizing the risk

$$\mathcal{R} = \sum_{i,j} -\log P_{Y|X}(i|x_i^j) \quad (2)$$

over the image dataset $\mathcal{D}$.

Even though it is a strong baseline, the softmax classifier is most successful for closed-set classification, where train and test object classes are the same. In general, the learned embedding f_θ does not have a good metric structure beyond these classes. For this reason, alternative approaches have been more successful for open set problems, such as image retrieval [10], face identification [52, 60], or person re-identification [70]. These are usually based on metric learning embeddings, such as pairwise [18] or triplet embeddings [52]. However, these techniques have problems of their own. Because there are many more example pairs or triplets than single examples in $\mathcal{D}$, they require sampling techniques that are not always easy to implement and lead to slow convergence.

3.3. Randomization

In this work, we explore an alternative approach to learn embeddings that generalize beyond the set of training classes, based on the softmax classifier of (1). This consists of randomizing the surrogate task and is inspired by previous work in low-shot learning [54], where meta-learning techniques re-sample the classes for which the embedding is trained. The intuition is that, when the task is changed, the metric structure of the embedding changes as well. This forces the embedding to have a good metric structure over larger regions of the feature space, therefore generalizing better to unseen classes. In this work, we consider randomization strategies that leverage the view richness of multiview datasets to achieve better generalization to unseen classes during training. This is critical in the LWUMOR setting, where the goal is to enable the learning of multiview embeddings without dense view datasets or even view labels. We propose to randomize the embedding by using random feature vectors as classifier parameters w_i in (1).

The idea is summarized in Fig. 2 for a problem where $N = 3$. Fig. 2 (a) shows the vectors w_i (solid arrows) learned in feature space with the combination of (1) and (2). Since cross-entropy minimization only aims to separate the seen instances, this embedding leads to feature distributions such as shown in Fig. 2 (a). Embeddings of images of the same object are not tightly clustered and can be close to those from other objects. A common procedure to encourage better metric structure (in this case Euclidean) is to normalize the embedding to unit norm [43, 51, 52, 60], i.e.

Algorithm 1 Randomization schedule

1: **Input** Threshold t
2: Use the view samplers $\nu_i, \forall i$ to select a set of random prototypes $\mathcal{W} = \{f_\theta(x_1^{\nu_1}), \ldots, f_\theta(x_N^{\nu_N})\}$ to use in (3).
3: **while** Not convergence **do**
4:    Minimize the risk of (2)
5:    **for all** $i \in N$ **do**
6:       $u \sim Unif(0,1)$
7:       **if** $u < t$ **then**
8:          Use ν_i to resample a new prototype $f_\theta(x_i^{\nu_i})$
9:          $w_i \leftarrow f_\theta(x_i^{\nu_i})$
10:       **end if**
11:    **end for**
12: **end while**

---

add a normalization layer at the output of $f_\theta(x)$ such that $||f_\theta(x)||_2 = 1$. As shown in Fig. 2 (b), this maps all feature vectors to the unit norm ball. For simplicity, $f_\theta(x)$ is assumed to be normalized in all that follows.

In this work, we propose to replace the classifier weight w_i by the embedding of a *randomly chosen view of object instance* o_i. This is implemented by defining a *view sampler* $\nu_i \in \{1, \ldots, V_i\}$ per object instance i, which outputs a number between 1 and V_i. This sampler is then used to draw a feature vector $f_\theta(x_i^{\nu_i})$ that serves as the parameter vector w_i of (1). The sampled feature vectors are called "prototypes" for o_i, as shown in Fig. 2 (c). A softmax temperature parameter τ is also introduced to control the sharpness of the posterior distribution. Larger temperatures originate sharper distributions, smaller temperatures originate more uniform ones. All these transform the softmax layer into

$$P_{Y|X}^s(i|x) = \frac{\exp(f_\theta(x_i^{\nu_i})^T f_\theta(x)/\tau)}{\sum_{k=1}^N \exp(f_\theta(x_i^{\nu_i})^T f_\theta(x)/\tau)}, \quad (3)$$

where $s = \{f_i^{\nu_i}\}_{i=1}^N$ denotes the set of prototypes used to compute the probability.

3.4. Multiview embeddings

The prototype classifier has the ability to learn a more stable multiview representation than the instance classifier. This, however, depends on the sampling of the prototypes $f_i^{\nu_i}$ of (3). To study the impact of prototype sampling, we consider different *randomization schedules*, where the view sampler ν is called more or less frequently during learning, using Algorithm 1. In this algorithm, the threshold $t \in [0, 1]$ controls the frequency with which prototypes are changed. If $t = 0$, prototypes are fixed, and the embedding is denoted a *prototype embedding* (PE). If $t = 1$, the prototypes can change at every iteration. Fig.2 (c-e) illustrates the idea. Starting from an initial prototype set (Fig.2 (c)), prototypes are randomized by choosing embeddings of different views

of each instance to play the role of prototypes (Fig.2 (d-e)) as training progresses.

Mathematically, prototypes belong to the set $\mathbb{S} = \prod_{i=1}^N \{f_i^j\}_{j=1}^{V_i}$ of all possible combinations of view embeddings across the N object instances. This set has cardinality $|\mathbb{S}| = \prod_{i=1}^N V_i$. The randomization of Algorithm 1 can thus be seen as replacing (1) by an ensemble of $|S|$ classifiers, during training. This is similar to the dropout [57], but applied to prototypes only. However, unlike dropout, the randomization is structured in the sense that all the prototypes used to replace w_i are embeddings $f_\theta(x_i^\nu)$ of views from the same object o_i. This ensembling over views makes $f_\theta(x)$ a more stable multiview representation. For this reason, the learned embedding is referred to as a *multiview stochastic prototype embedding* (MVSPE).

3.5. Multiview consistency regularization

The regularization above can be further strengthened by considering the posterior probability distributions of (1). During training, the feature embedding is guided to move toward the prototypes used at each iteration, as illustrated by the dashed arrows of Fig. 2 (c-e). This causes the variations in the distributions also shown in the figure. The magnitude of these variations is a measure of the view sensitivity of the embedding. For effective LWUMOR, the feature distributions should not vary significantly with the prototype. This would imply that the different views of the instance were effectively mapped into a view invariant representation. It follows that it should be possible to strengthen the invariance of the embeddings by minimizing these variations, i.e. encouraging the distributions $P_{Y|X}(i|x)$ to remain stable as the set of prototype is varied. This regularization can be enforced by minimizing the average Kullback-Leibler divergence [8]

$$L_{KL} = K \sum_{s_p, s_q \in \mathbb{S}, p \neq q} \sum_{k=1}^N P^{s_p}(k|x) \log\left(\frac{P^{s_p}(k|x)}{P^{s_q}(k|x)}\right), \quad (4)$$

where $K = \frac{2}{|\mathbb{S}|(|\mathbb{S}|-1)}$, between all pairs of distributions $P_{Y|X}^{s_p}(i|x)$ and $P_{Y|X}^{s_q}(i|x)$ of prototype sets s_p and s_q, where $p \neq q$. When this regularization is used, the resulting embedding is denoted as *view invariant stochastic prototype embedding* (VISPE).

3.6. Scalable Implementation

In practice, the number of instances in the unlabeled dataset $\mathcal{D}$ can be as large as 30,000. Given the memory capacity of current GPUs, it is impractical to load all object prototypes in memory at each each training iteration. One of the benefits of randomization as a regularization strategy is that it is fully compatible with the sampling of a small subset of object prototypes. In our implementation we use $m = 32$ object instances per minibatch. A

set $\mathcal{I} = \{\xi_1, \ldots, \xi_m\}$ of distinct instance indexes is randomly sampled, defining a subset of view embedding combinations $\mathbb{S}' = \prod_{i=1}^{m} \{f_{\xi_i}^j\}_{j=1}^{V_{\xi_i}}$ from which prototypes are drawn. Prototype sets are then defined as $s' = \{f_{\xi_i}^{\nu_{\xi_i}}\}_{i=1}^{m}$ and the posterior probabilities with

$$P_{Y|X}^{s'}(\xi_i|x) = \frac{\exp(f_\theta(x_{\xi_i}^{\nu_{\xi_i}})^T f_\theta(x)/\tau)}{\sum_{k=1}^{m} \exp(f_\theta(x_{\xi_i}^{\nu_{\xi_i}})^T f_\theta(x)/\tau)}, \quad (5)$$

where $i \in \{1, \ldots, m\}$. At each iteration, a pair of prototypes s_1' and s_2' is sampled from the subset of prototype combinations $\mathbb{S}'$, the risk of classifying a training view $x_{\xi_i}^j$ of object instance label ξ_i, using prototype set s_p', is computed with

$$L_{s_p'}(i,j) = -\log\left(P_{Y|X}^{s_p'}(\xi_i|x_{\xi_i}^j)\right) \quad (6)$$

for $p \in \{1, 2\}$, and the KL divergence with

$$L_{KL'} = \sum_{k=1}^{m} P^{s_1'}(k|x_{\xi_i}^j) \log\left(\frac{P^{s_1'}(k|x_{\xi_i}^j)}{P^{s_2'}(k|x_{\xi_i}^j)}\right). \quad (7)$$

Finally, the stochastic gradient descent (SGD) loss for training example $(x_{\xi_i}^j, \xi_i), i \in \{1, \ldots, m\}, j \in \{1, \ldots, V_{\xi_i}\}$ is

$$L = L_{s_1'} + L_{s_2'} + \alpha L_{KL'}. \quad (8)$$

In all experiments, we use a temperature $\tau = 0.05$ and $\alpha = 5$. The implementation is based on Pytorch [47], using the VGG16 [53] model as feature extractor and the output of the last layer as feature vector. A standard SGD was used with learning rate 0.001 to train the network for 300 epochs using batch size of 32.

4. Experiments

In this section, we evaluate the different self-supervised learning algorithms on three multiview datasets.

4.1. Dataset

Three datasets, Modelnet40 [71], Shapenet [7] and ModelNet-S, were used in all experiments. Instead of the rendering process of [24][1], we adopted the rendering approach and dataset[2] widely used in the multiview literature [14, 21, 23, 30, 38, 58]. Given a synthetic CAD model, 12 views are rendered around it at every 30 degrees. The virtual camera elevates 30 degrees and points to the center of the model. Please see [58] for more details.

Modelnet [71] is a synthetic dataset of 3,183 CAD models from 40 object classes. We follow the seen/unseen class split of [24], where unseen classes are those of Modelnet10,

a subset of Modelnet40. The standard training and testing partitions [14, 23, 30, 58] are adopted.

ShapeNet [7] is a synthetic dataset of 55 categories following the Wordnet [41] hierarchy. We use the rendered images from [58], which contains 35,764 training objects and 5,159 test objects. The seen/unseen class split procedure is identical to [24], using the 30 largest categories as seen and the remaining 25 as unseen classes.

Modelnet-S is sampled by ourselves to resemble a dataset with missing views. This is a subset of Modelnet and shares its train/test setup as well as seen/unseen classes.

4.2. Baselines

We consider SSL baselines that solve different surrogate tasks, ranging from context, to motion, view, data augmentation and sequence based, as discussed in Section 2. All baselines except Jigsaw puzzle [46] use the same backbone (VGG16 [53]) and features are extracted from the last network layer. All methods are trained from scratch. For [46], we refer to the original paper for architecture details and the use of pool5 feature. The implementation of these baselines is discussed below and more details can be found in the supplementary materials.

Pretrained [53] is a VGG16 model pre-trained on ImageNet [11] using class supervision.

Autoencoder [17] is trained to reconstruct the input image, using an L2 loss to measure the difference between input and reconstruction.

Egomotion [2] predicts the camera motion between 2 images. Given a pair of images, features are extracted, concatenated and fed into stacked fully connected layers to predict the relative view point difference. Assuming only V viewpoints exist in the dataset, the model will output $V - 1$ probabilities, corresponding to the $V - 1$ viewpoints differences. For architecture details see supplementary material.

Jigsaw puzzle [46] crops 9 patches from the 255×255 input images and shuffles them. The surrogate task is to solve the puzzle. Based on the public source code[3].

UEL [67] treats each image as a class and learns a data augmentation invariant feature. Based on the author's code[4].

ShapeCode [24] reconstructs the subsequent views given an object view. We use the loss function proposed in the original paper to train the network. To accommodate the different rendering conditions, the network inputs and generates 224×224 images instead of 32×32.

MVCNN [58] inputs all views of an object and averages their feature vectors, feeding the result to a fully connected classifier that predicts the object identity.

Triplet [52] is a metric learning approach that learns from triplets of examples: an anchor (input) image, a positive

---

[1]We do not have access to the rendered images.
[2]https://github.com/suhangpro/mvcnn

[3]https://github.com/bbrattoli/JigsawPuzzlePytorch
[4]https://github.com/mangye16/Unsupervised_Embedding_Learning

Table 1: KNN classification results for various baselines, solving different surrogate tasks. RSPE outperforms all self-supervised learning methods, VGG16 pretrained model and instance classifiers.

Datasets Methods / Classes	Surrogate Task	ModelNet seen	ModelNet unseen	ShapeNet seen	ShapeNet unseen	ModelNet-S seen	ModelNet-S unseen
Chance	N/A	3.3	10.0	3.3	4.0	3.3	10.0
Pretrained [53]	N/A	62.7	52.7	63.9	58.1	58.2	55.2
Autoencoder [17]	Context	31.8	37.2	29.8	26.3	34.7	38.8
Egomotion [2]	Motion	32.4	34.7	72.6	47.1	33.0	35.2
Puzzle [46]	Sequence	34.4	41.5	67.8	48.6	34.8	42.4
UEL [67]	Data Aug.	47.9	46.5	68.7	53.4	46.4	48.2
ShapeCode [24]	View	39.4	46.5	67.1	42.3	38.8	47.2
MVCNN [58]	N/A	39.6	48.1	30.3	32.4	36.7	44.8
Triplet [52]	N/A	70.1	62.4	81.2	61.2	64.7	62.1
Instance classifier	N/A	57.7	58.9	69.3	60.4	52.3	54.6
PE	Object	69.7	61.7	81.6	63.8	62.1	60.4
MVSPE	Object	70.3	63.2	82.4	64.6	64.6	62.1
VISPE	Object	**71.2**	**64.4**	**82.9**	**65.5**	**66.2**	**64.3**

Figure 3: TSNE visualization of **unseen class** embeddings. Each color represents a class. RSPE produces more structured embedding.

image (from the same object as the anchor) and a negative image (from a different object). Margin 1 performed best in this setting.

Instance classifier treats each object as a class and trains a VGG16 classifier to minimize (2).

4.3. Classification

All baselines are tested on the 3 datasets, using no labels for training. Inference is based on k nearest neighbor classification, where k is the number of images of the class with fewest objects in the dataset. This is 960, 468 and 500 for ModelNet, Shapenet and ModelNet-S, respectively. Each experiment is repeated for seen and unseen classes per dataset.

Table 1 shows that all previous surrogate tasks perform poorly for MV-SSL. All proposed methods outperform all baselines, regardless of surrogate task. The only competitive baselines are methods that distinguish objects: instance classifier and triplet embedding. The triplet loss has results comparable to MVSPE but, as discussed in Sec. 3.2 and shown in Fig. 4 (a), much slower training convergence. While all proposed methods converge in around 80 training epochs for ModelNet, it requires more than 200. Overall, VISPE has the best performance in all datasets, for both seen and unseen classes. This shows that the surrogate task of learning *object invariants* leads to more robust SSL for multiview data. Fig. 4 (b) shows the effect of the randomization threshold of Algorithm 1, presenting the average accuracy on unseen classes over ten experiments per threshold. Despite the variance of these results, it is clear that randomization during training strengthens model generalization to unseen classes.

4.4. Retrieval and Clustering

Ideally, the learned embedding should map images from the same class close together and images from different classes apart, *even for **unseen** classes*. To test this, Kmeans [20] is used to cluster the image embeddings of unseen classes. Two metrics are used to evaluate clustering quality: recall @ K and normalized mutual information (NMI) [40]. NMI is defined as $\frac{2I(A,C)}{H(A),H(C)}$, where I denotes mutual information, H entropy, $A = \{a_1, \ldots, a_n\}$ where a_i is the set of images assigned to class i, and $C = \{c_1, \ldots, c_n\}$, where c_j is the set of images of ground truth class j. Both metrics are popular in the metric learning literature [43, 55, 56].

Table 2 shows results for Modelnet. Again, triplet is the only baseline competitive with the proposed embeddings, although weaker, and VISPE clearly achieves the best performance. Its effectiveness is highlighted by the large NMI gains. The tightness of its clusters can also be verified in Fig. 3, which presents a TSNE visualization of the feature embeddings. Note how VISPE clustering better separates the different colors, which identify the different object classes.

4.5. Few-shot object recognition

The generalization strength of the different embeddings is further tested by experiments with few shot classification. Table 2 shows classification accuracy of unseen classes when k images are labeled per object class. A linear SVM is trained on the labelled feature vectors of Modelnet [71] unseen classes, and used to classify its test set. Similarly to the previous experiments, only the triplet embedding is competitive with MVSPE and VISPE, and VISPE achieves the best performance.

Table 2: Left: Recall @ k and NMI on Modelnet unseen classes. Right: low shot accuracy for k labeled images.

Methods	Retrieval and Clustering Recall					Low shot k Images		
	@1	@2	@4	@8	NMI	1	3	5
Pretrained [53]	94.5	96.6	98.2	**99.3**	46.7	34.3	46.8	51.2
Autoencoder [17]	81.7	86.8	92.3	95.2	25.4	25.0	30.0	28.0
Egomotion [2]	73.4	80.7	88.0	92.9	7.5	15.1	18.1	19.9
Puzzle [46]	77.8	84.1	89.8	94.0	21.9	21.1	26.8	29.3
UEL [67]	77.8	85.4	91.6	95.7	24.6	23.8	30.9	34.2
ShapeCode [24]	83.4	88.5	93.4	96.2	27.4	28.8	36.1	39.5
MVCNN [58]	80.3	86.7	91.7	95.0	19.3	21.6	27.0	29.5
Triplet [52]	90.8	94.7	97.4	98.8	48.2	41.4	50.3	54.5
Instance classifier	89.1	92.5	95.6	97.4	37.1	28.3	42.2	48.4
PE	91.2	95.0	97.2	98.5	48.2	40.2	49.7	52.9
MVSPE	92.4	95.4	97.7	98.9	48.4	41.5	50.8	54.2
VISPE	**95.5**	**97.7**	**98.6**	99.2	**51.1**	**43.1**	**52.5**	**55.9**

4.6. Dependence on number of objects and views

We next consider how many views and objects are needed to learn an embedding that generalizes to unseen classes. To study this, we sample a subset of ModelNet objects and a subset of views per object. The VISPE embedding is then trained on the sampled data and tested on the unseen classes. Fig. 5 (a) shows that classification accuracy saturates around 40 objects per class and 8 views per object. Interestingly, when the number of objects is small, capturing more views per object compensates for the lack of object diversity. This is of importance for applications, since it suggests that the embedding could be quickly retrained on a relatively small set of objects, e.g. when a robot has to be deployed on a totally new environment.

4.7. Trade-off of training with labels

Even though supervised learning requires more labeling effort, it performs better on predefined classes. For example, training VGG16 [53] with labels on seen classes of ModelNet and ShapeNet yields 84.1% and 87.5% on its test set. However, the performance of KNN on unseen classes drops significantly to 20.9% and 35.1%, which is much worse than most SSL results in Table 1. This begs the question of when SSL should be used. As shown in Fig. 5 (b), when few objects per class are available for Model-Net, VISPE is a better choice than supervised learning, because it generalizes much better ($> 30\%$) on unseen classes and maintains comparable performance ($< 10\%$) on seen classes.

5. Conclusion

In this work, we made several contributions to MV-SSL. We started by discussing the current impractical assumption of fully supervised multiview recognition, which re-
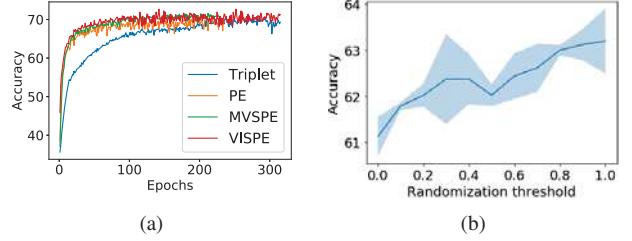
Figure 4: (a) Convergence rate of proposed methods and triplet loss. (b) Effect of different randomization threshold on unseen class accuracy.
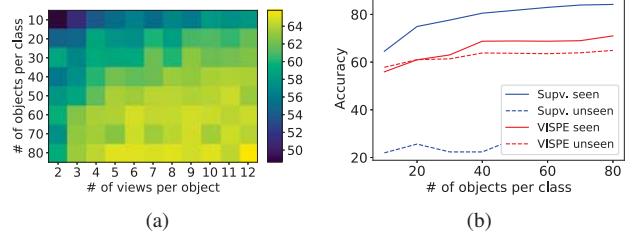
Figure 5: (a) Accuracy (represented by color) of VISPE on unseen classes, as a function of views per object and objects per class in training set. (b) Trade-off of training with labels as a function of object per class.

quires intensive labeling. We then relaxed this assumption by investigating MV-SSL methods, where only "free labels" (image to object association) are required. Embeddings that generalize to both seen and unseen data were then learned with variants of this MV-SSL surrogate task. These variants differ in the regularization used to encourage object invariant representations. We started by leveraging view information by choosing the embedding of a random object view as the object prototype. A randomization schedule was then proposed to sample prototypes stochastically. This can be seen as an ensembling over views, to encourage stable multiview embeddings. To strengthen the learning of object invariants, we finally proposed a multiview consistency constraint. The combination of all these contributions produced a new class of view invariant stochastic prototype embeddings (VISPE). These embeddings were shown to outperform other SSL methods on seen and unseen data for both multiview classification and retrieval. While we have not studied the semi-supervised setting, where few labels are provided, in great detail, this setting is also supported by VISPE. We believe that these are important contributions for the much needed extension of multiview recognition to the LWUMOR setting of Figure 1, which is of interest for many real world applications.

References

[1] A.Dosovitskiy, J.T.Springenberg, M.Riedmiller, and T.Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in Neural Information Processing Systems 27 (NIPS)*, 2014.

[2] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 37–45, Dec 2015.

[3] Unaiza Ahsan, Rishi Madhok, and Irfan A. Essa. Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. *CoRR*, abs/1808.07507, 2018.

[4] Lei Jimmy Ba and Brendan Frey. Adaptive dropout for training deep neural networks. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3084–3092, USA, 2013. Curran Associates Inc.

[5] Miguel Ángel Bautista, Artsiom Sanakoyeu, Ekaterina Tikhoncheva, and Björn Ommer. Cliquecnn: Deep unsupervised exemplar learning. In *NIPS*, 2016.

[6] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision*, 2018.

[7] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015.

[8] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, New York, NY, USA, 1991.

[9] Guoxian Dai, Jin Xie, and Yi Fang. Siamese cnn-bilstm architecture for 3d shape representation learning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 670–676. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[10] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2):5:1–5:60, May 2008.

[11] 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.

[12] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1422–1430, 2015.

[13] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. *CoRR*, abs/1809.09401, 2018.

[14] Yifan Feng, Zizhao Zhang, Xibin Zhao, Rongrong Ji, and Yue Gao. Gvcnn: Group-view convolutional neural networks for 3d shape recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[15] Zan Gao, Deyu Wang, Xiangnan He, and Hua Zhang. Group-pair convolutional neural networks for multi-view based 3d object retrieval. In *AAAI*, 2018.

[16] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *CoRR*, abs/1803.07728, 2018.

[17] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[18] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742, June 2006.

[19] Z. Han, H. Lu, Z. Liu, C. Vong, Y. Liu, M. Zwicker, J. Han, and C. L. P. Chen. 3d2seqviews: Aggregating sequential views for 3d global feature learning by cnn with hierarchical attention aggregation. *IEEE Transactions on Image Processing*, 28(8):3986–3999, Aug 2019.

[20] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[21] Xinwei He, Yang Zhou, Zhichao Zhou, Song Bai, and Xiang Bai. Triplet-center loss for multi-view 3d object retrieval. *CoRR*, abs/1803.06189, 2018.

[22] Vishakh Hegde and Reza Zadeh. Fusionnet: 3d object classification using multiple data representations. *CoRR*, abs/1607.05695, 2016.

[23] Chih-Hui Ho, Pedro Morgado, Amir Persekian, and Nuno Vasconcelos. Pies: Pose invariant embeddings. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[24] Dinesh Jayaraman, Ruohan Gao, and Kristen Grauman. Unsupervised learning through one-shot image-based shape reconstruction. *CoRR*, abs/1709.00505, 2017.

[25] Dinesh Jayaraman and Kristen Grauman. Slow and steady feature analysis: Higher order temporal coherence in video. *CoRR*, abs/1506.04714, 2015.

[26] Dinesh Jayaraman and Kristen Grauman. Learning image representations tied to egomotion from unlabeled video. *Int. J. Comput. Vision*, 125(1-3):136–161, Dec. 2017.

[27] Huaizu Jiang, Gustav Larsson, Michael Maire, Greg Shakhnarovich, and Erik Learned-Miller. Self-supervised relative depth learning for urban scene understanding. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 20–37, Cham, 2018. Springer International Publishing.

[28] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *CoRR*, abs/1902.06162, 2019.

[29] Jung-Eun Lee, Rong Jin, and A. K. Jain. Rank-based distance metric learning: An application to image retrieval. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[30] Asako Kanezaki. Rotationnet: Learning object classification using unsupervised viewpoint estimation. *CoRR*, abs/1603.06208, 2016.

[31] Dahun Kim, Donghyeon Cho, Donggeun Yoo, and In So Kweon. Learning image representations by completing damaged jigsaw puzzles. *CoRR*, abs/1802.01880, 2018.

[32] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Self-supervised learning of 3d human pose using multi-view geometry. *CoRR*, abs/1903.02330, 2019.

[33] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6402–6413. Curran Associates, Inc., 2017.

[34] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. *CoRR*, abs/1603.06668, 2016.

[35] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. *CoRR*, abs/1703.04044, 2017.

[36] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. *CoRR*, abs/1708.01246, 2017.

[37] Haebeom Lee, Juho Lee, Eunho Yang, and Sung Ju Hwang. Dropmax: Adaptive stochastic softmax. *CoRR*, abs/1712.07834, 2017.

[38] Zhaoqun Li, Cheng Xu, and Biao Leng. Angular triplet-center loss for multi-view 3d shape retrieval. *CoRR*, abs/1811.08622, 2018.

[39] C. Ma, Y. Guo, J. Yang, and W. An. Learning multi-view representation with lstm for 3-d shape recognition and retrieval. *IEEE Transactions on Multimedia*, 21(5):1169–1182, May 2019.

[40] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008.

[41] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, Nov. 1995.

[42] Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. Unsupervised learning using sequential verification for action recognition. *CoRR*, abs/1603.08561, 2016.

[43] Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. *CoRR*, abs/1703.07464, 2017.

[44] T. Nathan Mundhenk, Daniel Ho, and Barry Y. Chen. Improvements to context based self-supervised learning. *CoRR*, abs/1711.06379, 2017.

[45] Sanjeev Muralikrishnan, Vladimir G. Kim, Matthew Fisher, and Siddhartha Chaudhuri. Shape unicode: A unified shape representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[46] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. *CoRR*, abs/1603.09246, 2016.

[47] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[48] Deepak Pathak, Ross B. Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. *CoRR*, abs/1612.06370, 2016.

[49] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context encoders: Feature learning by inpainting. 2016.

[50] Charles Ruizhongtai Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. *CoRR*, abs/1604.03265, 2016.

[51] Hang Qi, Matthew Brown, and David G. Lowe. Learning with imprinted weights. *CoRR*, abs/1712.07136, 2017.

[52] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832, 2015.

[53] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[54] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. *CoRR*, abs/1703.05175, 2017.

[55] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1857–1865. Curran Associates, Inc., 2016.

[56] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. *CoRR*, abs/1511.06452, 2015.

[57] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, Jan. 2014.

[58] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. *CoRR*, abs/1505.00880, 2015.

[59] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Self-supervised 3d hand pose estimation through training by fitting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[60] Feng Wang, Xiang Xiang, Jian Cheng, and Alan L. Yuille. Normface: L_2 hypersphere embedding for face verification. *CoRR*, abs/1704.06369, 2017.

[61] Junyan Wang, Bingzhang Hu, Yang Long, and Yu Guan. Order matters: Shuffling sequence generation for video prediction. *CoRR*, abs/1907.08845, 2019.

[62] Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. *CoRR*, abs/1904.03597, 2019.

[63] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. *CoRR*, abs/1505.00687, 2015.

[64] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016.

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

[66] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan L. Yuille. Mitigating adversarial effects through randomization. *CoRR*, abs/1711.01991, 2017.

[67] Mang Ye, Xu Zhang, Pong C. Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. *CoRR*, abs/1904.03436, 2019.

[68] Haoxuan You, Yifan Feng, Rongrong Ji, and Yue Gao. Pvnet: A joint convolutional network of point cloud and multi-view for 3d shape recognition. *CoRR*, abs/1808.07659, 2018.

[69] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. *CoRR*, abs/1603.08511, 2016.

[70] Liang Zheng, Yi Yang, and Alexander G. Hauptmann. Person re-identification: Past, present and future. *CoRR*, abs/1610.02984, 2016.

[71] Zhirong Wu, S. Song, A. Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, June 2015.