CoKe: Localized Contrastive Learning for Robust Keypoint Detection

Yutong Bai*, Angtian Wang*, Adam Kortylewski†, Alan Yuille†
Johns Hopkins University
{ytongbai, alan.l.yuille}@gmail.com; {angtianwang,akortyl1}@jhu.edu

Abstract

Today's most popular approaches to keypoint detection involve very complex network architectures that aim to learn holistic representations of all keypoints. In this work, we take a step back and ask: Can we simply learn a local keypoint representation from the output of a standard backbone architecture? This will help make the network simpler and more robust, particularly if large parts of the object are occluded. We demonstrate that this is possible by looking at the problem from the perspective of representation learning. Specifically, the keypoint kernels need to be chosen to optimize three types of distances in the feature space: Features of the same keypoint should be similar to each other, while differing from those of other keypoints, and also being distinct from features from the background clutter. We formulate this optimization process within a framework, which we call CoKe, which includes supervised contrastive learning. CoKe needs to make several approximations to enable representation learning process on large datasets. In particular, we introduce a clutter bank to approximate non-keypoint features, and a momentum update to compute the keypoint representation while training the feature extractor. Our experiments show that CoKe achieves state-of-the-art results compared to approaches that jointly represent all keypoints holistically (Stacked Hourglass Networks, MSS-Net) as well as to approaches that are supervised by detailed 3D object geometry (StarMap). Moreover, CoKe is robust and performs exceptionally well when objects are partially occluded and significantly outperforms related work on a range of diverse datasets (PASCAL3D+, MPII, ObjectNet3D).

1. Introduction

Semantic keypoints, such as the joints of a human body, provide concise abstractions of visual objects in terms of their shape and pose. Accurate keypoint detections are of central importance for many visual understanding tasks, including viewpoint estimation [29], human pose estimation [4], action recognition [25], feature matching [23], image classification [44], and 3D reconstruction [16]. Today's most popular approaches to keypoint detection learn very complex deep network architectures [27, 17], that sometimes even require 3D supervision with CAD models [46]. These models aim to learn a holistic representation of all keypoints, which enables them to implicitly leverage the relative spatial geometry between keypoints to prevent false-positive detections due to local ambiguities. However, recent works [47, 20] have shown that deep vision systems are not as robust as humans to partial occlusion at image classification. Our experiments show that partial occlusion is also a fundamental challenge for state-of-the-art keypoint detectors that needs to be addressed.

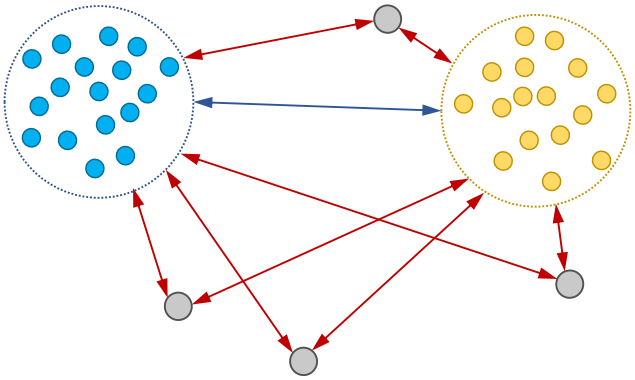


Figure 1. Intuition behind our approach for the contrastive learning of keypoint detectors. Feature representations of two different keypoints are drawn as blue and yellow circles. Clutter features are drawn as grey circles. Our goal is to learn a representation in which the distance between features of the same keypoint is small, *i.e.* they form tight clusters, while their distance to features of other keypoints (blue) and clutter features (red) should be large.

The advanced complexity of current deep network architectures and the lack of robustness motivates us to study if we can simply learn a deep network such that each convolution kernel in the last layer is a separate keypoint detector. Intuitively, the independent detection of each keypoint will avoid a holistic representation of all keypoints and

*Joint first authors
†Joint senior authors

hence naturally introduce robustness to partial occlusion. To achieve this, a main challenge is to reduce the false-positive detections of the independent detectors due to local ambiguities in the image. We consider this problem from the perspective of contrastive representation learning. In particular, the keypoint representations need to be learned such that they optimize three types of distances in the feature space (Figure 1): Features of the same keypoint should be similar to each other, while differing from those of other keypoints, and also being distinct from features from the background clutter. The main problem when applying contrastive learning in the context of keypoint detection is that the number of representations per image is large. In related work, such as face recognition [30] or unsupervised learning [10], each image is represented by a single one-dimensional representation vector. However, for keypoint detection the learned representation is a three-dimensional feature tensor. Moreover, the higher the resolution of the feature tensor, the more accurately can keypoints be detected, but, on the other hand, this also introduces an even larger computational cost. Our key contribution is that we introduce a contrastive keypoint learning framework (CoKe) that can manage this computational cost by making the following efficient approximations:

1) Maximizing the distance to clutter features (Figure 1 red) is important for reduce false-positive detections in the non-keypoint positions. However, most of the features in an image are clutter features and it is not feasible to compute the distance to all of them. Therefore, we introduce a *clutter bank* that keeps track of clutter features that are spatially close to keypoint features and hence are most difficult to be distinguished from.

2) Maximizing the distance between keypoints (Figure 1 blue) enforces keypoints to be different from each other. The number of distance comparisons between keypoint features is combinatorial in the number of keypoints and training images. To overcome this computational burden we introduce prototypical keypoint representations that are stored in a *keypoint bank*. During the training of the feature extractor, these are updated using cumulative moving average update [10]. The distance between features of different keypoints can then be approximated as the distance to the corresponding keypoint prototypes.

3) Computing the distance between all features of the same keypoint (indicated in Figure 1 by the dotted circle) is quadratic in their number. We instead approximate this by computing the distance of the features to their prototypical representation, reducing the computational cost to be linear.

We perform our experiments in two situations. First, we show on the PASCAL 3D+ dataset that CoKe performs mostly on par, and often even better, compared to state-of-the-art related work that have holistic representations (Stacked Hourglass Networks, MSS-Net [17]) and to approaches that use additional supervision in terms of detailed 3D object geometries (StarMap [46]). In addition, CoKe also outperforms SHG on the MpII and the ObjectNet3D dataset. These results are remarkable as CoKe performs keypoint detection independently without any geometric constraints between the keypoints. Also note that CoKe works on all these datasets while, for example, the best results on MpII are often achieved by architectures that are specialized for human keypoint detection. Second, we show that CoKe performs exceptionally well when objects are partially occluded, due to its localized keypoint representation. Our contributions in this work are as follows:

- We introduce a new approach to keypoint detection that is simple, works for a wide range of object classes, and is non-holistic.
- We introduce a general framework for contrastive representation learning in localization problems. Specifically, we generalize the current paradigm of learning a single vectorial representation of an image, towards learning tensor-shaped representations. We make this possible by significantly reducing the computational complexity through efficient approximations. Our generalized framework could be useful for other problems in vision, such as sub-pixel localization, part-based detection or image segmentation.
- For non-occluded objects, we outperform related work on PASCAL3D+[43], ObjectNet3D[42], MPII[1] by a small but significant margin, despite being much simpler and non-holistic.
- We achieve exceptional robustness to partial occlusion when compared to related work.

2. Related Work

Keypoint Detection. Keypoint detection, as a widely studied problem in computer vision, is usually discussed and studied in two categories – human joints detection [4, 27, 33, 34] and rigid objects' keypoint detection [40, 35, 29, 46]. Early approaches relied on local descriptors [8] that are distinctive and invariant [24]. While approaches using local descriptors have proven to be robust to occlusion and background clutter, they were outperformed by deep learning approaches that were trained end-to-end [27]. Toshev et al. [34] first trained a deep neural network for 2D human pose regression and Li et al. [21] extended this approach to 3D. Starting from the work of Tompson et al. [33], the heatmap representation became very popular for 2D keypoint estimation, achieving very good performances in both 2D human pose estimation [27]. However for rigid objects' keypoint detection, These works showed that predicting keypoints jointly with deep networks led to an

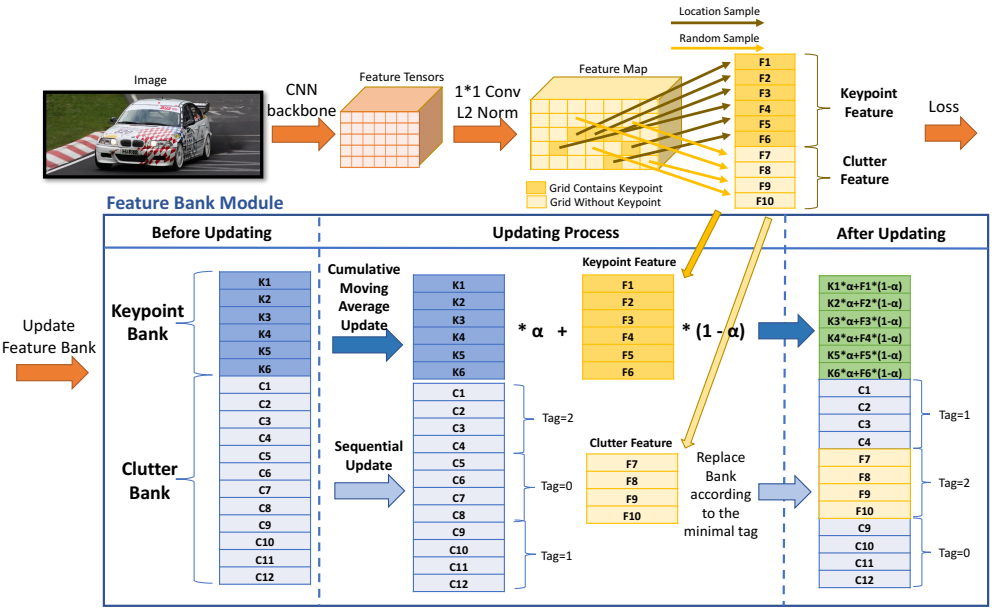


Figure 2. Illustration the Feature Bank Updating Process. At first, we extract a feature map for the input image. After dimensionality reduction and L_2 normalization, we retrieve the features for keypoint f_k^i for all keypoints and features for random selected clutter c_k^i. We refer to these features as 'Keypoint Feature' and 'Clutter Feature'. We compute the loss between these features w.r.t. the Feature Bank Module and update the feature bank accordingly. The feature bank consists of two parts: a Keypoint Bank and a Clutter Bank. The Keypoint Bank is updated using cumulative moving average update. The Clutter Bank is updated by replacing the oldest features in the Clutter Bank with c_k^i based on the time tag.

improved performance, as the implicitly encoded structural information between keypoints provides important cues to resolve locally ambiguous keypoint detections. Tulsiani et al. [35], proposed to integrate the structural information between keypoints explicitly by integrating 2D and 3D models, which inspired a number of follow-up works, in particular for rigid objects [46, 35, 29].

Supervised Contrastive Learning. Contrastive learning, originating from Metric Learning[5, 39] at first, involves the learning of a feature space by optimizing the similarities of sample pairs in a representation space. The general intuition underlying supervised contrastive learning is to transform the training data into a feature space where the distance of feature representations of samples from the same class is small, whereas it should be big for samples from different classes. Popular examples use pairs of samples for loss computation[9], triplets [30], or N-Pair tuples [31].

Recently, contrastive learning has attracted attention from the research community in self-supervised learning [6, 10, 41, 26, 12] . The main difference in the self-supervised learning setting is that positive examples are usually generated using data augmentations[7] or co-occurrence[12, 14, 32] of a query sample whereas negative examples are chosen as other images in the same mini-batch, while in supervised, the labels are used to guide the choice of positive and negative pairs.

While most of the supervised contrastive learning [18] focuses on learning a holistic representation of the complete image, in this paper, we target a more fine-grained task - keypoint detection. Keypoints are localized image patterns and therefore require the learning of local feature embeddings. The main challenge is that local image patterns can be highly ambiguous (e.g. the front and back tire of a car) and therefore require a contrastive learning framework that can learn to disambiguate local representations, while at the same time being able to learn a distinct representation that can be localized accurately.

3. CoKe: Contrastive Keypoint Learning

In this Section, we present our framework for contrastive keypoint learning. We discuss the intuition underlying our approach and the training pipeline in Section 3.1. Finally, we discuss how to perform keypoint detection during inference in Scetion 3.2.

3.1. Training CoKe

We use Φ to denote the feature extractor, that given input image $\mathbf{I}^i$ computes the feature map $\Phi(\mathbf{I}^i) = \mathbf{F}^i \in \mathbb{R}^{H \times W \times D}$, where i is the image index in the training data $\{\mathbf{I}^i | i \in \{1, \ldots, N\}\}$. Using the keypoint annotation can compute the correspond location of a keypoint on the feature map $\mathbf{F}^i$. We denote this position as k. Similarly, we can

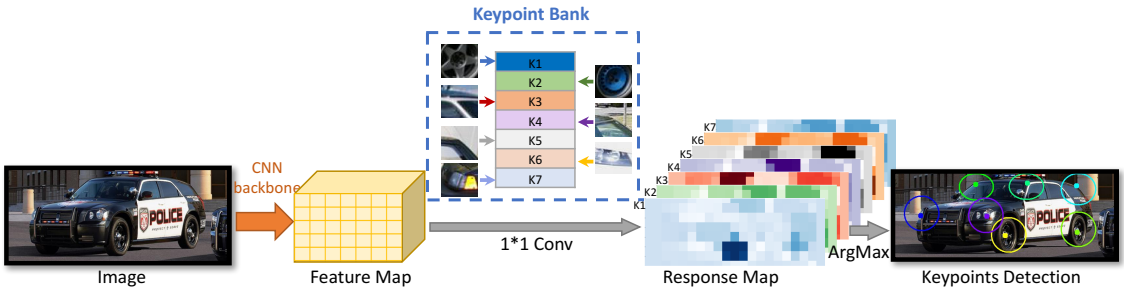


Figure 3. Keypoint detection with CoKe. We use a CNN backbone to extract feature representation at different positions of the input image. Each keypoint in the Keypoint Bank has an individual representation that is used as a convolution kernel to compute a response map for each keypoint. The location of maximum response is used as prediction result. The colored boxes show the ground truth and the dots illustrate the prediction result.

(randomly) select a non-keypoint location as clutter point c. We retrieve the corresponding keypoint feature $\mathbf{f}_k^i$ and clutter feature $\mathbf{f}_c^i$. We define distance between two features as $d(\cdot,\cdot)$. Our goal of training is to learn a feature extractor that optimizes the following distances in the feature space:

During training one of our goals is to **minimize the distance between features of the same keypoint** across all training images. For one keypoint feature $\mathbf{f}_k^i$ this involves computing the objective: $D_{within}(\mathbf{f}_k^i) = \sum_{j=1}^{N} d(\mathbf{f}_k^i, \mathbf{f}_k^j)$. However, as we described in the introduction it is unpractical to calculate all distances $d(\mathbf{f}_k^i, \mathbf{f}_k^j)$ for the whole dataset. To resolve this computational problem, we define a prototypical keypoint feature θ_k, that represents the average feature of keypoint k. Instead of computing the full objective, we approximate it via $\bar{D}_{within}(\mathbf{f}_k^i) = d(\mathbf{f}_k^i, \theta_k)$. We store the prototypical features of all keypoints in a *keypoint bank* and update them during training as described in Section 3.1.1.

Furthermore, we need to **maximize the distance between features of different keypoints**. To compute the distance between one particular keypoint feature $\mathbf{f}_k^i$ and all feature vectors of other keypoints we need to compute the distance $D_{between}(\mathbf{f}_k^i) = \sum_{j=1}^{N} \sum_{k' \in K \setminus k} d(\mathbf{f}_k^i, \mathbf{f}_{k'}^j)$. We approximate this distance by instead computing the distance to the prototypes of the respective keypoint features from the keypoint bank: $\bar{D}_{between}(\mathbf{f}_k^i) = \sum_{k' \in K \setminus k} d(\mathbf{f}_k^i, \theta_{k'})$.

Finally, we need to **maximize the distance between the keypoint features and all clutter features**. In the best case, this involves computing the distance between a keypoint feature $\mathbf{f}_k^i$ to every clutter feature $\mathbf{f}_c^i, \forall c \in C$ for all training images. To avoid the computation of this large amount of distances, we instead approximate this objective by storing a set of clutter features $\{\theta_c, c \in \{1, \ldots, C\}\}$ in a *clutter bank*. Which allows us to deduct the objective as $D_{clutter}(\mathbf{f}_k^i) = \sum_{c \in C} d(\mathbf{f}_k^i, \theta_c)$.

These approximations make it feasible to optimize the overall objective with a feasible computational load. However, during learning the parameters of the feature extractor will change, and hence the clutter features in the clutter bank as well as the prototypes θ_k need to be updated. To achieve this, we follow an EM-type optimization process. In particular, we initialize the clutter bank by randomly sampling clutter features from the training data. In addition, we initialize the prototypes as $\theta_k = \frac{\sum_{i=1}^{N} f_k^i}{N}$. Using these initial estimates, we can compute the overall objective and train the feature extractor (see Section 3.1.2). While training the feature extractor we update the clutter bank and keypoint bank as described in the next section. Then, we perform both updates in an alternating manner.

3.1.1 Feature Bank Update

Figure 2, illustrates the process of updating the keypoint prototypes and the clutter bank during training. Computing the prototypical keypoint features θ_k while learning the feature extractor is challenging, because we want to avoid to re-compute the prototypes as $\theta_k = \frac{\sum_{i=1}^{N} f_k^i}{N}$ after every gradient step. Instead, we approximate the sample mean as a cumulative moving average. Specifically, we update the θ_k using:

$$\theta_k \leftarrow \theta_k' * \alpha + \frac{\sum_{i=0}^{m} \mathbf{f}_k^i}{m} * (1 - \alpha) \tag{1}$$

where m is the batchsize. Similarly, for maintaining the clutter samples $\{\theta_c, c \in C\}$ we cannot extract the clutter features of all training images. Instead, we maintain a limited set of clutter features. In practice the size of the clutter bank depend on the GPU memory and we observe that the larger the bank, the better the training process (see the experiments Section 4). For updating the clutter bank during training, we replace the oldest clutter features in the bank with newly calculated ones from the current training batch based on a tag that indicates how long features have been stored in the bank.

3.1.2 Feature Extractor Update

When training the feature extractor, we fix the keypoint bank and clutter bank, and use them to compute the loss to calculate gradient updates for the weights of feature extractor. Note that all in our model are L2 normalized. To compute the distance between features, we use the L2 distance:

$$D_{within}(\mathbf{f}_k^i) = (\mathbf{f}_k^i - \theta_k)^2 = 2 * (1 - \mathbf{f}_k^i \cdot \theta_k) \quad (2)$$

Thus, we can minimize $D(\mathbf{f}_k^i)$ by maximizing $\mathbf{f}_k^i \cdot \theta_k$. Similarly, we can optimize $D_{between}(\mathbf{f}_k^i)$ and $D_{clutter}(\mathbf{f}_k^i)$ by minimizing $\{\mathbf{f}_k^i \cdot \theta_{k'} | \forall k' \in K \setminus k\}$ and $\{\mathbf{f}_k^i \cdot \theta_c \,|\forall c \in C\}$ respectively. To optimize those terms simultaneously, we use a non-parametric softmax as our loss function. Thus, the loss for each keypoint is calculated as:

$$\mathcal{L}(\mathbf{f}_{k'}^i, \{\theta_k\}, \{\theta_c\}) = \frac{e^{\mathbf{f}_{k'}^i \cdot \theta_{k'}}}{\sum_{k \in K} e^{\mathbf{f}_{k'}^i \cdot \theta_k} + \sum_{c \in C} e^{\mathbf{f}_{k'}^i \cdot \theta_c}}, \quad (3)$$

where $\{\theta_k\}$ is the keypoint bank and $\{\theta_c\}$ is the clutter bank. In practice, optimizing the loss function in Equation 3, the feature extractor will not effectively optimize the clutter features $\mathbf{f}_c^i$ to maximize $D(\mathbf{f}_k^i, \mathbf{f}_c^i)$. As a result, once the number C is large enough, θ_c will be uniformly spread across the unit hyper sphere. This makes $D_{clutter}(\mathbf{f}_k^i)$ hard to optimize. We found that we can solve this problem, by maximizing $D_{clutter}(\theta_k)$ during training. To achieve this, we propose a clutter loss:

$$\mathcal{L}(\mathbf{f}_{c'}^i, \{\theta_k\}) = \sum_{k \in K} \mathbf{f}_{c'}^i \cdot \theta_k \quad (4)$$

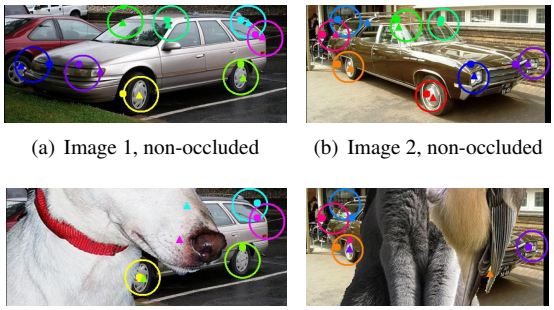
Thus the final loss is:

$$\mathcal{L}(\mathbf{F}^i, \{\theta_k\}, \{\theta_c\}) = \sum_{k' \in K} \mathcal{L}(\mathbf{f}_{k'}^i, \{\theta_k\}) + \sum_{c' \in C} \mathcal{L}(\mathbf{f}_{c'}^i, \{\theta_c\}) \quad (5)$$

3.2. Inference with the CoKe Model

In the following, we describe the model for keypoints of a single category. However, note that it can be easily extended to multiple categories. Figure 3 illustrates the inference process with the CoKe. During inference we will predict p_k, the predicted the keypoint position for keypoint k on the feature map. This can be simply achieved for a test image $\mathbf{I}$ using the following steps:

- Extract a feature map $\Phi(\mathbf{I}) = \mathbf{F}$ using the trained feature extractor Φ.
- Use the prototypes $\theta_k, k \in K$ as 1×1 convolution kernel on the feature map $\mathbf{F}$. The output will be a score map that contains at every position p and for every channel k the keypoint detection score $\mathbf{f}_p^i \cdot \theta_k$.



(a) Image 1, non-occluded (b) Image 2, non-occluded

(c) Image 1, occluded (d) Image 2, occluded

Figure 4. Negative effect of partial occlusion on keypoint localization. Colored Circles: Ground truth keypoint location; Dots: CoKe predictions; Triangles: SHG predictions. For non-occluded objects (a & b) both models perform well. However, partial occlusion (c & d) distorts the SHGs because of the entangled keypoint representation, while CoKe can still localize the visible keypoints well.

- To predict the location of keypoint k, we pick the position $p \in P$ that has the highest score.
- Project p back to the original image coordinates.

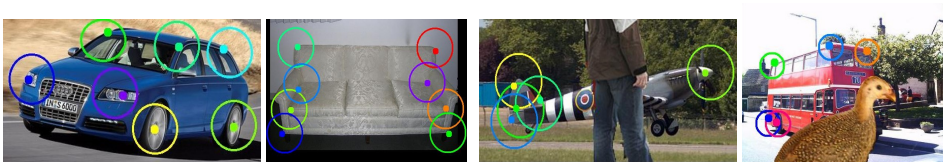
4. Experiments

In this section, we experimentally evaluate CoKe and compare its performance to related work that uses joint keypoint representations. We begin with describing the experimental setup and evaluate the robustness of CoKe and other methods to partial occlusion. Then we show some qualitative results and analyse the time and memory consumption. At last, we perform ablation studies for all the important components and design choices in CoKe.

4.1. Experimental Setup

Evaluation Protocol. Following the standard experimental setup, we use PCK=0.1 (percentage of correct keypoints), as the evaluation metric. PCK considers a keypoint to be correct if its L_2 pixel distance from the ground truth keypoint location is less than $0.1 \cdot max(h, w)$, where h and w are the object's bounding box pixel size. We evaluate each object category by computing the average accuracy on the visible keypoints over all the test images.

Training Setup. We use the standard train-val-test split for all the datasets. For training we use a batch size of 64. In each image, we randomly choose 20 clutter points as a group and the clutter bank contains 1024 groups. We choose the clutter features to be within two pixels distance to the keypoint annotation in the feature map. We choose to use the non-parametric softmax [41] to calculate the similarity between features and banks. The temperature parameter[13] that controls the concentration level of the



(a) Occlusion Level: 0 (b) Occlusion Level: 1

(c) Occlusion Level: 2 (d) Occlusion Level: 3

(e) MPII: Challenging Scenarios

Figure 5. Qualitative detection results under different levels of partial occlusion for artificially occluded objects from PASCAL3D+ (a-d) and humans from MPII (e). The dots visualize the detection result of CoKe. The colored circles in (a-d) indicate the ground-truth position within PCK=0.1. Note how CoKe is very robust even under strong occlusion.

distribution is set to $\tau = 0.7$. We compute the baseline following the basic settings of the Stacked Hourglass Network with 8 stacks.

PASCAL3D+ Dataset. We evaluate our approach on the PASCAL3D+ benchmark. The dataset contains 12 man-made object categories with totally 11045 images for training and 10812 images for evaluation. Different from some previous work [46, 35], we use all the images for evaluation, including the occluded and truncated ones. The number of keypoints ranges from 7 to 15 per category.

OccludedPASCAL3D+ Dataset. While it is important to evaluate algorithms on real images of partially occluded objects (see experiments in Section 4.2), simulating occlusion enables us to quantify the effects of partial occlusion more accurately. Inspired by the success of dataset with artificially generated partial occlusion in image classification [20, 19], part detection [37] and object detection [36] , we use an analogous dataset with artificial occlusion for keypoint detection. In particular, we use the *OccludedPASCAL3D+* dataset proposed in [36] for object detection. It contains contains all 12 classes of the original PASCAL 3D+ [43] at various levels of occlusion. The occluders, include humans, animals and plants, which were cropped from the MS-COCO dataset [22] are different from the 12 PASCAL3D+ classes. The dataset has a total of 3 occlusion levels, with Lv.1: 20-40%, Lv.2: 40-60% and Lv.3: 60-80% of the object area being occluded.

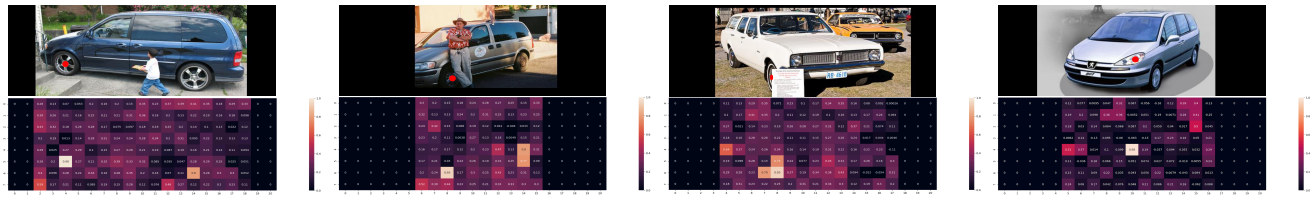
MPII Dataset. MPII Human Pose [1] has 25k images with annotations for multiple people providing 40k annotated samples (28k training,11k testing). MPII consists of images taken from a wide range of human activities with a challenging array of articulated poses. The keypoint visibility is annotated, enabling us to report numbers for the full dataset as well as partially occluded humans.

ObjectNet3D Dataset. ObjectNet3D [42] consists of common objects in daily life and is notably more difficult compared to PASCAL3D+ as it contains more rare viewpoints, shapes and truncated objects with occlusion. We use 7 chosen classes from ObjectNet3D in our evaluation based on the annotation accuracy for a reasonable result.

4.2. Performance on Various Datasets

4.2.1 PASCAL3D+.

Table 1 shows the keypoint detection results on the PASCAL3D+ dataset for CoKe models learned from three different backbones: ResNet-50 [11], Stacked-Hourglass-Network and Res-UNet [45]. The performance of the CoKe models with these very different backbones is constantly high. The highest performance is achieved with the most recently developed architecture Res-UNet. When compared to Stacked-Hourglass-Networks (SHGs) we can clearly observe a large gain in performance. Most notably, the performance difference is very prominent for strong occlusion. We also report the performance of StarMap [46] which uses explicit 3D models to jointly reason about the relative position of the keypoints. Note that StarMap reports only use images which are non-truncated and non-occluded,



(a) False Positives (b) Natural Occlusion (c) Rare Occlusion (d) Different Domain

(e) Different Subtype (f) Irregular Appearance (g) Irregular State (h) Rare Viewpoints & Self Occlusion

Figure 6. Eight examples of CoKe-Res50's representation visualization. For each sub-figure, top is the original image with keypoint annotation, labeled with red dot. Bottom is the response map, predicted by CoKe-Res50. It is worth noting that how all keypoints are detected accurately despite the difficulty from false positives, occlusions, rare viewpoints, different domain, irregular appearance and irregular state and rare viewpoints.

PASCAL3D+					
Occlusion Level	Lv.0	Lv.1	Lv.2	Lv.3	Avg
SHGs	68.0	46.5	43.2	39.9	49.4
MSS-Net	68.9	46.6	42.9	39.6	49.5
StarMap	78.6	-	-	-	-
CoKe-Res50	77.0	67.6	**59.9**	53.4	64.4
CoKe-SHG	78.3	66.3	58.4	52.3	63.8
CoKe-Res-UNet	**80.3**	**68.5**	59.1	**54.0**	**65.5**

Table 1. Keypoint detection results on PASCAL3D+ under different levels of partial occlusion (Lv.0:0%,Lv.1:20-40%,Lv.2:40-60%,Lv.3:60-80% of objects are occluded, L0 is the original dataset). CoKe models learned from several different backbones (ResNet50, Stacked-Hourglass, Res-UNet) are highly robust to partial occlusion. Furthermore, they outperform the models that leverage additional the structural information between the keypoints implicitly (SHGs, MSS-Net[17]) and explicitly (StarMap).

but we report results for all 11476 images without manually picking up. Overall, our results clearly highlight that CoKe is competitive with models that leverage geometric constraints between keypoints either explicitly or implicitly, while being highly robust to partial occlusion.

MPII. We compare Stacked-Hourglass-Networks and a CoKe model learned from the SHGs backbone on MPII in Table 3. CoKe-SHG outperforms many holistic algorithms which is very difficult to achieve. Also, in Table 3 we show that CoKe-SHG achieves higher results on occlusion scenarios compared with SHGs for both occluded and non-occluded keypoint detection on the MPII dataset.

ObjectNet3D. We compare Stacked-Hourglass-Networks and a CoKe model learned from the SHGs backbone on ObjectNet3D in Table 4.2.1. CoKe-SHG outperforms SHGs for object keypoint detection on

MPII	
RecurrentPose[2]	88.1
PoseMachines[38]	88.5
DeeperCut[15]	88.5
PartHeatmap[3]	89.7
SHG[]	90.9
DualPathNetworks[28]	91.2
PoseRegression[3]	91.2
CoKe-SHG	**91.4**

Table 2. Keypoint detection results on MPII compared with holistic algorithms. As a local detector, CoKe remains competitive compared with these baselines.

MPII		
	Full	Occluded
SHGs	90.1	84.3
CoKe-SHG	**91.4**	**86.7**

Table 3. Keypoint detection results on MPII and with a Stacked-Hourglass-Network (SHGs) and a CoKe model learned from the SHGs backbone. CoKe outperforms SHGs on MPII, not only on the original images but challenging scenarios such as occluded humans as well.

ObjectNet3D dataset on nine categories, which we picked based on their annotation accuracy since some categories on ObjectNet3D has some truncated objects due to the complexity of this dataset.

In summary, we observe that CoKe is a general purpose framework that constantly achieves a high performance for a wide range of backbone architectures and for a range of datasets with very different characteristics.

ObjectNet3D					
	coffee	dryer	kettle	jar	wash
SHG[27]	31.0	35.1	32.2	41.9	33.9
CoKe-SHG	**36.4**	**37.6**	**37.8**	**45.2**	**36.1**
	can	calc	eyegls	guitar	mean
SHG[27]	66.8	52.3	44.6	45.8	42.62
CoKe-SHG	**70.2**	**57.7**	**51.3**	**49.6**	**46.88**

Table 4. Keypoint Detection for novel categories results on ObjectNet3D+ [42]. Here, we provide results on 9 categories which doesn't have too much annotation errors and truncated objects.

Occlusion Level	Lv.0	Lv.1	Lv.2	Lv.3
No clutter	79.3	75.4	71.8	65.8
Image-specific clutter	92.8	82.7	76.5	69.2
Clutter Bank (64 groups)	93.0	83.6	**80.1**	**73.3**
Clutter Bank (256 groups)	94.3	84.3	77.7	71.0
Clutter Bank (1024 groups)	**95.5**	**85.9**	79.0	70.6
Clutter Bank w/o clutter loss	94.2	83.1	76.8	68.0

Table 5. Ablation study on PASCAL3D+ with different settings for contrastive learning: no clutter features, image-specific clutter features (using 20 features from the same image as negative examples), our proposed Clutter Bank with different number of groups(each group contains 20 features) and deactivating clutter loss. Note the benefit of using clutter features in general, and in particular using a large clutter bank, as well as the importance of the clutter loss.

4.3. Qualitative Results.

Detection Results Visualization. We visualize qualitative results in Figure 5. Overall, the illustrations demonstrate the robustness of CoKe to partial occlusions. Any keypoints that are not in the vicinity of occluders are correctly detected and not affected by the occlusion. Furthermore, keypoints that are partially occluded (e.g. the wheel) can still be located robustly, although the detections tend to move away from the occluder. Importantly, we do not observe false positive detections at locally ambiguous keypoints. This demonstrates that CoKe leverages the large receptive field to disambiguate keypoints, while still being able to localize individual keypoints accurately.

Feature Map Visualization. Here we also provide some visualization of the feature map from CoKe-Res50. All of them are from Car category which has more images and more complex situations in PASCAL3D+. From the Figure 6 we can observe that CoKe is robust to challenging scenarios. It is worth noting that how all keypoints are detected accurately despite the difficulty from false positives, occlusions, rare viewpoints, different domain, irregular appearance and irregular state and rare viewpoints.

Inference Time and Memory Consumption. During inference, CoKe-Res50 (params: 23M, acc: 77%) takes 0.01s per image, while (SHGs) (params: 25M, acc: 68%) needs 0.06s per image. For the memory consumption, CoKe-Res50 needs 715MB, SHGs 786MB when batch size equals to 1. CoKe has an advantage of the inference time while maintaining the competence of the memory consumption.

4.4. Ablation Study

4.4.1 Clutter Bank Mechanism

In Table 5, we study the influence of the clutter features and the clutter loss on the contrastive learning result. In particular, it shows the keypoint detection results for CoKe-Res-UNet on the car category of the PASCAL3D+ dataset. We observe that the performance decreases significantly when no clutter features are used during training. An extension of this basic setup is to use image-specific clutter features but without maintaining the features in a Clutter Bank. In particular, we select the clutter features from the same image from which the keypoint features are sampled using the same hard negative sampling mechanism as in our standard setup. From the results we observe that using the image-specific clutter features increases the performance significantly. However, the best performance is achieved using our proposed Clutter Bank mechanism. In particular, the results show a general trend that the more features we store in the bank, the higher the performance becomes. Notably, lower occlusion scenarios benefit from a larger clutter bank while for stronger occlusion a smaller clutter bank is more beneficial. Finally, our ablation shows that explicitly regularizing the clutter to be distinct from the Keypoint Bank using the clutter loss is highly beneficial.

4.4.2 Cumulative Moving Average Update

We also study the importance of the cumulative moving average update. Here we provide another possible method: average approximation. In particular, we calculate an average over the whole dataset for $\{\theta\}$ each 10 epochs. We report the result using CoKe-Res-UNet on the car category of the PASCAL3D+ dataset. We observe a deduction of keypoint detection accuracy as L0: $94.2 \rightarrow 88.7$, L1: $83.1 \rightarrow 80.0$, L0: $76.8 \rightarrow 75.0$, L0: $68.0 \rightarrow 68.9$. Conclusively, the best performance is achieved using our proposed cumulative moving average update mechanism.

5. Conclusion

The advanced complexity of current approaches for keypoint detection and their lack of robustness motivated us to develop a simple yet effective and robust approach for keypoint detection. In particular, we considered problem of keypoint detection from the perspective of contrastive representation learning. We generalized the current paradigm of learning a single vectorial representation of an image, towards learning tensor-shaped representations. We made this possible by significantly reducing the computational

complexity through efficient approximations. Our proposed keypoint detection framework outperforms related work by a small but significant margin on popular datasets. Moreover, it simple, works for a wide range of object classes, and is exceptionally robust to partial occlusion.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 2, 6

[2] Vasileios Belagiannis and Andrew Zisserman. Recurrent human pose estimation. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 468–475. IEEE, 2017. 7

[3] Adrian Bulat and Georgios Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *European Conference on Computer Vision*, pages 717–732. Springer, 2016. 7

[4] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7291–7299, 2017. 1, 2

[5] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 11–14. Springer, 2009. 3

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 3

[7] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018. 3

[8] Nicolas Gourier, Daniela Hall, and James L Crowley. Facial features detection robust to pose, illumination and identity. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, volume 1, pages 617–622. IEEE, 2004. 2

[9] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 3

[10] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 2, 3

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[12] Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019. 3

[13] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 5

[14] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018. 3

[15] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision*, pages 34–50. Springer, 2016. 7

[16] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–386, 2018. 1

[17] Lipeng Ke, Ming-Ching Chang, Honggang Qi, and Siwei Lyu. Multi-scale structure-aware network for human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 713–728, 2018. 1, 2, 7

[18] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020. 3

[19] Adam Kortylewski, Ju He, Qing Liu, and Alan Yuille. Compositional convolutional neural networks: A deep architecture with innate robustness to partial occlusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 6

[20] Adam Kortylewski, Qing Liu, Huiyu Wang, Zhishuai Zhang, and Alan Yuille. Combining compositional models and deep networks for robust object classification under occlusion. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1333–1341, 2020. 1, 6

[21] Sijin Li and Antoni B Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *Asian Conference on Computer Vision*, pages 332–347. Springer, 2014. 2

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 6

[23] Jonathan L Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In *Advances in neural information processing systems*, pages 1601–1609, 2014. 1

[24] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 2

[25] Ross Messing, Chris Pal, and Henry Kautz. Activity recognition using the velocity histories of tracked keypoints. In *2009 IEEE 12th international conference on computer vision*, pages 104–111. IEEE, 2009. 1

[26] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020. 3

[27] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016. 1, 2, 8

[28] Guanghan Ning and Zhihai He. Dual path networks for multi-person human pose estimation. *arXiv preprint arXiv:1710.10192*, 2017. 7

[29] Georgios Pavlakos, Xiaowei Zhou, Aaron Chan, Konstantinos G Derpanis, and Kostas Daniilidis. 6-dof object pose from semantic keypoints. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2011–2018. IEEE, 2017. 1, 2, 3

[30] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 2, 3

[31] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems*, pages 1857–1865, 2016. 3

[32] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 3

[33] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–656, 2015. 2

[34] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014. 2

[35] Shubham Tulsiani and Jitendra Malik. Viewpoints and keypoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1510–1519, 2015. 2, 3, 6

[36] Angtian Wang, Yihong Sun, Adam Kortylewski, and Alan Yuille. Robust object detection under occlusion with context-aware compositionalnets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 6

[37] Jianyu Wang, Cihang Xie, Zhishuai Zhang, Jun Zhu, Lingxi Xie, and Alan Yuille. Detecting semantic parts on partially occluded objects. *arXiv preprint arXiv:1707.07819*, 2017. 6

[38] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016. 7

[39] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(2), 2009. 3

[40] Jiajun Wu, Tianfan Xue, Joseph J Lim, Yuandong Tian, Joshua B Tenenbaum, Antonio Torralba, and William T Freeman. Single image 3d interpreter network. In *European Conference on Computer Vision*, pages 365–382. Springer, 2016. 2

[41] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 3, 5

[42] Yu Xiang, Wonhui Kim, Wei Chen, Jingwei Ji, Christopher Choy, Hao Su, Roozbeh Mottaghi, Leonidas Guibas, and Silvio Savarese. Objectnet3d: A large scale database for 3d object recognition. In *European Conference on Computer Vision*, pages 160–176. Springer, 2016. 2, 6, 8

[43] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE winter conference on applications of computer vision*, pages 75–82. IEEE, 2014. 2, 6

[44] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. In *European conference on computer vision*, pages 834–849. Springer, 2014. 1

[45] Zhengxin Zhang, Qingjie Liu, and Yunhong Wang. Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 15(5):749–753, 2018. 6

[46] Xingyi Zhou, Arjun Karpur, Linjie Luo, and Qixing Huang. Starmap for category-agnostic keypoint and viewpoint estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 318–334, 2018. 1, 2, 3, 6

[47] Hongru Zhu, Peng Tang, Jeongho Park, Soojin Park, and Alan Yuille. Robustness of object recognition under extreme occlusion in humans and computational models. *CogSci Conference*, 2019. 1