Learning from Synthetic Animals

Jiteng Mu*, Weichao Qiu*, Gregory Hager, Alan Yuille
Johns Hopkins University
jitengmu@jhu.edu, hager@cs.jhu.edu, {qiuwch, alan.l.yuille}@gmail.com

Abstract

Despite great success in human parsing, progress for parsing other deformable articulated objects, like animals, is still limited by the lack of labeled data. In this paper, we use synthetic images and ground truth generated from CAD animal models to address this challenge. To bridge the domain gap between real and synthetic images, we propose a novel consistency-constrained semi-supervised learning method (CC-SSL). Our method leverages both spatial and temporal consistencies, to bootstrap weak models trained on synthetic data with unlabeled real images. We demonstrate the effectiveness of our method on highly deformable animals, such as horses and tigers. Without using any real image label, our method allows for accurate keypoint prediction on real images. Moreover, we quantitatively show that models using synthetic data achieve better generalization performance than models trained on real images across different domains in the Visual Domain Adaptation Challenge dataset. Our synthetic dataset contains 10+ animals with diverse poses and rich ground truth, which enables us to use the multi-task learning strategy to further boost models' performance.

1. Introduction

Thanks to the presence of large scale annotated datasets and powerful Convolutional Neural Networks(CNNs), the state of human parsing has advanced rapidly. By contrast, there is little previous work on parsing animals. Parsing animals is important for many tasks, including, but not limited to monitoring wild animal behaviors, developing bio-inspired robots, building motion capture systems, etc.

One main problem for parsing animals is the limit of datasets. Though many datasets containing animals are built for classification, bounding box detection, and instance segmentation, only a small number of datasets are built for parsing animal keypoints and parts. Annotating large scale datasets for animals is prohibitively expensive. Therefore, most existing approaches for parsing humans, which often require enormous annotated data [1, 32], are less suited for parsing animals.

In this work, we use synthetic data to address this challenge. Many works [34, 29] show that by jointly using synthetic images and real images, models can yield supreme results. In addition, synthetic data also has many unique advantages compared to real-world datasets. First, rendering synthetic data with rich ground truth at scale is easier and cheaper compared with capturing and annotating real-world images. Second, synthetic data can also provide accurate ground truth for cases where annotations are hard to acquire for natural images, such as labeling optical flow [11] or under occlusion and low-resolution. Third, real-world datasets usually suffer from the long-tail problem where rare cases are less represented. Generated synthetic datasets can avoid this problem by sampling rendering parameters.

However, there are large domain gaps [7, 38, 14] between synthetic images and real images, which prevent models trained on synthetic data from generalizing well to real-world images. Moreover, synthetic data is also limited by object diversity. ShapeNet [6] has been created to include diverse 3D models and SMPL [24] has been built for humans. Nevertheless, creating such diverse synthetic models is a difficult task, which requires capturing the appearance and attaching a skeleton to the object. Besides, considering the number of animal categories in the world, creating diverse synthetic models along with realistic textures for each animal is almost infeasible.

In this paper, we propose a method where models are trained using synthetic CAD models. Our method can achieve high performance with only a single CAD animal model. We generate pseudo-labels on unlabeled real images for semi-supervised learning. To handle noisy pseudo-labels, we design three consistency-check criteria to evaluate the quality of the predicted labels, which we refer to as consistency-constrained semi-supervised learning (CC-SSL). Through extensive experiments, we show that our models achieve similar performance to models trained on real data, but without using any annotation of real images. It also outperforms other domain adaptation meth-

* Indicates equal contributions.

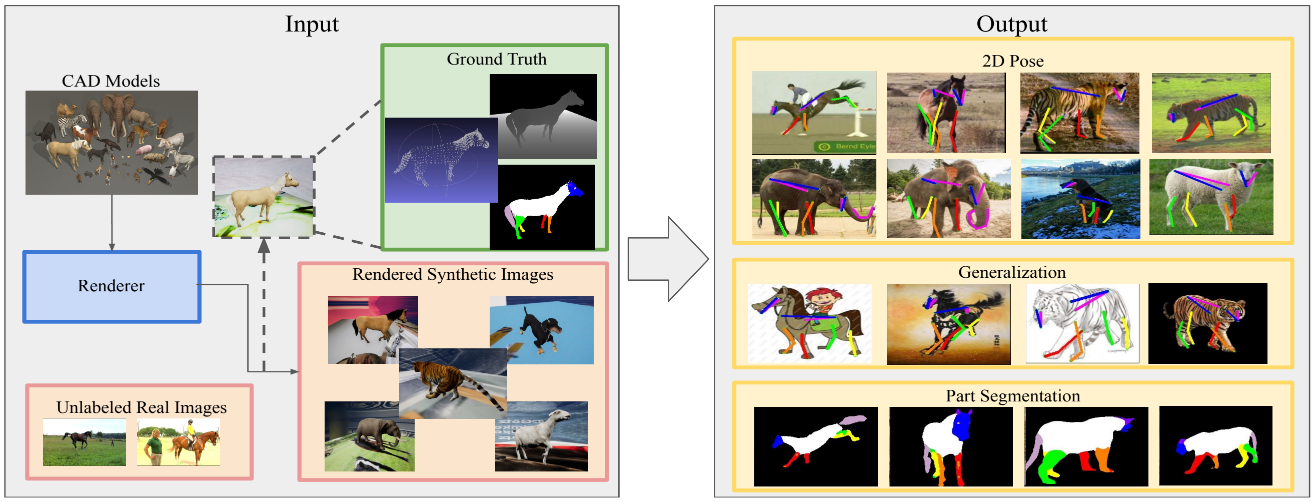

Figure 1. Overview. We generate a synthetic animal dataset by randomly sampling rendering parameters including camera viewpoints, lighting, textures and poses. The dataset contains 10+ animals along with rich ground truth, such as dense 2D keypoints, part segmentation and depth maps. With the synthetic dataset, we propose an effective method which allows for accurate keypoint prediction across domains. In addition to 2D pose estimation, we also show models can predict accurate part segmentation.

ods by a large margin. Providing real image annotations, the performance can be further improved. Furthermore, we demonstrate models trained with synthetic data show better domain generalization performance in multiple visual domains compared with those trained on real data. The code is available at https://github.com/JitengMu/Learning-from-Synthetic-Animals.

We summarize the contributions of our paper as follows. First, we propose a consistency-constrained semi-supervised learning framework (CC-SSL) to learn a model with one single CAD object. We show that models trained with synthetic data and unlabeled real images allow for accurate keypoint prediction on real images. Second, when using real image labels, we show that models trained jointly on synthetic and real images achieve better results compared to models trained only on real images. Third, we evaluate the generalizability of our learned models across different visual domains in the Visual Domain Adaptation Challenge dataset and we quantitatively demonstrate that models trained using synthetic data show better generalization performance than models trained on real images. Lastly, we generate an animal dataset with 10+ different animal CAD models and we demonstrate the data can be effectively used for 2D pose estimation, part segmentation, and multi-task learning.

2. Related Work

2.1. Animal Parsing

Though there exists large scale datasets containing animals for classification, detection, and instance segmentation, there are only a small number of datasets built for pose estimation [28, 39, 5, 27, 20] and animal part segmentation [8]. Besides, annotating keypoints or parts is time-consuming and these datasets only cover a tiny portion of animal species in the world.

Due to the lack of annotations, synthetic data has been widely used to address the problem [43, 3, 44, 45]. Similar to SMPL models [24] for humans, [45] proposes a method to learn articulated SMAL shape models for animals. Later, [44] extracts more 3D shape details and is able to model new species. Unfortunately, these methods are built on manually extracted silhouettes and keypoint annotations. Recently, [43] proposes to copy texture from real animals and predicts 3D mesh of animals in an end-to-end manner. Most related to our method is [3], where authors propose a method to estimate animal poses on real images using synthetic silhouettes. Different from [3] which requires an additional robust segmentation model for real images during inference, our strategy does not require any additional models.

2.2. Unsupervised Domain Adaptation

Unsupervised domain adaptation focuses on learning a model that works well on a target domain when provided with labeled source samples and unlabeled target samples. A number of image-to-image translation methods [22, 40, 15] are proposed to transfer images from different domains. Another line of work studies how to explicitly minimize some measure of feature difference, such as maximum mean discrepancy [37, 23] or correlation distances [33, 35]. [4] proposes to explicitly partition features into a shared space and a private space. Recently, adversarial loss [36, 14] is used to learn domain invariant features, where a domain classifier is trained to distinguish the source and target distributions. [36] proposes a general framework to bring features from different domains closer. [14, 25] extend this idea with cycle consistency to improve results.

Recent works have also investigated how to use these techniques to advance deformable objects parsing. [7] studies using synthetic human images combined with domain adaptation to improve human 3D pose estimation. [38] renders 145 realistic synthetic human models to reduce the domain gap. Different from previous works where a large amount of realistic synthetic models are required, we show that models trained on one CAD model can learn domain-invariant features.

2.3. Self-training

Self-training has been proved effective in semi-supervised learning. Early work [19] draws the connection between deep self-training and entropy regularization. However, since generated pseudo-labels are noisy, a number of methods [17, 10, 41, 42, 18, 12, 21, 9, 30, 31] are proposed to address the problem. [41, 42] formulate self-training as a general EM algorithm and proposes a confidence regularized self-training framework. [18] proposes a self-ensembling framework to bootstrap models using unlabeled data. [12] extends the previous work to unsupervised domain adaptation and demonstrate its effectiveness in bridging domain gaps.

Closely related to our work on 2D pose estimation is [30], where the authors propose a simple method for omni-supervised learning that distills knowledge from unlabeled data and demonstrate its effectiveness on detection and pose estimation. However, under large domain discrepancy, the assumption that the teacher model assigns high-confidence pseudo-labels is not guaranteed. To tackle the problem, we introduce a curriculum learning strategy [2, 13, 16] to progressively increase pseudo-labels and train models in iterations. We also extend [30] by leveraging both spatial and temporal consistencies.

3. Approach

We first formulate a unified image generation procedure in Section 3.1 built on the low dimension manifold assumption. In Section 3.2, we define three consistencies and discuss how to take advantage of these consistencies during pseudo-label generation process. In Section 3.3, we propose a Pseudo-Label Generation algorithm using consistency-check. Then in Section 3.4 we present our consistency-constrained semi-supervised learning algorithm and discuss the iterative training pipeline. Lastly, in Section 3.5, we explain how our synthetic datasets are generated.

We consider the problem under unsupervised domain adaptation framework with two datasets. We name our synthetic dataset as the source dataset (X_s, Y_s) and real images as the target dataset X_t. The goal is to learn a model f to predict labels for the target data X_t. We simply start with learning a source model f_s using paired data (X_s, Y_s) in a fully supervised way. Then we bootstrap the source model using target dataset with consistency-constrained semi-supervised learning. An overview of the pipeline is presented in Figure 2.

3.1. Formulate Image Generation Procedure

In order to learn a model using synthetic data that can generalize well to real data, one needs to assume that there exists some essential knowledge shared between these two domains. Take animal 2D pose estimation as an example, though synthetic and natural images look differently by textures and background, they are quite similar in terms of poses and shape. Actually, these are exactly what we hope a model trained on synthetic data can learn. So an ideal model should be able to capture these essential factors and ignore those less relevant ones, such as lighting and background.

Formally, we introduce a generator G that transforms poses, shapes, viewpoints, textures, etc, into an image. Mathematically, we group all these factors into two categories, task-related factors α, which is what a model cares about, and others β, which are irrelevant to the task at hand. So we parametrize the image generation process as follows,

$$X = G(\alpha, \beta) \tag{1}$$

where X is a generated image and G denotes the generator. Specifically, for 2D pose estimation, α represents factors related to the 2D keypoints, such as pose and shape; β indicates factors independent of α, which could be textures, lighting and background.

3.2. Consistency

Based on the formulation in Section 3.1, we define three consistencies and discuss how to take advantage of these consistencies for the pseudo-label generation process.

Since model-generated labels on the target dataset are noisy, one needs to tell the model which predictions are correct and which are wrong. Intuitively, an ideal 2D keypoint detector should generate consistent predictions on one image no matter how the background is perturbed. In addition, if one rotates the image, the prediction should change accordingly as well. Based on these intuitions, we propose to use consistency-check to reduce false positives.

In the following paragraphs, we will introduce invariance consistency, equivariance consistency and temporal consistency. We will discuss how to use consistency-check to generate pseudo-labels, which serves as the basis for the proposed semi-supervised learning method.

The transformation applied to an image can be considered as directly transforming the underlying factors in Equation 1. We define a general tensor operator, T :

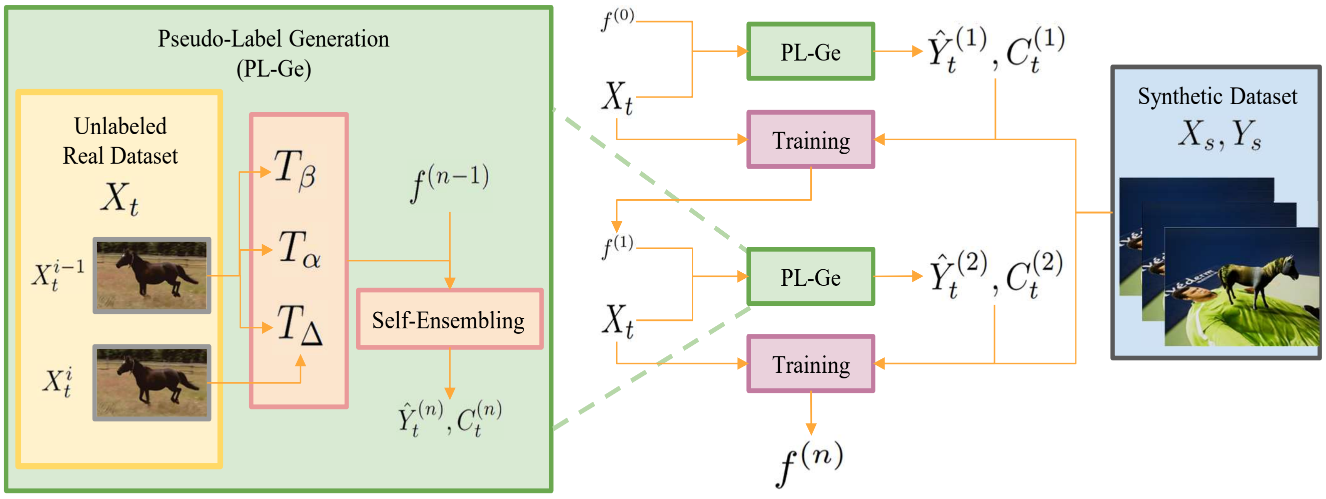
Pseudo-Label Generation (PL-Ge)
Unlabeled Real Dataset X_t
X_t^{i-1}
X_t^i
T_β
T_α
T_Δ
$f^{(n-1)}$
Self-Ensembling
$\hat{Y}_t^{(n)}, C_t^{(n)}$
$f^{(0)}$
X_t
PL-Ge
Training
$\hat{Y}_t^{(1)}, C_t^{(1)}$
$f^{(1)}$
$\hat{Y}_t^{(2)}, C_t^{(2)}$
$f^{(n)}$
Synthetic Dataset X_s, Y_s

Figure 2. Consistency-constrained semi-supervised learning pipeline. T_β indicates the invariance consistency, T_α indicates the equivariance consistency and T_Δ indicates the temporal consistency. The training procedure can be described as follows: we start with training a model only using synthetic data and obtain an initial model $f^{(0)}$. Then we iterate the following procedure. For the nth iteration, we first use the proposed Pseudo-Label Generation Algorithm 1 to generate labels $\hat{Y}_t^{(n)}$. Next, we train the model using (X_s, Y_s) and $(X_t, \hat{Y}_t^{(n)})$ jointly.

$\mathbb{R}^{H\times W} \rightarrow \mathbb{R}^{H\times W}$. In addition, we introduce τ_α corresponding to operations that would affect α and τ_β to represent operations independent of α. Then Equation 1 can be expressed as following,

$$T(X) = G(\tau_\alpha(\alpha), \tau_\beta(\beta)) \tag{2}$$

We use $f : \mathbb{R}^{H\times W} \rightarrow \mathbb{R}^{H\times W}$ to denote a perfect 2D pose estimation model. When f is applied to Equation 2, it is obvious that, $f[T(X)] = f[G(\tau_\alpha(\alpha), \tau_\beta(\beta))]$.

Invariance consistency: If the transform T does not change factors associated with the task, the model's prediction is expected to be the same. The idea here is that a well-behaved model should be invariant to operations on β. For example, in 2D pose estimation, adding noise to the image or perturbing colors should not affect the model's prediction. We name these transforms *invariance transform* T_β, as shown in Equation 3.

$$f[T_\beta(X)] = f(X) \tag{3}$$

If we apply multiple invariance transforms to the same image, the predictions on these transformed images should be consistent. This consistency can be used to verify whether the prediction is correct, which we refer to as *invariance consistency*.

Equivariance consistency: Besides invariance transform, there are other cases where the task related factors are changed. We use T_α to denote transforms related to operations τ_α. There are special cases where we can easily get the corresponding T_α. One easy case is that, sometimes, the effect of τ_α only cause geometric transformations in 2D images, which we refer to as *equivariance transform* T_α. Actually, this is essentially similar to what [30] proposes. Therefore, we have *equivariance consistency* as shown in Equation 4.

$$f[T_\alpha(X)] = T_\alpha[f(X)] \tag{4}$$

It is also easy to show that $f(X) = T_\alpha^{-1}[f[T_\alpha(X)]]$, which means that, after applying the inverse transform T_α^{-1}, a good model should give back the original prediction.

Temporal consistency: It is difficult to model transformations between frames in a video. This transform T_Δ does not satisfy the invariance and equivariance properties described above. However, T_Δ is still induced by variations of underlying factors α and β. It is reasonable to assume that, in a real-world video, these factors do not change dramatically between neighboring frames.

$$f[T_\Delta(X)] = f(X) + \Delta \tag{5}$$

So we assume the keypoints shifting between two frames is relatively small as shown in Equation 5. Intuitively, this means that the keypoint prediction for the same joint in consecutive frames should not be too far away, otherwise it is likely to be incorrect.

For 2D keypoint estimation, we observe that T_Δ can be approximated by optical flow to get Δ, which allows us to use optical flow to propagate pseudo-labels from confident frames to less confident ones.

Although we define these three consistencies for 2D pose estimation, they can be easily extended to other problems. For example, in 3D pose estimation, α can be factors related to 3D pose. Then the invariance consistency is still the same, but the equivariance consistency no longer holds, since the mapping of 3D pose to 2D pose is not a one-to-one mapping and there are ambiguities in the depth dimension. However, one can still use it as a constraint for the other two dimensions, which means the projected poses should still satisfy the same consistency. So it is easy to see that though corresponding consistencies may change for different tasks, they all follow the same philosophy.

Algorithm 1 Pseudo-Label Generation Algorithm

```
Input: Target dataset $X_t$; model $f^{(n-1)}$; decay factor $\lambda_{decay}$.
Intermediate Result: $P_\beta$, $P_\alpha$ are predictions after applying invariance and equivariance transform.
Output: Pseudo-labels $\hat{Y}_t^{(n)}$; confidence score $C_t^{(n)}$.
 1: for $X_t^i$ in $X_t$ do
 2:                                              ▷ Invariance Consistency
 3:     $P_\beta = f^{(n-1)}(T_\beta(X_t^i))$
 4:                                            ▷ Equivariance Consistency
 5:     $P_\alpha = T_\alpha^{-1}[f^{(n-1)}(T_\alpha(X_t^i))]$
 6:                                                  ▷ Self-Ensembling
 7:     Ensemble $P_\beta$ and $P_\alpha$ to get ($\hat{Y}_t^{(n),i}$, $C_t^{(n),i}$)
 8:                                              ▷ Temporal Consistency
 9:     if $C_t^{(n),i}/C_t^{(n),i-1} < \lambda_{decay}$ then
10:         $\hat{Y}_t^{(n),i} = (\hat{Y}_t^{(n),i-1}) + \Delta$
11:         $C_t^{(n),i} = \lambda_{decay} * C_t^{(n),i-1}$
12:     end if
13: end for
14: Sort $C_t^{(n)}$ and obtain $C_{thresh}$ based on a fixed curriculum learning policy.
15: Set $C_t^{(n),i} = \mathbb{1}(C_t^{(n),i} \geq C_{thresh}), \quad \forall i$
```

3.3. Pseudo-Label Generation

In this section, we explain in details how to apply these consistencies in practice for generating pseudo-labels and propose the pseudo-label generation method as in Algorithm 1.

We address the noisy label problem in two ways. First, we develop an algorithm to generate pseudo-labels using consistency-check to remove false positives, assuming that labels generated using the correct information always satisfy these consistencies. Second, we apply the curriculum learning idea to gradually increase the number of training samples and learn models in an iterative fashion.

For the nth iteration, with the previous model $f^{(n-1)}$ obtained from the $(n-1)$th iteration, we iterate through each image X_t^i in the target dataset X_t. $f^{(n-1)}$ is not updated in this process. First, for each image, we apply multiple invariance transform T_β, equivariance transform T_α to X_t^i, and ensemble all predictions P_β and P_α to get a pair of estimated labels and confidence scores ($\hat{Y}_t^{(n),i}$, $C_t^{(n),i}$).

Second, we use temporal consistency to update weak predictions. For each keypoint, we check whether the current confidence score $C_t^{(n),i}$ is strong compared to the one in the previous frame $C_t^{(n),i-1}$ with respect to a decay factor λ_{decay}. If the current frame prediction is confident, we simply keep it; otherwise, we replace the prediction $\hat{Y}_t^{(n),i}$ with the flow prediction Δ plus the previous frame prediction and replace $C_t^{(n),i}$ with previous frame confidence multiplied by a decay factor λ_{decay}. Temporal consistency is optional and can be used if videos are available.

To this end, the algorithm has generated labels and confidence scores for all images. The last step is to iterate through the target dataset again to select C_{thresh} using a curriculum learning strategy, which determines the percentage of labels used for training. The idea here is to use keypoints with high confidence first and gradually include more keypoints after iterations. In practice, we use a policy to include top 20% ranking keypoints at the beginning, 40% for the second iteration, until hitting 80%.

3.4. Consistency-Constrained Semi-Supervised Learning (CC-SSL)

For the nth iteration, the loss function $L^{(n)}$ is defined to be the Mean Square Error on heatmaps of both the source data and target data, as in Equation 6. γ is used to balance the loss between source and target datasets.

$$L^{(n)} = \sum_i L_{MSE}(f^{(n)}(X_s^i), Y_s^i) + \gamma \sum_j L_{MSE}(f^{(n)}(X_t^j), \hat{Y}_t^{(n-1),j}) \tag{6}$$

To this end, we present our Consistency-Constrained Semi-Supervised Learning (CC-SSL) approach as follows: we start with training a model only using synthetic data and obtain an initial weak model $f^{(0)} = f_s$. Then we iterate the following procedure. For the nth iteration, we first use Algorithm 1 to generate labels $\hat{Y}_t^{(n)}$. With the generated labels, we simply train the model using (X_s, Y_s) and $(X_t, \hat{Y}_t^{(n)})$ jointly using $L^{(n)}$.

3.5. Synthetic Dataset Generation

In order to create a diverse combination of animal appearances and poses, we collect a synthetic animal dataset containing 10+ animals. Each animal comes with several animation sequences. We use Unreal Engine to collect rich ground truth and enable nuisance factor control. The implemented factor control includes randomizing lighting, textures, changing viewpoints and animal poses.

The pipeline for generating synthetic data is as follows. Given a CAD model along with a few animation sequences, an animal with random poses and random texture is rendered from a random viewpoint for some random lighting and a random background image. We also generate ground truth depth maps, part segmentation and dense joint locations (both 2D and 3D). See Figure 1 for samples from the synthetic dataset.

4. Experiments

First, we quantitatively test our approach on the TigDog dataset [28] in Section 4.2. We compare our method with other popular unsupervised domain adaptation methods, such as CycleGAN [40], BDL [21] and CyCADA [14]. We also qualitatively show keypoint detection of other animals where no labeled real images are available, such as elephants, sheep and dogs. Second, in order to show the domain generalization ability, we annotated the keypoints of animals from Visual Domain Adaptation Challenge dataset (VisDA2019). In Section 4.3, we evaluate our models on these images from different visual domains. Third, the rich ground truth in synthetic data enables us to do more tasks beyond 2D pose estimation, so we also visualize part segmentation on horses and tigers and demonstrate the effectiveness of multi-task learning in Section 4.4.

4.1. Experiment Setup

Network Architecture. We use Stacked Hourglass [26] as our backbone for all experiments. Architecture design is not our main focus and we strictly follow parameters from the original paper. Each model is trained with RMSProp for 100 epochs. The learning rate starts with $2.5e^{-4}$ and decays twice at 60 and 90 epoches respectively. Input images are cropped with the size of 256×256 and augmented with scaling, rotation, flipping and color perturbation.

Synthetic Datasets. We explain the details of our data generation parameters as follows. The virtual camera has a resolution of 640×480 and field of view of 90. We randomize synthetic animal textures and backgrounds using Coco val2017 dataset. We does not use any segmentation annotation from coco val2017. For each animal, we generated 5,000 images with random texture and 5,000 images with the texture coming with the CAD model, to which we refer as the original texture. We split the training set and validation set with a ratio of 4:1, resulting in 8,000 images for training and 2,000 for validation. We also generate rich ground truth including part segmentation, depth maps and dense 2D and 3D keypoints. For part segmentation, we define nine parts for each animal, which are eyes, head, ears, torso, left-front leg, left-back leg, right-front leg, right-back leg and tail. The parts definition follows [8] with a minor difference that we distinguish front from back legs. CAD models used in this paper are purchased from UE4 marketplace[1].

[1]https://www.unrealengine.com/marketplace/en-US/product/animal-pack-ultra-01

CC-SSL In our experiments, we pick scaling and rotation from T_α and obtain Δ using optical flow. λ_{decay} is set to 0.9 and we train one model for 10 epochs and re-generate pseudo labels with the new model. Models are trained for 60 epochs with γ set to be 10.0.

TigDog Dataset The TigDog dataset is a large dataset containing 79 videos for horses and 96 videos for tigers. In total, for horse, we have 8380 frames for training and 1772 frames for testing. For tigers, we have 6523 frames for training and 1765 frames for testing. Each frame is provided with 19 keypoint annotations, which are defined as eyes(2), chin(1), shoulders(2), legs(12), hip(1) and neck(1). The neck keypoint is not clearly distinguished for left and right, so we leave it out in all experiments.

4.2. 2D Pose Estimation

Results Analysis. Our main results are summarized in Table 1. We present our results in two different setups: the first one is under the unsupervised domain adaptation setting where real image annotations are not available; the second one is when labeled real images are available.

When annotations of real images are not available, our proposed **CC-SSL** surpasses other methods by a significant margin. The PCK@0.05 accuracy of horses reaches 70.77, which is close to models trained directly on real images. For tigers, the proposed method achieves 64.14. It is worth noticing that these results are achieved without accessing any real image annotation, which demonstrated the effectiveness of our proposed method.

We also visualize the predicted keypoints in Figure 3. Even for some extreme poses, such as horse riding and lying on the ground, the method still generate accurate predictions. The observations for tigers are similar.

When annotations of real images are available, our proposed **CC-SSL-R** achieves 82.43 for horses and 84.00 for tigers, which are noticeably better than models trained on real images only. CC-SSL-R is achieved simply by further finetuning the CC-SSL models using real image labels.

In addition to horses and tigers, we apply the method to other animals as well. The method can be easily transferred to other animal categories and we qualitatively show keypoint prediction results for other animals, as shown in Figure 4. Notice that our method can also detect trunks of elephants.

We empirically find the performance does not improve much with CycleGAN. We conjecture that one reason is that CycleGAN in general requires a large number of real images to work well. However, in our case, the diversity of real images is limited. Another reason is that animal shapes of transferred images are not maintained well. We also try different adversarial training strategies. Though BDL works quite well for semantic segmentation, we find the improvements on keypoints detection is small. CyCADA also suf-

	Horse Accuracy								Tiger Accuracy							
	Eye	Chin	Shoulder	Hip	Elbow	Knee	Hoove	Mean	Eye	Chin	Shoulder	Hip	Elbow	Knee	Hoove	Mean
synthetic + real																
Real	79.04	89.71	**71.38**	91.78	82.85	80.80	72.76	78.98	**96.77**	93.68	65.90	94.99	67.64	80.25	81.72	81.99
CC-SSL-R	**89.39**	**92.01**	69.05	**92.28**	**86.39**	**83.72**	**76.89**	**82.43**	95.72	**96.32**	**74.41**	**91.64**	**71.25**	**82.37**	**82.73**	**84.00**
synthetic only																
Syn	46.08	53.86	20.46	32.53	20.20	24.20	17.45	25.33	23.45	27.88	14.26	52.99	17.32	16.27	19.29	21.17
CycleGAN [40]	70.73	84.46	56.97	69.30	52.94	49.91	35.95	51.86	71.80	62.49	29.77	61.22	36.16	37.48	40.59	46.47
BDL [21]	74.37	86.53	64.43	75.65	63.04	60.18	51.96	62.33	77.46	65.28	36.23	62.33	35.81	45.95	54.39	52.26
CyCADA [14]	67.57	84.77	56.92	76.75	55.47	48.72	43.08	55.57	75.17	69.64	35.04	65.41	38.40	42.89	48.90	51.48
CC-SSL	**84.60**	**90.26**	**69.69**	**85.89**	**68.58**	**68.73**	**61.33**	**70.77**	**96.75**	**90.46**	**44.84**	**77.61**	**55.82**	**42.85**	**64.55**	**64.14**

Table 1. Horse and tiger 2D pose estimation accuracy PCK@0.05. Synthetic data are with randomized background and textures. Synthetic only shows results when no real image label is available, Synthetic + Real are cases when real image labels are available. In both scenarios, our proposed CC-SSL based methods achieve the best performance.

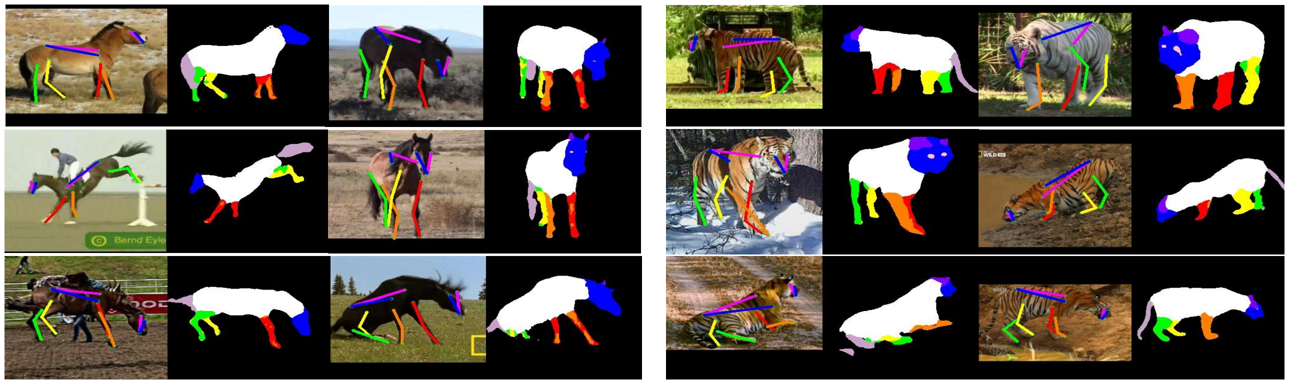

Figure 3. Visualization of horse and tiger 2D pose estimation and part segmentation prediction. The 2D pose estimations are predicted using CC-SSL as described in Section 4.2 and part segmentation predictions are generated using the multi-task learning as described in Section 4.4. Best viewed in color.

fers from the same problem as CycleGAN. In comparison, CC-SSL does not suffer from those problems and it can work well even with limited diversity of real data.

We use the same set of augmentations as in [26] for baselines **Real** and **Syn**. We use a different set of augmentations for other experiments, which we refer to as Strong Augmentation. In addition to what [26] used, Strong Augmentation also includes Affine Transform, Gaussian Noise and Gaussian Blurring.

4.3. Generalization Test on VisDA2019

In this section, we test model generalization on images from Visual Domain Adaptation Challenge dataset (VisDA2019). The dataset contains six domains: real, sketch, clipart, painting, infograph, and quickdraw. We pick up sketch, painting and clipart for our experiments since inforgraph and quickdraw are not suitable for 2D pose estimation. For each of these three domains, we manually annotate images for horses and tigers. Evaluation results are summarized in Table 2. Same as before, we use **Real** as our baseline. **CC-SSL** and **CC-SSL-R** are used for comparison.

For both animals, we observe that models trained using synthetic data achieve best performance in all settings. We present our results under two settings. Visible Keypoints Accuracy only accounts for keypoints that are directly visible whereas Full Keypoints Accuracy shows results with self-occluded keypoints.

Under all settings, CC-SSL-R is better than Real. More interestingly, notice that even without using real image labels, our CC-SSL method yields better performance than Real in almost all domains. The only one exception is the painting domain of tigers. We hypothesize that this is because texture information (yellow and black stripes) in paintings is still well preserved so models trained on real images can still “generalize”. For sketches and cliparts, appearances are more different from real images and models trained on synthetic data show better results.

4.4. Part Segmentation

Since the synthetic animal dataset is generated with rich ground truth, our task is not limited to 2D pose estimation. We also experiment with part segmentation in a multi-task learning setting. All models are trained on synthetic images with Strong Augmentation and tested on TigDog dataset directly.

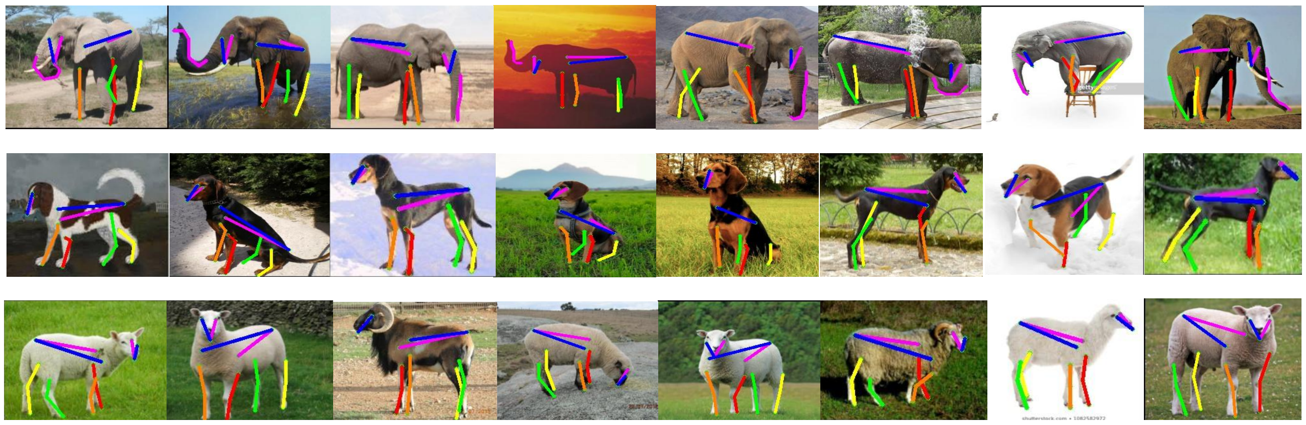
Figure 4. Visualization of 2D pose estimation of other animals. Our method can be easily generalized to elephants' trunks. Best viewed in color.

	Horse						Tiger					
	Visible Kpts Accuracy			Full Kpts Accuracy			Visible Kpts Accuracy			Full Kpts Accuracy		
	Sketch	Painting	Clipart	Sketch	Painting	Clipart	Sketch	Painting	Clipart	Sketch	Painting	Clipart
Real	65.37	64.45	64.43	61.28	58.19	60.49	48.10	61.48	53.36	46.23	53.14	50.92
CC-SSL	72.29	73.71	**73.47**	**70.31**	**71.56**	**72.24**	53.34	55.78	59.34	52.64	48.42	54.66
CC-SSL-R	**73.25**	**74.56**	71.78	67.82	65.15	65.87	**54.94**	**68.12**	**63.47**	**53.43**	**58.66**	**59.29**

Table 2. Horse and tiger 2D pose estimation accuracy PCK@0.05 on VisDA2019. We present our results under two settings: Visible Kpts Accuracy only accounts for visible keypoints; Full Kpts Accuracy also includes self-occluded keypoints. Under all settings, our proposed methods achieve better performance than baseline Real.

As shown in Table 3, we observe that models, trained on keypoints and part segmentation jointly, can generalize better on real images for both animals, compared to the baseline where models are only trained with keypoints. Since we cannot quantitatively evaluate part segmentation predictions, we visualize the part segmentation results on TigDog dataset as shown in Figure 3.

In the multi-task learning setting, we only make minor changes to the original Stacked Hourglass architecture, where we add a branch parallel to the original keypoint prediction one for part segmentation.

Models	Horse	Tiger
Baseline	60.84	50.26
+Part segmentation	**62.25**	**51.69**

Table 3. Horse and tiger 2D pose estimation PCK@0.05 with multi-task learning. We show models can generalize better to real images trained jointly using 2D keypoints and part segmentation.

5. Conclusions

In this paper, we present a simple yet efficient method using synthetic images to parse animals. To bridge the domain gap, we present a novel consistency-constrained semi-supervised learning (CC-SSL) method, which leverages both spatial and temporal constraints. We demonstrate the effectiveness of the proposed method on horses and tigers in the TigDog Dataset. Without any real image label, our model can detect keypoints reliably on real images. When using real image labels, we show that models trained jointly on synthetic and real images achieve better results compared to models trained only on real images. We further demonstrate that the models trained using synthetic data achieve better generalization performance across different domains in the Visual Domain Adaptation Challenge dataset. We build a synthetic dataset contains 10+ animals with diverse poses and rich ground truth and show that multi-task learning is effective.

Acknowledgements

Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DOI/IBC) contract number D17PC00342. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government. The authors would like to thank Chunyu Wang, Qingfu Wan, Yi Zhang for helpful discussions.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter V. Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, pages 3686–3693, 2014. 1

[2] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, pages 41–48, 2009. 3

[3] Benjamin Biggs, Thomas Roddick, Andrew W. Fitzgibbon, and Roberto Cipolla. Creatures great and SMAL: recovering the shape and motion of animals from video. *CoRR*, abs/1811.05804, 2018. 2

[4] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *NeurIPS*, pages 343–351, 2016. 2

[5] Jinkun Cao, Hongyang Tang, Haoshu Fang, Xiaoyong Shen, Cewu Lu, and Yu-Wing Tai. Cross-domain adaptation for animal pose estimation. *CoRR*, abs/1908.05806, 2019. 2

[6] 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. 1

[7] Wenzheng Chen, Huan Wang, Yangyan Li, Hao Su, Zhenhua Wang, Changhe Tu, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. Synthesizing training images for boosting human 3d pose estimation. In *3DV*, pages 479–488, 2016. 1, 3

[8] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan L. Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *CVPR*, pages 1979–1986, 2014. 2, 6

[9] Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation. *CoRR*, abs/1909.00589, 2019. 3

[10] Yifan Ding, Liqiang Wang, Deliang Fan, and Boqing Gong. A semi-supervised two-stage approach to learning from noisy labels. In *2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018*, pages 1215–1224, 2018. 3

[11] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, pages 2758–2766, 2015. 1

[12] Geoffrey French, Michal Mackiewicz, and Mark H. Fisher. Self-ensembling for visual domain adaptation. In *ICLR*, 2018. 3

[13] Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R. Scott, and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. In *ECCV*, pages 139–154, 2018. 3

[14] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, pages 1994–2003, 2018. 1, 2, 3, 6, 7

[15] Xun Huang, Ming-Yu Liu, Serge J. Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, pages 179–196, 2018. 2

[16] Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G. Hauptmann. Self-paced curriculum learning. In *AAAI*, pages 2694–2700, 2015. 3

[17] Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. NLNL: negative learning for noisy labels. *CoRR*, abs/1908.07387, 2019. 3

[18] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017. 3

[19] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2, 2013. 3

[20] Shuyuan Li, Jianguo Li, Weiyao Lin, and Hanlin Tang. Amur tiger re-identification in the wild. *CoRR*, abs/1906.05586, 2019. 2

[21] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *CVPR*, pages 6936–6945, 2019. 3, 6, 7

[22] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *NeurIPS*, pages 700–708, 2017. 2

[23] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *ICML*, pages 97–105, 2015. 2

[24] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: a skinned multi-person linear model. *ACM Trans. Graph.*, 34(6):248:1–248:16, 2015. 1, 2

[25] Zak Murez, Soheil Kolouri, David J. Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *CVPR*, pages 4500–4509, 2018. 3

[26] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII*, pages 483–499, 2016. 6, 7

[27] David Novotný, Diane Larlus, and Andrea Vedaldi. I have seen enough: Transferring parts across categories. In *BMVC*, 2016. 2

[28] Luca Del Pero, Susanna Ricco, Rahul Sukthankar, and Vittorio Ferrari. Articulated motion discovery using pairs of trajectories. In *CVPR*, pages 2151–2160, 2015. 2, 6

[29] Aayush Prakash, Shaad Boochoon, Mark Brophy, David Acuna, Eric Cameracci, Gavriel State, Omer Shapira, and Stan Birchfield. Structured domain randomization: Bridging the reality gap by context-aware synthetic data. In *ICRA*, pages 7249–7255, 2019. 1

[30] Ilija Radosavovic, Piotr Dollár, Ross B. Girshick, Georgia Gkioxari, and Kaiming He. Data distillation: Towards omni-supervised learning. In *CVPR*, pages 4119–4128, 2018. 3, 4

[31] Aruni Roy Chowdhury, Prithvijit Chakrabarty, Ashish Singh, SouYoung Jin, Huaizu Jiang, Liangliang Cao, and Erik G. Learned-Miller. Automatic adaptation of object detectors to

new domains using self-training. In *CVPR*, pages 780–790, 2019. 3

[32] Benjamin Sapp and Ben Taskar. MODEC: multimodal decomposable models for human pose estimation. In *CVPR*, pages 3674–3681, 2013. 1

[33] Baochen Sun and Kate Saenko. Deep CORAL: correlation alignment for deep domain adaptation. In *ECCV Workshops*, pages 443–450, 2016. 2

[34] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *CVPR*, pages 969–977, 2018. 1

[35] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *ICCV*, pages 4068–4076, 2015. 2

[36] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, pages 2962–2971, 2017. 2, 3

[37] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474, 2014. 2

[38] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, pages 4627–4635, 2017. 1, 3

[39] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 2

[40] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV 2017*, pages 2242–2251, 2017. 2, 6, 7

[41] Yang Zou, Zhiding Yu, B. V. K. Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, pages 297–313, 2018. 3

[42] Yang Zou, Zhiding Yu, Xiaofeng Liu, B. V. K. Vijaya Kumar, and Jinsong Wang. Confidence regularized self-training. *CoRR*, abs/1908.09822, 2019. 3

[43] Silvia Zuffi, Angjoo Kanazawa, Tanya Y. Berger-Wolf, and Michael J. Black. Three-d safari: Learning to estimate zebra pose, shape, and texture from images ”in the wild”. *CoRR*, abs/1908.07201, 2019. 2

[44] Silvia Zuffi, Angjoo Kanazawa, and Michael J. Black. Lions and tigers and bears: Capturing non-rigid, 3d, articulated shape from images. In *CVPR*, pages 3955–3963, 2018. 2

[45] Silvia Zuffi, Angjoo Kanazawa, David W. Jacobs, and Michael J. Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *CVPR*, pages 5524–5532, 2017. 2