Learning Landmarks from Unaligned Data using Image Translation

Tomas Jakab
VGG, University of Oxford
tomj@robots.ox.ac.uk

Ankush Gupta
DeepMind, London
ankushgupta@google.com

Hakan Bilen
School of Informatics
University of Edinburgh
hbilen@ed.ac.uk

Andrea Vedaldi
VGG, University of Oxford
Facebook AI Resesarch, London
vedaldi@robots.ox.ac.uk

Abstract

We introduce a method for learning landmark detectors from unlabelled video frames and unpaired labels. This allows us to learn a detector from a large collection of raw videos given only a few example annotations harvested from existing data or motion capture. We achieve this by formulating the landmark detection task as one of image translation, learning to map an image of the object to an image of its landmarks, represented as a skeleton. The advantage is that this translation problem can then be tackled by CycleGAN. However, we show that a naive application of CycleGAN confounds appearance and pose information, with suboptimal keypoint detection performance. We solve this problem by introducing an analytical and differentiable renderer for the skeleton image so that no appearance information can be leaked in the skeleton. Then, since cycle consistency requires to reconstruct the input image from the skeleton, we supply the appearance information thus removed by conditioning the generator with a second image of the same object (e.g. *another frame from a video). Furthermore, while CycleGAN uses two cycle consistency constraints, we show that the second one is detrimental in this application and we discard it, significantly simplifying the model. We show that these modifications improve the quality of the learned detector leading to state-of-the-art unsupervised landmark detection performance in a number of challenging human pose and facial landmark detection benchmarks. Project page:* http://www.robots.ox.ac.uk/~vgg/research/unsupervised_pose/

1. Introduction

Modern machine learning methods can solve complex image labelling tasks such as pose recognition with good accuracy, but at the cost of collecting large annotated datasets for training. The cost of these manual annotations is a major obstacle to deploying machine learning to new tasks. Removing the annotation bottleneck is thus one of the key objectives of current research in computer vision.

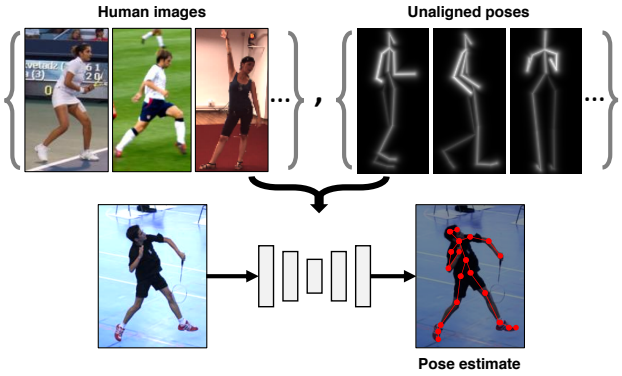

Figure 1. **Learning landmark detectors from unaligned data.** We learn to estimate object landmarks from a collection of unlabelled videos given only prior information on the distribution of human poses, which may be harvested independently from synthetic data, motion capture or small amounts of manually annotations. We demonstrate our approach outperforms recent methods [59, 76] for unsupervised landmark localisation for human pose and faces across a number of benchmarks (section 5).

Recent advances in unsupervised representation learning [32, 46, 74] have focussed on learning generic feature extractors via pretext tasks. These features can then be transferred to an end task such as pose recognition, but this requires to finetune the model using manually-sourced labels. Very few works have attempted to solve a task such as ego-pose (and depth) from unlabelled data directly [40].

In this paper, we introduce a method that can learn to recognize pose from a large collection of unlabelled images given only unpaired annotations. We build our approach on the success of recent *image-to-image translation* methods such as CycleGAN [79]. Following [79], we cast image labelling as the problem of translating a natural image into an image of the corresponding label. For pose recognition, the label image thus look like a rendering of a skeleton with the same pose as the provided input image (see fig. 1). For other objects such as faces, the rendering may look instead as a collection of 2D keypoints (fig. 2 and section 5). Either way, CycleGAN can learn such a mapping given only unpaired samples from each domain, so that none of the ex-

ample images supplied to the network for training needs to be labelled.

However, a limitation of CycleGAN is that the mappings between domains are assumed to be one-to-one, which is not the case for most image labelling tasks. In fact, while there is a single plausible mapping from an image of a person to its skeleton, the same skeleton can be mapped to many different people. In order to remove this ambiguity, a naive application of CylceGAN introduces in the skeleton subtle artefacts that leak appearance information along with pose. This phenomenon, discussed in detail in [11] and visualised in fig. 2, was found to hinder the ability of CycleGAN to extract robust pose information from images (see section 5.3).

To address this issue, we propose two important enhancements. First, we introduce a *tight bottleneck* in the part of the model that generates the label image, preventing undesired "leakage" of appearance information in the label. The bottleneck is designed to extract the location of a certain number of 2D object keypoints, which are then used to re-render the skeleton image using an hand-crafted differentiable function. The small dimensionality of the space used to represent pose (2D keypoints) and the absence of learnable parameters in the rendering function makes it very difficult to leak appearance information in the label image, thus better factoring appearance and geometry.

The downside of this "clean" label is that reconstructing an image from it becomes much more ambiguous. To address this issue, we further modify CycleGAN to use a *conditional image generator*. This generator combines the geometric information contained in the label with appearance information extracted from *another* image of the same object. Crucially, we ensure that this auxiliary image has the same appearance of the first, but a different pose, which further contributes to factor geometry and appearance information. Such image pairs can be cheaply obtained as frames in a video.

We show empirically that the resulting approach can disentangle appearance and geometry given only videos of people or faces. We can thus achieve excellent performance in pose recognition without any manual annotations for the video frames used to train the model, achieving state-of-the-art pose recognition performance from unlabelled images on standard benchmarks such as Human3.6M or 300-W, and significantly outperform the previous methods.

Note that our method still requires the use of some pose annotations to learn the space of possible configurations of the landmarks. However, such a prior can be partially or fully hand-crafted, and, if learned, it can be extracted from a different dataset. In the face experiments, for example, we show that we can train the model from a complex but unlabelled dataset of video faces (VoxCeleb) using a prior on keypoints extracted from a much simpler and smaller dataset (MultiPIE) to achieve state-of-the-art keypoint detection performance on a third dataset (300-W), demonstrating the very strong generalization capabilities of our technique (fig. 6).

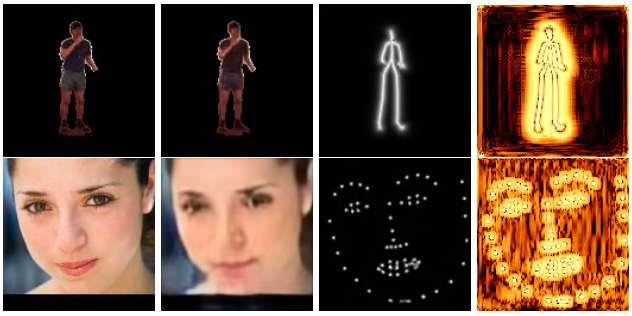

Figure 2. **Appearance leakage.** From left to right: input image, reconstruction, pose represented as the image of a skeleton or keypoints, local-contrast normalised (LCN) pose image (in log scale). In order to satisfy the cycle-consistency constraint, CycleGAN [79] must reconstruct the input image from the pose image, and thus leaks appearance information in the latter [11], reducing pose recognition performance (section 5.3). The visually imperceptible leaked information is highlighted in the LCN pose images as artifacts. We avoid this via conditional image generation and a tight bottleneck (section 3), learning a better keypoint detector.

2. Related work

Supervised pose estimation. Estimation of articulated human limbs from a single image is a well established problem in computer vision, explored primarily in the supervised setting, *i.e.*, where images and *corresponding* ground-truth annotations (*e.g.*, joint locations) are available. Early methods [3, 45, 48, 49, 55, 72] cast this as inference in tree-structured graphical models with priors connecting limbs under the *pictorial structures* framework [13]. Toshev *et al*. [62] propose to directly regress keypoint coordinates from deep CNN features [31].

Tompson *et al*. [61] and Chen *et al*. [10] regress spatial confidence *heatmaps* for joint locations instead, and model geometric relationships between joint locations. Recent works use multi-stage, very deep networks for sequentially refining the heatmaps in both single [4, 7, 9, 44, 47, 60, 68] and multiple person settings [8, 20]. The success of these methods however heavily rely on large annotated datasets such as MS COCO Keypoints [35], Human3.6M [21], MPII [2] and LSP [25]. We instead focus on a more challenging case and propose to learn only from unlabeled images containing humans, and a separate *unaligned* set of anthropomorphically plausible human pose skeletons which can simply be harvested from human joint-limit datasets (*e.g.*, PosePrior [1]), as also explored by Kanazawa *et al*. [26] for lifting 2D keypoints to 3D.

Weakly supervised pose estimation. Recently several works [26, 53, 63, 71] have emerged in the literature that

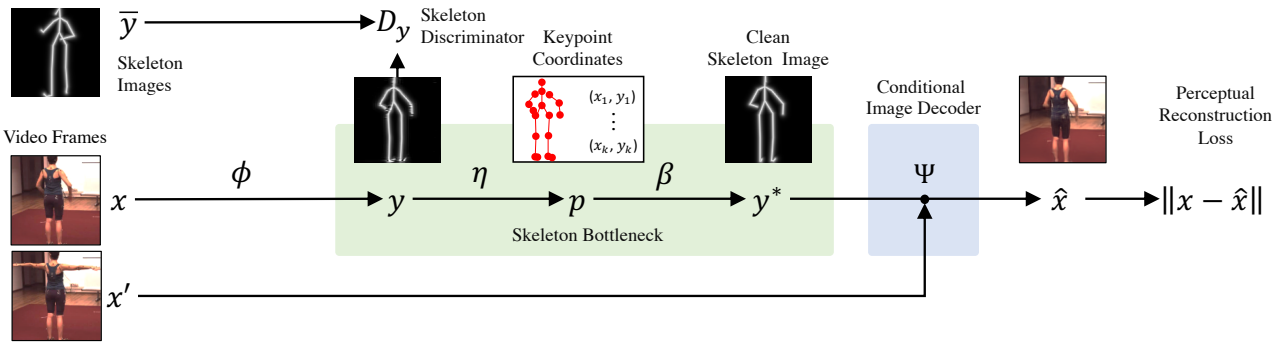

Figure 3. **Model architecture.** Our model learns to regress pose parameters $\boldsymbol{p}$ (keypoint coordinates) given only video frames $\boldsymbol{x}, \boldsymbol{x}'$ and *unpaired* skeleton images $\bar{\boldsymbol{y}}$. It a function $\Phi : \mathcal{X} \rightarrow \mathcal{Y}$ mapping image to skeletons and a second function $\Psi : \mathcal{Y} \times \mathcal{X} \rightarrow \mathcal{X}$ mapping skeletons to images, *conditioned* on a second *style* image $\boldsymbol{x}'$ form the same video. To suppress leakage of appearance information through the generated skeleton image $\boldsymbol{y}$, we introduce a tight *bottleneck* as a skeleton auto-encoder. The latter is formed by a pre-trained encoder η mapping skeleton images to pose coordinates $\boldsymbol{p}$, and an *analytical* skeleton image renderer β that does the opposite. The key difference with respect to methods such as *CycleGAN* are highlighted with coloured backgrounds (see section 3).

propose to learn pose estimation from coarser or reduced labels. The concurrent works [26, 63, 71] propose weakly supervised pose neural networks that learn to reconstruct a full 3D mesh of human body from a single RGB image and 2D joint locations by incorporating feedback from a discriminator to match distribution of predictions with ground-truth factors. Similar to ours, these methods use unpaired training data, 2D joint positions and 3D meshes, however, our method does not require any manual annotations for 2D pose estimation. Ronchi *et al*. [53] make use of less detailed annotations, human annotated relative depth information from images to learn 3D human pose estimation. Liu and Ferrari [36] propose an active learning approach for human pose estimation with the goal of maximising performance while minimising annotation effort.

Unsupervised pose estimation. There are also unsupervised pose estimation techniques [27, 52, 58, 59, 69] that leverage the relative transformation between different instances of same object as supervisory signal to learn 2D pose estimation. WarpNet [27] and geometric matching networks [52] learn to match object pairs by predicting relative transformations between them. Thewlis *et al*. [58, 59] exploit the principle of equivariance and distinctiveness to factorise viewpoint and deformation changes and to learn object structure via landmarks [59] and dense labelling [58]. Sundermeyer *et al*. [57] propose a self-supervised 3D object pose estimator for rigid objects by training it only on synthetic views of a 3D model. Similar to ours, at test time the method finds the most similar synthetic image to the given real 2D image to determine its pose. However, our method does not require realistic renderings of synthetic images in contrast to [57]. Three recent works by Zhang *et al*. [76], Wiles *et al*. [69], and Jakab *et al*. [23] propose using conditional image generation to learn landmark prediction in a unsupervised manner. Concretely, Wiles *et al*. [69] learn a dense deformation field for faces. Zhang *et al*. [76] develop an autoencoding formulation to discover landmarks as explicit structural representations for a given image and use them to reconstruct the original image. Jakab *et al*. [23] propose using frame pairs that differ by a viewpoint or deformation change to factorise appearance and geometry while conditionally generating one frame from another one. The authors show that the discovered landmarks can be further fed into a regressor to learn semantic ones. As a matter of fact, our method also relies on conditional image generation. However, ours differs from [23, 76] significantly. Unlike [23, 76], we learn to generate cross-modal examples *i.e*. from RGB to skeleton image and skeleton to RGB image. Thus our method can be used to directly go from an RGB image to pose configuration without requiring any further training of a regressor. We demonstrate that our method outperforms prior works [23, 76] in 2D landmark detection.

Adversarial learning from unaligned data. Adversarial learning methods are shown to be effective for image labelling tasks [15, 18, 19, 64, 65] and generation tasks [17, 79] in the presence of domain shift between different domains and between generated and real images. Ganin and Lempitsky [15] and Tzeng *et al*. [64] concurrently propose using a confusion loss with the goal that feature statistics of multiple domains are similar. The Generative Adversarial Network (GAN) [17] proposes an adversarial loss to encourage the network to generate realistic images by capturing the data distribution from real images. Isola *et al*. [22] propose an image-to-image translation framework that can learn the mapping from input image to output image, but requires paired data. Most related to ours, CycleGAN [79] relax the requirement of aligned image pairs and learn the mapping between input and output images from unaligned pairs. As a matter of fact, we build our method on CycleGAN, however we extend it in a significant way. CycleGAN

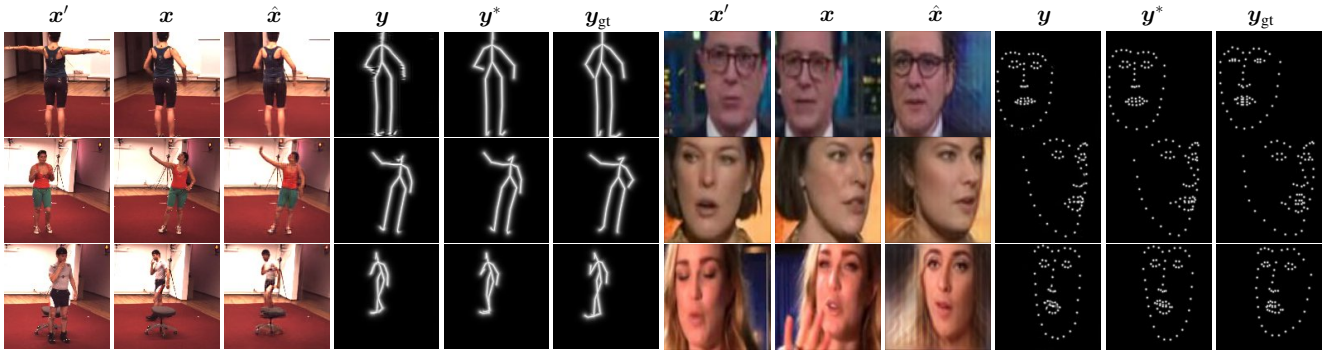

Figure 4. **Training data flow.** Data flowing through our model (fig. 3) during training on the Human3.6M (human pose) and VoxCeleb (face) datasets. Two types of *skeleton images* ($\boldsymbol{y}$) are visualised: stick-figures for human pose and keypoints for faces. $\boldsymbol{y}_{\text{gt}}$ (*g.t.*) refers to ground-truth pose; $\boldsymbol{y}, \boldsymbol{y}^*$ are our predictions.

suffers ambiguity in the generation process when there are multiple possible mappings from one domain to another. For instance, while a skeleton sketch can be mapped to generate different identities. Therefore we condition the generation on an embedding extracted from the given person image.

3. Method

Our aim is to learn a function that maps an image of an object $\boldsymbol{x} \in \mathcal{X} = \mathbb{R}^{3\times H\times W}$ to the object's pose parameters $\boldsymbol{p}$, represented as K 2D keypoints $\boldsymbol{p} = (p_1, \ldots, p_K) \in \Omega^K = \mathbb{R}^{2\times K}$. We wish to learn this function given only pairs of example images $\{(\boldsymbol{x}_i, \boldsymbol{x}'_i)\}_{i=1}^N$ that show the same object with different poses, such as frames in a video. We also assume to have examples $\{\boldsymbol{p}_j\}_{j=1}^M$ of the pose parameters, but, importantly, these are *unaligned*, in the sense that they are not annotations of the training images. The practical advantage is that these pose annotations can be fully or partly synthesized and/or ported from one dataset to another for free.

Inspired by CycleGAN [79], we propose to formulate the unaligned learning problem as an *unsupervised* image-to-image translation task [37, 38, 79] after representing the 2D keypoints as a *skeleton image* $\boldsymbol{y} \in \mathcal{Y} = \mathbb{R}^{H\times W}$. CycleGAN then learns two mappings between domains: $\Phi : \mathcal{X} \to \mathcal{Y}$, translating an image into its skeleton, and its inverse $\Psi : \mathcal{Y} \to \mathcal{X}$.

In order to learn from unaligned data samples $\boldsymbol{x}$ and $\boldsymbol{y}$, CycleGAN uses two ideas. The first is that functions Φ and Ψ should transform the data *distributors* $p(\boldsymbol{x})$ and $p(\boldsymbol{y})$ one into the other. This is achieved by letting Φ and Ψ compete against adversarial discriminators $D_{\mathcal{Y}}$ and $D_{\mathcal{X}}$ [17], ensuring that the distributions of generated samples $\Phi(\boldsymbol{x})$ and $\Psi(\boldsymbol{y})$ match the distributions real samples $\boldsymbol{y}$ and $\boldsymbol{x}$, respectively. However, this alone does not guarantee that individual samples are translated in a meaningful way. Hence, the second idea is to encourage the mappings to be one-to-one by enforcing the *cycle-consistency* conditions $\Psi\circ\Phi(\boldsymbol{x}) \approx \boldsymbol{x}$ and $\Phi \circ \Psi(\boldsymbol{y}) \approx \boldsymbol{y}$.

Unfortunately, in our case the cycle consistency condition $\Psi \circ \Phi(\boldsymbol{x}) \approx \boldsymbol{x}$ cannot be satisfied as there are many images $\boldsymbol{x}$ that have the same skeleton $\boldsymbol{y}$ (section 6). Enforcing this condition causes CycleGAN to encode appearance in the skeleton image $\boldsymbol{y}$ [11], leading to sub-optimal pose detection (see fig. 2 and section 5.3).

We overcome this issue by means of three innovations (fig. 3). First, we avoid leaking appearance in the skeleton image by passing it through a tight *bottleneck* (section 3.1), implemented using an analytical differentiable renderer. Second, we supplant the lack of appearance information in the skeleton by extending the image generator Ψ to take as input a second conditioning image $\boldsymbol{x}'$, which has the same appearance but different pose from the input $\boldsymbol{x}$ — usually $\boldsymbol{x}$ and $\boldsymbol{x}'$ are two frames from a video (section 3.2). Third, while the other cycle consistency condition $\Phi \circ \Psi(\boldsymbol{y}) \approx \boldsymbol{y}$ is applicable to our case, we show that it is not only redundant but also detrimental to accurate landmark detection (sections 4 and 5.3), hence we dispense with it, simplifying the model considerably.

We give details of the various model components below and in fig. 3. Unless otherwise specified, all such components are implemented as convolutional neural networks [33].

3.1. Skeleton bottleneck

The skeleton bottleneck is a mechanism that maps the skeleton image to interpretable 2D keypoint coordinates (or pose parameters) and prevents leaking appearance information in the skeleton images. The bottleneck is implemented as an autoencoder comprising the *skeleton encoder* $\eta : \mathcal{Y} \to \Omega^K$, which maps a skeleton image $\boldsymbol{y}$ to its pose parameters $\boldsymbol{p} = \eta(\boldsymbol{y})$, and the *skeleton generator* $\beta : \Omega^K \to \mathcal{Y}$, which does the opposite.

Crucially, the generator β is *not* learned as a neural net-

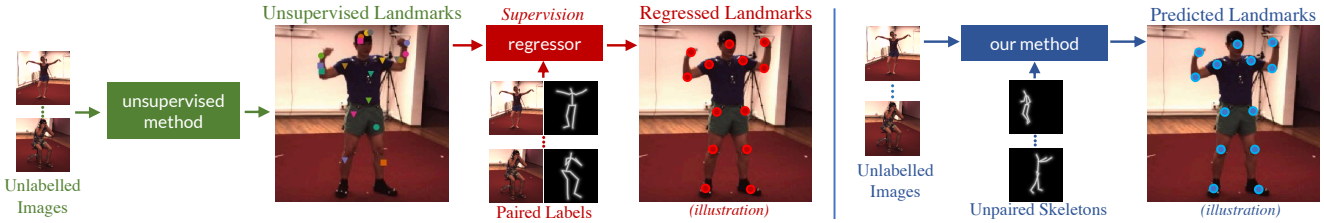

Figure 5. **Comparison with unsupervised methods.** Unsupervised landmark detectors [23, 59, 76] (green) learn to discover landmarks using unlabelled images. However, such landmarks are not aligned with standard labels. In order to align them, they need to learn a map (red) from the unsupervised (discovered) keypoints to those labelled by humans using *paired* supervised samples (image, annotation). In contrast, our method (blue) can learn to directly predict landmarks requiring only unlabelled images and *unpaired* annotations.

work, but is handcrafted to be a differentiable function that renders the skeleton $\boldsymbol{y}$ from the joint coordinates $\boldsymbol{p}$. Let E be a set of keypoint pairs (i, j) connected by a skeleton edge and let $u \in \{1, \dots, H\} \times \{1, \dots, W\}$ be an image pixel. Then,

$$\beta(\boldsymbol{p})_u = \exp\left(-\gamma \min_{(i,j)\in E, r\in[0,1]} \|u - r\boldsymbol{p}_i - (1-r)\boldsymbol{p}_j\|^2\right) \tag{1}$$

is the an exponentially-weighted version of the distance transform of the skeleton. This renderer is appropriate for objects such as human bodies. For others such as human faces, we render the "skeleton" as a set of 2D blobs representing keypoints:

$$\beta(\boldsymbol{p})_u = \sum_{i=1}^{K} \exp\left(-\frac{1}{2\sigma^2}\|u - \boldsymbol{p}_i\|^2\right). \tag{2}$$

The skeleton generator is used to pre-train the encoder η independently from the image autoencoder, using the reconstruction constraint $\eta(\beta(\boldsymbol{p})) = \boldsymbol{p}$ from unaligned pose data,

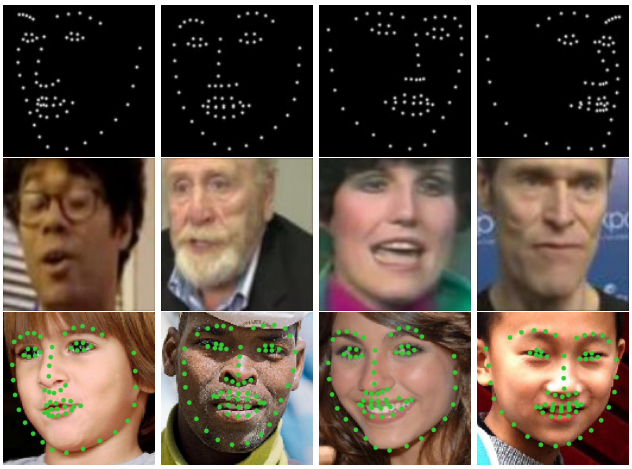

Figure 6. **Unaligned transfer.** We leverage the small number ($\approx$ 4000) of landmark annotations, without using corresponding images, in the MultiPIE dataset [56] **[top]** and unlabelled images from the the large-scale VoxCeleb [43] **[middle]** (32 million frames, 1300 identities) to train a detector that we test on the 300-W dataset [54] **[bottom]** (predictions in green) with state-of-the-art results (table 3). More qualitative results can be found in the supplementary.

e.g. from synthetic or motion-capture datasets. It is also used to introduce a *tight bottleneck* in the image autoencoding process, discussed next.

3.2. Conditional image autoencoder

The cycle consistency constraint $\Psi \circ \Phi(\boldsymbol{x}) \approx \boldsymbol{x}$ in CycleGAN can be viewed as an autoencoder for the images, where the image code $\Phi(\boldsymbol{x})$ is a skeleton. Our modification retains the encoder $\Phi : \mathcal{X} \to \mathcal{Y}$ as a map from image $\boldsymbol{x}$ to skeleton $\boldsymbol{y}$, but the decoder $\Psi : \mathcal{Y} \times \mathcal{X} \to \mathcal{X}$ now reconstructs the input image $\boldsymbol{x}$ given the skeleton image $\boldsymbol{y}$ as well as a second *conditioning* image $\boldsymbol{x}'$, supplanting the appearance information which is missing in the skeleton due to the bottleneck.

Since the generator performs the inverse operation of the encoder, for a pair of images $(\boldsymbol{x}, \boldsymbol{x}')$ differing only by pose, the two functions satisfy the modified cycle consistency constraint:

$$\Psi(\Phi(\boldsymbol{x}), \boldsymbol{x}') = \boldsymbol{x}. \tag{3}$$

The encoder is learned by competing against an adversarial discriminator $D_{\mathcal{Y}}$, whose purpose is to distinguish between the generated skeleton images $\{\boldsymbol{y} = \Phi(\boldsymbol{x})\}$ and samples $\{\boldsymbol{y}\}$ of real skeleton images obtained from unaligned pose data [17]. Note that neither constraint (3) nor the discriminator require to label images with pose information; instead, they only require unlabelled images and unaligned skeletons.

The bottleneck of section 3.1 is introduced in the autoencoder (3) as follows:

$$\Psi(\beta \circ \eta \circ \Phi(\boldsymbol{x}), \boldsymbol{x}') = \boldsymbol{x}. \tag{4}$$

In this equation, the skeleton autoencoder (η, β) is pre-trained and remains fixed as the rest of the model is learned. Together with the fact that the skeleton generator is handcrafted, this prevents the image encoder Φ from leaking any appearance information through the skeleton image. In particular, the skeleton autoencoder $\beta \circ \eta = 1$ acts as the identity if the input is a proper skeleton image $\boldsymbol{y}$; however, when $\boldsymbol{y} = \Phi(\boldsymbol{x})$ is imperfect because it is produced by the image encoder, then $\beta \circ \eta$ effectively reprojects $\boldsymbol{y}$ to a "clean" version of skeleton $\boldsymbol{y}^* = \beta(\eta(\boldsymbol{y}))$.

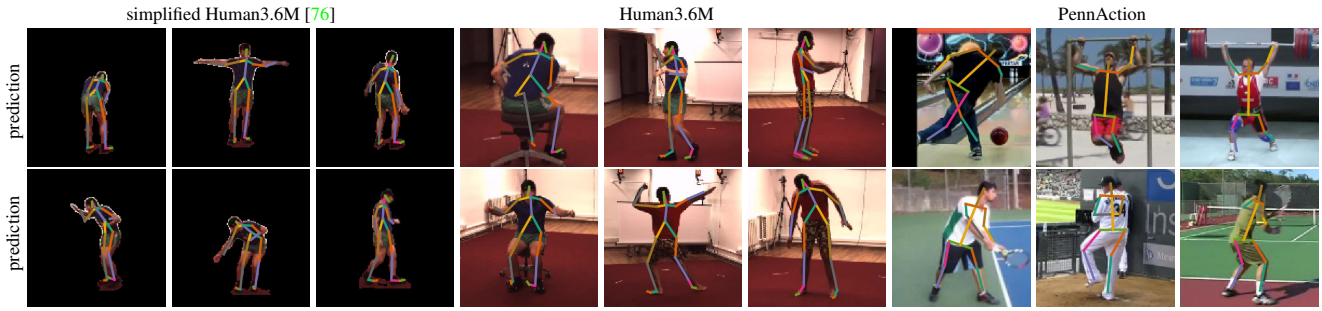

Figure 7. **Human pose predictions.** 2D Keypoint predictions (visualised as connected limbs) on the simplified [76] (with no background), full Human3.6M [21], and PennAction [75] test sets. Our method detects keypoints in complex poses, however, it might not always disambiguate between the left and right sides due to bilateral symmetry of the human body. More samples are included in the supplementary.

We employ eq. (4) as the primary signal for learning our model, as explained in the next section.

3.3. Learning objective

Equation (4) is enforced via a *perceptual loss* [6, 12, 16, 24] $L_{\text{perc}}(\hat{\boldsymbol{x}}, \boldsymbol{x})$ where $\hat{\boldsymbol{x}} = \Psi(\beta \circ \eta \circ \Phi(\boldsymbol{x}), \boldsymbol{x}')$ is the result of the conditional autoencoding process. In addition to the perceptual loss, we employ an adversarial loss to align the distributions of skeletons $\boldsymbol{y} = \Phi(\boldsymbol{x})$ generated by the image encoder and of genuine skeletons $\bar{\boldsymbol{y}}$, utilising discriminator network $D_{\mathcal{Y}}$. We use the squared difference adversarial loss as proposed in [41]:

$$L_{\text{disc}}(\boldsymbol{y}, \bar{\boldsymbol{y}}) = \mathop{\mathbb{E}}_{\bar{\boldsymbol{y}} \sim \mathcal{Y}}[(1 - D_{\mathcal{Y}}(\bar{\boldsymbol{y}}))^2] + \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{X}}[D_{\mathcal{Y}}(\Phi(\boldsymbol{x}))^2]. \tag{5}$$

An iteration of training the model consists of first sampling two frames $(\boldsymbol{x}, \boldsymbol{x}') \sim p(\boldsymbol{x}, \boldsymbol{x}')$ from the same video as well as a random skeleton image $\bar{\boldsymbol{y}} \sim p(\boldsymbol{y})$. Then, the autoencoders are evaluated to obtain $\boldsymbol{y} = \Phi(\boldsymbol{x})$, $\boldsymbol{y}^* = \beta(\eta(\boldsymbol{y}))$, and $\hat{\boldsymbol{x}} = \Psi(\boldsymbol{y}^*, \boldsymbol{x}')$. Losses are combined in the following joint objective:

$$L_{\text{perc}}(\hat{\boldsymbol{x}}, \boldsymbol{x}) + \lambda L_{\text{disc}}(\boldsymbol{y}, \bar{\boldsymbol{y}}). \tag{6}$$

4. Implementation details

Our model comprises the image encoder Φ, the image decoder Ψ, skeleton encoder η, the keypoint encoder β and the skeleton discriminator β (see fig. 3). These are implemented using the neural network modules described below (see also fig. 3 and the supplementary material for further details). The source code and models will be made available.

Modules. The *downsampling module* takes an image (*e.g.* skeleton / appearance image) and encodes it as a tensor with $\frac{1}{8}$ the spatial resolution. The module consists of 4 convolutional blocks with stride 2, each followed by batch normalization and ReLU layers. All the convolutional layers contain 3×3 kernels except for the first one which is 7×7. The blocks are then followed by 1×1 convolution. The number of filters is set to start with 32 and doubled at each block.

The *upsampling module* takes a code tensor and outputs a higher spatial resolution tensor (i.e. skeleton or image). It consists of 4 blocks, each containing two convolutional layers with 3×3 filters. All but the first block start by doubling the resolution of the input using bilinear upsampling. The first block halves the number of feature channels and only the very last outputs the desired number of channels — 1 for the skeleton and 3 for an image.

Components. The *image encoder* Φ takes an image $\boldsymbol{x}$ as input and returns skeleton image $\boldsymbol{y}$. It contains a downsampling and upsampling module.

The *image generator* Ψ takes clean skeleton $\boldsymbol{y}^*$ and an image $\boldsymbol{x}'$ as input and generates image $\hat{\boldsymbol{x}}$. It contains two downsampling and one upsampling module. The first downsampling module takes $\boldsymbol{x}'$ and the second one $\boldsymbol{y}^*$. Their outputs are then concatenated and fed into the upsampling module.

The *skeleton encoder* η uses the downsampling module to take in a generated skeleton image and outputting K heatmaps. The locations of keypoints are further obtained as in [23] by converting each heatmap into a 2D probability distribution. The expectation of this probability distribution corresponds to the location of the keypoints. The *skeleton generator* β takes K 2D keypoints from the output of η and generates a clean skeleton image. The spatial coordinates are normalised to the $[-1, 1]$ range; correspondingly, we use $\gamma = \frac{1}{0.04}$ and $\sigma = 0.02$ in eqs. (1) and (2) respectively. As explained in section 3.1, this function does not contain any learnable parameters. It is differentiable and thus we can backpropagate the error signal through this function during training of the complete model.

The *skeleton discriminator* $D_{\mathcal{Y}}$ follows the discriminator architecture of [79]. It takes a set of generated and real skeleton images $\bar{\boldsymbol{y}}$ and $\Phi(\boldsymbol{x})$, aims to distinguish between them and outputs a scalar score for each image. We use three such discriminators each for a different scale of the input image. We resize the input images by 1, $\frac{1}{2}$, and $\frac{1}{4}$ factors.

	method	all	wait	pose	greet	direct	discuss	walk	eat	phone	purchase	sit	sit down	smoke	take photo	walk dog	walk together
supervised	hourglass [44]	20.22	16.42	14.55	17.58	16.70	20.92	14.11	15.47	19.31	20.45	26.18	40.93	19.68	23.13	22.43	15.41
		19.52	15.53	13.88	17.14	15.81	19.55	**13.74**	15.33	18.81	19.88	25.85	39.07	19.40	22.24	21.58	14.96
	ours	22.30	19.24	21.19	20.23	18.09	22.64	21.69	17.81	22.01	22.32	24.49	28.66	22.83	23.17	26.81	26.36
		19.35	16.12	16.31	16.69	15.66	19.17	13.94	16.41	20.47	19.43	**23.86**	**26.32**	20.96	21.99	21.78	18.57
	ours	26.62	22.74	24.74	23.84	23.03	22.03	24.19	22.20	25.43	28.69	29.10	45.99	25.80	27.20	26.02	28.35
	3DHP skeletons	21.41	18.87	18.71	19.39	18.40	19.26	18.06	17.89	21.35	23.37	24.84	34.03	21.25	23.18	22.47	20.00
	ours	23.32	18.81	15.17	17.98	14.90	18.26	21.69	20.83	25.53	19.96	32.98	42.24	23.55	23.57	23.83	25.19
	H3.6M skeletons	**17.61**	**14.16**	**11.88**	**13.78**	**12.62**	**16.44**	17.88	**14.50**	**17.56**	**18.31**	24.05	33.36	**17.65**	**20.10**	**20.42**	**14.01**

Table 1. **2D Human landmark detection.** Comparison on the full Human3.6M test set with supervised baselines — (1) Stacked Hourglass [44], and (2) our model trained with supervision. We report the MSE in pixels for each activity. Shaded rows show error for the original predictions of the model (*no flip* in section 5.1), while unshaded rows represent the minimum of the errors obtained with and without flipping the predictions against the axis of bilateral symmetry (*with flips* in section 5.1). We highlight the minimum error across all models in bold.

Second cycle constraint and discriminator. CycleGAN enforces two cycle constraints $\Psi \circ \Phi(\boldsymbol{x}) \approx \boldsymbol{x}$ and $\Phi \circ \Psi(\boldsymbol{y}) \approx \boldsymbol{y}$. Our model implements a conditional version (4) of the first, while the second can be written as $\Phi(\Psi(\bar{\boldsymbol{y}}, \boldsymbol{x}')) \approx \bar{\boldsymbol{y}}$. CycleGAN also utilizes a discriminator $D_{\mathcal{X}}$ on images $\hat{\Psi}(\boldsymbol{y})$ generated from skeletons to match their distribution to images $\boldsymbol{x}$; the same discriminator applies here, except that images are generated conditionally $\Psi(\bar{\boldsymbol{y}}, \boldsymbol{x}')$ and they are tested against the distribution of images $\boldsymbol{x}$ from the same video, so $\mathcal{D}_{\mathcal{X}}(\Psi(\bar{\boldsymbol{y}}, \boldsymbol{x}'), \boldsymbol{x}')$ is conditional too. The architecture of the discriminator is based on [28] and detailed in the supplementary. We found the additional cycle constraint and discriminator to bring negligible performance benefit, while increasing the complexity of the model significantly, so we do not include them in our "gold-standard" version of the model. However, we ablate these components in the experiments.

Training details. We first train the skeleton encoder η in an offline step using unpaired keypoints and corresponding synthetically generated pose images through an ℓ_2-loss regression loss on the keypoints. The learning rate is decreased once by a factor of 10 when the error plateaus. Once the skeleton encoder (η) is trained, we freeze its parameters and incorporate the skeleton auto-encoder ($\beta \circ \eta$) into our model to train the mappings between images and pose images (Φ, Ψ) by minimising the objective in eq. (6) ($\lambda = 10$). For both the stages, we use the Adam optimiser [30] with a learning rate of 2e-4, $\beta_1 = 0.5$ and $\beta_2 = 0.999$; batch size is set to 16 and the norm of the gradients is clipped to 1.0 for stability. We train for 3 million iterations for human pose experiments and 300k for faces. All the network parameters are trained from scratch.

5. Experiments

We evaluate our method on the task of 2D landmark detection for human pose (section 5.1) and faces (section 5.2), and outperform the state-of-the-art methods (tables 1 to 3) for both. Finally, we examine the relative contribution of various components of our model in a detailed ablation study (section 5.3).

5.1. Human pose

Datasets. **Human3.6M [21]** is a large-scale dataset that contains 3.6M accurate 2D and 3D human pose annotations for human subjects doing 17 different activities, imaged under 4 different viewpoints and a static background. For training, we use subjects 1, 5, 6, 7, and 8, and subjects 9 and 11 for evaluation, as in [67]. **Simplified Human3.6M** introduced by Zhang *et al*. [76] for evaluating unsupervised pose recognition, contains 6 activities in which human bodies are mostly upright; it comprises 800k training and 90k testing images. **PennAction [75]** contains 2k challenging consumer videos of 15 sports categories. **MPI-INF-3DHP [42]** contains videos from 8 subjects performing 8 activities in complex exercise poses and covers a wider range of poses than the Human3.6M dataset. There are 28 joints annotated.

We split datasets into two *disjoint* parts for sampling image pairs $(\boldsymbol{x}, \boldsymbol{x}')$ (cropped to the provided bounding-boxes), and keypoints $(\bar{\boldsymbol{y}})$ respectively to ensure that the data does not contain labels corresponding to the training images. For Human3.6M datasets, we split the videos in half, while for PennAction, we split in half the set of videos from each action category. In some experiments, we use skeletons from MPI-INF-3DHP dataset as the only source of a skeleton prior.

Evaluation. We report 2D landmark detection performance on the full and simplified Human3.6M datasets. For full Human3.6M, we follow the standard protocol and report the mean error in pixels over 17 of the 32 joints (but the model estimates all 32 joints). For simplified Human3.6M we follow [76] and report the error for all 32 joints normalized by the image size. To demonstrate learning from unaligned labels, we consider two settings for sourcing the images and unpaired landmarks – (1) *different* datasets: im-

method	all	wait	pose	greet	direct	discuss	walk
supervised							
hourglass [44]	**2.16**	**1.88**	**1.92**	**2.15**	**1.62**	**1.88**	**2.21**
unsupervised + supervised linear regression (with flips)							
Thewlis *et al*. [59]	7.51	7.54	8.56	7.26	6.47	7.93	5.40
Zhang *et al*. [76]	4.14	5.01	4.61	4.76	4.45	4.91	4.61
ours *no flips*	7.52	6.86	7.96	7.29	8.34	7.25	7.44
with flips	**2.91**	**2.92**	**2.60**	**2.96**	**2.54**	**2.45**	**3.99**

Table 2. **Simplified 2D human landmark detection.** Comparison with state-of-the-art methods for 2D human landmark detection on the Simplified Human3.6 dataset [76]. We report %-MSE normalised by image size for each activity. *no flips / with flips* represent errors with and without taking the minimum across the axis of bilateral symmetry (see section 5.1).

ages from the full Human3.6M and landmarks from MPI-INF-3DHP. (2) *same* datasets: images and landmarks are sampled from a *disjoint* split (as explained above) of the Human3.6M datasets. When using MPI-INF-3DHP dataset as the source of skeletons, we predict 28 joints, but use 17 joints that are common with Human3.6M for evaluation. We train our method from scratch and compare its performance with both supervised and unsupervised methods.

Results. Table 1 summarises results on the full Human3.6M test set. As noted in [23, 76], for unsupervised methods it may be difficult to distinguish the frontal and dorsal views of a person. Hence, following Zhang *et al*. [76] we also report the minimum error between the estimate obtained by the model and its symmetric version (labelled *with flips*). We compare against supervised baselines — (1) the state-of-the-art Stacked Hourglass [44] (using 1-stack), and (2) our model architecture trained with labelled data. With *flips*, our unsupervised model outperforms the baselines overall, however, is worse on challenging activities like sitting and walking. Notably, due to limited variability in the dataset, the supervised baselines overfit on the training set, while our method does not, the training / test error (*with flips*) are — supervised hourglass: 14.61 / 19.52, ours supervised: 9.67 / 19.35, ours unsupervised : 18.00 / 17.61. When training with different sources of skeletons (MPI-INF-3DHP) and images (full Human3.6M), the error is marginally ($\approx$4 MSE points) higher: 21.41 (w/ flip), 26.62 (no flip). This is due to the domain gap between the two datasets.

Table 2 summarises results on the Simplified Human3.6M. Our model significantly outperforms previous methods [59, 76] on all activities, even *without any supervised linear regression* as required by them (see fig. 5).

5.2. Human faces

Datasets. **VoxCeleb [43]** is a large-scale audio-visual dataset consisting of 100k short clips of human speech, obtained from interview videos uploaded to YouTube. **MultiPIE [56]** is a dataset of 750k facial images of 337 people under 15 viewpoints and 19 illumination conditions. It contains 68 labelled facial landmarks for 6k images. **300-W [54]** is a challenging dataset of facial images obtained by combining multiple datasets [5, 50, 78] as described in [51, 59]. As in MultiPIE, 300-W contains 68 annotated facial landmarks. We keep 300-W as our test dataset and follow the evaluation protocol in [51].

method	300-W
supervised	
LBF [51]	6.32
CFSS [80]	5.76
cGPRT [34]	5.71
DDN [73]	5.65
TCDCN [77]	5.54
RAR [70]	4.94
Wing Loss [14]	**4.04**
self-supervised + supervised regression	
Thewlis *et al*. [58]	9.30
Thewlis *et al*. [59]	7.97
Wiles *et al*. [69]	5.71
ours	
unaligned *(VoxCeleb / MultiPIE)*	9.85
+ supervised regression	5.56
unaligned *(VoxCeleb / VoxCeleb)*	7.86
+ supervised regression	**5.37**

Table 3. **Facial landmark detection.** Comparison with state-of-the-art methods on 2D facial landmark detection. We report the inter-ocular distance normalised keypoint localisation error [77] (in %; $\downarrow$ is better) on the 300-W test set. Datasets in parenthesis refer to the source of unaligned examples of *(images / landmarks)* respectively.

Results. As for human pose, we study two scenarios for generating a training set with unpaired face and landmark images. In the first, images and facial landmarks are sourced from *different* datasets, VoxCeleb and MultiPIE (6k landmarks) respectively (fig. 6). In the second, we source from both from VoxCeleb but from different identities. For VoxCeleb, pseudo-labels were obtained by running the dlib facial landmark detector [29]. We train our method from scratch for each case and report its performance on 300-W in table 3. Our method performs well even without any fine-tuning on the target 300-W. As expected, our method performs better when the unpaired images come from a single dataset, where we also outperform the unsupervised methods of [58, 59]. When we learn a supervised linear regressor (on 300-W training set) as also in [69], we outperform all the unsupervised and even supervised methods except [14, 70]. Table 4 demonstrates that a very small number (=50) of unpaired landmarks are sufficient to retain the performance of our method.

# unaligned samples	6k	1k	500	50
300-W error	9.85	10.10	10.28	10.31

Table 4. **Varying # of unaligned MultiPIE annotation samples.** We train our method using varying numbers of MultiPIE annotations and evaluate the performance on 300-W dataset. We show that decreasing the number of annotations to 50 does not result in significatly worse performance.

5.3. Ablation study

We study the relative contribution of each of the proposed components, (1) conditional image generator Ψ, (2) skeleton bottleneck $\beta \circ \eta$, and (3) removing the second cycle-consistency constraint, for both human pose and facial landmark detection, and report the results in table 5. We train our model on simplified Human3.6M [76] for human pose and on VoxCeleb/MultiPIE (see table 3) data for faces (no fine-tuning on test sets).

We start from CycleGAN as our base model and modify it by conditioning its image generator on a second appearance image. This conditioning partially ameliorates appearance leakage through the skeleton image (fig. 2) and reduces landmark detection error rate for humans from 4.39% to 4.07%. Next, adding the skeleton bottleneck further decouples appearance from pose, resulting in a significant improvement for both for humans (3.01% *vs.* 4.39% CycleGAN) and faces (10.10% *vs.* 18.51% CycleGAN).

Finally, we remove the second cycle-consistency constraint which simplifies the network architecture and the training procedure drastically. This simplification again results in a marked for both: $\Delta = 0.1\%$ for humans, and $\Delta = 0.25\%$ for faces. This is because the second cycle imposes a contrariant task — generating the appearance identity in the pose represented in the *independently* sampled landmarks image, such that the landmarks can be recovered from the generated image. Since the landmarks encode the facial shape, they are not perfectly decoupled from identity, making this difficult. Abandoning the reconstruction constraint relieves the model from this distress.

method	humans	faces
CycleGAN	4.39	18.51
+ conditional generator	4.07	–
+ skeleton-bottleneck	3.01	10.10
− 2nd cycle = **ours**	2.91	9.85

Table 5. **Ablation study.** We start with the CycleGAN [79] model and sequentially augment it with — (1) conditional image generator (Ψ), (2) skeleton bottleneck ($\beta \circ \eta$), and (3) remove the second cycle-constraint (see section 4) resulting in our proposed model. We report 2D landmark detection error ($\downarrow$ is better) on the simplified Human3.6M (*with flips*; see section 5.1) for human pose, and on the 300-W (section 5.2) for faces.

6. Conclusion

We presented an unsupervised method that can learn to predict pose from unaligned pairs of human and skeleton/keypoint images. We showed that recent unsupervised image-to-image translation techniques such as CycleGAN are not well suited to pose estimation problem where the mapping from a skeleton to a human image is not unique. To this end, we proposed multiple technical innovations including a conditioning technique and an analytical and differentiable bottleneck that enables an decouples the appearance and style information. Our method achieves the best landmark detection accuracy on multiple benchmarks and narrows down the gap between supervised and unsupervised methods.

Acknowledgements. We are grateful for the support of ERC 638009-IDIU, and the Clarendon Fund scholarship. We would like to thank Triantafyllos Afouras for immense support, and Relja Arandjelovi for helpful advice.

References

[1] I. Akhter and M. J. Black. Pose-conditioned joint angle limits for 3d human pose reconstruction. In *Proc. CVPR*, pages 1446–1455, 2015.

[2] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proc. CVPR*, pages 3686–3693, 2014.

[3] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Proc. CVPR*, pages 1014–1021. IEEE, 2009.

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

[5] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. *TPAMI*, 35(12):2930–2940, 2013.

[6] J. Bruna, P. Sprechmann, and Y. LeCun. Super-resolution with deep convolutional sufficient statistics. In *Proc. ICLR*, 2016.

[7] A. Bulat and G. Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *Proc. ECCV*, pages 717–732. Springer, 2016.

[8] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proc. CVPR*, pages 7291–7299, 2017.

[9] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human pose estimation with iterative error feedback. In *Proc. CVPR*, pages 4733–4742, 2016.

[10] X. Chen and A. L. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Proc. NIPS*, pages 1736–1744, 2014.

[11] C. Chu, A. Zhmoginov, and M. Sandler. Cyclegan, a master of steganography, 2017.

[12] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances*

in Neural Information Processing Systems, pages 658–666, 2016.

[13] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International journal of computer vision*, 61(1):55–79, 2005.

[14] Z.-H. Feng, J. Kittler, M. Awais, P. Huber, and X.-J. Wu. Wing loss for robust facial landmark localisation with convolutional neural networks. In *Proc. CVPR*, 2018.

[15] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015.

[16] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proc. CVPR*, pages 2414–2423, 2016.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proc. NIPS*, pages 2672–2680, 2014.

[18] A. Gupta, A. Vedaldi, and A. Zisserman. Learning to read by spelling: Towards unsupervised text recognition. In *Proc. ICVGIP*, 2018.

[19] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017.

[20] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *Proc. ECCV*, pages 34–50. Springer, 2016.

[21] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *TPAMI*, 36(7):1325–1339, jul 2014.

[22] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proc. CVPR*, 2017.

[23] T. Jakab, A. Gupta, H. Bilen, and A. Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *Proc. NIPS*, 2018.

[24] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proc. ECCV*, pages 694–711. Springer, 2016.

[25] S. Johnson and M. Everingham. Learning effective human pose estimation from inaccurate annotation. In *Proc. CVPR*, pages 1465–1472. IEEE, 2011.

[26] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In *Proc. CVPR*, 2018.

[27] A. Kanazawa, D. W. Jacobs, and M. Chandraker. Warpnet: Weakly supervised matching for single-view reconstruction. In *Proc. CVPR*, pages 3253–3261, 2016.

[28] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[29] D. E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10(Jul):1755–1758, 2009.

[30] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. NIPS*, pages 1097–1105, 2012.

[32] G. Larsson, M. Maire, and G. Shakhnarovich. Learning representations for automatic colorization. In *European Conference on Computer Vision*, pages 577–593. Springer, 2016.

[33] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[34] D. Lee, H. Park, and C. D. Yoo. Face alignment using cascade gaussian process regression trees. In *Proc. CVPR*, 2015.

[35] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Proc. ECCV*, pages 740–755. Springer, 2014.

[36] B. Liu and V. Ferrari. Active learning for human pose estimation. In *Proc. CVPR*, pages 4363–4372, 2017.

[37] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, pages 700–708, 2017.

[38] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016.

[39] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013.

[40] R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proc. CVPR*, pages 5667–5675, 2018.

[41] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley. Least squares generative adversarial networks. In *Proc. ICCV*, pages 2794–2802, 2017.

[42] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *2017 International Conference on 3D Vision (3DV)*, pages 506–516. IEEE, 2017.

[43] A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612*, 2017.

[44] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *Proc. ECCV*, 2016.

[45] W. Ouyang, X. Chu, and X. Wang. Multi-source deep learning for human pose estimation. In *Proc. CVPR*, pages 2329–2336, 2014.

[46] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proc. CVPR*, pages 2536–2544, 2016.

[47] T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. In *Proc. CVPR*, pages 1913–1921, 2015.

[48] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Poselet conditioned pictorial structures. In *Proc. CVPR*, pages 588–595, 2013.

[49] V. Ramakrishna, D. Munoz, M. Hebert, J. A. Bagnell, and Y. Sheikh. Pose machines: Articulated pose estimation via inference machines. In *Proc. ECCV*, pages 33–47. Springer, 2014.

[50] D. Ramanan and X. Zhu. Face detection, pose estimation, and landmark localization in the wild. In *Proc. CVPR*, 2012.

[51] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *Proc. CVPR*, 2014.

[52] I. Rocco, R. Arandjelovic, and J. Sivic. Convolutional neural network architecture for geometric matching. In *Proc. CVPR*, volume 2, 2017.

[53] M. R. Ronchi, O. Mac Aodha, R. Eng, and P. Perona. It's all relative: Monocular 3d human pose estimation from weakly supervised data. In *BMVC*, 2018.

[54] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: Database and results. *Image and vision computing*, 47:3–18, 2016.

[55] B. Sapp, C. Jordan, and B. Taskar. Adaptive pose priors for pictorial structures. In *Proc. CVPR*, pages 422–429, 2010.

[56] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression (pie) database. In *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, pages 53–58. IEEE, 2002.

[57] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proc. ECCV*, pages 712–729. Springer, 2018.

[58] J. Thewlis, H. Bilen, and A. Vedaldi. Unsupervised learning of object frames by dense equivariant image labelling. In *Proc. NIPS*, 2017.

[59] J. Thewlis, H. Bilen, and A. Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *Proc. ICCV*, 2017.

[60] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. In *Proc. CVPR*, pages 648–656, 2015.

[61] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in neural information processing systems*, pages 1799–1807, 2014.

[62] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proc. CVPR*, pages 1653–1660, 2014.

[63] H.-Y. F. Tung, A. W. Harley, W. Seto, and K. Fragkiadaki. Adversarial inverse graphics networks: Learning 2d-to-3d lifting and image-to-image translation from unpaired supervision. In *Proc. ICCV*, volume 2, 2017.

[64] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *Proc. CVPR*, pages 4068–4076, 2015.

[65] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *Proc. CVPR*, 2017.

[66] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[67] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee. Learning to generate long-term future via hierarchical prediction. In *Proc. ICML*, 2017.

[68] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *Proc. CVPR*, pages 4724–4732, 2016.

[69] O. Wiles, A. S. Koepke, and A. Zisserman. Self-supervised learning of a facial attribute embedding from video. In *Proc. BMVC*, 2018.

[70] S. Xiao, J. Feng, J. Xing, H. Lai, S. Yan, and A. Kassim. Robust facial landmark detection via recurrent attentive-refinement networks. In *Proc. ECCV*, 2016.

[71] W. Yang, W. Ouyang, X. Wang, J. Ren, H. Li, and X. Wang. 3d human pose estimation in the wild by adversarial learning. In *Proc. CVPR*, volume 1, 2018.

[72] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Proc. CVPR*, pages 1385–1392. IEEE, 2011.

[73] X. Yu, F. Zhou, and M. Chandraker. Deep deformation network for object landmark localization. In *Proc. ECCV*. Springer, 2016.

[74] R. Zhang, P. Isola, and A. A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, volume 1, page 5, 2017.

[75] W. Zhang, M. Zhu, and K. G. Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2248–2255, 2013.

[76] Y. Zhang, Y. Guo, Y. Jin, Y. Luo, Z. He, and H. Lee. Unsupervised discovery of object landmarks as structural representations. In *Proc. CVPR*, pages 2694–2703, 2018.

[77] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Learning deep representation for face alignment with auxiliary attributes. *TPAMI*, 38(5):918–930, 2016.

[78] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin. Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 386–391, 2013.

[79] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. CVPR*, 2018.

[80] S. Zhu, C. Li, C. Change Loy, and X. Tang. Face alignment by coarse-to-fine shape searching. In *Proc. CVPR*, 2015.

Appendix

This supplementary material provides further technical details, illustrations and analysis Here we first detail how our method factorizes appearance and geometry (appendix A). We also show how this property can be used for style transfer and keypoint-conditioned image editing. We then provide extended version of qualitative results on facial landmarks detection (appendix B) and human pose estimation (appendix C). We show training progression over multiple checkpoints in terms of both keypoint estimation and conditioned image generation (appendix D). Finally, detailed description of the network architectures is provided (appendix E).

We also provide additional test results on facial landmarks detection, human pose estimation, and keypoint-conditioned image editing in the form of videos contained in `videos` folder accompanying this supplementary material.
File `human36m.mp4` shows results on Human3.6M test set, `voxceleb.mp4` results on VoxCeleb test set for the model trained using unpaired landmarks from MultiPIE dataset, and `editing.mp4` contains keypoint-driven animated image editing.

A. Appearance and geometry factorization

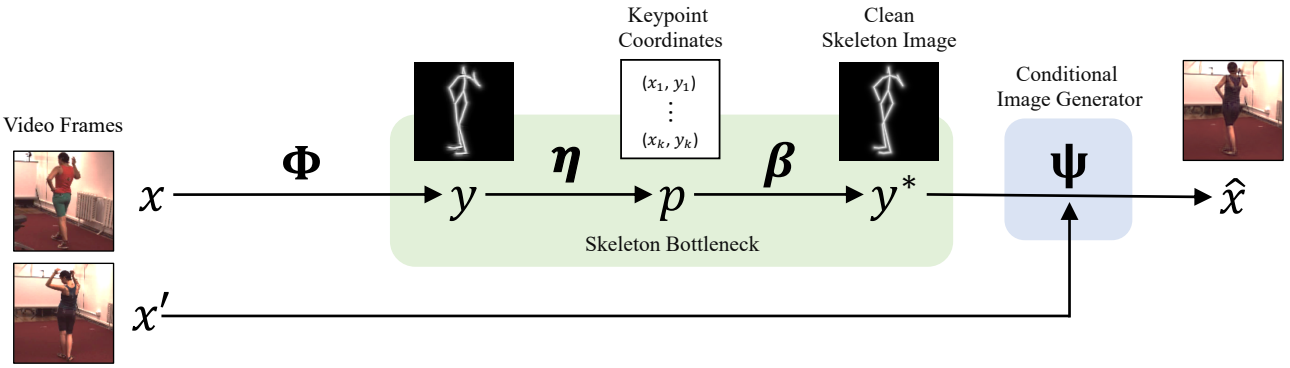

Figure 8. **Disentangling appearance and geometry.** During test time, the target image $\boldsymbol{x}$ and the style image $\boldsymbol{x}'$ contain different identity and viewpoint. The reconstructed image $\hat{\boldsymbol{x}}$ then inherits the geometry (pose) from the target image $\boldsymbol{x}$ and appearance from the style image $\boldsymbol{x}'$. More examples shown in fig. 9.

The conditional image generator $\Psi : (\boldsymbol{y}^*, \boldsymbol{x}') \mapsto \hat{\boldsymbol{x}}$ is tasked with generating a colour image ($\hat{\boldsymbol{x}}$) from the *clean* pose ($\boldsymbol{y}^*$) and a second appearance image ($\boldsymbol{x}'$). Due to our skeleton bottleneck ($\beta \circ \eta$), the pose image is devoid of any appearance information. Hence, the generator learns to pool the appearance information from the conditioning image, thereby *factorising* geometry (pose) and appearance.

While the image pairs $(\boldsymbol{x}, \boldsymbol{x}')$ are selected from same videos during training, here we sample the image pair $(\boldsymbol{x}, \boldsymbol{x}')$ with *different* appearances (*e.g.* from different videos) to better demonstrate the factorization of pose and appearance and show successful transfer of appearance from one to another (see fig. 8). Note, this also demonstrates significant generalisation over the training setting where the image pairs have the *same* appearance, *i.e.* are sampled from the same video. Figure 9 visualises this swapping. In fig. 10, we further leverage the disentanglement of geometry and appearance, and show fine-grained control of image generation through the pose keypoints.

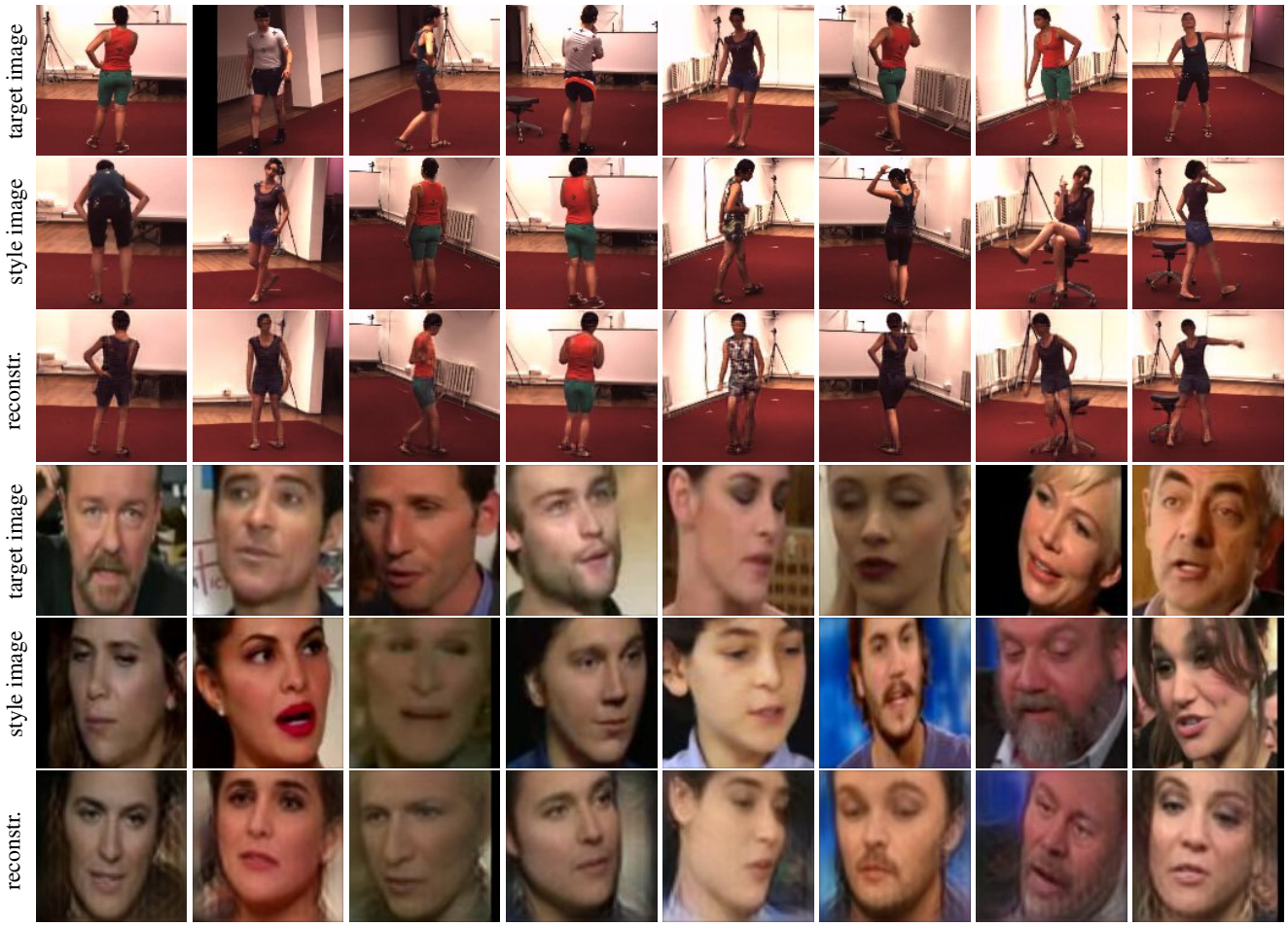

Figure 9. **Factorization of appearance and geometry.** *Reconstructed* image inherits appearance from the *style* image and geometry from the *target* image. **[top]:** human pose samples from Human3.6M. **[bottom]:** face samples from VoxCeleb.

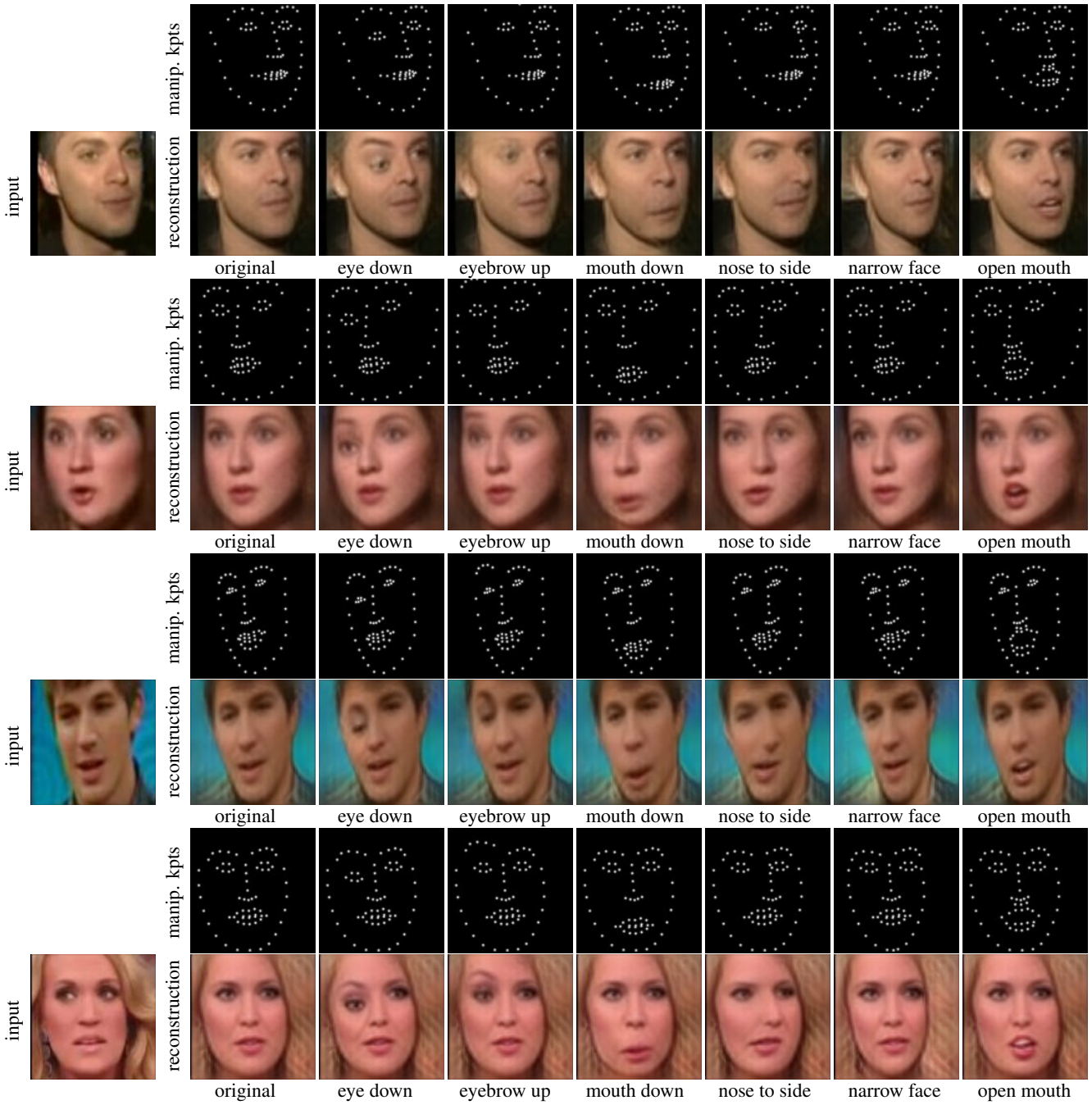

Figure 10. **Image editing using detected landmarks.** We show fine-grained control over the generated image by manipulating the coordinates of detected landmarks (*manip. kpts*). For example, we pick landmarks corresponding to an eye and move them down [second column], or open the mouth [last column] (note, the generator fills in the teeth absent in the input images). The resulting changes are localised and allow for fine-grained control. Apart from demonstrating successful disentanglement of appearance and geometry, this also suggests that the model assigns correct semantics to the detected landmarks. We provide further animations in `videos/editing.mp4`.

B. Facial landmarks detections

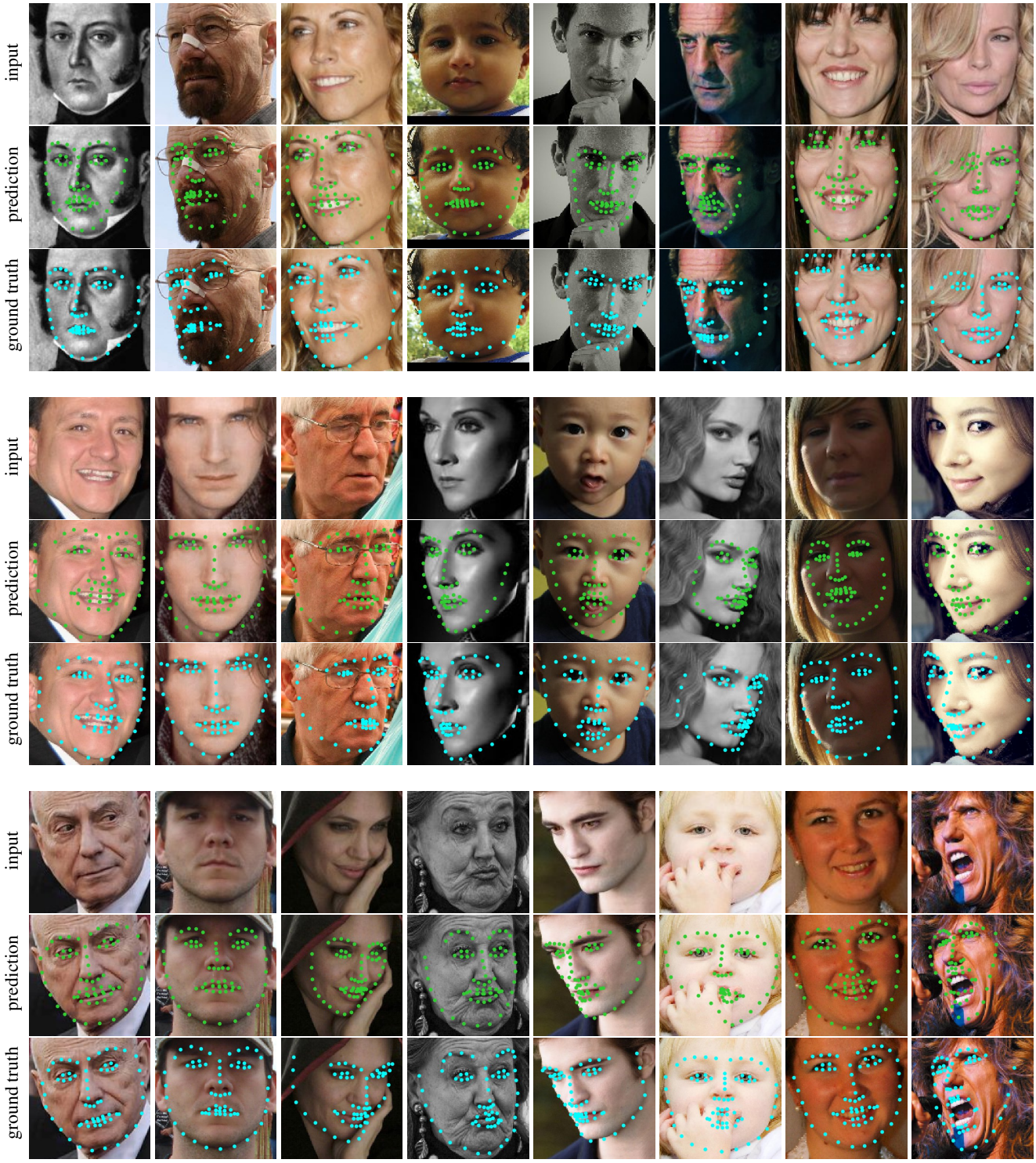

Figure 11. **Facial landmark detections on 300-W.** Randomly sampled predictions from 300-W test set. The model was trained with unlabeled images from VoxCeleb face videos dataset and unpaired landmarks sampled from MultiPIE dataset, hence shows significant generalisation. Green markers denote our detections, blue correspond to the ground truth.

C. Human pose estimation

C.1. Pose detection on Human3.6M

Figure 12. **Pose estimation on Human3.6M.** Randomly sampled results from Human3.6M test set. The model is trained with unpaired images and skeletons from Human3.6M. We show predictions on videos in `videos` folder accompanying this supplementary material.

C.2. Pose detection on Simplified Human3.6M

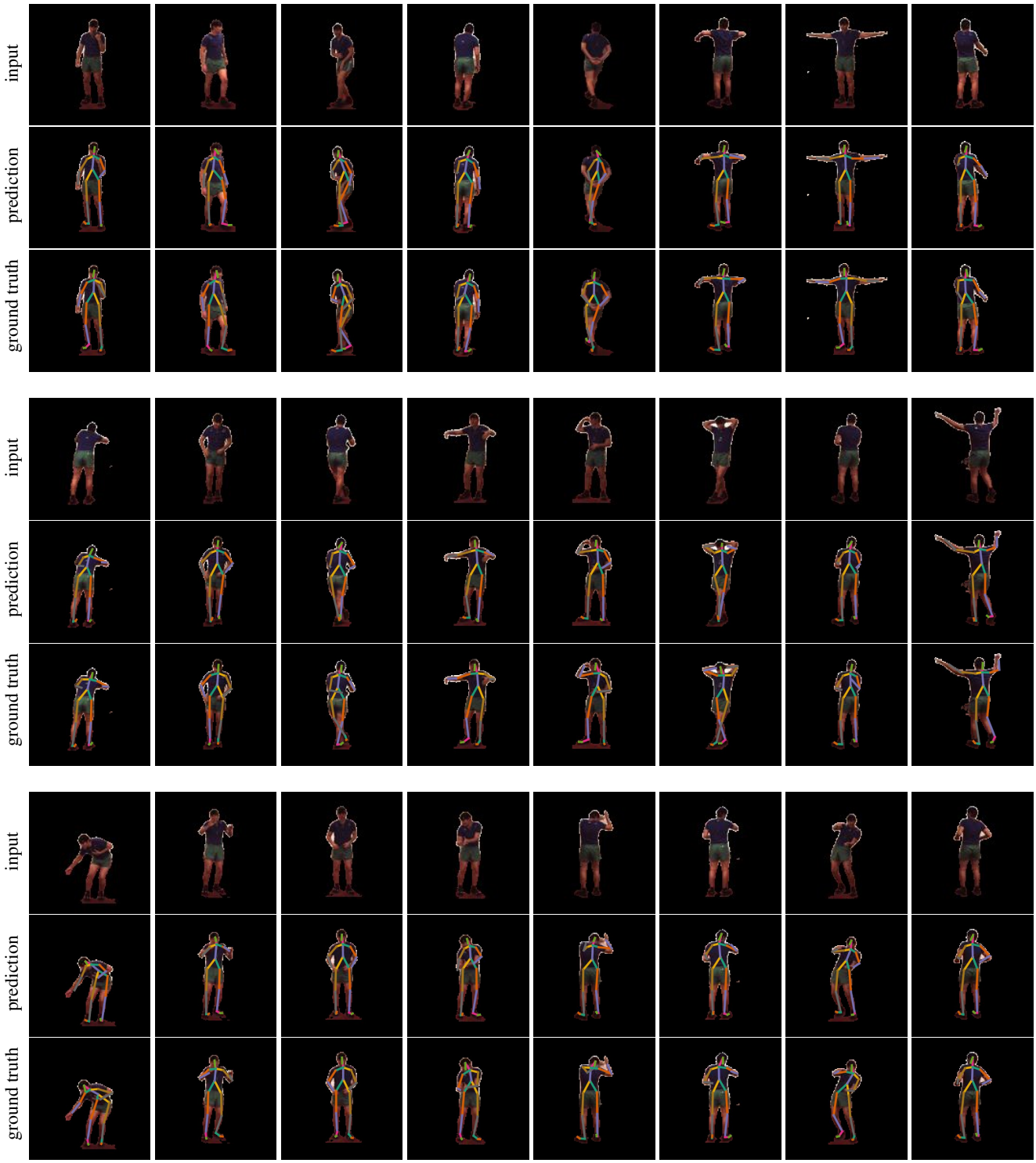

Figure 13. **Pose estimation on the Simplified Human3.6M.** Randomly sampled results from the Simplified Human3.6M test set. The model is trained with unpaired images and skeletons from Simplified Human3.6M.

C.3. Pose detection on PennAction

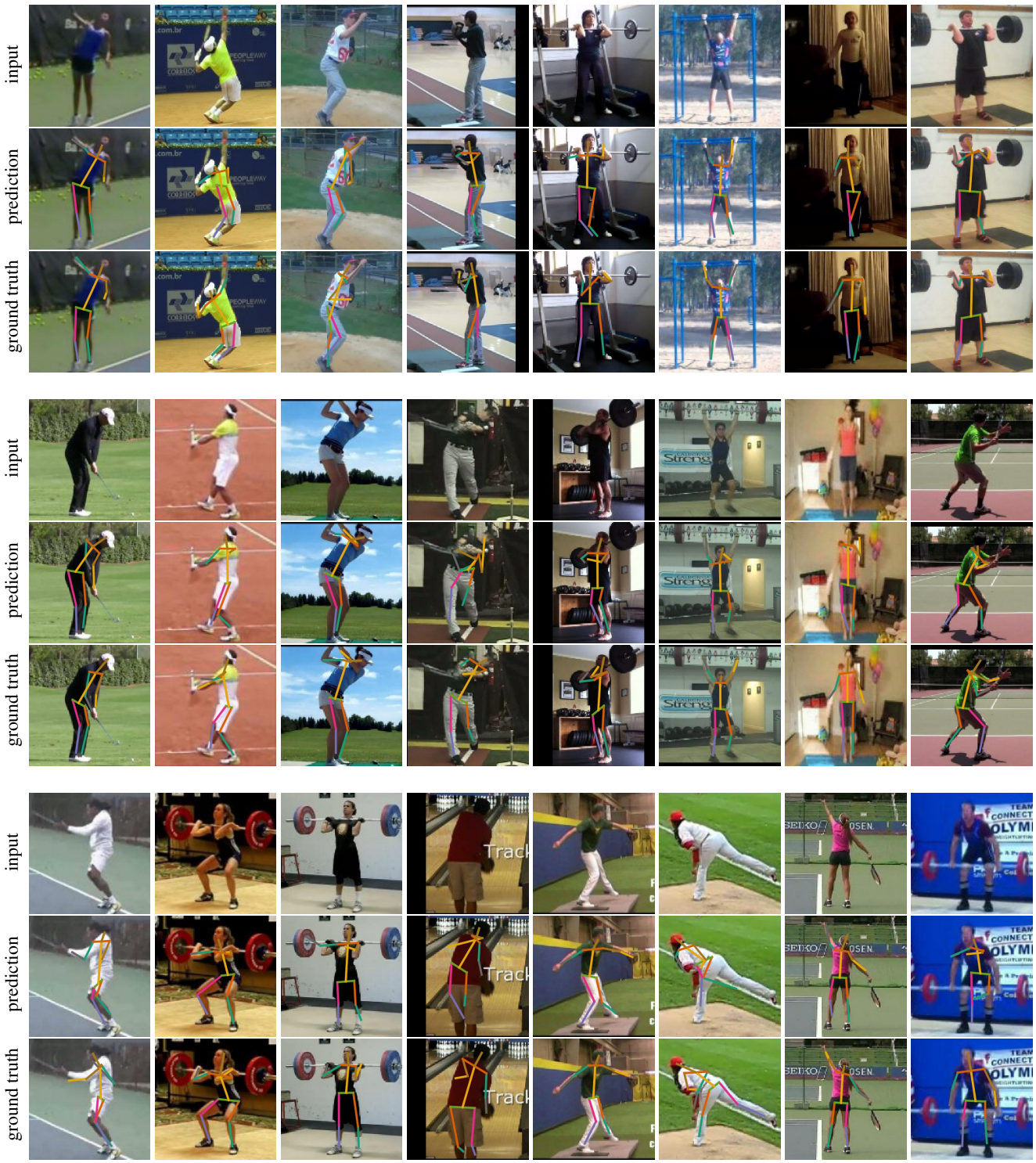

Figure 14. **Pose estimation on PennAction.** Randomly sampled results from PennAction test set. The model is trained with unpaired images and skeletons from PennAction.

D. Training progress

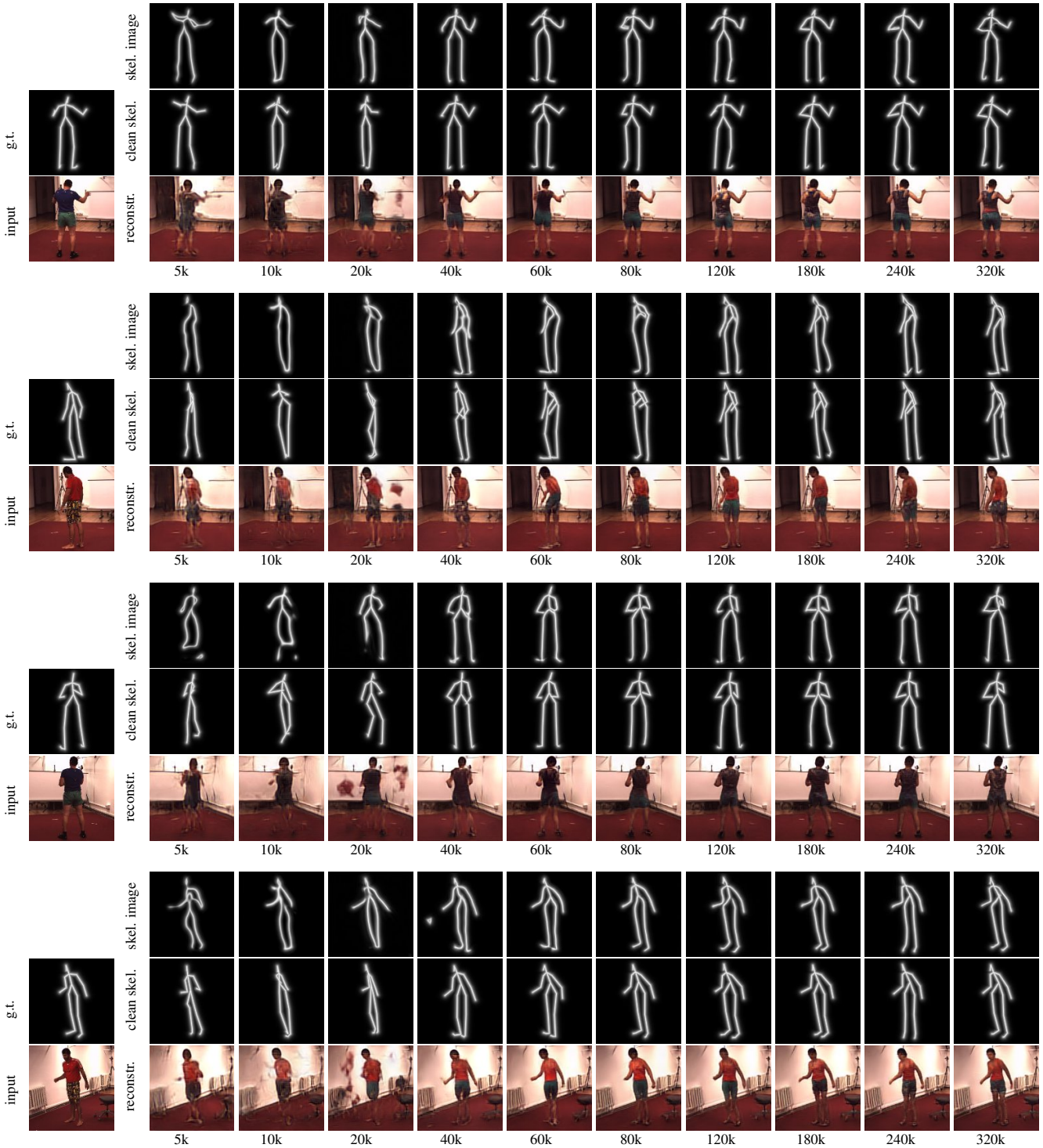

Figure 15. **Human pose training progression.** Samples from Human3.6M test set through the training iterations. Numbers below images denote elapsed training iterations. Our model learns to output plausible looking skeletons after about 10k-20k iterations, and learns to to align them with the input at convergence.

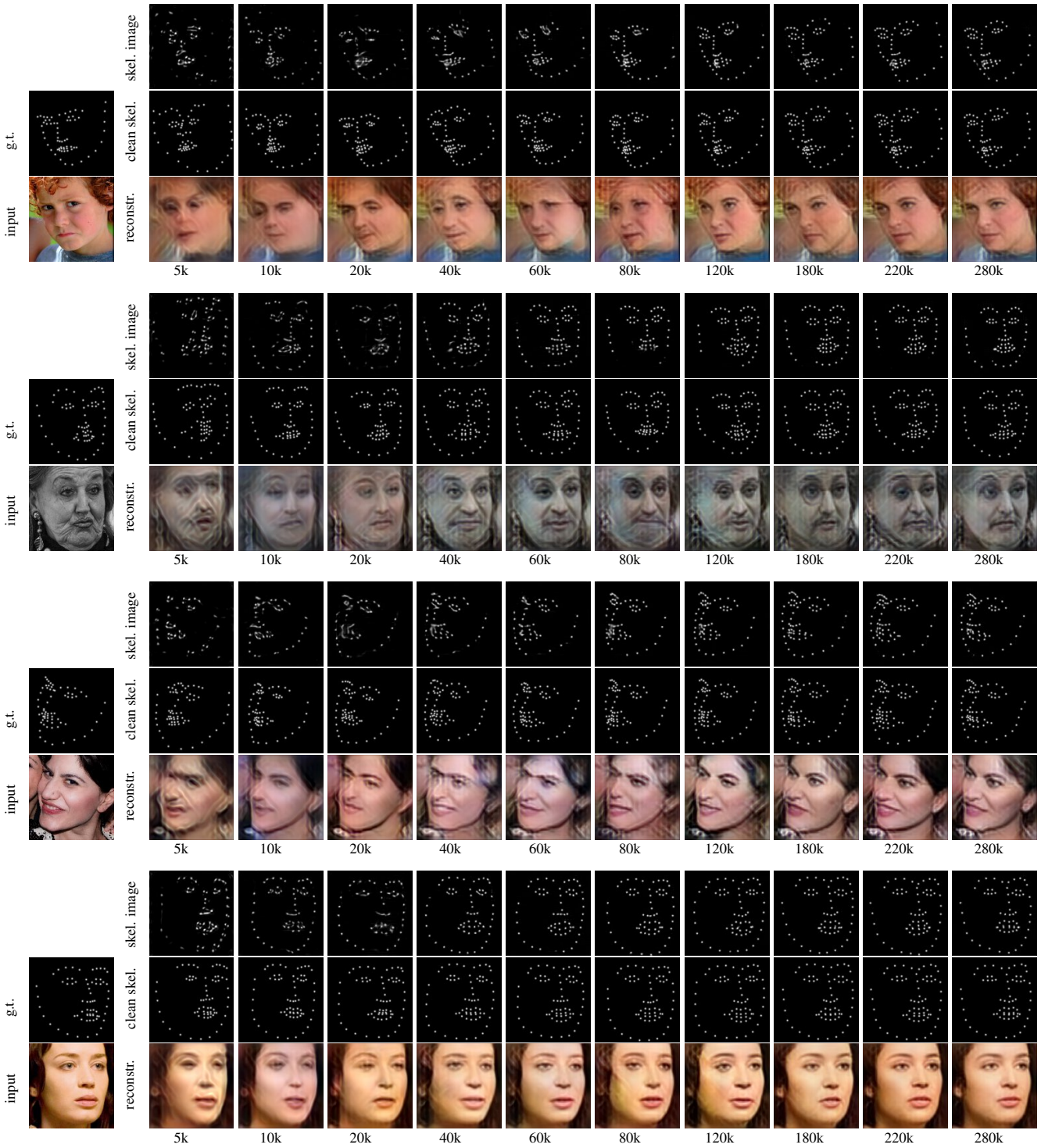

Figure 16. **Facial landmarks training progression.** Samples from the 300-W testset through the training iterations. Model is trained on unlabeled images from VoxCeleb dataset and unpaired landmarks from MultiPIE dataset. Numbers below images denote elapsed training iterations. Reasonably aligned landmark predictions are learned only after about 10k iterations.

E. Architectures

Type	Kernel	Stride	Output channels	Output size	Norm.	Activation
Input $\boldsymbol{x}$	-	-	3	128	-	-
Conv	7	1	32	128	Batch	ReLU
Conv	3	1	32	128	Batch	ReLU
Conv	3	2	64	64	Batch	ReLU
Conv	3	1	64	64	Batch	ReLU
Conv	3	2	128	32	Batch	ReLU
Conv	3	1	128	32	Batch	ReLU
Conv	3	2	256	16	Batch	ReLU
Conv	3	1	256	16	Batch	ReLU
Conv	1	1	256	16	None	None
Conv	3	1	256	16	Batch	ReLU
Conv	3	1	256	16	Batch	ReLU
Bilinear upsampl.	-	-	128	32	-	-
Conv	3	1	128	32	Batch	ReLU
Conv	3	1	128	32	Batch	ReLU
Bilinear upsampl.	-	-	64	64	-	-
Conv	3	1	64	64	Batch	ReLU
Conv	3	1	64	64	Batch	ReLU
Bilinear upsampl.	-	-	32	128	-	-
Conv	3	1	32	128	Batch	ReLU
Conv	3	1	1	128	None	None

Figure 17. **Image encoder Φ.** The network is composed of the encoder and decoder network from [23].

Type	Kernel	Stride	Output ch.	Output size	Norm	Activ
Input $\boldsymbol{x}'$	-	-	3	128	-	-
Conv	7	1	32	128	Batch	ReLU
Conv	3	1	32	128	Batch	ReLU
Conv	3	2	64	64	Batch	ReLU
Conv	3	1	64	64	Batch	ReLU
Conv	3	2	128	32	Batch	ReLU
Conv	3	1	128	32	Batch	ReLU
Conv	3	2	256	16	Batch	ReLU
Conv	3	1	256	16	Batch	ReLU
Conv	1	1	256	16	None	None

Type	Kernel	Stride	Output ch.	Output size	Norm	Activ
Input $\boldsymbol{y}^*$	-	-	1	128	-	-
Conv	7	1	32	128	Batch	ReLU
Conv	3	1	32	128	Batch	ReLU
Conv	3	2	64	64	Batch	ReLU
Conv	3	1	64	64	Batch	ReLU
Conv	3	2	128	32	Batch	ReLU
Conv	3	1	128	32	Batch	ReLU
Conv	3	2	256	16	Batch	ReLU
Conv	3	1	256	16	Batch	ReLU
Conv	1	1	256	16	None	None

Type	Kernel	Stride	Output ch.	Output size	Norm.	Activ.
Concat	-	-	512	16	-	-
Conv	3	1	256	16	Batch	ReLU
Conv	3	1	256	16	Batch	ReLU
Bi. upsampl.	-	-	128	32	-	-
Conv	3	1	128	32	Batch	ReLU
Conv	3	1	128	32	Batch	ReLU
Bi. upsampl.	-	-	64	64	-	-
Conv	3	1	64	64	Batch	ReLU
Conv	3	1	64	64	Batch	ReLU
Bi. upsampl.	-	-	32	128	-	-
Conv	3	1	32	128	Batch	ReLU
Conv	3	1	32	128	None	None

Figure 18. **Image decoder Ψ.** Image encoder first processes the conditioning image $\boldsymbol{x}'$ and the skeleton $\boldsymbol{y}^*$ in two separate independent branches before it concatenates them into a single stream. The design follows [23].

Type	Kernel size	Stride	Output channels	Output size	Norm.	Activation
Input $\boldsymbol{y}$	-	-	3	128	-	-
Conv	7	1	32	128	Batch	ReLU
Conv	3	1	32	128	Batch	ReLU
Conv	3	2	64	64	Batch	ReLU
Conv	3	1	64	64	Batch	ReLU
Conv	3	2	128	32	Batch	ReLU
Conv	3	1	128	32	Batch	ReLU
Conv	3	2	256	16	Batch	ReLU
Conv	3	1	256	16	Batch	ReLU
Conv	1	1	*n keypoints*	16	None	None

Figure 19. **Skeleton encoder η.** The architecture is based on the encoder from [23]. The last layer has as many output channels as the number of keypoints to predict.

Type	Kernel size	Stride	Output channels	Output size	Norm.	Activation
Input ($\bar{\boldsymbol{y}}$ or $\boldsymbol{y}$)	-	-	1	128	-	-
Conv	4	2	64	64	Instance	LReLU
Conv	4	2	128	32	Instance	LReLU
Conv	4	2	256	16	Instance	LReLU
Conv	4	1	512	15	Instance	LReLU
Conv	4	1	1	14	None	None

Figure 20. **Skeleton discriminator $D_{\mathcal{Y}}$.** The architecture follows [79]. LReLU stands for Leaky Rectified Linear Unit [39] that is used with 0.2 negative slope. Instance normalization [66] is used before every activation.

Type	Kernel	Stride	Output ch.	Output size	Norm.	Activ.
Input ($\Psi(\bar{\boldsymbol{y}}, \boldsymbol{x}')$ or $\boldsymbol{x}$)	-	-	3	128	-	-
Conv	3	1	16	128	Instance	LReLU
Conv	3	1	32	128	Instance	LReLU
Conv	3	1	32	128	Instance	LReLU
Avg. pool.	2	2	32	64	-	-
Conv	3	1	64	64	Instance	LReLU
Conv	3	1	64	64	Instance	LReLU
Avg. pool.	2	2	64	32	-	-
Conv	3	1	128	32	Instance	LReLU
Conv	1	1	128	32	Instance	LReLU
Avg. pool.	2	2	128	16	-	-
Conv	3	1	256	16	Instance	LReLU
Conv	1	1	256	16	Instance	LReLU
Avg. pool.	2	2	256	8	-	-

weight sharing

Type	Kernel	Stride	Output ch.	Output size	Norm.	Activ.
Input $\boldsymbol{x}'$	-	-	3	128	-	-
Conv	3	1	16	128	Instance	LReLU
Conv	3	1	32	128	Instance	LReLU
Conv	3	1	32	128	Instance	LReLU
Avg. pool.	2	2	32	64	-	-
Conv	3	1	64	64	Instance	LReLU
Conv	3	1	64	64	Instance	LReLU
Avg. pool.	2	2	64	32	-	-
Conv	3	1	128	32	Instance	LReLU
Conv	1	1	128	32	Instance	LReLU
Avg. pool.	2	2	128	16	-	-
Conv	3	1	256	16	Instance	LReLU
Conv	1	1	256	16	Instance	LReLU
Avg. pool.	2	2	256	8	-	-

Type	Kernel size	Stride	Output channels	Output size	Norm.	Activation
Concat	-	-	512	8	-	-
Conv	1	1	512	8	Instance	LReLU
Conv	3	1	512	8	Instance	LReLU
Conv	3	1	512	8	Instance	LReLU
Avg. pool.	2	2	512	4	-	-
Conv	3	1	512	4	Instance	LReLU
Conv	1	1	512	4	Instance	LReLU
Conv	4	1	512	1	Instance	LReLU
Conv	1	1	1	1	None	None

Figure 21. **Conditional image discriminator $D_{\mathcal{X}}$.** Conditional image discriminator starts with a Siamese architecture until the two streams are concatenated. When the version without conditioning is required, the second branch in the Siamese part is simply omitted. LReLU stands for Leaky Rectified Linear Unit [39]. We set the negative slope to 0.2. Every activation is preceded by instance normalization [66]. The architecture is loosely based on [28].

F. Architectures for experiments with MPI-INF-3DHP

In experiments that use MPI-INF-3DHP as the source of skeletons and Human3.6M as the source of unlabelled images, we employ modified architectures for some parts of the model as described below.

Type	Kernel	Stride	Output channels	Output size	Norm.	Activation
Input $\boldsymbol{x}$	-	-	3	128	-	-
Conv	7	1	32	128	Batch	ReLU
Conv	3	1	32	128	Batch	ReLU
Conv	3	2	64	64	Batch	ReLU
Conv	3	1	64	64	Batch	ReLU
Conv	3	2	128	32	Batch	ReLU
Conv	3	1	128	32	Batch	ReLU
Conv	3	2	256	16	Batch	ReLU
Conv	3	1	256	16	Batch	ReLU
Conv	1	1	256	16	None	None
Conv	3	1	256	16	Batch	ReLU
Conv	3	1	256	16	Batch	ReLU
Bilinear upsampl.	-	-	128	32	-	-
Conv	3	1	128	32	Batch	ReLU
Conv	3	1	128	32	Batch	ReLU
Bilinear upsampl.	-	-	64	64	-	-
Conv	3	1	64	64	Batch	ReLU
Conv	3	1	64	64	Batch	ReLU
Bilinear upsampl.	-	-	32	128	-	-
Conv	3	1	32	128	Batch	ReLU
Conv	3	1	32	128	Batch	ReLU
Conv	3	1	1	128	None	None

Figure 22. **Image encoder Φ.** The network is based of the encoder and decoder network from [23]. Arrows on the side denotes skip connection that are concatenated to the other input.

Type	Kernel	Stride	Output ch.	Output size	Norm.	Activ.
Input $\boldsymbol{x}'$	-	-	3	128	-	-
Conv	7	1	32	128	Batch	ReLU
Conv	3	1	32	128	Batch	ReLU
Conv	3	1	64	128	Batch	ReLU
Conv	3	1	64	128	Batch	ReLU
Conv	3	2	128	64	Batch	ReLU
Conv	3	1	128	64	Batch	ReLU
Conv	1	1	# limbs * 64	64	None	None
Avg. pool	64	1	# limbs * 64	1	None	None
Group channels	-	-	# limbs x 64	1	-	-

Type	Kernel	Stride	Output ch.	Output size	Norm.	Activ.
Input $\boldsymbol{y}^*$	-	-	# limbs	128	-	-

Mask features with detected limbs

Type	Kernel	Stride	Output ch.	Output size	Norm.	Activ.
Input	-	-	# limbs x 64	128	-	-

Type	Kernel	Stride	Output ch.	Output size	Norm.	Activ.
Max across limbs	-	-	64	128	-	-
Concat	-	-	67	128	-	-

Bilinear downsampling

Type	Kernel	Stride	Output channels	Output size	Norm.	Activation
Conv	3	1	256	32	Batch	ReLU
Conv	3	1	256	32	Batch	ReLU
Bilinear upsampl.	-	-	256	64	-	-
Conv	3	1	128	64	Batch	ReLU
Conv	3	1	128	64	Batch	ReLU
Bilinear upsampl.	-	-	128	128	-	-
Conv	3	1	64	128	Batch	ReLU
Conv	3	1	64	128	Batch	ReLU
Conv	3	1	64	128	None	None

Figure 23. **Image decoder Ψ.** Image decoder first encodes the conditioning image $\boldsymbol{x}'$ and then it uses limbs from skeleton $\boldsymbol{y}^*$ to mask features of the encoded conditioning image $\boldsymbol{x}'$. Arrows on the sides denote skip connections that are concatenated to the other input.