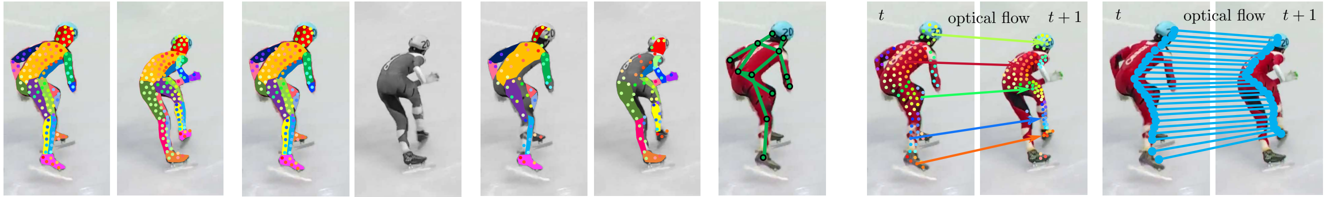
Slim DensePose: Thrifty Learning from Sparse Annotations and Motion Cues

Natalia Neverova[1] James Thewlis[2*] Rıza Alp Güler[3] Iasonas Kokkinos[3*] Andrea Vedaldi[1]
[1]Facebook AI Research [2]University of Oxford [3]Ariel AI



full annotations limited dense annotations sparse annotations keypoints

(a) different strategies of reducing the annotation load

GT propagation equivariance

(b) additional sources self-supervision

Figure 1: The goal of this work is to discover effective and cost-efficient data annotation strategies for the task of learning dense correspondences in the wild (DensePose). We significantly reduce the annotation effort by exploiting (a) sparse subsets of the DensePose labels augmented with cheaper kinds of annotations, such as object masks or keypoints, and (b) temporal information in videos to propagate ground truth and enforce dense spatio-temporal equivariance constraints.

Abstract

DensePose supersedes traditional landmark detectors by densely mapping image pixels to body surface coordinates. This power, however, comes at a greatly increased annotation cost, as supervising the model requires to manually label hundreds of points per pose instance. In this work, we thus seek methods to significantly slim down the DensePose annotations, proposing more efficient data collection strategies. In particular, we demonstrate that if annotations are collected in video frames, their efficacy can be multiplied for free by using motion cues. To explore this idea, we introduce DensePose-Track, a dataset of videos where selected frames are annotated in the traditional DensePose manner. Then, building on geometric properties of the DensePose mapping, we use the video dynamic to propagate ground-truth annotations in time as well as to learn from Siamese equivariance constraints. Having performed exhaustive empirical evaluation of various data annotation and learning strategies, we demonstrate that doing so can deliver significantly improved pose estimation results over strong baselines. However, despite what is suggested by some recent works, we show that merely synthesizing motion patterns by applying geometric transformations to isolated frames is significantly less effective, and that motion cues help much more when they are extracted from videos.

1. Introduction

The analysis of people in images and videos is often based on landmark detectors, which only provide a sparse description of the human body via keypoints such as the hands, shoulders and ankles. More recently, however, several works have looked past such limitations, moving towards a combined understanding of object categories, fine-grained deformations [18, 26, 7, 23] and *dense geometric structure* [13, 32, 9, 12, 20, 19, 29]. Such an understanding may arise via fitting complex 3D models to images or, as in the case of DensePose [12], in a more data-driven manner, by mapping images of the object to a dense UV frame describing its surface.

Despite these successes, most of these techniques need large quantities of annotated data for training, proportional to the complexity of the model. For example, in order to train DensePose, the authors introduced an intricate annotation framework and used it to crowd-source manual annotations for 50K people, marking a fairly dense set of landmark points on each person, for a grand total of 5M manually-labelled 2D points. The cost of the DensePose dataset is estimated to be 30K $. This cost is justified for visual objects such as people that are particularly important in applications, but these methods cannot reasonably scale up to a dense understanding of the whole visual world.

Aiming at solving this problem, papers such as [29, 27] have proposed models similar to DensePose, but replacing manual annotations with self-supervision [29] or even

[*] James Thewlis and Iasonas Kokkinos were with Facebook AI Research (FAIR) during this work.

no supervision [27]. The work of [29], in particular, has demonstrated that a dense object frame mapping can be learned for simple objects such as human and pet faces using nothing more than the compatibility of the mapping with synthetic geometric transformations of the image, a property formalized as the *equivariance* of the learned mapping. Nevertheless, these approaches typically fail to learn complex articulated objects such as people.

In this paper, we thus examine the interplay between weakly-supervised and self-supervised learning with the learning of complex dense geometric models such as DensePose (fig. 1). Our goal is to identify a strategy that will allow us to use the least possible amount of supervision, so as to eventually scale models like DensePose to more non-rigid object categories.

We start by exploring the use of sources of weaker supervision, such as semantic segmentation masks and human keypoints. In fact, one of the key reasons why collecting annotations in DensePose is so expensive is the sheer amount of points that need to be manually clicked to label image pixels with surface coordinates. By contrast, masks and keypoints do not require establishing correspondences and as such are a lot cheaper to collect. We show that, even though keypoints and masks alone are insufficient for establishing correct UV coordinate systems, they allow us to substantially sparsify the number of image-to-surface correspondences required to attain a given performance level.

We then extend the idea of sparsifying annotations to the temporal domain and turn to annotating selected video frames in a video instead of still images as done by [12]. For this we introduce DensePose-Track, a large-scale dataset consisting of dense image-to-surface correspondences gathered on the sequences of frames comprising the PoseTrack dataset [16]. While the cost of manually annotating a video frame is no different than the cost of annotating a similar still image, videos contain motion information that, as we demonstrate, can be used to multiply the efficacy of the annotations. In order to do so, we use an off-the-shelf algorithm for optical flow [14] to establish reliable *dense correspondence* between different frames in a video. We then use these correspondences in two ways: to *transfer annotations* from one frame to another and to enforce an *equivariance constraint* similar to [29].

We compare this strategy to the approach adopted by several recent papers [29, 31, 30] that use for this purpose synthesized image transformations, thus replacing the actual object deformation field with simple rotations, affine distortions, or thin-plate splines (TPS). Crucially, we demonstrate that, while synthetic deformations are not particularly effective for learning a model as complex as DensePose, data-driven flows work well, yielding a strong improvement over the strongest existing baseline trained solely with manually collected static supervision.

2. Related work

Several recent works have aimed at reducing the need for strong supervision in fine-grained image understanding tasks. In semantic segmentation for instance [25, 22, 21] successfully used weakly- or semi- supervised learning in conjunction with low-level image segmentation techniques. Still, semantic segmentation falls short of delivering a surface-level interpretation of objects, but rather acts as a dense, ‘fully-convolutional’ classification system.

Starting from a more geometric direction, several works have aimed at establishing dense correspondences between pairs [5] or sets of RGB images, as e.g. in the recent works of [32, 9]. More recently, [29] use the equivariance principle in order to align sets of images to a common coordinate system, while [27] showed that autoencoders could be trained to reconstruct images in terms of templates deformed through UV maps. More recently, [20] showed that silhouettes and landmarks suffice to recover 3D shape information when used to train a 3D deformable model. These approaches bring unsupervised, or self-supervised learning closer to the deformable template paradigm [11, 6, 2], that is at the heart of the connecting images with surface coordinates. Along similar lines, equivariance to translations was recently proposed in the context of sparse landmark localization in [8], where it was shown that it can stabilize network features and the resulting detectors.

3. Method

We first summarise the DensePose model and then discuss two approaches to significantly decreasing the cost of collecting annotations for supervising this model.

3.1. UV maps

DensePose can be described as a dense body landmark detector. In landmark detection, one is interested in detecting a discrete set of body landmarks $u = 1, \dots, U$, such as the shoulders, hands, and knees. Thus, given an image $I : \mathbb{R}^2 \to \mathbb{R}^3$ that contains a person (or several), the goal is to tell for each pixel $p \in \mathbb{R}^2$ whether it contains any of the U landmarks and, if so, which ones.

DensePose generalizes this concept by considering a dense space of landmarks $\mathcal{U} \subset \mathbb{R}^2$, often called a UV-space. It then learns a function Φ (a neural network in practice) that takes an image I as input and returns an association of each pixel p to a UV point $u = \Phi_p(I) \in \mathcal{U} \cup \{\phi\}$. Since some pixels may belong to the background region instead of a person, the function can also return the symbol ϕ to denote background. The space $\mathcal{U}$ can be thought of as a “chart” of the human body surface; for example, a certain point $u \in \mathcal{U}$ in the chart may correspond to “left eye” and another to “right shoulder”. In practice the body is divided into multiple charts, with a UV map predicted per part.

While DensePose is more powerful than a traditional landmark detector, it is also more expensive to train. In traditional landmark detectors, the training data consists of a dataset of example images I where landmarks are manually annotated; the conceptually equivalent annotations for DensePose are UV associations $\Phi_p^{\text{gt}}(I) \in \mathcal{U}$ collected densely for every pixel p in the image. It is then possible to train the DensePose model Φ via minimization of a loss of the type $\|\Phi(I) - \Phi^{\text{gt}}(I)\|$.

In practice, it is only possible to manually annotate a discretized version of the UV maps. Even so, this requires annotators to click hundreds of points per person instance, while facing issue with ambiguities in labeling pixels that are not localized on obvious human features (e.g. points on the abdomen). A key innovation of the DensePose work [12] was a new system to help human annotators to collect efficiently such data. Despite these innovations, the DensePose-COCO training dataset consists of 50k people instances, for which 5 million points had to be clicked manually. Needless to say, this required effort makes DensePose difficult to apply to new object categories.

3.2. Geometric properties of the UV maps

Brute force manual labelling can be reduced by exploiting properties of the UV maps that we know must be satisfied a-priori. Concretely, consider two images I and I' and assume that pixels p and p' in the respective images contain the same body point (e.g. a left eye). Then, by definition, the map Φ must send pixels p and p' to the same UV point, so that we can write:

$$\Phi_p(I) = \Phi_{p'}(I'). \tag{1}$$

Consider now the special case where I and I' are frames of a video showing people deforming smoothly (where viewpoint changes are a special case of 3D deformation). Then, barring self-occlusions and similar issues, corresponding pixels (p, p') in the two images are related by a *correspondence field* $g : \mathbb{R}^2 \to \mathbb{R}^2$ such that we can write $p' = g(p)$. To a first approximation (i.e. assuming Lambertian reflection and ignoring occlusions, cast shadows, and other complications) image I' is a deformation gI of image I (i.e. $\forall p' : (gI)(p') = I(g^{-1}(p'))$). In this case, the compatibility equation (1) can be rewritten as the so-called *equivariance constraint*

$$\Phi_p(gI) = \Phi_{g(p)}(I) \tag{2}$$

which says that the geometric transformation g "pops-out" the function Φ.

Next, we discuss how equivariance can be used in different ways to help supervise the DensePose model. There are two choices here: (1) how the correspondence field g can be obtained (section 3.2.1) and (2) how it can be incorporated as a constraint in learning (section 3.2.2).

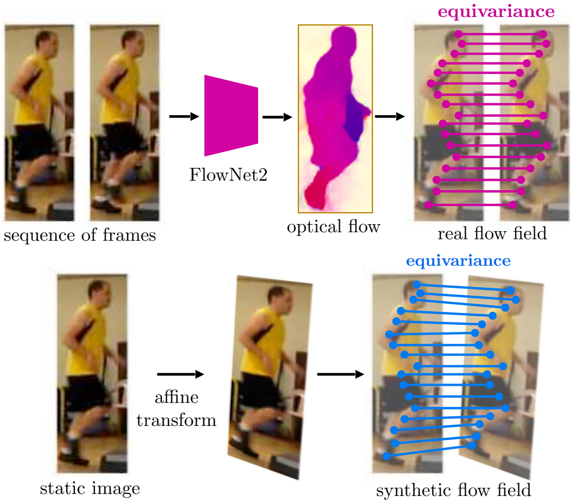


Figure 2: Real (top) and synthetic (bottom) transformation fields exploited to enforce equivariance constraints.

3.2.1 Correspondence fields: synthesized vs real

Annotating the correspondence field g in (2) is no easier than collecting the DensePose annotations in the first place. Thus, (2) is only useful if correspondences can be obtained in a cheaper manner. In this work, we contrast two approaches: synthesizing correspondences or measuring them from a video (see fig. 2).

The first approach, adopted by a few recent papers [29, 31, 30], *samples* g at random from a distribution of image warps. Typical transformations include affinities and thin-plate splines (TPS). Given the warp g, a training triplet $t = (g, I, I')$ is then generated by taking a random input image I and applying to it the warp to obtain $I' = gI$.

The second approach is to *estimate* a correspondence field from data. This can be greatly simplified if we are given a video sequence, as in this case low-level motion cues can be integrated over time to give us correspondences. The easiest way to do so is to apply to the video an off-the-shelf optical flow method, possibly integrating its output over a short time. Then, a triplet is formed by taking the first frame I, the last I', and the integrated flow g.

The synthetic approach is the simplest and most general as it does not require video data. However, sampled transformations are at best a coarse approximation to correspondence fields that may occur in nature; in practice, as we show in the experiments, this severly limits their utility. On the other hand, measuring motion fields is more complex and requires video data, but results in more realistic flows, which we show to be a key advantage.

3.2.2 Leveraging motion cues

Given a triplet $t = (g, I, I')$, we now discuss two different approaches to generating a training signal: transferring ground-truth annotations and Siamese learning.

The first approach assumes that the ground-truth UV map $\Phi^{\mathrm{gt}}_{p'}(I')$ is known for image I', as for the DensePose-Track dataset that will be introduced in section 4. Then, eq. (2) can be used to recover the ground-truth mapping for the first frame I as $\Phi^{\mathrm{gt}}_{p}(I) = \Phi^{\mathrm{gt}}_{g(p)}(I')$. In this manner, when training DP, the loss term $\|\Phi^{\mathrm{gt}}(I') - \Phi(I')\|$ can be augmented with the term $\|\Phi^{\mathrm{gt}}(I) - \Phi(I)\|$.

The main restriction of the approach above is that the ground-truth mapping must be available for one of the frames. Otherwise, we can still use eq. (2) and enforce the constraint $\Phi_p(I) = \Phi_{gp}(I')$. This can be encoded in a loss term of the type $\|\Phi(I) - \Phi_g(I'))\|$ where $\Phi_g(I')$ is the warped UV map of the second image. Note that both terms in the loss are output by the learned model Φ, which makes this a Siamese neural network configuration.

Another advantage of the equivariance constaint eq. (2) is that it can be applied to *intermediate layers* of the deep convolutional neural network Φ as in fact the nature of the output of the function is irrelevant. In the experiments, equivariance is applied to the features preceding the output layers at each hourglass stack as this was found to work best. Thus, denote by $\Psi(I)$ the tensor output obtained at the appropriate layer of network Φ with input I and let Ψ_g be the warped tensor. We encode the equivariance constraint via the cosine-similarity loss of the embedding tensors $\mathcal{L}_{\cos} = 1 - \rho(\Psi(I), \Psi_g(I'))$, where ρ is the cosine similarity $\rho(x, y) = \langle x, y\rangle / (\|x\|\|y\|)$ of vectors x and y.

4. DensePose-Track

We introduce the DensePose-Track dataset, based on the publicly available version of the PoseTrack dataset [16], which contains 10 339 images and 76 058 annotations. PoseTrack annotations are provided densely for 30 frames that are located temporally in the middle of the video. The DensePose-Track dataset has 250 training videos and 50 validation videos. In order to allow a more diverse evaluation of long range articulations, every fourth frame is additionally annotated for the validation set.

Since subsequent frames in DensePose-Track can be highly correlated, we temporally subsample the tracks provided in the PoseTrack dataset using two different sampling rates. Firstly, in order to preserve the diversity and capture slower motions, we annotate every eighth frame. Secondly, in order to capture faster motions we sample every second frame for four frames in each video.

Each person instance in the selected images is cropped based on a bounding box obtained from the keypoints and histogram-equalized. The skeleton is superimposed on the cropped person images to guide the annotators and identify the person in occlusion cases. The collection of correspondences between the cropped images and the 3D model is done using the efficient annotation process analogous to the one described in [12].

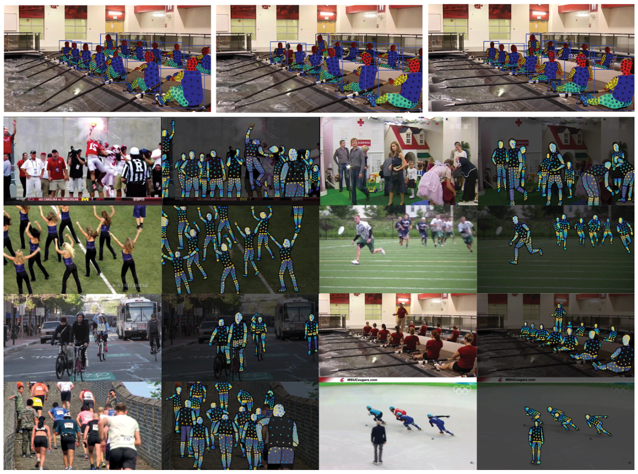


Figure 3: Annotations in the collected DensePose-Track dataset. Top row: Parts and points. Bottom rows: Images and collected points colored based on the 'U' value [12], indicating one of the two coordinates in a part-based, locally planar parameterization of the human body surface.

The PoseTrack videos contain rapid motions, person occlusions and scale variation which leads to a very challenging annotation task. Especially due to motion blur and small object sizes, in many of the cropped images the visual cues are enough to localize the keypoints but not the detailed surface geometry. To cope with this we have filtered the annotation candidates. Firstly, the instances with less then six visible keypoints are filtered out. This is followed by manual elimination of samples that are visually hard to annotate.

The DensePose-Track dataset training/validation sets have 1680 / 782 images labelled in total with dense correspondences for 8274 / 4753 instances, resulting in a total of 800 142 / 459 348 point correspondences, respectively. Sample image-annotation pairs are visualized in Fig 3.

Beyond the purpose of self-supervised training through optical flow, PoseTrack contains information that could be used to asses dense pose estimation systems in time, or improve them through spatio-temporal processing at test time. Static datasets do not capture the effects of occlusions caused by multi-person interactions, e.g. when dancing. Recent datasets for pose estimation in time focus on more challenging, multi-person videos as e.g. [17, 15], but are smaller in scale — in particular due to the challenging nature of the task. Regarding establishing dense correspondences between images and surface-based body models DensePose-COCO was introduced in [12], providing annotations for 50K images of humans appearing in the COCO dataset. Still, this dataset only contains individual frames, and as such cannot be used to train models that exploit temporal information. We intend to explore these research avenues in future work, and focus here on studying how to best exploit temporal information as a means of supervision.

Model	Train	Test	**AP**	**AP**$_{50}$	**AP**$_{75}$	**AP**$_M$	**AP**$_L$	**AR**	**AR**$_{50}$	**AR**$_{75}$	**AR**$_M$	**AR**$_L$
DensePose-RCNN	DP-COCO	DP-COCO	55.5	89.1	60.8	50.7	56.8	63.2	92.6	69.6	51.8	64.0
Hourglass	DP-COCO	DP-COCO	57.3	88.4	63.9	57.6	58.2	65.8	92.6	73.0	59.6	66.2
DensePose-RCNN	DP-COCO	DP-Track	30.1	61.3	26.4	4.5	32.2	37.5	67.3	36.9	5.7	39.7
Hourglass	DP-COCO	DP-Track	39.3	70.7	38.9	22.4	40.6	48.7	78.3	50.8	33.2	49.8
+ GT prop. + equiv.	All	DP-Track	40.3	72.3	39.7	23.3	41.6	49.4	79.5	51.6	34.1	50.5

Table 1: Comparison with the state-of-the-art of dense pose estimation in a multi-person setting on DensePose-COCO (`DP-COCO`) and DensePose-Track (`DP-Track`) datasets. The DensePose-RCNN baseline is based on a ResNeXt-101 backbone, Hourglass has 6 stacks. In all cases we use real bounding box detections produced by DensePose-RCNN.

Data	5 cm	10 cm	20 cm
Human (*)	65	92	98
DensePose-RCNN	51.16	68.21	78.37
Hourglass – 1 stack	50.38	77.97	89.80
2 stacks	55.78	82.34	92.55
8 stacks	**58.23**	**84.06**	**93.57**

Table 2: **Baseline architectures.** Comparison of different DensePose architectures on the DensePose-COCO dataset: the original ResNeXt-based RCNN network of [12] and the Hourglass architecture [24]. Accuracy on the DensePose-COCO dataset increases with the number of hourglass stacks. However, deeper models overfit the biases of the COCO dataset used for pretraining, so that the best performance when transferred to DensePose-Track is at 6 stacks. (*) evaluated on manually annotated synthetic images[12].

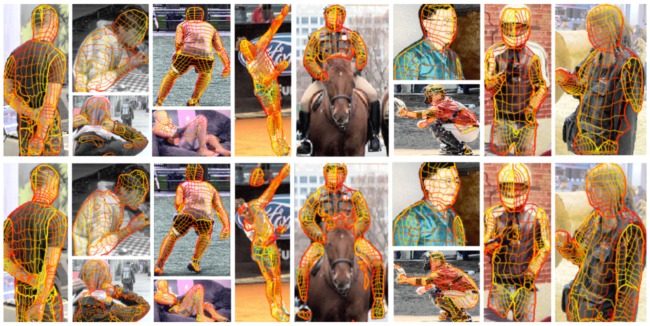


Figure 4: **Qualitative results.** Hourglass (bottom) vs DensePose-RCNN [12] (top). The advantages of the fully convolutional Hourglass include better recall and spatial alignment of predictions with the input, at cost of higher sensitivity to high-frequency variations in textured inputs.

5. Experiments

In the first part of the experiments (section 5.1), we discuss the baseline DensePose architecture and find out a new "gold-standard" setup for this problem.

In the second part (section 5.2), we use the DensePose-COCO dataset to ablate the amount and type of supervision that is needed to learn dense pose estimation in static images. In this manner, we clarify how much data annotations can be reduced without major changes to the approach.

Finally, in the last part (section 5.3) we explore the interplay with temporal information on the DensePose-Track dataset and study how optical flow can help increase the accuracy of dense pose estimation in ways which go beyond generic equivariance constraints.

5.1. Baseline architectures

In most of the following experiments we consider the performance of dense pose estimation obtained on ground-truth bounding boxes in a single person setting (including the DensePose-RCNN evaluation). This allows us to isolate any issues related to object detection performance, and focus exclusively on the task of dense image-surface alignment. We further introduce the use of Hourglass networks [24] as a strong baseline that is trained from scratch on the task of dense pose estimation. This removes any dependency on pretraining on ImageNet, and allows us to have an orderly ablation of our training choices. In this setting, we evaluate the performance in terms of ratios of points localized within 5 cm, 10 cm and 20 cm from the ground truth position measured along the surface of the underlying 3D model (geodesic distance) [12].

Starting from the results in table 2, we observe that we get substantially better performance than the system of [12] which relies on the DensePose-RCNN architecture. We note that the system of [12] was designed to execute both detection and dense pose estimation and operates at multiple frames per second; as such the numbers are not directly comparable. We do not perform detection, and instead report all results on images pre-cropped around the subject. Still, it is safe to conclude that Hourglass networks provide us with a strong baseline (see fig. 4 for illustrations).

For completeness, in table 1 we also report performance of both architectures (DensePose-RCNN and Hourglass) in the multi-person setting, expressed in COCO metrics and obtained using the real bounding box detections produced by DensePose-RCNN with a ResNeXt-101 backbone.

5.2. Ablating annotations

We first examine the impact of reducing the amount of DensePose supervision; we also consider using simpler annotations related to semantic part segmentation that are faster to collect than DensePose chart annotations.

	Data	5 cm	10 cm	20 cm
(i)	Full dataset	55.78	82.34	92.55
(ii)	Segmentation only	3.53	13.25	48.21
(iii)	50% $(k+u)$	52.49	79.45	90.40
image	5% $(k+u)$	36.27	64.58	79.93
subsampling	1% $(k+u)$	14.11	32.06	50.21
(iv)	100% k + 50% u	53.50	80.29	90.86
image	100% k + 5% u	40.80	69.04	83.15
subsampling	100% k + 1% u	36.16	66.59	83.14
(v)	50% $(k+u)$	54.06	81.24	91.92
point	5% $(k+u)$	47.68	76.34	88.86
subsampling	1% $(k+u)$	37.65	68.25	84.37

Table 3: Reduced supervision on DensePose-COCO, k stands for body part index and u for UV coordinates (fig. 5b additionally illustrates experiments (i), (iii) and (v)).

	Data	5 cm	10 cm	20 cm
	Full dataset	55.78	82.34	92.55
	1% u	37.65	68.25	84.37
	keypoints	36.60	63.03	76.81
	1% u + keypoints	39.17	68.78	85.12

Table 4: The positive effect of augmenting sparse DensePose-COCO annotations with skeleton keypoints.

Reduced supervision. Recall that DensePose annotations break down the chart $\mathcal{U} = \cup_{k=1}^{K} \mathcal{U}_k \subset \mathbb{R}^2$ into K parts and, for each pixel p, provide the chart index $k(p)$ (segmentation masks) and the specific chart point $u(p) \in \mathcal{U}_{k(p)}$ within it ($u(p)$ is in practice normalized in the range $[0,1]^2$). The neural network $\Phi_p(I) \approx (k(p), u(p))$ is tasked with estimating both quantities in parallel, optimizing a classification and a regression loss respectively.

We first observe (rows (i) vs (ii) of table 3) that supervising only by using the segmentation masks (thus discarding the regression term in the loss) is not very useful, which is not surprising since they do not carry any surface-related information. However, part masks can result in a much more graceful drop in performance when removing DensePose supervision. To show this, in experiment (iii) we use only a subset of DensePose-COCO images for supervision (using complete part-point annotations (k,u)), whereas in (iv) we add back the other images, but only providing the cheaper part k annotations for the images we add back. We see that performance degrades much more slowly, suggesting that, given an annotation budget, it is preferable to collect coarse annotations for a large number of images while collecting detailed annotations for a smaller subset.

The final experiment (v) in table 3 and fig. 5b is similar, but instead of reducing the number of images, we reduce the number of pixels p for which we provide chart point supervision $u(p)$ (thus saving a corresponding number of annotator "clicks"). For a comparable reduction in annotations, this yields higher accuracy as the network is exposed to a broader variety of poses during training. Hence, for a fixed budget of annotator "clicks" one should collect fewer correspondences per image for a large number of images.

Keypoint supervision. Traditional landmark detectors use keypoint annotations only, which is even cheaper than collecting part segmentations. Thus, we examine whether keypoint annotations are complementary to part segmentations as a form of coarse supervision. In fact, since a keypoint associates a small set of surface points with a single pixel, this type of supervision could drive a more accurate image-surface alignment result. Note that not only keypoints are sparse, but they are also easier to collect from an annotator than an image-to-surface correspondence u, since they do not not require presenting to the annotator a clickable surface interface as done in [12].

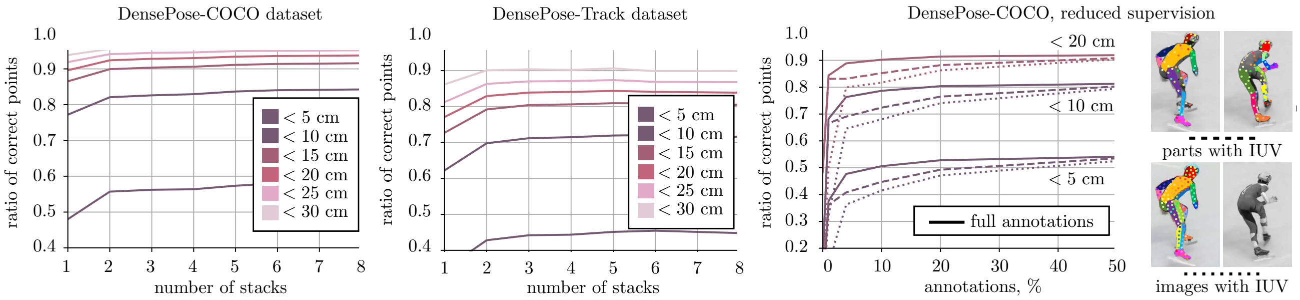
Table 4 replicates the experiment (v.a) of table 3, repeats it but this time providing only keypoint annotations instead of u annotations, and then combines the two. We see that the two annotations types are indeed complementary, especially for highly-accurate localization regimes.

5.3. Paced learning

Next, we examine statistical differences between the DensePose-COCO and DensePose-Track datasets (discarding for now dynamics) and their effect on training DensePose architectures. We show that DensePose-Track does improve performance when used in combination with DensePose-COCO; however, it is substantially harder and thus must be learned in a paced manner, after the DensePose model has been initialized on the easier COCO data.

The details on this group of experiments are given in table 5. In all cases, we train a 6-stack Hourglass model, using the best performing architecture identified in the previous section. Stage I means that the model is first initialized by training on the stated dataset and Stage II, where applicable, means that the model is fine-tuned on the indicated data. We observe that training on DensePose-Track (row (i) of table 5) yields worse performance than training on an equiparable subset or the full DensePose-COCO dataset (ii-iii), even when the model is evaluated on DensePose-Track. We assume that this is due to both the larger variability of images in the COCO training set, as well as the cleaner nature of COCO images (blur-free, larger resolution), which is known to assist training [1]. This assumption is further supported by row (iv), where it is shown than training simultaneously on the union of COCO and PoseTrack yields worse results than training exclusively on COCO.

By contrast, we observe that a two-stage procedure, where we first train on DensePose-COCO and then finetune on DensePose-Track yields substantial improvements. The best results are obtained by fine-tuning on the union of both datasets – even giving an improvement on the DensePose-



(a) performance of the Hourglass network as a function of a number of stacks (b) training with reduced amount of supervision

Figure 5: (a) Performance of the Hourglass architecture on the DensePose-COCO dataset monotonically increases with the number of stacks, but peaks at 6 stacks for the DensePose-Track dataset. (b) Given a fixed annotation budget, it is beneficial to partially annotate a large number of images, rather than collect full annotations on a subset of the dataset.

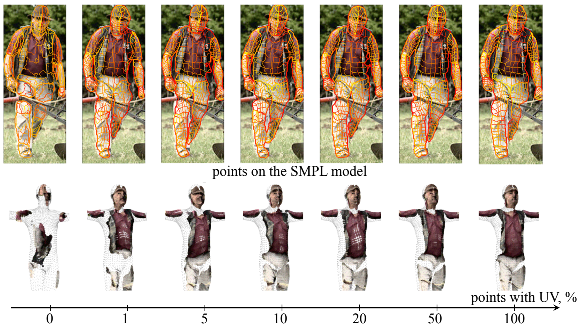


Figure 6: **Reduced supervision.** Top: effect of training with a reduced percentage of points with UV annotations. Bottom: the texture map displayed on the SMPL model [3] shows the quality of the learned mapping.

COCO test set. This is again aligned with curriculum learning [1], which suggests first training on easy examples and including harder examples in a second stage.

5.4. Leveraging motion cues

Having established a series of increasingly strong baselines, we now turn to validating the contribution of flow-based training when combined with the strongest baseline.

Flow computation. For optical flow computation we use the competitive neural network based method of FlowNet2 [14], which has been trained on synthetic sequences. We run this model on Posetrack and MPII Pose (video version), computing for each frame T the flow to frames $T - 3$ to $T + 3$ (where available). For MPII Pose we start with about a million frames and obtain 5.8M flow fields. For DensePose-Track we have 68k frames and 390k flow fields. Note that a subset of MPII Pose clips are used in DensePose-Track, although the Posetrack versions contain more frames of context. For DensePose-Track, we propagate the existing DensePose annotations according to the flow fields, leading to 48K new cropped training images from the original 8K (12% of frames have manual labels).

In order to propagate annotations across frames, we simply translate the annotation locations according to the flow field (fig. 2). Because optical flow can be noisy, especially in regions of occlusion, we use a forward-backward consistency check. If translating forward by the forward flow then back again using the backward flow gives an offset greater than 5 pixels we ignore that annotation. On MPII pose, we use the annotations of rough person centre and scale.

Results. We compare the baseline results obtained in the previous section to different ways of augmenting training using motion information. There are two axes of variations: whether motion is randomly synthesized or measured from a video using optical flow (section 3.2.1) and whether motion is incorporated in training by propagating ground-truth labels or via the equivariance constraint (section 3.2.2).

Rows (i-iv) of table 6 compare using the baseline supervision via the available annotations in DensePose-Track to their augmentation using GT propagation, equivariances and the combination of the two. For each combination, the table also reports results using both synthetic (random TPS) and real (optical flow) motion. Rows (v-viii) repeat the experiments, but this time starting from a network pre-trained on DensePose-COCO instead of a randomly initialized one.

There are several important observations. First, both GT propagation and equivariance improve the results, and the best result is obtained via their combination. GT propagation performs at least a little better than equivariance (but it cannot be used if no annotations are available).

Second, augmenting via real motion fields (optical flow) works a lot better than using synthetic transformations, suggesting that realism of motion augmentation is key to learn complex articulated objects such as people.

Third, the benefits of motion augmentation are particularly significant when one starts from a randomly-initialized network. If the network is pre-trained on DensePose-COCO, the benefits are still non-negligible.

It may seem odd that GT propagation works better than equivariance since both are capturing similar constraints. After analyzing the data, we found out that the reason is that equivariance optimized for *some* charting of the human

	Training data		Tested on DensePose-Track			Tested on DensePose-COCO		
	Stage I	*Stage II*	5 cm	10 cm	20 cm	5 cm	10 cm	20 cm
(i)	DensePose-Track	—	21.06	42.94	59.54	20.34	41.24	57.29
(ii)	DensePose-COCO subset (*)	—	33.67	58.79	73.45	47.10	74.06	86.27
(iii)	DensePose-COCO	—	44.89	71.52	83.71	55.78	82.34	92.55
(iv)	DensePose-COCO & Track	—	41.76	69.94	83.60	55.27	82.05	92.37
(v)	DensePose-COCO	DensePose-Track	45.57	73.35	85.77	53.70	81.34	92.03
(vi)	DensePose-COCO	all	**46.04**	**73.41**	**85.79**	**58.01**	**84.06**	**93.64**

Table 5: **Training strategies.** Effect of training and testing on DensePose-COCO vs DensePose-Track in various combinations. The best performing model (vi) is first trained on the cleaner COCO data and then fine tuned on a union of datasets. (*) a random subset of the DensePose-COCO training images of size of the DensePose-Track dataset.

	Training strategy	Training data		Synthetic (TPS)			Real (Optical Flow)		
		Stage I	*Stage II*	5 cm	10 cm	20 cm	5 cm	10 cm	20 cm
(i)	Baseline	—	DensePose-Track	21.06	42.94	59.54	21.06	42.94	59.54
(ii)	GT propagation			22.33	45.30	62.08	32.85	60.07	75.95
(iii)	Equivariance			21.57	44.17	61.27	23.12	45.85	62.22
(iv)	GT prop. + equivariance			**22.41**	**45.53**	**62.71**	**34.50**	**61.70**	**77.20**
(v)	Baseline	DensePose-COCO	DensePose-Track	45.57	73.35	83.71	45.57	73.35	83.71
(vi)	GT propagation			45.77	73.65	86.13	47.36	75.17	87.47
(vii)	Equivariance			45.67	73.47	85.93	45.76	73.54	86.06
(viii)	GT prop. + equivariance			**45.81**	**73.70**	**86.14**	**47.45**	**75.21**	**87.56**
(ix)	Baseline	DensePose-COCO	all	46.04	73.41	85.79	46.04	73.41	85.79
(x)	GT prop. + equivariance			-	-	-	**47.62**	**75.80**	**88.12**

Table 6: **Leveraging real and synthetic flow fields.** The best performing model (x) is trained on a combination DensePose-COCO+Track by exploiting the real flow for GT propagation between frames and enforcing equivariance.

Features	Synthetic (TPS)			Real (Optical Flow)		
	5 cm	10 cm	20 cm	5 cm	10 cm	20 cm
0	45.74	73.62	86.14	45.90	73.71	86.10
1	**46.08**	**73.85**	**86.29**	45.91	73.74	86.15
2	45.97	73.82	**86.29**	45.92	73.64	86.04
3	45.85	73.55	86.05	45.97	73.81	**86.30**
4, all	45.98	73.62	86.15	45.84	73.42	85.86
4, segm.	46.02	73.74	86.20	**45.98**	**73.85**	86.20
4, UV	45.78	73.76	86.26	45.95	73.64	86.16
none	45.57	73.35	83.71	45.57	73.35	83.71

Table 7: Training with applying synthetic and optical flow warp-based equivariance at different feature levels (pre-training on DensePose-COCO, tuning and testing on DensePose-Track). Level 4 corresponds to the output of each stack, level 0 – to the first layer. 'Segm.' denotes the segmentation part of the output, 'UV' – the UV coordinates.

body, but that, since many charts are possible, this needs not to be the same that is constructed by the annotators. Bridging this gap between manual and unsupervised annotation statistics is an interesting problem that is likely to be of relevance whenever such techniques are combined.

Equivariance at different feature levels. Finally, we analyze the effect of applying equivariance losses to different layers of the network, using synthetic or optical flow based transformations (see table 7). The results show benefits of imposing these constraints on the intermediate feature levels in the network, as well as on the subset of the output scores representing per-class probabilities in body parsing.

6. Conclusion

In this work we have explored different methods of improving supervision for dense human pose estimation tasks by leveraging on weakly-supervised and self-supervised learning. This has allowed us to exploit temporal information to improve upon strong baselines, delivering substantially more advanced dense pose estimation results when compared to [12]. We have also introduced a novel dataset DensePose-Track, which can facilitate further research at the interface of dense correspondence and time.

Despite this progress, applying such models to videos on a frame-by-frame basis can reveal some of their shortcomings, including flickering, missing body parts or false detections over the background (as witnessed in the hardest of the supplemental material videos). These problems can potentially be overcome by exploiting temporal information, along the lines pursued in the pose tracking problem, [28, 4, 15, 16, 10]. For instance, motion blur, or partial occlusion can result in erroneous correspondences at a given image position; however, we can recover from such failures by combining complementary information from adjacent frames where the same structure is better visible. We intend to further investigate this direction in future research.

References

[1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, 2009. 6, 7

[2] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3D morphable model. *PAMI*, 25(9):1063–1074, 2003. 2

[3] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *ECCV*, 2016. 7

[4] Christoph Bregler, Jitendra Malik, and Katherine Pullen. Twist based acquisition and tracking of animal and human kinematics. *International Journal of Computer Vision*, 56(3):179–194, 2004. 8

[5] Hilton Bristow, Jack Valmadre, and Simon Lucey. Dense semantic correspondence where every pixel is a classifier. In *ICCV*, 2015. 2

[6] Timothy Cootes, Gareth Edwards, and Christopher Taylor. Active appearance models. In *ECCV*, 1998. 2

[7] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 1

[8] Xuanyi Dong, Shoou-I Yu, Xinshuo Weng, Shih-En Wei, Yi Yang, and Yaser Sheikh. Supervision-by-registration: An unsupervised approach to improve the precision of facial landmark detectors. In *CVPR*, 2018. 2

[9] Utkarsh Gaur and B. S. Manjunath. Weakly supervised manifold learning for dense semantic object correspondence. In *ICCV*, 2017. 1, 2

[10] Rohit Girdhar, Georgia Gkioxari, Lorenzo Torresani, Manohar Paluri, and Du Tran. Detect-and-track: Efficient pose estimation in videos. In *CVPR*, 2018. 8

[11] U. Grenander, Y. Chow, and D. M. Keenan. *Hands: A Pattern Theoretic Study of Biological Shapes*. Springer-Verlag, Berlin, Heidelberg, 1991. 2

[12] Riza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *CVPR*, 2018. 1, 2, 3, 4, 5, 6, 8

[13] Rıza Alp Güler, George Trigeorgis, Epameinondas Antonakos, Patrick Snape, Stefanos Zafeiriou, and Iasonas Kokkinos. Densereg: Fully convolutional dense shape regression in-the-wild. In *CVPR*, 2017. 1

[14] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. 2, 7

[15] Eldar Insafutdinov, Mykhaylo Andriluka, Leonid Pishchulin, Siyu Tang, Evgeny Levinkov, Bjoern Andres, and Bernt Schiele. Arttrack: Articulated multi-person tracking in the wild. In *CVPR*, 2017. 4, 8

[16] Umar Iqbal, Anton Milan, Mykhaylo Andriluka, Eldar Ensafutdinov, Leonid Pishchulin, Juergen Gall, and Bernt Schiele. PoseTrack: A benchmark for human pose estimation and tracking. In *CVPR*, 2018. 2, 4, 8

[17] Umar Iqbal, Anton Milan, and Juergen Gall. Posetrack: Joint multi-person pose estimation and tracking. In *CVPR*, 2017. 4

[18] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2015. 1

[19] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018. 1

[20] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018. 1, 2

[21] Anna Khoreva, Rodrigo Benenson, Jan Hendrik Hosang, Matthias Hein, and Bernt Schiele. Weakly supervised object boundaries. In *CVPR*, 2016. 2

[22] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, 2016. 2

[23] Natalia Neverova and Iasonas Kokkinos. Mass displacement networks. In *BMVC*, 2018. 1

[24] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 5

[25] George Papandreou, Liang-Chieh Chen, Kevin Murphy, and Alan L. Yuille. Weakly- and semi-supervised learning of a DCNN for semantic image segmentation. In *CVPR*, 2016. 2

[26] George Papandreou, Iasonas Kokkinos, and Pierre-André Savalle. Modeling local and global deformations in deep learning: Epitomic convolution, multiple instance learning, and sliding window detection. In *CVPR*, 2015. 1

[27] Zhixin Shu, Mihir Sahasrabudhe, Riza Alp Güler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos. Deforming autoencoders: Unsupervised disentangling of shape and appearance. In *ECCV*, 2018. 1, 2

[28] Hedvig Sidenbladh, Michael J. Black, and David J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *ECCV*, 2000. 8

[29] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object frames by dense equivariant image labelling. In *NIPS*, 2017. 1, 2, 3

[30] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *ICCV*, 2017. 2, 3

[31] Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee. Unsupervised discovery of object landmarks as structural representations. In *CVPR*, 2018. 2, 3

[32] Tinghui Zhou, Philipp Krähenbühl, Mathieu Aubry, Qi-Xing Huang, and Alexei A. Efros. Learning dense correspondence via 3d-guided cycle consistency. In *CVPR*, 2016. 1, 2