Self-supervised Learning of Structural Representations of Visual Objects

Tomas Jakab
New College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Trinity 2021

Acknowledgements

I want to express my deepest gratitude to my supervisor Andrea Vedaldi for his support throughout my DPhil journey. He has been a constant source of advice and guidance, and his passion for exploring challenging topics encouraged me to push through when things were not working.

I want to thank my examiners Andrew Zisserman and Ming-Hsuan Yang for their valuable feedback and insights during the discussion of my thesis.

I am grateful to my friend Ankush Gupta for our research discussions that led to my first paper, as well as for his friendship and support, especially at the beginning of my DPhil.

I would also like to thank my internship hosts at Google Research for their mentorship. I am especially grateful to Angjoo Kanazawa that made an extra effort in mentoring me.

I am also thankful to the rest of my excellent collaborators that I was lucky to work with: Hakan Bilen, Noah Snavely, Jiajun Wu, Ameesh Makadia, Richard Tucker, Shangzhe Wu, and Christian Rupprecht.

I am grateful to my family for their love, support, and that they have always believed in me. My father inspired my curiosity in computer science and my grandfather inspired me to pursue research. My mother and grandmother were a source of unlimited support throughout the whole journey.

I am also incredibly grateful to Marta for her support and patience with me. She has been always on my side and this journey would be much more difficult without her.

I want to also thank the excellent members of VGG for creating a supportive and inspiring environment. It has been a privilege to be surrounded by such fine people.

Finally, I thank the generous Clarendon Fund for their financial support.

Abstract

This thesis explores how a computer can learn the structure of visual objects in the absence of strong supervision using self-supervised learning. We demonstrate that we can learn structural representations of objects using an autoencoding framework with reconstruction as the key learning signal. We do this by engineering bottlenecks that disentangle object structure from other factors of variation. Moreover, we design the bottlenecks to represent the object structure in the form of 2D and 3D object landmarks or 3D mesh. Specifically, we develop a method that automatically discovers 2D object landmarks without any annotations using a conditional autoencoder with 2D keypoint bottleneck that disentangles pose, represented as 2D keypoints, and appearance. Despite the ability of self-supervised learning methods to learn stable object landmarks, the automatically discovered landmarks are not aligned with landmarks that would be annotated by human annotators. To address this, we present a method that can inject an unpaired empirical prior into a conditional autoencoder by introducing a novel landmark autoencoding that can leverage powerful image discriminators used in adversarial learning. A by-product of these conditional autoencoding methods is that the generation can be interactively controlled by manipulating the keypoints in the bottleneck. We leverage this feature in a novel method for interactive 3D shape deformation. The method is trained in a self-supervised way to use automatically discovered 3D landmarks to align pairs of 3D shapes. In the test time, the method allows the user to interactively deform the object shape via the discovered 3D object landmarks. Finally, we present a method that uses a photo-geometric autoencoder to recover 3D shape of an object category without any 3D annotations. It uses videos for training and learns to disentangle an image input into a rigid pose, texture and deformable shape model.

Contents

Chapter 1

Introduction

Learning structural representations of objects is a fundamental problem in computer vision. Structural representations such as object landmarks, parts, or 3D meshes are useful as intermediate representations of objects that allow agents to reason about them as they reflect the underlying physical state while being invariant to other factors like appearance or camera pose. Moreover, if the learned structural representations are human-interpretable, they can be directly used by humans in various applications. Examples include object landmarks in animal behavior studies [80], 3D meshes for artists in computer graphics. With the emergence of deep learning, there has been a great success in learning such representations in a supervised way. However, supervised learning requires costly manual annotations that make applications on novel data expensive. This work studies how a machine can learn structural representations of objects, specifically 2D and 3D object landmarks and 3D meshes, while relaxing the requirement for expensive annotations.

Specifically, we leverage deep neural networks following their universal successes in computer vision. A deep neural network is essentially a composition of parametric functions that works as a function approximator trained by minimizing a loss function on its outputs by gradient descent optimization. Training a neural network usually requires training data consisting of input and target pairs. The target is the desired output of the neural network given the input. For example, the input can be an image of a person and the target are 2D coordinates of landmarks that we are interested in, such as the coordinates of the head, left and right hand and so on. The network then takes the input and produces an output that is compared to the target from the training data. This comparison is done by a loss function that measures how far is the output from the target. The network is then optimized by gradient descent that adjusts the network parameters in such a way that minimizes the loss, which brings the output closer to the target. This procedure is called supervised learning

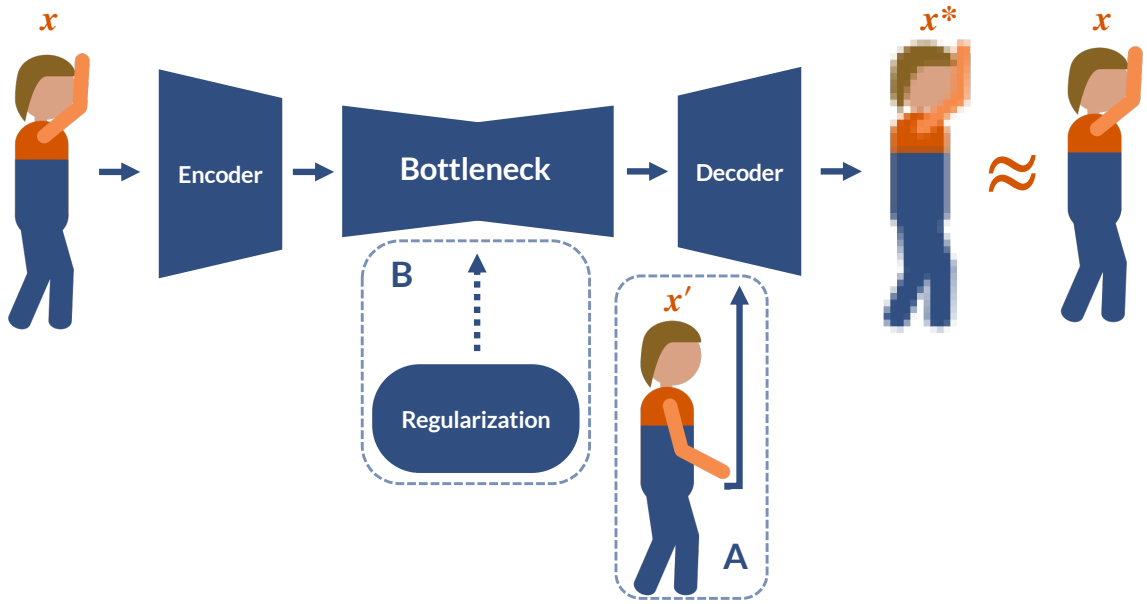


Figure 1.1: **Autoencoding framework.** The encoder takes in input $\boldsymbol{x}$ and produces an encoding that is passed through the bottleneck. The decoder then produces a reconstruction $\boldsymbol{x}^*$ the original input $\boldsymbol{x}$. (A) We also show a conditional autoencoding version. If the bottleneck removes all the information except for the structure, such as 2D keypoints, we provide a supplemental input $\boldsymbol{x}'$ for the decoder. The supplemental input, usually referred to as the source, differs in the property that we want the bottleneck to capture, in this case pose, but remains the same in the rest of the properties. This has two benefits: the bottleneck does not have to encode for the rest of the properties, in this case appearance, but it has to efficiently encode for the property that is unique for the target. If the bottleneck consists of multiple representations that capture all factors of variations, conditioning is not necessary as done in our photometric autoencoder, fig. 1.7 and chapter 6. (B) We can further impose regularization on the bottleneck as we do in chapters 4 to 6 to steer the learned representation.

and usually requires a large number of training pairs in order to be able to generalize on unseen data. These training pairs are in most of the cases produced manually by human annotators labelling input examples, such as images of humans annotated with landmarks. This is a very time consuming and costly process as it is usually required to annotate thousands of images. For example, annotating a single image in the popular COCO dataset [74] took 19 minutes, which is 39k human hours in total for the 123k images. Obtaining 3D ground-truth for shape reconstruction of objects in the wild also poses many challenges. One approach is to use Structure-from-Motion (SfM) [18, 37, 97] that can reconstruct 3D scene from videos. This requires the data-collectors to obtain detailed videos of objects capturing them from all sides. That alone does not ensure successful 3D reconstruction. Reizenstein et al. [97] used this technique to create a dataset of common objects and they note that out of 22.6k collected video sequences, only 5.6k resulted in accurate 3D reconstructions. However, this approach is not applicable to non-rigid objects such as people or animals. This limitation led authors of [73] to use videos where people were asked to stay frozen in various poses while the camera moves through the scene. While this could be a viable approach to collect human data, we can hardly ask animals to stay still in their poses.

Despite the existence of many large scale annotated datasets, they will be always limited to a fraction of object categories that exist in the world. This means that in practice it is still often required to collect annotations for particular applications. For example, animal researchers interested in detecting and tracking animal landmarks to study their behavior need to first manually annotate a large number of images and repeat this whenever the animal appearance, environment or lighting condition changes. This waste their valuable time that they could have otherwise spent on actual research [80].

These limitations of supervised learning lead to the question whether we can train a neural network to produce desired outputs in the absence of annotated training examples. One approach to do this is to design a proxy learning task for which we can use some part of inputs or transformed inputs as the targets. This is called self-supervised learning [1]. A simple version of a proxy task is where we use the original input as the target and the neural network is tasked to reconstruct it. While this could be a trivial task as the neural network could just simply copy the input to the output, the key is to impose an information bottleneck inside the network. This leads to an autoencoder neural network that consists of an encoder that transforms

[1]This can be also often called unsupervised learning in the sense that it does not use annotated data.

the input into a lower-dimensional code serving as the bottleneck and a decoder that transforms this code back to be as close as possible to the original input. Since the bottleneck should not have enough capacity to copy the input, the network needs to learn an encoding that would be the most efficient in capturing the input information. The neural networks can be designed to have high capacity in the encoder and the decoder which means the bias of the dataset can be stored in the weights of the network. This allows the code in the bottleneck to only encode the differences in the data. For example, if the data consist of images of a single person in different poses, the code would only need to encode the different poses but not the appearance of the person. This way the learned code can become discriminative of the person's pose. With more elaborative bottleneck designs and training strategies, we can learn more complex representations. Moreover, as we demonstrate in this thesis, the bottleneck can be designed to directly represent structural representations such as 2D/3D object landmarks or 3D meshes.

1.1 Key ideas and motivation

In this section, we discuss the main ideas and motivation behind the methods introduced in this thesis. The unifying principle is the use of an autoencoding framework for learning structural representations of objects as illustrated in fig. 1.1. We extend this autoencoding framework by engineering bottlenecks that disentangle the object structure from other factors of variation, such as the pose from the appearance. Moreover, we design the bottlenecks and regularizers to enforce the learned structural representation the well-established form of 2D and 3D object landmarks or 3D shape represented as a mesh that are easily interpretable and can be readily used for other applications. Examples of such a engineered bottlenecks are shown in figs. 1.2, 1.5 and 1.7. Overall, we present four methods that deal with 2D object landmark estimation, 3D object landmarks for shape deformation and 3D shape reconstruction:

1. A conditional autoencoder with 2D keypoint bottleneck, fig. 1.2, that disentangles pose, represented by 2D object landmarks, and appearance without any further regularization.

2. A conditional autoencoder with a dual 2D keypoint representation that enables the use of an image discriminator trained on an unpaired landmark prior that

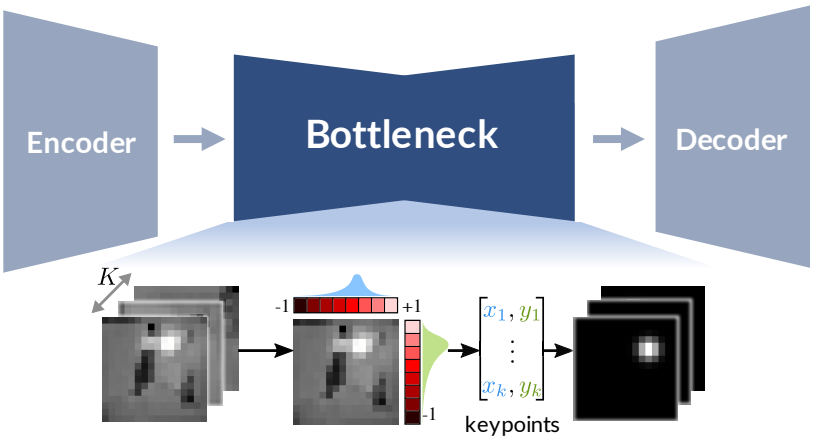


Figure 1.2: **2D keypoint bottleneck.** A differentiable bottleneck engineered to extract 2D keypoints introduced in chapter 3.

steers the self-supervised landmark learning to learn landmarks that are human-interpretable.

3. A conditional autoencoder that extracts 3D object landmarks from 3D shapes and learns to use them for interactive shape deformation. The learned object landmarks are regularized by a novel keypoint regularizer based on the farthest point sampling algorithm.

4. An autoencoder with a photo-geometric bottleneck that disentangles rigid pose, deformable shape and texture, fig. 1.7. We leverage temporal consistency presented in videos during training for regularization.

We now discuss the key ideas and motivations behind each of the individual methods. The methods are then presented in their full detail in chapters 3 to 6.

1.1.1 2D object landmarks

As discussed above, structural representations in the form of object landmarks are useful for many applications but obtaining annotation for supervised learning is expensive. Hence we wish to learn 2D object landmarks in a self-supervised way. In order to achieve that, we extend the autoencoding framework by engineering a 2D keypoint bottleneck fig. 1.2 that allows only the information about the 2D keypoints coordinates to be passed through. The decoder reconstructing the original image from the supplied 2D keypoints was inspired by [120] that demonstrate the use of keypoints for conditional image generation. Their conditional image generator takes

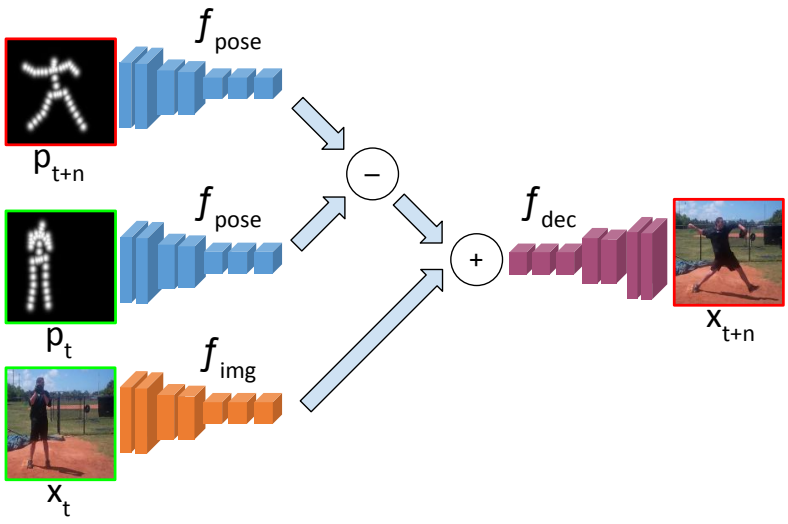
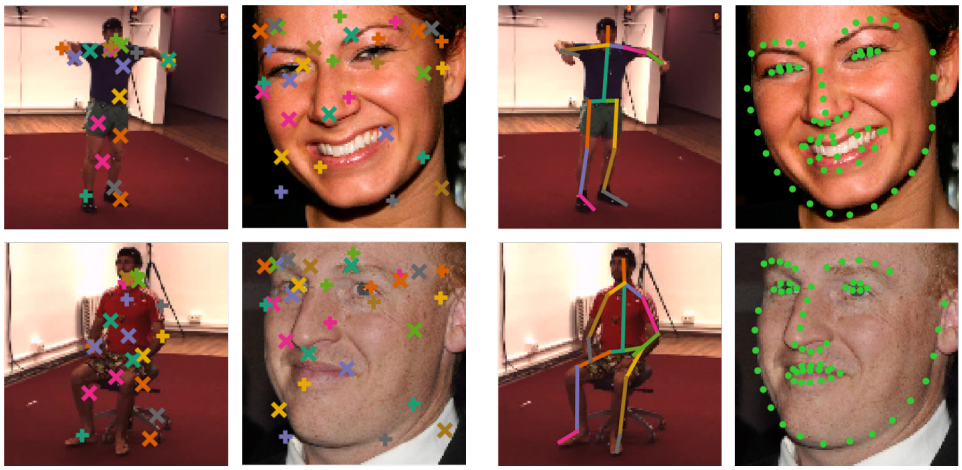


Figure 1.3: **Conditional image generator.** Conditional image generator proposed by [120]. The generator is conditioned on an image x_t and an alternative pose p_{t+n} supplied as an image and generates a new image with the pose from p_{t+n} and the appearance from the x_t. Our conditional generator proposed in chapter 3 was inspired by this approach, but we do not use p_t and most importantly, the pose is in the form of unsupervised landmarks that our method automatically discovers.

the keypoints in the form of a keypoint image and another image of an object and generates a novel image where the object follows the pose as dictated by the keypoints fig. 1.3. Inspired by this, our generator also takes keypoints in the form of a keypoint image and is conditioned on the appearance image that is supplied from another frame. This simplifies the job for the generator as it has to only perform image-to-image translation and translate the 2D keypoint image into a novel image conditioned on the appearance image. Convolutional neural networks are indeed well-suited for image-to-image translation as demonstrated in [50]. The conditioning on the appearance image is crucial as the keypoints should not carry the information about the appearance. At the same time, the appearance image should not contain the information about the object pose which should be only encoded by the keypoints. For this purpose, we sample the appearance image from another frame when training from video sequences or we synthetically warp the input image when learning from still images. This approach is efficient in discovering stable object landmarks and does not require any other regularization as the concurrent work [136] that uses a similar bottleneck but no conditioning. Their method has to specifically enforce the keypoints to be spread in order to prevent them from collapsing. Our method also does not require access to known correspondences between the training images as



discovered landmarks human-interpretable landmarks

Figure 1.4: **Discovered and human-interpretable landmarks.** Self-supervised landmark detectors [112, 136, 51, 77] discover landmarks **[left]** that stable but not human-interpretable (predictions from [51]).

needed by [112, 111, 110, 136]. Other works using image generation to learn about pose [125, 103] learn a latent code that is discriminative of the object pose but they cannot readily obtain object landmarks from this code. Our method is specifically designed to directly output object landmarks in the 2D space of the image.

1.1.2 Human-interpretable 2D object landmarks

While our method for self-supervised discovery of object landmarks introduced above learns object landmarks that are invariant to the object identity, view and lighting, the landmarks are far from what human annotators would label as illustrated in fig. 1.4. This is also common for other self-supervised learning methods [112, 111, 110, 136, 77]. While this is not a problem for applications where the landmarks are used by other learning algorithms as their intermediate representation for object structure [104, 65, 82, 128], other applications might require the learned landmarks to follow some established labeling style that is usually dictated by humans. We might be for example interested in landmarks located at the position of eyes for faces or joint locations for human or animal bodies. Our idea is to guide the self-supervised landmark discovery with unpaired landmark prior that is cheaper to obtain than large-scale annotated datasets. For example, the unpaired prior can be automatically collected in a laboratory setting as done with motion capture. Inspired by the successes of unpaired image-to-image translation [141] using adversarial learning, we would like to leverage

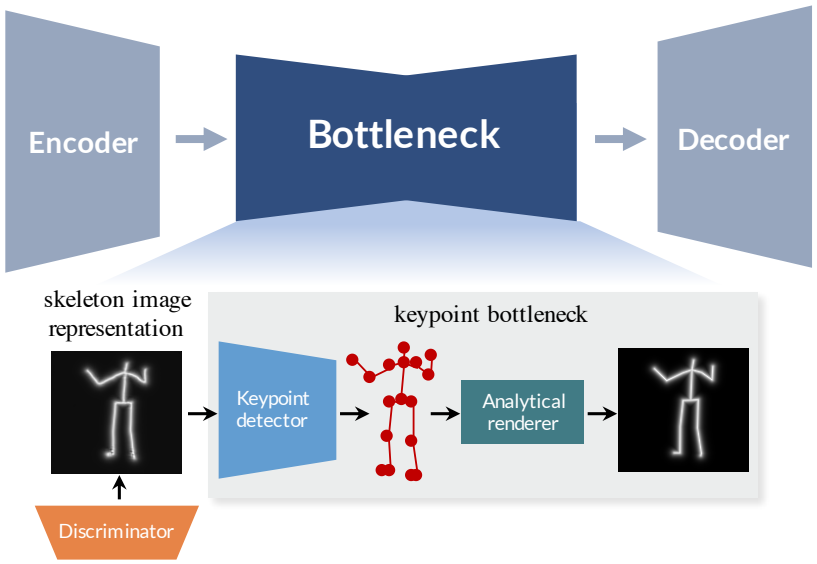


Figure 1.5: **Dual representation 2D keypoint bottleneck.** A differentiable bottleneck with dual keypoint representation that allows for the use of an image discriminator to steer 2D object landmark discovery. This bottleneck is used by the method introduced in chapter 4.

powerful image discriminators to impose the prior. For this purpose, we propose a novel keypoint autoencoding with a dual representation of landmarks:

1. An image-based keypoint representation produced by neural network encoder that can be directly regularized by an image discriminator trained on an unpaired landmark prior.

2. An analytical keypoint representation based on 2D keypoint coordinates that serve as the bottleneck.

To make this end-to-end differentiable, we translate from the image-based representation to the keypoint-based representation by a keypoint detector that is a neural network pre-trained on analytically rendered keypoints or skeletons as illustrated in fig. 1.5. This approach learns to discover correctly aligned landmarks without any further regularization.

An interesting property of this form of conditional generation shared with the previous work is that it can be used to control image generation by manipulating the landmarks in the bottleneck as illustrated in fig. 1.6.

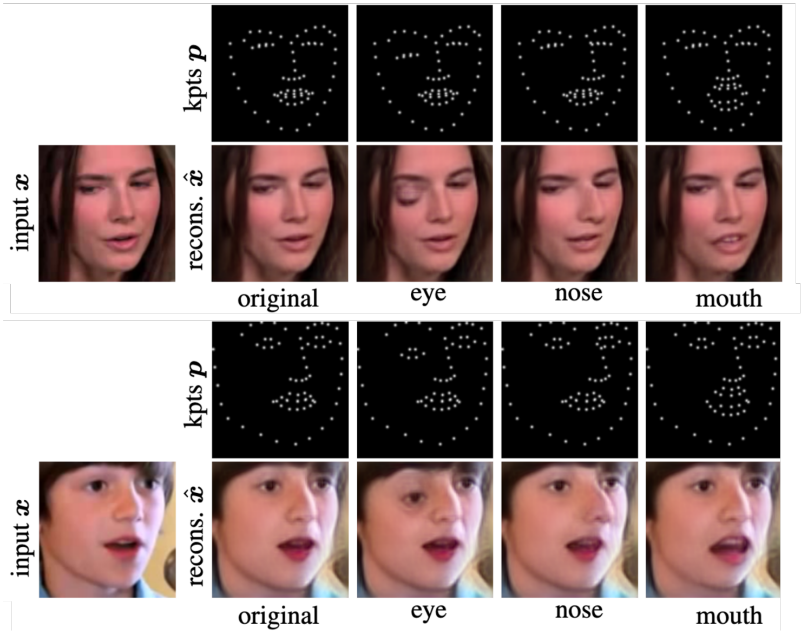


Figure 1.6: **Conditional image editing using detected landmarks.** As a by-product of the conditional autoencoding used by methods from chapters 3 and 4, we can control the generated image by manipulating the coordinates of detected keypoints (*kpts*). Figure comes from [52].

1.1.3 3D landmarks for interactive shape deformation

Production of virtual environments for computer games, virtual or augmented reality comes with a demand for a large number of 3D models. Editing existing 3D models for specific applications could be an effective way to meet this demand. However, editing 3D models still requires expert knowledge and can be time-consuming. In order to make this task more accessible and efficient, we propose a machine-assisted 3D content creation method for deforming 3D shapes that is simple and intuitive. Our method uses 3D keypoints as semantically meaningful control points for object deformation. Since supervision for keypoints and deformation would be difficult to obtain, we wish to learn the keypoints and deformation in a self-supervised way. For this purpose, we adapt the conditional autoencoding framework and design a proxy learning task where the goal is to align a source shape with a target shape by deforming the source shape. The method uses a 3D keypoint bottleneck that extracts 3D keypoints from the source and target shapes. The extracted 3D keypoint representation of the shapes is then used to infer the deformation that would align the source shape into the target shape. The deformation is modeled using cages-based deformation [132] that preserves the details of the shape. During the test time, thanks

to conditional autoencoding, the deformation of the source shape can be controlled by the target keypoints without the need for an exemplar target shape. The user can directly manipulate the target keypoints in the bottleneck to guide the deformation of the source shape. The motivation for this design is inspired by our previously introduced methods for 2D landmarks where the conditional editing ability is learned as a by-product as shown in fig. 1.6, but here this is the primary goal of this method.

Since it is difficult to obtain ground-truth training pairs that would be a direct deformation of each other, our method has to be able to work with distinct pairs of instances from the same object category that are present in common datasets such as ShapeNet [7]. As one-to-one correspondences are not available in this case, we use the Chamfer distance between the deformed source shape and the target shape as the reconstruction loss. We also found that we need to use further regularization to learn useful landmarks as compared to the 2D landmarks where the task itself induces a strong learning signal. We design a farthest point sampling regularizer that encourages keypoints to be evenly distributed and focus on significant locations. The object landmarks learned by the method are semantically consistent across largely different instances of an object category.

1.1.4 Deformable objects from videos

A 3D mesh is a powerful structural representation of objects and if predicted in a canonical frame, it is invariant to viewpoint changes. Not only it fully captures the underlying 3D shape of the object but it can be also used to implicitly obtain dense correspondences between different viewpoints. Moreover, if different instances of the object category share the same mesh structure, such as through a prior/canonical mesh, one can also establish dense correspondences between different instances. Dense correspondences can be considered as a superset of object landmarks or object parts. If object landmarks or parts predictions are still required, they can be easily obtained by annotating a single mesh with keypoints or object parts. In this sense, a 3D mesh can be considered as a more powerful structural representation than landmarks or object parts.

However, reconstructing a 3D mesh of an object from a single image is an extremely ill-posed task. With learning-based methods, this task can become tractable as they can capture shape priors for an object-category that help them to reconstruct a 3D shape from a single image [13, 26, 121, 32, 61, 27, 58]. Many of these approaches have the disadvantage that they require access to ground-truth 3D shapes during training [13, 26, 121, 32, 61]. Obtaining 3D ground-truth data for objects

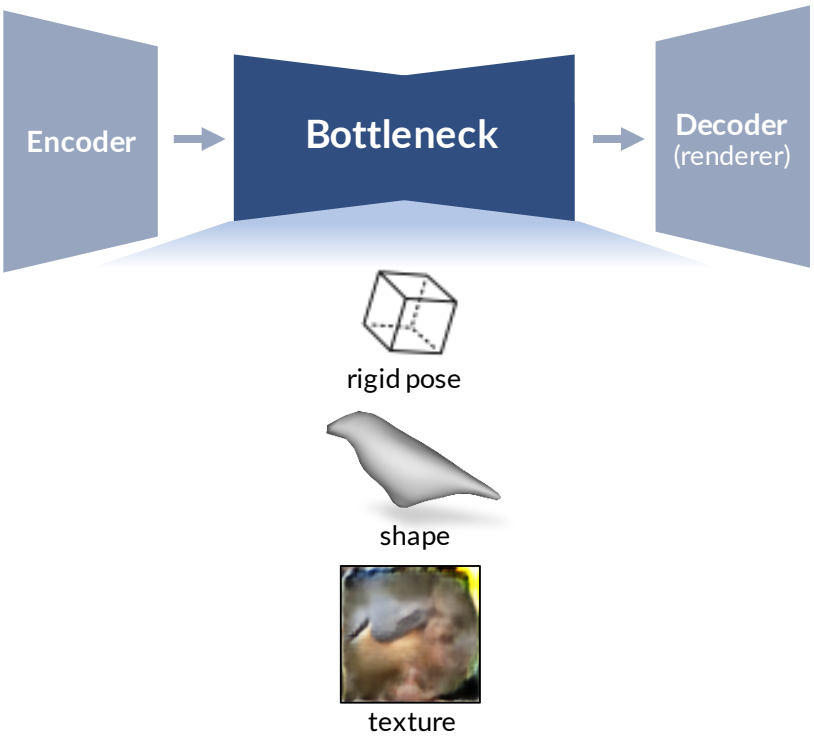


Figure 1.7: **Photo-geometric bottleneck.** Can be thought of as multiple bottlenecks each encoding for a different factor of variation: rigid pose, mesh, texture that are then fed into a differentiable mesh renderer serving as the decoder reconstructing the input.

in the wild is very challenging and even more when the objects are deformable like animals or humans. Using videos of objects has the potential to sufficiently constrain the problem in the absence of ground-truth 3D data. Videos can provide different views of the object when the object moves or by moving the camera around the object. Structure-from-Motion methods [18, 37] then establish correspondences across the different views and recover the 3D shape of the object provided that it is rigid. Learning-based methods [88, 73, 41] use a similar principle and learn to reconstruct 3D shapes from a single or a few images but they are also limited to only rigid objects. However, many objects such as animals, humans, machines are not rigid but deformable. Using videos of deformable objects for supervision is challenging as it is difficult to disambiguate what changes in the observed images have arisen from the rigid motion of the object/camera and the deformation of the object when establishing correspondences.

Our approach addresses these challenges by jointly learning to recover the rigid pose, object shape and the per-frame deformation of the shape and by exploiting the natural temporal consistency of videos. Specifically, we base our method on the autoencoding framework with a bottleneck that factorizes the encoding of the image input into a rigid pose, 3D shape, a per-frame deformation, and a texture as illustrated in fig. 1.7. The decoder is in the form of an analytical mesh renderer [61, 75, 96] that reconstructs the original image from these components. We also leverage the temporal consistency of videos to guide the learning in two ways. First, as we can assume that the identity and appearance of the object do not change in the video, we can enforce constant texture and per-sequence shape while the deformation is being factored out by a deformation model. Second, learning from videos allows us to use a generic optical flow estimator to obtain local short-term correspondences that can be used as a part of the supervision. As the video sequences are not guaranteed to observe the object from all sides during learning, we need to impose further regularization on the shape. We do that by incorporating a learned per-category shape prior that is then deformed into individual per-sequence object instances. This also brings another benefit. As all the predicted shapes are related through the shape prior, we can not only establish correspondences between different views or deformations of the same instance but also between different instances of an object category. All this regularization, enabled by the use of videos, allows us to use less supervision than other image-based weakly-supervised methods that require ground-truth 2D keypoints [58, 72], 3D viewpoints [58, 72] or initial prior shapes [29] while our shape reconstructions are also more consistent and accurate across novel viewpoints. UMR

method presented in [71] also uses less category-specific supervision than the above methods, but it still requires a weak-supervision for correspondences obtained from a network pre-trained on ImageNet dataset [63, 106, 38, 45].

1.2 Publications

The work in this thesis was presented in the following publications:

- Chapter 3: Unsupervised Learning of Object Landmarks through Conditional Image Generation. Tomas Jakab*, Ankush Gupta*, Hakan Bilen, Andrea Vedaldi. *Proceedings of 32nd Conference on Neural Information Processing Systems (NeurIPS), 2018.*

- Chapter 4: Self-supervised Learning of Interpretable Keypoints from Unlabelled Videos. Tomas Jakab, Ankush Gupta, Hakan Bilen, Andrea Vedaldi. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.*

- Chapter 5: KeypointDeformer: Unsupervised 3D Keypoint Discovery for Shape Control. Tomas Jakab, Richard Tucker, Ameesh Makadia, Jiajun Wu, Noah Snavely, Angjoo Kanazawa. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021.*

- Chapter 6: DOVE: Learning Deformable 3D Objects by Watching Videos. Shangzhe Wu*, Tomas Jakab*, Christian Rupprecht, Andrea Vedaldi. *Preprint arXiv:2107.10844, 2021.*

* denotes equal contribution.

Chapter 2

Literature Review

Despite the success of supervised learning, the key limitation remains: it requires large-scale annotated datasets. Self-supervised learning methods can on the other hand learn for unannotated data that are much easier to obtain. Self-supervised learning has been mainly focused on general feature learning, also called representational learning. The learned features can then be used to train a supervised model for the desired down-stream task with a much smaller amount of annotations. Early examples include training autoencoders [42, 67]. Later, more complex proxy tasks were proposed leading to better performance. This includes works that propose to extract patches from images and predict the spatial relationships between them [15, 87]. In a similar spirit but in the temporal domain, [83, 21] use videos and task their network to predict the temporal relationships between sampled frames. Larsson et al. [66] suggest using features obtained from a network trained on image colorization [135]. Gidaris et al. [25] rotate an image and train their network to predict the rotation that was applied. DeepCluster [5] use feature clustering and iteratively alternates between clustering learned features into groups and training a neural network to predict the cluster assignments. Recently introduced contrastive learning [90] learns features by predicting the future in the feature space in temporal or spatial domain and leveraging contrastive losses. Rather than learning general image representations, this thesis deals with self-supervised training of neural networks that can directly predict structural representations of objects such as 2D/3D object landmarks or 3D meshes. Specifically, we review self-supervised methods for object landmarks, shape deformation and category mesh reconstruction.

2.1 Object landmarks

Learning object landmarks was predominantly studied using supervised learning that relies on annotated datasets for training such as MS COCO Keypoints [74], Human3.6M [49], MPII [1] and LSP [54]. Prior to the deep learning era, a common way was to use pictorial structures [19] to model the object poses. With the advent of deep convolutional neural networks, it became feasible to directly predict object landmarks from the input image. Initially this was done by directly regressing the keypoint coordinates [114] but methods predicting keypoint heatmaps proved to be the most successful [124, 84, 2, 93, 6, 3, 113, 48, 4, 80].

Since deep neural networks are notorious for requiring a large amount of training data, there has been growing interest in leveraging self-supervised learning for pose estimation.

2.1.1 Pose embeddings

Initially, a line of self-supervised learning methods for pose learned features that are discriminative of object pose, but they could not directly predict 2D/3D object landmarks. Kanazawa et al. [57] and Rocco et al. [98] align pairs of images containing objects from the same category by predicting the deformation field to warp the source image. Their methods are supervised by known correspondences from synthetically warped images. Shu et al. [103] and Wiles et al. [125] propose using autoencoders that factor out an appearance and deformation modeled as a deformation field. Shu et al. [103] train an autoencoder that disentangles appearance represented as texture in a canonical coordinate frame and a deformation field which warps the texture to reconstruct the input. The appearance information has to be passed through a tight bottleneck to prevent trivial solutions which lower the quality of reconstruction. In contrast, Wiles et al. [125] use videos and train a conditional autoencoder that obtains the appearance information from a second frame. Conditioning on a second frame prevents the network from learning trivial solutions as in [103] and leads to better performance.

2.1.2 2D object landmarks

Other line of works focuses on directly predicting object landmarks. Works by Thewlis et al. [112, 111, 110] learn sparse and dense landmarks and are supervised by known transform between training image pairs using equivariance. The transformation is obtained by synthetically warping images as in [57, 98] or from optical flow. Zhang

et al. [136] proposed an autoencoder to discover landmarks that serve as an explicit structural representation inside an autoencoder where they are used to pool features extracted from the input image. The input image is reconstructed from these landmark-pooled features. However, this is insufficient to learn geometry and requires them to also use the principle of equivariance [112] and other regularizers such as distinctiveness that require balancing to achieve optimal results. Our method [51] introduced in chapter 3 uses a conditional autoencoder that extracts a small amount of information from a given target video frame via a tight bottleneck which retains pose information while discarding appearance. The appearance information is obtained from a second source frame. The method does not require any further regularization such as in the form of equivariance or distinctiveness as in the previous works while producing more accurate landmarks. Mallis et al. [79] propose a self-training approach that transforms generic landmarks into object landmarks. The work shows better performance across large viewpoint changes that is perhaps due to the strong initialization from generic keypoints that were obtained by Superpoint method [14] trained on a large-scale synthetic dataset. Our other work [52], chapter 4, notice that landmarks discovered by self-supervised methods are not semantically aligned with human annotations and propose to use an unpaired prior to guide the automatic landmark discovery.

2.1.3 3D object landmarks

Self-supervised 3D object landmarks are less explored than 2D landmarks. Most of the works focus on lifting 2D annotations to 3D space [117, 140, 130, 56, 99, 89]. Many of these works utilize adversarial 3D landmark prior obtained from mocap data [117, 130, 56]. Suwajanakorn et al. [109] learn 3D object landmarks for objects from image pairs differing by a known 3D rigid transformation. Self-supervised learning of 3D keypoints for 3D shapes represented as meshes or point clouds has been mostly studied from the interest point detection perspective rather than learning semantic 3D object landmarks [11, 20, 53]. This includes handcrafted methods [16, 10, 138, 134] and only recently deep learning [69] methods. Chen et al. [11] propose a point cloud autoencoder that outputs a structured 3D representation to obtain sparse or dense shape correspondences for object category. Fernandez et al. [20] learn to predict 3D landmarks using symmetric linear shape basis and a set of desirable keypoint properties as a regularization. Our work [53] introduced in chapter 5 also learns 3D landmarks by training an encoder that outputs an order set of keypoints regularized by a farthest point sampling keypoint regularizer.

2.1.4 Object parts

Works on self-supervised learning of object parts are also related as they often build on similar principles as methods for object landmarks. Lorenz et al. [77] propose a similar autoencoder as [136] but instead of using isotropic gaussian point representation inside the bottleneck that is used to transport the image features they use anisotropic gaussian that is better suited for object parts but the detected shapes are limited to be mostly elliptical. Hung et al. [45] propose learning object part segmentation utilizing multiple geometric, equivariance and semantic consistency constraints. The semantic consistency utilizes image features obtained from a network pre-trained on ImageNet [63, 106, 38]. Liu et al. [76] propose a conditional autoencoder that transfers appearance features from a source image to a target image via predicted object part segmentation.

2.1.5 Adversarial learning

Also related is adversarial learning as we utilize this to impose empirical pose prior on our self-supervised learning of landmarks chapter 4. Adversarial learning was proposed as a generative model [31] trained on a set of unlabeled data, for example images. The model consists of two networks, generator and discriminator. Generator transforms an input, usually noise, into an output that is in the domain of the training data. The discriminator is trained to verify whether a sample belongs to the training data or is generated by the generator while the generator is trained to fool the discriminator. This leads the generator to produce outputs that follow the distribution of the training data. When trained on large image datasets, this technique can generate realistically looking novel images that are not contained in the training data [59]. Apart from image generation, adversarial learning was later applied to image labelling [23, 43, 118, 119, 34] and unpaired image-to-image translation [141]. Unpaired image-to-image translation uses two datasets of images, for example, aerial photos and image patches of maps, and learns a neural network that can translate between them. This is done by training a generator that takes an image from the first dataset as the input and produces an output that is enforced to follow a distribution of another dataset by the discriminator. Zhu et al. [141] demonstrate that image-to-image translation can be trained to preserve the structure in the images when translating between different domains, for example, roads in aerial photos are preserved when translated into the map domain.

2.2 Shape deformation

Here we review two related areas of shape deformation in computer graphics: interactive shape deformation and shape alignment. In interactive shape deformation, a user defines cues or constraints of how a shape (usually mesh) should be deformed, and the method has to deform the mesh as realistically as possible while respecting the user-defined constraints. In shape alignment, the deformation of a source shape is guided by a target shape and the goal is to align the source shape with the target shape while preserving the details and realism of the deformed shape.

2.2.1 Interactive deformation

With optimization-based methods, the user manipulates a sparse set of vertices and the method has to adjust the rest of the vertices in such a way that the resulting deformation is as realistic as possible. This is usually achieved by preserving local Laplacian properties of the mesh. Examples include Laplacian-based shape editing [107] and As-Rigid-As-Possible shape deformation [108]. These methods are typically used for organic shapes or deformable objects.

In another line of works, the user does not manipulate the vertices of the object mesh directly. With cages-based methods, the user interacts with vertices of a coarse mesh structure, called cage, enclosing the original object mesh [55]. The cage and mesh vertices are tied together through a linear mapping and manipulating the cage vertices results in an interpolated deformation of the enclosed mesh. In skeleton-based deformation [78, 68], the mesh is rigged to a skeleton consisting of bones. The user then transforms a bone which transforms the vertices that are associated with it. To increase the realism of deformation, the mesh vertices are usually soft-associated with multiple bones and the deformations are interpolated. Both techniques are often used for character animation.

Thanks to the advancements in machine learning many learning-based approaches were proposed. Yumer et al. [133] use semantic attribute annotations for shapes and learn a mapping from semantic attributes to the shape geometry. In a test time, sliders are used to control attributes of the shape. Deep neural networks have recently led to self-supervised methods using generative models. Many of them learn a generative model of shape primitives together with the mapping to the high-fidelity shape. The shape then primitives serve as an abstraction of the shape that the user can interactively manipulate. Tulsiani et al. [116] parse shape into a set of cuboids. This was extended by Paschalidou et al. [92] that use superquadrics instead

of cuboids. Gadelha et al. [22] learn more granular shape handles by utilizing external decomposition annotation. Hao et al. [36] learn a dual representation of a shape — sphere-based shape primitives and a high-fidelity shape. The mapping between the shape primitives and the high-fidelity shape is achieved by learning a generative model for each but with shared latent space. This results in tight coupling between representations that allows the user to interact with the high-fidelity shape through the sphere-based shape primitives. Our method [53] uses automatically discovered semantic 3D object landmarks to control the shape deformation model that is trained in a self-supervised way on a pair-wise shape alignment task.

2.2.2 Shape alignment

Shape alignment has been a long-studied problem. Traditional techniques include Iterative Closest Point (ICP) [100], which alternates between point correspondence estimation and rigid deformation. Non-rigid ICP [44] extends this to non-rigid deformations between source and target shapes. The success of deep neural networks has led to the increased popularity of learning based-methods based on a conditional autoencoder that does not require correspondences and is trained with reconstruction loss that is usually in the form of Chamfer distance as the point-to-point correspondences are not available. Hanocka et al. [35] propose a conditional autoencoder that takes source and target objects and predicts a freeform deformation that warps the source shape to match the target shape. Wang et al. [122] also use a conditional autoencoder but their method directly predicts the per-vertex offsets. The predicted deformation is further regularized by feature-preserving losses such as mesh laplacian loss. Groueix et al. [33] also perform per-vertex deformation but use cycle-consistency loss together with reconstruction loss which leads to smoother deformations. Yifan et al. [132] leverage detail-preserving cage-based deformation and instead of predicting the per-vertex offsets of the mesh, they predict per-vertex offsets for the cage vertices.

2.3 3D shape reconstruction

Recovering 3D shapes of objects from images has been traditionally done using multiple-view geometry. Traditional methods like Structure-from-Motion (SfM) [18, 37] require multiple views of the object across which they estimate correspondences in order to infer the camera poses and recover the underlying 3D shape. In this thesis, we are interested in recovering 3D shape of an object from a single image which is an extremely ill-posed problem. With learning-based approaches that use

deep neural networks, methods can capture shape priors during training which can supplement the information missing in the single image during reconstruction. Many methods that were initially proposed require 3D ground-truth shape annotations for training [13, 26, 121, 32, 61]. Obtaining good 3D ground-truth is a challenging task. One way to do this is to create a synthetic dataset by rendering 3D models. However, due to the domain gap, models trained on synthetic data do not perform well on in-the-wild data. Collecting 3D ground-truth for in-the-wild objects can be done by using Structure-from-Motion pipeline but that requires manually capturing every object on a camera from all sides which is a laborious process [97] and works only for rigid objects. This inspired the development of self-supervised and weakly supervised methods that do not need 3D shape ground-truth annotations. They are mostly based on an autoencoding framework using reconstruction loss as the main source of their supervision.

2.3.1 Depth

Initially, self-supervised methods focused primarily on depth recovery. Many of these works exploit multi-view consistency as the source of the supervisory signal. The works of Garg et al. [24] and Godard et al. [28] use conditional autoencoder that is trained on binocular stereo image pairs. The autoencoder encodes the first image as a single-view disparity (inverse depth map) that is used to warp the second image to match the first image. SfMLearner [139] is based on a similar principle but instead of using binocular images, it uses monocular video footage which means that it has to also learn to estimate the ego-motion between the frames. Unsup3D [127] learns to estimate the 3D shape of an object category by learning an autoencoder that decomposes the input image depth, albedo, viewpoint and lighting. They do not need multiple views in the supervision, instead they exploit the object symmetry for supervision. Their method works only with limited viewpoint variation and does not recover the full shape.

2.3.2 Point clouds and voxels

Insafutdinov et al. [47] propose a conditional autoencoder that encodes source and target images containing different views of an object. The source image is encoded as a point cloud and the target image as a camera pose. Using the source point cloud and the target camera pose, the point cloud is rendered in a differentiable way on the image plane. The autoencoder is supervised by reconstruction loss between the point

cloud rendering and the target silhouette. The rendering does not work with colours and thus requires silhouettes for supervision. A similar approach is used by [115], but instead of point clouds, it predicts voxel occupancy. It also does not work with colours and the rendering only produces silhouettes.

2.3.3 Meshes

More recently, a group of autoencoding methods, including ours [126], use 3D meshes as their shape representation [61, 58, 75, 12, 60, 121, 91, 39, 30, 71, 72, 137]. This was enabled by the invention of differentiable mesh renderers [61, 75, 12]. Many of these works do not use multiple views for supervision, instead, they learn from still images of object category. Since this makes the problem more challenging, they require additional sources of supervision such as 2D keypoints or shape priors. CMR [58] and VMR [70] require 2D keypoints to construct initial shape and estimate viewpoints using Structure-from-Motion. U-CMR [30] improves upon CMR by removing the 2D keypoints requirement, but the method needs a category template shape for initialization. UMR [71] uses the least amount of supervision out of these methods as it replaces 2D keypoints with weakly-supervised part segmentation's from SCOPS [46] to obtain correspondences for supervision. Our work [126] learns from videos and apart from optical flow obtained from a generic optical flow estimator, it does not require any of the supervision mentioned above. All these works, including ours, still require object segmentation masks.

2.3.4 Adversarial learning and optimization

Methods using adversarial training [64, 9, 40, 85, 86, 131, 102, 137] use a discriminator trained on an image collection of object category. During training, they sample novel views of the object or a scene and render them. Since they do not have access to multiple views of the object, they use a discriminator that encourages the sampled view to follow the distribution of training images that are expected to contain multiple views of object category but not necessary of the same object instance. The reconstructions tend to be coarse and produce inconsistent shapes and textures across different views.

Most of the works using multiple views during inference focus on optimization for a single instance or a scene reconstruction. Recently introduced Neural Radiance Fields (NeRF) [81] synthesize novel views by optimizing a volumetric scene function from densely sampled views of a single scene. D-NeRF [95] extends NeRF to optimize

representations of dynamic scenes from multi-view videos. LASR [129] optimizes a deformable 3D mesh on an individual video sequence of an object.

Chapter 3

Unsupervised Learning of Object Landmarks through Conditional Image Generation

Unsupervised Learning of Object Landmarks through Conditional Image Generation

Tomas Jakab[1*] **Ankush Gupta**[1*] **Hakan Bilen**[2] **Andrea Vedaldi**[1]

[1] Visual Geometry Group
University of Oxford
{tomj,ankush,vedaldi}@robots.ox.ac.uk

[2] School of Informatics
University of Edinburgh
hbilen@ed.ac.uk

Abstract

We propose a method for learning landmark detectors for visual objects (such as the eyes and the nose in a face) without any manual supervision. We cast this as the problem of generating images that combine the appearance of the object as seen in a first example image with the geometry of the object as seen in a second example image, where the two examples differ by a viewpoint change and/or an object deformation. In order to factorize appearance and geometry, we introduce a tight bottleneck in the geometry-extraction process that selects and distils geometry-related features. Compared to standard image generation problems, which often use generative adversarial networks, our generation task is conditioned on both appearance and geometry and thus is significantly less ambiguous, to the point that adopting a simple perceptual loss formulation is sufficient. We demonstrate that our approach can learn object landmarks from synthetic image deformations or videos, all without manual supervision, while outperforming state-of-the-art unsupervised landmark detectors. We further show that our method is applicable to a large variety of datasets — faces, people, 3D objects, and digits — without any modifications.

1 Introduction

There is a growing interest in developing machine learning methods that have little or no dependence on manual supervision. In this paper, we consider in particular the problem of learning, without external annotations, detectors for the landmarks of object categories, such as the nose, the eyes, and the mouth of a face, or the hands, shoulders, and head of a human body.

Our approach learns landmarks by looking at images of deformable objects that differ by acquisition time and/or viewpoint. Such pairs may be extracted from video sequences or can be generated by randomly perturbing still images. Videos have been used before for self-supervision, often in the context of future frame prediction, where the goal is to generate future video frames by observing one or more past frames. A key difficulty in such approaches is the high degree of ambiguity that exists in predicting the motion of objects from past observations. In order to eliminate this ambiguity, we propose instead to condition generation on two images, a source (past) image and a target (future) image. The goal of the learned model is to reproduce the target image, given the source and target images as input. Clearly, without further constraints, this task is trivial. Thus, we pass the target through a tight bottleneck meant to *distil the geometry of the object* (fig. 1). We do so by constraining the resulting representation to encode spatial locations, as may be obtained by an object landmark detector. The source image and the encoded target image are then passed to a generator network which reconstructs the target. Minimising the reconstruction error encourages the model to learn landmark-like representations because landmarks can be used to encode the *geometry* of the object,

[*] equal contribution.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

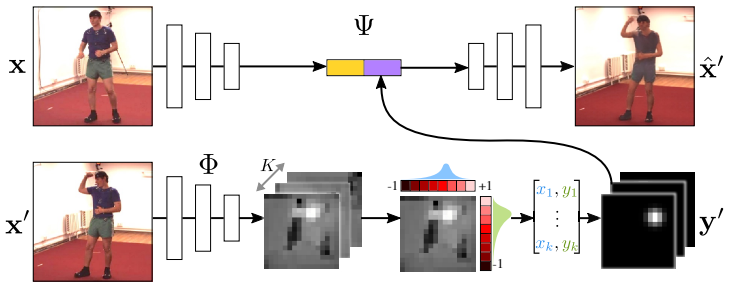


Figure 1: **Model Architecture.** Given a pair of source and target images $(\mathbf{x}, \mathbf{x}')$, the pose-regressor Φ extracts K heatmaps from $\mathbf{x}'$, which are then marginalized to estimate coordinates of keypoints, to limit the information flow. 2D Gaussians $(\mathbf{y}')$ are rendered from these keypoints and stacked along with the image features extracted from $\mathbf{x}$, to reconstruct the target as $\Psi(\mathbf{x}, \mathbf{y}') = \hat{\mathbf{x}}'$. By restricting the information-flow our model learns semantically meaningful keypoints, without any annotations.

which changes between source and target, while the appearance of the object, which is constant, can be obtained from the source image alone.

The key advantage of our method, compared to other works for unsupervised learning of landmarks, is the simplicity and generality of the formulation, which allows it to work well on data far more complex than previously used in unsupervised learning of object landmarks, *e.g.* landmarks for the highly-articulated human body. In particular, unlike methods such as [45, 44, 55], we show that our method can learn from synthetically-generated image deformations as well as raw videos as it *does not* require access to information about correspondences, optical-flow, or transformation between images.

Furthermore, while image generation has been used extensively in unsupervised learning, especially in the context of (variational) auto-encoders [22] and Generative Adversarial Networks (GANs [13]; see section 2), our approach has a key advantage over such methods. Namely, conditioning on both source and target images simplifies the generation task considerably, making it much easier to learn the generator network [18]. The ensuing simplification means that we can adopt the direct approach of minimizing a perceptual loss as in [10], without resorting to more complex techniques like GANs. Empirically, we show that this still results in excellent image generation results and that, more importantly, semantically consistent landmark detectors are learned without manual supervision (section 4). Project code and details are available at: `http://www.robots.ox.ac.uk/~vgg/research/unsupervised_landmarks/`

2 Related work

The recent approaches of [45, 44] learn to extract landmarks based on the principles of equivariance and distinctiveness. In contrast to our work, these methods are not generative. Further, they rely on known correspondences between images obtained either through optical flow or synthetic transformations, and hence, cannot leverage video data directly. Since the principle of equivariance is orthogonal to our approach, it can be incorporated as an additional cue in our method.

Unsupervised learning of representations has traditionally been achieved using auto-encoders and restricted Boltzmann machines [14, 47, 15]. InfoGAN [6] uses GANs to disentangle factors in the data by imposing a certain structure in the latent space. Our approach also works by imposing a latent structure, but using a *conditional*-encoder instead of an auto-encoder.

Learning representations using conditional image generation via a bottleneck was demonstrated by Xue *et al*. [52] in variational auto-encoders, and by Whitney *et al*. [50] using a discrete gating mechanism to combine representations of successive video frames. Denton *et al*. [8] factor the pose and identity in videos through an adversarial loss on the pose embeddings. We instead design our bottleneck to explicitly shape the features to resemble the output of a landmark detector, without any adversarial training. Villegas *et al*. [46] also generate future frames by extracting a representation of appearance and human pose, but, differently from us, require ground-truth pose annotations. Our method essentially *inverts* their analogy network [36] to output landmarks given the source and target image pairs.

Several other generative methods [42, 40, 37, 48, 32] focus on video extrapolation. Srivastava *et al*. [40] employ Long Short Term Memory (LSTM) [16] networks to encode video sequences into fixed-length representation and decode it to reconstruct the input sequence. Vondrick *et al*. [48] propose a GAN for videos, also with a spatio-temporal convolutional architecture that disentangles foreground and background to generate realistic frames. Video Pixel Networks [20] estimate the discrete joint distribution of the pixel values in a video by encoding different modalities such as time, space and colour information. In contrast, we learn a *structured embedding* that explicitly encodes the spatial location of object landmarks.

A series of concurrent works propose similar methods for unsupervised learning of object structure. Shu *et al*. [38] learn to factor a single object-category-specific image into an appearance template in a canonical coordinate system, and a deformation field which warps the template to reconstruct the input, as in an auto-encoder. They encourage this factorisation by controlling the size of the embeddings. Similarly, Wiles *et al*. [51] learn a dense deformation field for faces but obtain the template from a second related image, as in our method. Suwajanakorn *et al*. [43] learn 3D-keypoints for objects from two images which differ by a known 3D transformation, by enforcing equivariance [45]. Finally, the method of Zhang *et al*. [55] shares several similarities with ours, in that they also use image generation with the goal of learning landmarks. However, their method is based on generating a single image from *itself* using landmark-transported features. This, we show is insufficient to learn geometry and requires, as they do, to also incorporate the principle of equivariance [45]. This is a key difference with our method, as ours results in a much simpler system that does *not* require to know the optical-flow/correspondences between images, and can learn from raw videos directly.

3 Method

Let $\mathbf{x}, \mathbf{x}' \in \mathcal{X} = \mathbb{R}^{H\times W\times C}$ be two images of an object, for example extracted as frames in a video sequence, or synthetically generated by randomly deforming $\mathbf{x}$ into $\mathbf{x}'$. We call $\mathbf{x}$ the source image and $\mathbf{x}'$ the target image and we use Ω to denote the image domain, namely the $H\times W$ lattice.

We are interested in learning a function $\Phi(\mathbf{x}) = \mathbf{y} \in \mathcal{Y}$ that captures the "structure" of the object in the image as a set of K object landmarks. As a first approximation, assume that $\mathbf{y} = (u_1, \dots, u_K) \in \Omega^K = \mathcal{Y}$ are K coordinates $u_k \in \Omega$, one per landmark.

In order to learn the map Φ in an unsupervised manner, we consider the problem of conditional image generation. Namely, we wish to learn a generator function

$$\Psi : \mathcal{X} \times \mathcal{Y} \to \mathcal{X}, \qquad (\mathbf{x}, \mathbf{y}') \mapsto \mathbf{x}'$$

such that the target image $\mathbf{x}' = \Psi(\mathbf{x}, \Phi(\mathbf{x}'))$ is reconstructed from the *source image* $\mathbf{x}$ and the *representation* $\mathbf{y}' = \Phi(\mathbf{x}')$ of the *target image*. In practice, we learn both functions Φ and Ψ jointly to minimise the expected reconstruction loss $\min_{\Psi,\Phi} E_{\mathbf{x},\mathbf{x}'}\left[\mathcal{L}(\mathbf{x}', \Psi(\mathbf{x}, \Phi(\mathbf{x}')))\right]$. Note that, if we do not restrict the form of $\mathcal{Y}$, then a trivial solution to this problem is to learn identity mappings by setting $\mathbf{y}' = \Phi(\mathbf{x}') = \mathbf{x}'$ and $\Psi(\mathbf{x}, \mathbf{y}') = \mathbf{y}'$. However, given that $\mathbf{y}'$ has the "form" of a set of landmark detections, the model is strongly encouraged to learn those. This is explained next.

3.1 Heatmaps bottleneck

In order for the model $\Phi(\mathbf{x})$ to learn to extract keypoint-like structures from the image, we terminate the network Φ with a layer that forces the output to be akin to a set of K keypoint detections. This is done in three steps. First, K heatmaps $S_u(\mathbf{x}; k), u \in \Omega$ are generated, one for each keypoint $k = 1, \dots, K$. These heatmaps are obtained in parallel as the channels of a $\mathbb{R}^{H\times W\times K}$ tensor using a standard convolutional neural network architecture. Second, each heatmap is renormalised to a probability distribution via (spatial) $\mathrm{Softmax}$ and condensed to a point by computing the (spatial) expected value of the latter:

$$u_k^*(\mathbf{x}) = \frac{\sum_{u\in\Omega} u e^{S_u(\mathbf{x};k)}}{\sum_{u\in\Omega} e^{S_u(\mathbf{x};k)}} \tag{1}$$

Third, each heatmap is replaced with a Gaussian-like function centred at u_k^* with a small fixed standard deviation σ:

$$\Phi_u(\mathbf{x}; k) = \exp\left(-\frac{1}{2\sigma^2}\|u - u_k^*(\mathbf{x})\|^2\right) \tag{2}$$

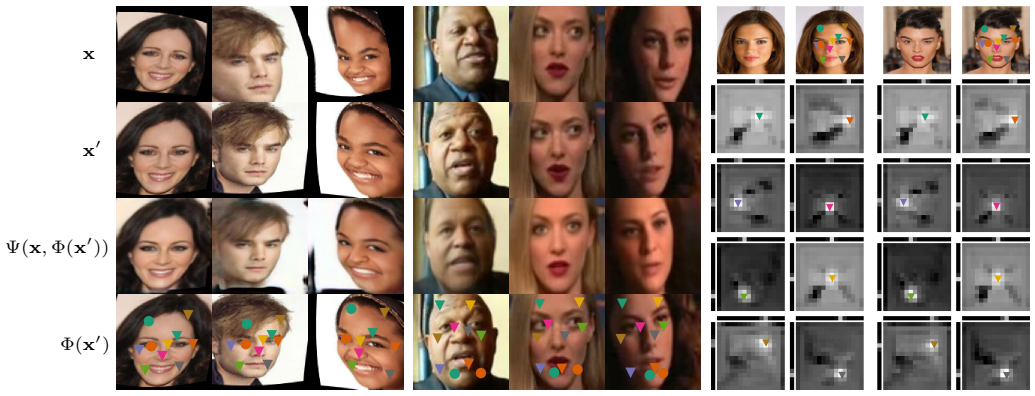


Figure 2: **Unsupervised Landmarks. [left]:** CelebA images showing the synthetically transformed source $\mathbf{x}$ and target $\mathbf{x}'$ images, the reconstructed target $\Psi(\mathbf{x}, \Phi(\mathbf{x}'))$, and the unsupervised landmarks $\Phi(\mathbf{x}')$. **[middle]:** The same for video frames from VoxCeleb. **[right]:** Two example images with selected (8 out of 10) landmarks u_k overlaid and their corresponding 2D score maps $S_u(\mathbf{x}; k)$ (see section 3.1; brighter pixels indicate higher confidence).

The end result is a new tensor $\mathbf{y} = \Phi(\mathbf{x}) \in \mathbb{R}^{H \times W \times K}$ that encodes as Gaussian heatmaps the location of K maxima. Since it is possible to recover the landmark locations exactly from these heatmaps, this representation is equivalent to the one considered above (2D coordinates); however, it is more useful as an input to a generator network, as discussed later.

One may wonder whether this construction can be simplified by removing steps two and three and simply consider $S(\mathbf{x})$ (possibly after re-normalisation) as the output of the encoder $\Phi(\mathbf{x})$. The answer is that these steps, and especially eq. (1), ensure that very little information from $\mathbf{x}$ is retained, which, as suggested above, is key to avoid degenerate solutions. Converting back to Gaussian landmarks in eq. (2), instead of just retaining 2D coordinates, ensures that the representation is still utilisable by the generator network.

Separable implementation. In practice, we consider a separable variant of eq. (1) for computational efficiency. Namely, let $u = (u_1, u_2)$ be the two components of each pixel coordinate and write $\Omega = \Omega_1 \times \Omega_2$. Then we set

$$u^*_{ik}(\mathbf{x}) = \frac{\sum_{u_i \in \Omega_i} u_i e^{S_{u_i}(\mathbf{x};k)}}{\sum_{u_i \in \Omega_i} e^{S_{u_i}(\mathbf{x};k)}}, \qquad S_{u_i}(\mathbf{x}; k) = \sum_{u_j \in \Omega_j} S_{(u_1,u_2)}(\mathbf{x}; k),$$

where $i = 1, 2$ and $j = 2, 1$ respectively. Figure 2 visualizes the source $\mathbf{x}$, target $\mathbf{x}'$ and generated $\Psi(\mathbf{x}, \Phi(\mathbf{x}'))$ images, as well as $\mathbf{x}'$ overlaid with the locations of the unsupervised landmarks $\Phi(\mathbf{x}')$. It also shows the heatmaps $S_u(\mathbf{x}; k)$ and marginalized separable softmax distributions on the top and left of each heatmap for $K = 10$ keypoints.

3.2 Generator network using a perceptual loss

The goal of the generator network $\hat{\mathbf{x}}' = \Psi(\mathbf{x}, \mathbf{y}')$ is to map the source image $\mathbf{x}$ and the distilled version $\mathbf{y}'$ of the target image $\mathbf{x}'$ to a reconstruction of the latter. Thus the generator network is optimised to minimise a reconstruction error $\mathcal{L}(\mathbf{x}', \hat{\mathbf{x}}')$. The design of the reconstruction error is important for good performance. Nowadays the standard practice is to learn such a loss function using adversarial techniques, as exemplified in numerous variants of GANs. However, since the goal here is not generative modelling, but rather to induce a representation $\mathbf{y}'$ of the object geometry for reconstructing a *specific* target image (as in an auto-encoder), a simpler method may suffice.

Inspired by the excellent results for photo-realistic image synthesis of [4], we resort here to use the "content representation" or "perceptual" loss used successfully for various generative networks [12, 1, 9, 19, 27, 30, 31]. The perceptual loss compares a set of the activations extracted from multiple layers of a deep network for both the reference and the generated images, instead of the only raw pixel values. We define the loss as $\mathcal{L}(\mathbf{x}', \hat{\mathbf{x}}') = \sum_l \alpha_l \|\Gamma_l(\mathbf{x}') - \Gamma_l(\hat{\mathbf{x}}')\|_2^2$, where $\Gamma(\mathbf{x})$ is an off-the-shelf pre-trained neural network, for example VGG-19 [39], Γ_l denotes the output of the l-th sub-network (obtained by chopping Γ at layer l). As our goal is to have a purely-unsupervised learning, we pre-train the network by using a self-supervised approach, namely colorising grayscale images [25].

n supervised	Thewlis [45]	Ours selfsup
1	10.82	12.89 ± 3.21
5	9.25	8.16 ± 0.96
† 10	8.49	7.19 ± 0.45
100	—	4.29 ± 0.34
500	—	2.83 ± 0.06
1000	—	2.73 ± 0.03
5000	—	2.60 ± 0.00
All (19,000)	7.15	$2.58\pm$ N/A

Figure 3: **Sample Efficiency for Supervised Regression on MAFL. [left]:** Supervised linear regression of 5 keypoints (bottom-row) from 10 unsupervised (top-row) on MAFL test set. Centre of the white-dots correspond to the ground-truth location, while the dark ones are the predictions. Both unsupervised and supervised landmarks show a good degree of equivariance with respect to head rotation (columns 2, 4) and invariance to headwear or eyewear (columns 1, 3). **[right]:** MSE ($\pm\sigma$) (normalised by inter-ocular distance (in %)) on the MAFL test-set for varying number (n) of supervised samples from MAFL training set used for learning the regressor from 30 unsupervised landmarks. †: we outperform the previous state-of-the-art [45] with only 10 labelled examples.

We also test using a VGG-19 model pre-trained for image classification in ImageNet. All other networks are trained from scratch. The parameters $\alpha_l > 0, l = 1, \ldots, n$ are scalars that balance the terms. We use a linear combination of the reconstruction error for 'input', 'conv1_2', 'conv2_2', 'conv3_2', 'conv4_2' and 'conv5_2' layers of VGG-19; $\{\alpha_l\}$ are updated online during training to normalise the expected contribution from each layer as in [4]. However, we use the ℓ_2 norm instead of their ℓ_1, as it worked better for us.

4 Experiments

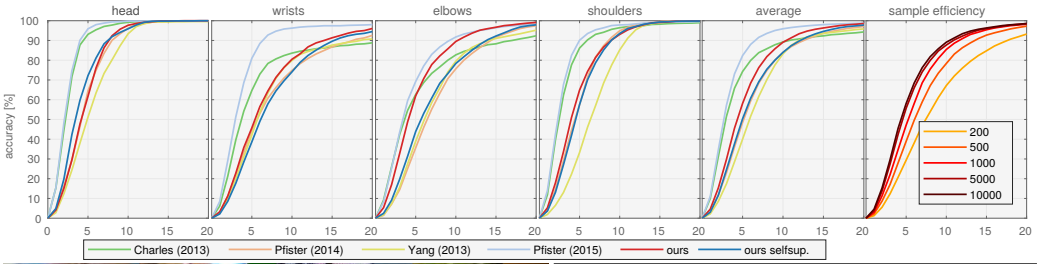
In section 4.1 we provide the details of the landmark detection and generator networks; a common architecture is used across all datasets. Next, we evaluate landmark detection accuracy on faces (section 4.2) and human-body (section 4.3). In section 4.4 we analyse the invariance of the learned landmarks to various nuisance factors, and finally in section 4.5 study the factorised representation of object style and geometry in the generator.

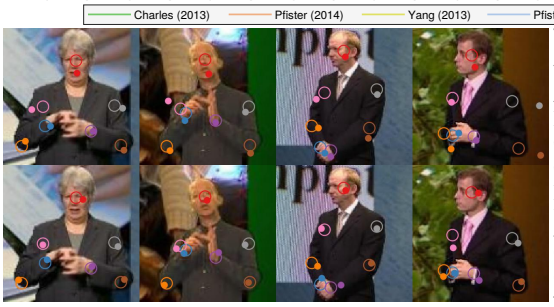
4.1 Model details

Landmark detection network. The landmark detector ingests the image $\mathbf{x}'$ to produce K landmark heatmaps $\mathbf{y}'$. It is composed of sequential blocks consisting of two convolutional layers each. All the layers use 3×3 filters, except the first one which uses 7×7. Each block doubles the number of feature channels in the previous block, with 32 channels in the first one. The first layer in each block, except the first block, downsamples the input tensor using stride 2 convolution. The spatial size of the final output, outputting the heatmaps, is set to 16×16. Thus, due to downsampling, for a network with $n-3$, $n \geq 4$ blocks, the resolution of the input image is $H\times W = 2^n\times2^n$, resulting in $16\times16\times(32\cdot2^{n-3})$ tensor. A final 1×1 convolutional layer maps this tensor to a $16\times16\times K$ tensor, with one layer per landmark. As described in section 3.1, these K feature channels are then used to render $16\times16\times K$ 2D-Gaussian maps $\mathbf{y}'$ (with $\sigma = 0.1$).

Image generation network. The image generator takes as input the image $\mathbf{x}$ and the landmarks $\mathbf{y}' = \Phi(\mathbf{x}')$ extracted from the second image in order to reconstruct the latter. This is achieved in two steps: first, the image $\mathbf{x}$ is encoded as a feature tensor $\mathbf{z} \in \mathbb{R}^{16\times16\times C}$ using a convolutional network with exactly the same architecture as the landmark detection network except for the final 1×1 convolutional layer, which is omitted; next, the features $\mathbf{z}$ and the landmarks $\mathbf{y}'$ are stacked together (along the channel dimension) and fed to a regressor that reconstructs the target frame $\mathbf{x}'$.

The regressor also comprises of sequential blocks with two convolutional layers each. The input to each successive block, except the first one, is upsampled two times through bilinear interpolation, while the number of feature channels is halved; the first block starts with 256 channels, and a minimum of 32 channels are maintained till a tensor with the same spatial dimensions as $\mathbf{x}'$ is obtained. A final convolutional layer regresses the three RGB channels with no non-linearity. All





BBC Pose Accuracy (%) at d = 6 pixels

	Head	Wrsts	Elbws	Shldrs	Avg.
Pfister *et al*. [35]	98.00	88.45	77.10	93.50	88.01
Charles *et al*. [3]	95.40	72.95	68.70	90.30	79.90
Chen *et al*. [5]	65.9	47.9	66.5	76.8	64.1
Pfister *et al*. [34]	74.90	53.05	46.00	71.40	59.40
Yang *et al*. [53]	63.40	53.70	49.20	46.10	51.63
Ours (selfsup.)	81.10	49.05	53.05	70.10	60.79
Ours	76.10	56.50	70.70	74.30	68.44

Figure 4: **Learning Human Pose.** 50 unsupervised keypoints are learnt on the BBC Pose dataset. Annotations (empty circles in the images) for 7 keypoints are provided, corresponding to — head, wrists, elbows and shoulders. Solid circles represent the predicted positions; in **[fig-top]** these are raw discovered keypoints which correspond maximally to each annotation; in **[fig-bottom]** these are regressed (linearly) from the discovered keypoints. **[table]:** Comparison against *supervised* methods; %-age of points within d= 6-pixels of ground-truth is reported. **[top-row]:** accuracy-vs-distance d, for each body-part; **[top-row-rightmost]:** average accuracy for varying number of supervised samples used for regression.

layers use 3×3 filters and each block has two layers similarly to the landmark network.
All the weights are initialised with random Gaussian noise ($\sigma = 0.01$), and optimised using Adam [21] with a weight decay of $5 \cdot 10^{-4}$. The learning rate is set to 10^{-2}, and lowered by a factor of 10 once the training error stops decreasing; the ℓ_2-norm of the gradients is bounded to 1.0.

4.2 Learning facial landmarks

Setup. We explore extracting source-target image pairs $(\mathbf{x}, \mathbf{x}')$ using either (1) synthetic transformations, or (2) videos. In the first case, the pairs are obtained as $(\mathbf{x}, \mathbf{x}') = (g_1\mathbf{x}_0, g_2\mathbf{x}_0)$ by applying two random thin-plate-spline (TPS) [11, 49] warps g_1, g_2 to a given sample image $\mathbf{x}_0$. We use the 200k CelebA [24] images after resizing them to 128×128 resolution. The dataset provides annotations for 5 facial landmarks — eyes, nose and mouth corners, which we *do not* use for training. Following [45] we exclude the images in MAFL [57] test-set from the training split and generate synthetically-deformed pairs as in [45, 55], but the transformations themselves are not required for training. We discount the reconstruction loss in the regions of the warped image which lie outside the original image to avoid modelling irrelevant boundary artefacts.

In the second case, $(\mathbf{x}, \mathbf{x}')$ are two frames sampled from a video. We consider VoxCeleb [28], a large dataset of face tracks, consisting of 1251 celebrities speaking over 100k English language utterances. We use the standard training split and remove any overlapping identities which appear in the test sets of MAFL and AFLW. Pairs of frames from the same video, but possibly belonging to different utterances are randomly sampled for training. By using video data for training our models we eliminate the need for engineering synthetic data.

Figure 5: **Unsupervised Landmarks on Human3.6M. [left]:** an example quadruplet source-target-reconstruction-keypoint (left to right) from Human3.6M. **[right]:** learned keypoints on a test video sequence. The landmarks consistently track the legs, arms, torso and head across frames.

Qualitative results. Figure 2 shows the learned heatmaps and source-target-reconstruction-keypoints quadruplets $\langle \mathbf{x}, \mathbf{x}', \Psi\left(\mathbf{x}, \Phi(\mathbf{x}')\right), \Phi(\mathbf{x}')\rangle$ for synthetic transformations and videos. We note that the method extracts keypoints which consistently track facial features across deformation and identity changes (*e.g.*, the green circle tracks the lower chin, and the light blue square lies between the eyes). The regressed semantic keypoints on the MAFL test set are visualised in fig. 3, where they are localised with high accuracy. Further, the target image $\mathbf{x}'$ is also reconstructed accurately.

Quantitative results. We follow [45, 44] and use unsupervised keypoints learnt on CelebA and VoxCeleb to regress manually-annotated keypoints in the MAFL and AFLW [23] test sets. We freeze the parameters of the unsupervised detector network (Φ) and learn a *linear* regressor (without bias) from our unsupervised keypoints to 5 manually-labelled ones from the respective training sets. Model selection is done using 10% validation split of the training data.

We report results in terms of standard MSE normalised by the inter-ocular distance expressed as a percentage [57], and show a few regressed keypoints in fig. 3. Before evaluating on AFLW, we finetune our networks pre-trained on CelebA or VoxCeleb on the AFLW training set. We do not use any labels during finetuning.

Sample efficiency. Figure 3 reports the performance of detectors trained on CelebA as a function of the number n of supervised examples used to translate from unsupervised to supervised keypoints. We note that $n = 10$ is already sufficient for results comparable to the previous state-of-the-art (SoA) method of Thewlis *et al*. [45], and that performance almost saturates at $n = 500$ (vs. 19,000 available training samples).

Vs. SoA. Table 1 compares our regression results to the SoA. We experiment regressing from $K{=}\{10, 30, 50\}$ unsupervised landmarks, using the self-supervised and the supervised perceptual loss networks; the number of samples n used for regression is maxed out ($= 19000$) to be consistent with previous works. On both MAFL and AFLW datasets, at 2.58% and 6.31% error respectively (for $K = 30$), we significantly outperform all the supervised and unsupervised methods. Notably, we perform better than the concurrent work of Zhang *et al*. [55] (MAFL: 3.16%; AFLW: 6.58%), while using a simpler method. When synthetic warps are removed from [55], so that the *equivariance constraint cannot be employed*, our method is significantly better (2.58% vs 8.42% on MAFL). We are also significantly better than many SoA *supervised* detectors [54, 41, 57] using only $n = 100$ supervised training examples, which shows that the approach is very effective at exploiting the unlabelled data. Finally, training with VoxCeleb video frames degrades the performance due to domain gap; including a bias in the linear regressor improves the performance.

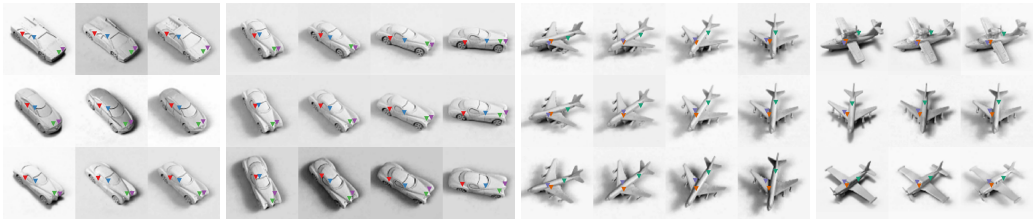
Method	K	MAFL	AFLW
Supervised			
RCPR [2]		–	11.60
CFAN [54]		15.84	10.94
Cascaded CNN [41]		9.73	8.97
TCDCN [57]		7.95	7.65
RAR [41]		–	7.23
MTCNN [56]		5.39	6.90
Unsupervised / self-supervised			
Thewlis [45]	30	7.15	–
	50	6.67	10.53
Thewlis [44](frames)	–	5.83	8.80
Shu † [38]	–	5.45	–
Zhang [55]	10	3.46	7.01
w/ equiv.	30	3.16	6.58
w/o equiv.	30	8.42	–
Wiles ‡ [51]	–	3.44	–
Ours, training set: CelebA			
loss-net: selfsup.	10	3.19	6.86
	30	2.58	**6.31**
	50	**2.54**	6.33
loss-net: sup.	10	3.32	6.99
	30	2.63	6.39
	50	2.59	6.35
Ours, training set: VoxCeleb			
loss-net: selfsup.	30	3.94	6.75
w/ bias	30	3.63	–
loss-net: sup.	30	4.01	7.10

Table 1: **Comparison with state-of-the-art on MAFL and AFLW.** K is the number of unsupervised landmarks. †: train a 2-layer MLP instead of a *linear* regressor. ‡: use the larger VoxCeleb2 [7] dataset for unsupervised training, and include a bias term in their regressor (through batch-normalization). Normalised $\%$-MSE is reported (see fig. 3).

fc-layer (d) $\rightarrow$	10	20	60	ours $K{=}30$
MAFL	20.60	21.94	28.96	**2.58**

loss $\rightarrow$	ℓ_1	adv.+ ℓ_1	ℓ_2	adv.+ ℓ_2	content (ours)
MAFL ($K{=}30$)	3.64	3.62	2.84	2.80	**2.58**

Table 2: **Abalation Study. [left]:** The keypoint bottleneck when replaced with a low d-dimensional, $d = \{10, 20, 60\}$, *fully-connected* (fc) layer leads to significantly worse landmark detection performance ($\%$-MSE) on the MAFL dataset. **[right]:** Replacing the *content* loss with ℓ_1, ℓ_2 losses on the images, optionally paired with an *adversarial* loss (*adv.*) also degrades the performance.



(a) (b) (c) (d)

Figure 6: **Invariant Localisation.** Unsupervised keypoints discovered on smallNORB test set for the *car* and *airplane* categories. Out of 20 learned keypoints, we show the most geometrically stable ones: they are invariant to pose, shape, and illumination. **[b–c]:** elevation-vs-azimuth; **[a, d]:** shape-vs-illumination (y-axis-vs-x-axis).

Ablation study. In table 2 we present two ablation studies, first on the keypoint bottleneck, and second where we compare against adversarial and other image-reconstruction losses. For both the settings, we take the best performing model configuration for facial landmark detection on the MAFL dataset.

Keypoint bottleneck. The keypoint bottleneck has two functions: (1) it provides a differentiable and distributed representation of the location of landmarks, and (2) it restricts the information from the target image to spatial locations only. When the bottleneck is replaced with a generic low dimensional fully-connected layer (as in a conventional auto-encoder) the performance degrades significantly This is because the continuous vector embedding is not encouraged to encode geometry explicitly.

Reconstruction loss. We replace our content/perceptual loss with ℓ_1 and ℓ_2 losses on generated pixels; the losses are also optionally paired with an *adversarial* term [13] to encourage verisimilitude as in [18]. All of these alternatives lead to worse landmark detection performance (table 2). While GANs are useful for aligning image distributions, in our setting we reconstruct a *specific* target image (similar to an auto-encoder). For this task, it is enough to use a simple content/perceptual loss.

4.3 Learning human body landmarks

Setup. Articulated limbs make landmark localisation on human body significantly more challenging than faces. We consider two *video* datasets, BBC-Pose [3], and Human3.6M [17]. BBC-Pose comprises of 20 one-hour long videos of sign-language signers with varied appearance, and dynamic background; the test set includes 1000 frames. The frames are annotated with 7 keypoints corresponding to head, wrists, elbows, and shoulders which, as for faces, we use only for quantitative evaluation, not for training. Human3.6M dataset contains videos of 11 actors in various poses, shot from multiple viewpoints. Image pairs are extracted by randomly sampling frames from the same video sequence, with the additional constraint of maintaining the time difference within the range 3-30 frames for Human3.6M. Loose crops around the subjects are extracted using the provided annotations and resized to 128×128 pixels. Detectors for $K = 20$ and $K = 50$ keypoints are trained on Human3.6M and BBC-Pose respectively.

Qualitative results. Figure 4 shows raw unsupervised keypoints and the regressed semantic ones on the BBC-Pose dataset. For each annotated keypoint, a maximally matching unsupervised keypoint is identified by solving bipartite linear assignment using mean distance as the cost. Regressed keypoints consistently track the annotated points. Figure 5 shows $\langle \mathbf{x}, \mathbf{x}', \Psi\left(\mathbf{x}, \Phi(\mathbf{x}')\right), \Phi(\mathbf{x}')\rangle$ quadruplets, as for faces, as well as the discovered keypoints. All the keypoints lie on top of the human actors, and consistently track the body across identities and poses. However, the model cannot discern frontal and dorsal sides of the human body apart, possibly due to weak cues in the images, and no explicit constraints enforcing such consistency.

Quantitative results. Figure 4 compares the accuracy of localising the 7 keypoints on BBC-Pose against *supervised* methods, for both self-supervised and supervised perceptual loss networks. The accuracy is computed as the the $\%$-age of points within a specified pixel distance d. In this case, the top two supervised methods are better than our unsupervised approach, but we outperform [33, 53] using 1k training samples (vs. 10k); furthermore, methods such as [35] are specialised for videos and

Figure 7: **Disentangling Style and Geometry.** Image generation conditioned on *spatial* keypoints induces disentanglement of representations for style and geometry in the generator. Source image ($\mathbf{x}$) imparts style (*e.g.* colour, texture), while the target image ($\mathbf{x}'$) influences the geometry (*e.g.* shape, pose). Here, during inference, $\mathbf{x}$ **[middle]** is sampled to have a different *style* than $\mathbf{x}'$ **[top]**, although during training, image pairs with *consistent* style were sampled. The generated images **[bottom]** borrow their style from $\mathbf{x}$, and geometry from $\mathbf{x}'$. **(a) SVHN Digits:** the foreground and background colours are swapped. **(b) AFLW Faces:** pose of the style image $\mathbf{x}$ is made consistent with $\mathbf{x}'$. **(c) Human3.6M:** the background, hat, and shoes are retained from $\mathbf{x}$, while the pose is borrowed from $\mathbf{x}'$. All images are sampled from respective test sets, never seen during training.

leverage temporal smoothness. Training using the supervised perceptual loss is understandably better than using the self-supervised one. Performance is particularly good on parts such as the elbow.

4.4 Learning 3D object landmarks: pose, shape, and illumination invariance

We train our unsupervised keypoint detectors on the SmallNORB [26] dataset, comprising 5 object categories with 10 object instances each, imaged from regularly spaced viewpoints and under different illumination conditions. We train category-specific detectors for $K = 20$ keypoints using image-pairs from neighbouring viewpoints and show results in fig. 6 for *car* and *airplane* (see supplementary material for visualisation of other object categories). Keypoints most invariant to various factors are visualised. These landmarks are especially robust to changes in illumination and elevation angle. They are also invariant to smaller changes in azimuth ($\pm 80°$), but fail to generalise beyond that. Most interesting, they localise structurally similar regions, even when there is a large change in object shape (*e.g.* fig. 6-(d)); such landmarks could thus be leveraged for viewpoint-invariant semantic matching.

4.5 Disentangling appearance and geometry

In fig. 7 we show that our method can be interpreted as disentangling appearance from geometry. Generator/ keypoint networks are trained on SVHN digits [29], AFLW faces, and Human3.6M people. The generator network is capable of retaining the geometry of an image, and substituting the style with any other image in the dataset, including unrelated image pairs never seen during training. For example, in the third column we re-render the number 3 by mixing its geometry with the appearance of the number 5. This generalises significantly from the training examples, which only consist of pairs of digits sampled from the *same* house number instance, sharing a common style.

5 Conclusions

In this paper we have shown that a simple network trained for conditional image generation can be utilised to induce, without manual supervision, a object landmark detectors. On faces, our method outperforms previous unsupervised as well as supervised methods for landmark detection. The method can also extend to much more challenging data, such as detecting landmarks of people, and diverse data, such as 3D objects and digits.

Acknowledgements. We are grateful for the support provided by EPSRC AIMS CDT, ERC 638009-IDIU, and the Clarendon Fund scholarship. We would like to thank James Thewlis for suggestions and support with code and data, and David Novotný and Triantafyllos Afouras for helpful advice.

References

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

[2] X. P. Burgos-Artizzu, P. Perona, and P. Dollár. Robust face landmark estimation under occlusion. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1513–1520. IEEE, 2013.

[3] J. Charles, T. Pfister, D. Magee, D. Hogg, and A. Zisserman. Domain adaptation for upper body pose tracking in signed TV broadcasts. In *Proc. BMVC*, 2013.

[4] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. In *Proc. ICCV*, volume 1, 2017.

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

[6] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proc. NIPS*, pages 2172–2180, 2016.

[7] J. S. Chung, A. Nagrani, and A. Zisserman. VoxCeleb2: Deep speaker recognition. In *INTERSPEECH*, 2018.

[8] E. L. Denton and V. Birodkar. Unsupervised learning of disentangled representations from video. In *Proc. NIPS*. 2017.

[9] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *Proc. NIPS*, 2016.

[10] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *Proc. NIPS*, pages 658–666, 2016.

[11] J. Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive theory of functions of several variables*. 1977.

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

[13] 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*, 2014.

[14] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[15] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

[17] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *PAMI*, 2014.

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

[19] J. Johnson, A. Alahi, and F. Li. Perceptual losses for real-time style transfer and super-resolution. In *Proc. ECCV*, 2016.

[20] N. Kalchbrenner, A. Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu. Video pixel networks. *arXiv preprint arXiv:1610.00527*, 2016.

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

[22] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[23] M. Koestinger, P. Wohlhart, P. M. Roth, and H. Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *ICCV Workshops*, 2011.

[24] Z. L., P. L., X. W., and X. T. Deep learning face attributes in the wild. In *Proc. ICCV*, 2015.

[25] G. Larsson, M. Maire, and G. Shakhnarovich. Learning representations for automatic colorization. In *Proc. ECCV*, 2016.

[26] Y. LeCun, F. J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proc. CVPR*, 2004.

[27] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *Proc. CVPR*, 2017.

[28] A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*, 2017.

[29] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS DLW*, volume 2011, 2011.

[30] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Proc. NIPS*, 2016.

[31] A. Nguyen, J. Yosinski, Y. Bengio, A. Dosovitskiy, and J. Clune. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proc. CVPR*, 2017.

[32] V. Patraucean, A. Handa, and R. Cipolla. Spatio-temporal video autoencoder with differentiable memory. In *ICLR Workshop*, 2015.

[33] T. Pfister, J. Charles, and A. Zisserman. Large-scale learning of sign language by watching TV (using co-occurrences). In *Proc. BMVC*, 2013.

[34] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman. Deep convolutional neural networks for efficient pose estimation in gesture videos. In *Proceedings of the Asian Conference on Computer Vision*, 2014.

[35] T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. In *Proc. ICCV*, 2015.

[36] S. E. Reed, Y. Zhang, Y. Zhang, and H. Lee. Deep visual analogy-making. In *Proc. NIPS*, 2015.

[37] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee. Learning what and where to draw. In *Proc. NIPS*, pages 217–225, 2016.

[38] Z. Shu, M. Sahasrabudhe, A. Guler, D. Samaras, N. Paragios, and I. Kokkinos. Deforming autoencoders: Unsupervised disentangling of shape and appearance. In *Proc. ECCV*, 2018.

[39] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[40] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *Proc. ICML*, pages 843–852, 2015.

[41] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *Proc. CVPR*, 2013.

[42] I. Sutskever, G. E. Hinton, and G. W. Taylor. The recurrent temporal restricted boltzmann machine. In *Proc. NIPS*, pages 1601–1608, 2009.

[43] S. Suwajanakorn, N. Snavely, J. Tompson, and M. Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. In *Proc. NIPS*, 2018.

[44] J. Thewlis, H. Bilen, and A. Vedaldi. Unsupervised object learning from dense invariant image labelling. In *Proc. NIPS*, 2017.

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

[46] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee. Learning to generate long-term future via hierarchical prediction. *arXiv preprint arXiv:1704.05831*, 2017.

[47] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proc. ICML*, pages 1096–1103. ACM, 2008.

[48] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *Proc. NIPS*, pages 613–621, 2016.

[49] G. Wahba. *Spline models for observational data*, volume 59. Siam, 1990.

[50] W. F. Whitney, M. Chang, T. Kulkarni, and J. B. Tenenbaum. Understanding visual concepts with continuation learning. In *ICLR Workshop*, 2016.

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

[52] T. Xue, J. Wu, K. L. Bouman, and W. T. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Proc. NIPS*, 2016.

[53] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Proc. CVPR*, 2011.

[54] J. Zhang, S. Shan, M. Kan, and X. Chen. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In *Proc. ECCV*, 2014.

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

[56] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In *Proc. ECCV*, pages 94–108. Springer, 2014.

[57] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Learning Deep Representation for Face Alignment with Auxiliary Attributes. *PAMI*, 2016.

Appendix

We first present more detailed results on MAFL dataset comparing performance of different versions of our method. Then we show extended versions of figures presented in the paper. The sections are organized by the datasets used.

A MAFL

	Training set →	CelebA			VoxCeleb
K landmarks	Regression set	Thewlis [45]	sup.	selfsup.	sup.
10	MAFL	7.95	3.32	3.19	—
30		7.15	2.63	2.58	4.17
50		6.67	2.59	2.54	3.59
10	CelebA	6.32	3.32	3.19	—
30		5.76	2.63	2.57	4.14
50		5.33	2.59	2.53	3.55

Table 3: **Results on MAFL face-landmarks test-set.** Varying number (K) of unsupervised landmarks are learnt on two training-sets — random-TPS warps on CelebA [24], and face-videos from the VoxCeleb [28]. These landmarks are regressed onto 5 manually-annotated landmarks in the MAFL [57] test set, using either CelebA or MAFL training sets. Mean squared-error (MSE) normalised by the inter-ocular distance is reported.

B Boundary Discounting

When TPS warping is used during training some pixels in the resulting image may lie outside the original image. Since reconstructing these empty pixels is irrelevant we ignore them in the reconstruction loss. We additonaly ignore 10 pixels on the edges of the original image and use a smooth step over the next 20 pixels. This is to further discourage reconstruction of the empty pixels as they can influence the perceptual loss when a convolutional neural network with a large receptive field is used.

C MAFL and AFLW Faces

Figure 8: Supervised linear regression of 5 keypoints (bottom rows) from 10 unsupervised (top rows) on MAFL (above) and AFLW (below) test sets. Centre of the white-dots correspond to the ground-truth location, while the dark ones are the predictions. The models were trained on random-TPS warped image-pairs; self-supervised peceptual-loss network was used.

D VoxCeleb

Figure 9: Training with video frames from VoxCeleb. **[rows top-bottom]:** (1) source image $\mathbf{x}$, (2) target image $\mathbf{x}'$, (3) generated target image $\Psi(\mathbf{x}, \Phi(\mathbf{x}'))$, (4) unsupervised landmarks $\Phi(\mathbf{x}')$ superimposed on the target image. The landmarks consistently track facial features.

E BBCPose

Figure 10: **Learning Human Pose.** 50 unsupervised keypoints are learnt. Annotations (empty circles) for 7 keypoints are provided, corresponding to — head, wrists, elbows and shoulders. Solid circles represent the predicted positions; Top rows show raw discovered keypoints which correspond maximally to each annotation; bottom rows show linearly regressed points from the discovered keypoints. **[above]:** randomly sampled frames for different actors **[below]:** frames from a video track.

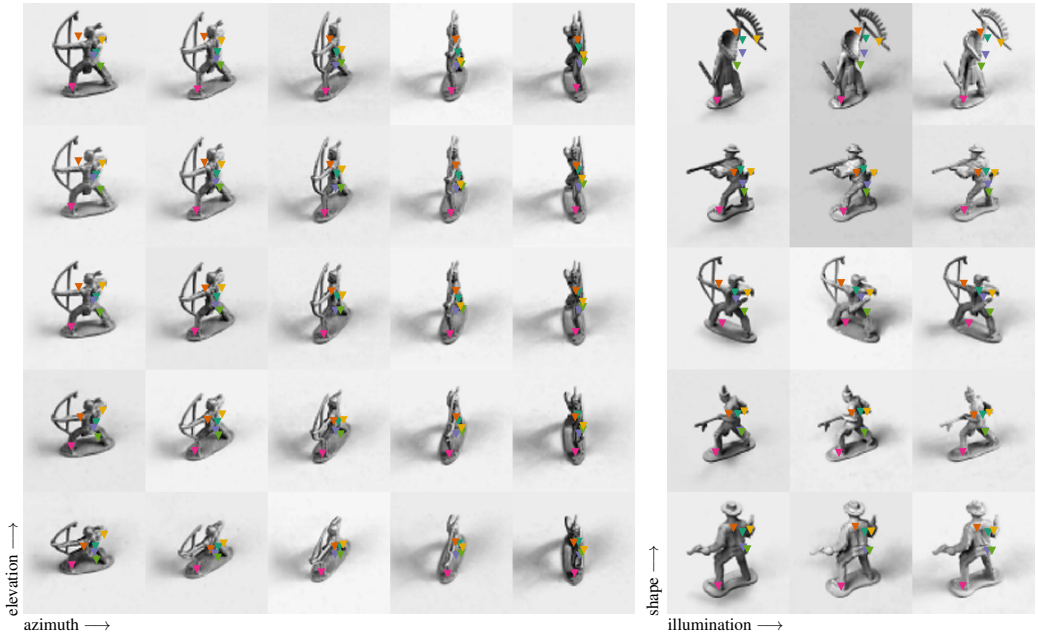
F Human3.6M

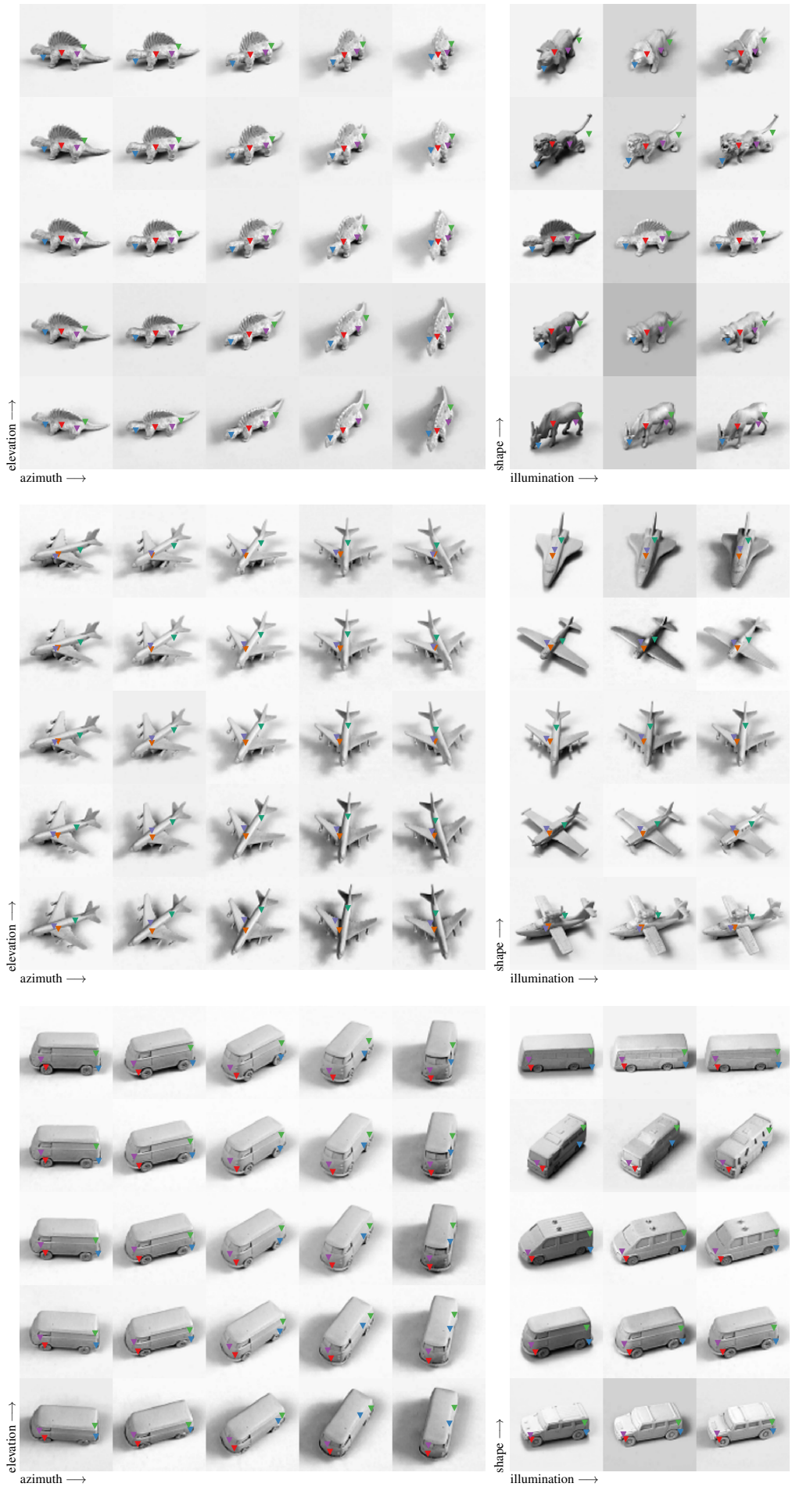
Figure 11: **Unsupervised Landmarks on Human3.6M.** Video of two actors (S1, S11) "posing", from the Human3.6M test set. **(rows)** (1) source, (2) target, (3) generated, (4) landmarks, (5) landmarks on frames from a different view, (6–7) landmarks on two views of the second actor. The landmarks consistently track the legs, arms, torso and head across frames, views and actors. However, the model confounds the frontal and dorsal sides.

G smallNORB 3D Objects: pose, shape, and illumination invariance

Object-category specific keypoint detectors are trained on the 5 categories in the smallNORB dataset — *human, car, animal, airplane,* and *truck*. Training is performed on pairs of images, which differ only in their viewpoints, but have the same object instance (or shape), and illumination.

Keypoints invaraint to viewpoint, illumniation, and object shape are visualised for object instances in the test set. The training set consists of only 5 object instances per category, yet the detectors generalise to novel object instances in the test set, and correspond to structurally similar regions across instances.




elevation ⟶
azimuth ⟶
shape ⟶
illumination ⟶
elevation ⟶
azimuth ⟶
shape ⟶
illumination ⟶
elevation ⟶
azimuth ⟶
shape ⟶
illumination ⟶

H Disentangling appearance and geometry

The generator substitutes the appearance of the target image ($\mathbf{x}'$) with that of the source image ($\mathbf{x}$). Instead of sampling image pairs ($\mathbf{x}, \mathbf{x}'$) with *consistent* style, as done during training, we sample pairs with *different* styles at inference, resulting in compelling transfer across different object categories — SVHN digits, Human3.6M humans, and AFLW faces.

Figure 12: **SVHN digits.** Target, source, and generated image triplets $\langle \mathbf{x}', \mathbf{x}, \Psi(\mathbf{x}, \Phi(\mathbf{x}')) \rangle$ from the SVHN test set. The digit shape is swapped out, while colours, shadows, and blur are retained.

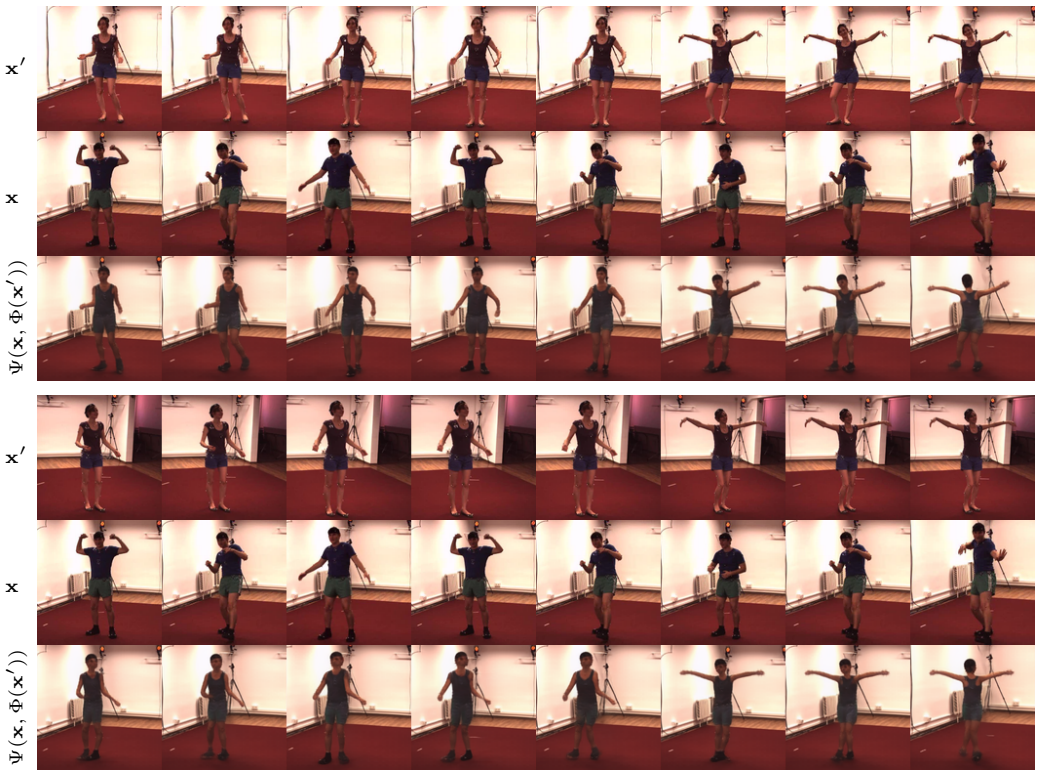


Figure 13: **Human3.6M humans.** Transfer across actors and viewpoints. **[top]:** different actors in various poses, imaged from the same viewpoint; the pose is swapped out, while appearance characteristics like shoes, clothing colour, and hat are retained. **[bottom]:** successful transfer even when the target is imaged from a different viewpoint (same poses as above).

Figure 14: **AFLW Faces.** The source image $\mathbf{x}$ is rendered with the pose from the target image $\mathbf{x}'$; the identity is retained.

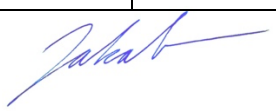
Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Unsupervised learning of object landmarks through conditional image generation	
Publication Status	☑ Published ☐ Submitted for Publication	☐ Accepted for Publication ☐ Unpublished and unsubmitted work written in a manuscript style
Publication Details	Jakab, T., Gupta, A., Bilen, H. and Vedaldi, A., 2018, December. Unsupervised learning of object landmarks through conditional image generation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (pp. 4020-4031).	

Student Confirmation

Student Name:	Tomas Jakab		
Contribution to the Paper	• Conception and development of the method • Extensive experimentation • Evaluation on human faces, bodies • Ablations • Manuscript • Supplemental material		
Signature		Date	17. 09. 2021

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Andrea Vedaldi		
This is a faithful representation of Tomas' contributions.		
Signature 	Date	17 September 2021

This completed form should be included in the thesis, at the end of the relevant chapter.

Chapter 4

Self-supervised Learning of Interpretable Keypoints from Unlabelled Videos

Self-supervised Learning of Interpretable Keypoints from Unlabelled Videos

Tomas Jakab
Visual Geometry Group
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
Visual Geometry Group
University of Oxford
vedaldi@robots.ox.ac.uk

Abstract

We propose KeypointGAN, a new method for recognizing the pose of objects from a single image that for learning uses only unlabelled videos and a weak empirical prior on the object poses. Video frames differ primarily in the pose of the objects they contain, so our method distils the pose information by analyzing the differences between frames. The distillation uses a new dual representation of the geometry of objects as a set of 2D keypoints, and as a pictorial representation, i.e. *a skeleton image. This has three benefits: (1) it provides a tight 'geometric bottleneck' which disentangles pose from appearance, (2) it can leverage powerful image-to-image translation networks to map between photometry and geometry, and (3) it allows to incorporate empirical pose priors in the learning process. The pose priors are obtained from unpaired data, such as from a different dataset or modality such as mocap, such that no annotated image is ever used in learning the pose recognition network. In standard benchmarks for pose recognition for humans and faces, our method achieves state-of-the-art performance among methods that do not require any labelled images for training. Project page:* http://www.robots.ox.ac.uk/~vgg/research/unsupervised_pose/

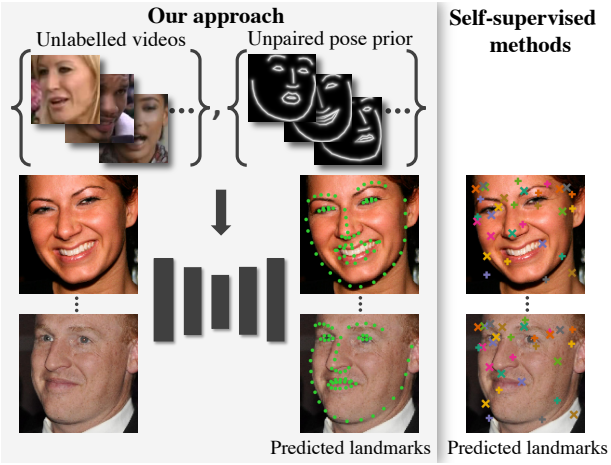


Figure 1. **Learning landmark detectors from unpaired data.** We learn to directly predict human-interpretable landmarks of an object using only unlabelled videos and a prior on the possible landmark configurations **[left]**. The prior can be obtained from unpaired supervision or from a different modality, such as mocap data. Our method, KeypointGAN, obtains state-of-the-art landmark detection performance for approaches that use unlabelled images for supervision. In contrast, self-supervised landmark detectors [25, 35, 57, 78] can only learn to discover keypoints **[right]** that are not human-interpretable (predictions from [25]) and require supervised post-processing.

1. Introduction

Learning with limited or no external supervision is one of the most significant open challenges in machine learning. In this paper, we consider the problem of learning the 2D geometry of object categories such as humans and faces using raw videos and as little additional supervision as possible. In particular, given as input a number of videos centred on the object, the goal is to learn automatically a neural network that can predict the *pose* of the object from a single image.

Learning from unlabelled images requires a suitable supervisory signal. Recently, [25] noted that during a video an object usually maintains its intrinsic appearance but changes its pose. Hence, the concept of pose can be learned by modelling the differences between video frames. They formulate this as *conditional image generation*. They extract a small amount of information from a given target video frame via a tight bottleneck which retains pose information while discarding appearance. For supervision, they reconstruct the target frame from the extracted pose, similar to an auto-encoder. However, since pose alone does not contain sufficient information to reconstruct the appearance of the object, they also pass to the generator a second video frame from which the appearance can be observed.

In this paper, we also consider a conditional image generation approach, but we introduce a whole new design for the model and for the 'pose bottleneck'. In particular, we adopt a dual representation of pose as a set of 2D object coordinates, and as a pictorial representation of the 2D coordinates in the

form of a skeleton image. We also define a differentiable skeleton generator to map between the two representations.

This design is motivated by the fact that, by encoding pose labels as images we can leverage *powerful image-to-image translation networks* [81] to map between photometry and geometry. In fact, the two sides of the translation process, namely the input image and its skeleton, are spatially aligned, which is well known to simplify learning by a Convolutional Neural Network (CNN) [81]. At the same time, using 2D coordinates provides a very tight bottleneck that allows the model to efficiently separate pose from appearance.

The pose bottleneck is further controlled via a discriminator, learned adversarially. This has the advantage of injecting prior information about the possible object poses in the learning process. While acquiring this prior may require some supervision, this is separate from the unlabelled videos used to learn the pose recognizer — that is, our method is able to leverage *unpaired supervision*. In this way, our method outputs poses that are directly interpretable. We refer to our proposed method as *KeypointGAN*. By contrast, state-of-the-art self-supervised keypoint detectors [25, 53, 57, 71, 78] do not learn "semantic" keypoints and, in post-processing, they need at least some *paired supervision* to output human-interpretable keypoints. We highlight this difference in fig. 1.

Overall, we make three significant contributions:

1. We introduce a new conditional generator design combining image translation, a new bottleneck using a dual representation of pose, and an adversarial loss which significantly improve recognition performance.
2. We learn, for the first time, to directly predict human-interpretable landmarks without requiring any labelled images.
3. We obtain state-of-the-art unsupervised landmark detection performance even when compared against methods that use paired supervision in post-processing.

We test our approach using videos of people, faces, and cat images. On standard benchmarks such as Human3.6M [23] and 300-W [50], we achieve state-of-the-art pose recognition performance for methods that learn only from unlabelled images. We also probe generalization by testing whether the empirical pose prior can be extracted independently from the videos used to train the pose recognizer. We demonstrate this in two challenging scenarios. First, we use the mocap data from MPI-INF-3DHP [38] as prior and we learn a human pose recognizer on videos from Human3.6M. Second, we use the MultiPIE [54] dataset as prior to learn a face pose recognizer on VoxCeleb2 [10] videos, and achieve state-of-the-art facial keypoint detection performance on 300-W.

2. Related work

We consider pose recognition, intended as the problem of predicting the 2D pose of an object from a single image. Approaches to this problem must be compared in relation to (1) the type of supervision, and (2) which priors they use. There are three broad categories for supervision: *full supervision* when the training images are annotated with the same labels that one wishes to predict; *weak supervision* when the predicted labels are richer than the image annotations; and *no supervision* when there are no image annotations. For the prior, methods can use a *prior model* learned from any kind of data or supervision, an *empirical prior*, or *no prior* at all.

Based on this definition, our method is *unsupervised* and uses an *empirical prior*. Next, we relate our work to others, dividing them by the type of supervision used (our method falls in the last category).

Full supervision. Several fully-supervised methods leverage large annotated datasets such as MS COCO Keypoints [33], Human3.6M [23], MPII [2] and LSP [27]. They generally do not use a separate prior as the annotations themselves capture one empirically. Some methods use pictorial structures [12] to model the object poses [1, 40, 43, 44, 51, 74]. Others use a CNN to directly regress keypoint coordinates [62], keypoint confidence maps [61], or other relations between keypoints [9]. Others again apply networks iteratively to refine heatmaps for single [3, 6, 8, 39, 42, 60, 70] and multi-person settings [7, 22]. Our method does not use any annotated image to learn the pose recognizer.

Weak supervision. A typical weakly-supervised method is the one of Kanazawa et al. [29]: they learn to predict dense 3D human meshes from sparse 2D keypoint annotations. They use two priors: SMPL [34] parametric human mesh model, and a prior on 3D poses acquired via adversarial learning from mocap data. Analogous works include [16, 17, 18, 49, 52, 64, 69, 73].

All such methods use a prior trained on unpaired data, as we do. However, they also use additional paired annotations such as 2D keypoints or relative depth relations [49]. Furthermore, in most cases they use a fully-fledged 3D prior such as SMPL human [34] or Basel face [41] models, while we only use an empirical prior in the form of example 2D keypoints configurations.

No supervision. Other methods use no supervision, and some no data-driven prior either. The works of [28, 48, 53, 71] learn to match pairs of images of an object, but they do not learn geometric invariants such as keypoints. [57, 58, 59] do learn sparse and dense landmarks, also without any annotation. The method of [56] does not use image annotations, but uses instead synthetic views of 3D models as prior, which we do not require.

Some of these methods use conditional image genera-

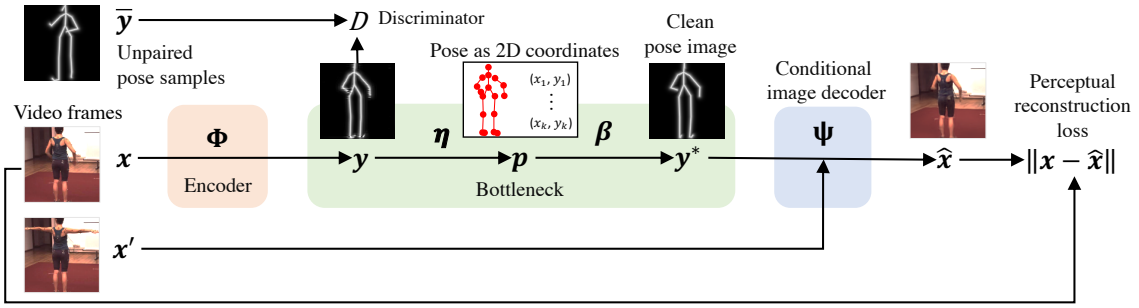


Figure 2. **KeypointGAN architecture.** We learn an encoder Φ that maps an image $\boldsymbol{x}$ to its pose $\boldsymbol{y}$, represented as a skeleton image. This is done via conditional auto-encoding, learning also a decoder Ψ that reconstruct the input $\boldsymbol{x}$ from its pose $\boldsymbol{y}$ and a second auxiliary video frame $\boldsymbol{x}'$. A bottleneck $\beta \circ \eta$ is used to drop appearance information that may leak in the pose image $\boldsymbol{y}$. A discriminator D is used to match the distribution of predicted poses to a reference prior distribution, represented by unpaired pose samples $\bar{\boldsymbol{y}}$.

tion as we do. Jakab & Gupta et al.[25], the most related, is described in the introduction. Lorenz et al. [35], Zhang et al. [78] develop an auto-encoding formulation to discover landmarks as explicit structural representations for a given image and use them to reconstruct the original image. Shu et al. [53], Wiles et al. [71] learn a dense deformation field for faces. Our method differs from those in the particular nature of the model and geometric bottleneck; furthermore, due to our use of a prior, we are able to learn out-of-the-box landmarks that are 'semantically meaningful'; on the contrary, these approaches must rely on at least some paired supervision to translate between the unsupervised and 'semantic' landmarks. We also outperform these approaches in landmark detection quality.

Adversarial learning. Our method is also related to adversarial learning, which has proven to be useful in image labelling [14, 20, 21, 65, 66] and generation [19, 81], including bridging the domain shift between real and generated images. Most relevant to our work, Isola et al. [24] propose an image-to-image translation framework using paired data, while CycleGAN [81] can do so with unpaired data. Our method also uses a image-to-image translation networks, but compared to CycleGAN our use of conditional image generation addresses the logical fallacy that an image-like label (a skeleton) does not contain sufficient information to generate a full image — this issue is discussed in depth in section 4.

Appearance and geometry factorization. Recent methods for image generation conditioned on object attributes, like viewpoint [47], pose [63], and hierarchical latents [55] have been proposed. Our method allows for similar but more fine-grained conditional image generation, conditioned on an appearance image or object landmarks. Many unsupervised methods for pose estimation [25, 35, 53, 71, 78] share similar ability. However, we can achieve more accurate and predictable image editing by manipulating semantic parts in the image through their corresponding landmarks.

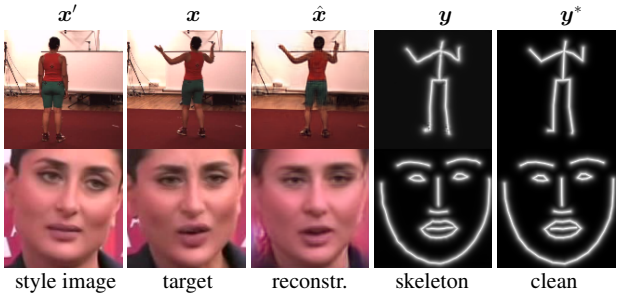


Figure 3. **Training data flow.** Data flowing through our model (fig. 2) during training on the Human3.6M (human pose) and VoxCeleb2 (face) datasets. $\boldsymbol{y}, \boldsymbol{y}^*$ are our predictions.

3. Method

Our goal is to learn a network $\Phi : \boldsymbol{x} \mapsto \boldsymbol{y}$ that maps an image $\boldsymbol{x}$ containing an object to its pose $\boldsymbol{y}$. To avoid having to use image annotations, the network is trained using an auto-encoder formulation. Namely, given the pose $\boldsymbol{y} = \Phi(\boldsymbol{x})$ extracted from the image, we train a decoder network Ψ that reconstructs the image from the pose. However, since pose lacks appearance information, this reconstruction task is ill posed. Hence, we also provide the decoder with a *different* image $\boldsymbol{x}'$ of the same object to convey its appearance. Formally, the image $\boldsymbol{x}$ is reconstructed from the pose $\boldsymbol{y}$ and the auxiliary image $\boldsymbol{x}'$ via a *conditional decoder network*

$$\boldsymbol{x} = \Psi(\Phi(\boldsymbol{x}), \boldsymbol{x}'). \tag{1}$$

Unfortunately, without additional constraints, this formulation fails to learn pose properly [25]. The reason is that, given enough freedom, the encoder $\Phi(\boldsymbol{x})$ may simply decide to output a copy of the input image $\boldsymbol{x}$, which allows it to trivially satisfy constraint (1) without learning anything useful (this issue is visualized in section 4 and fig. 4). The formulation needs a mechanism to force the encoder Φ to 'distil' only pose information and discard appearance.

We make two key contributions to address these issues.

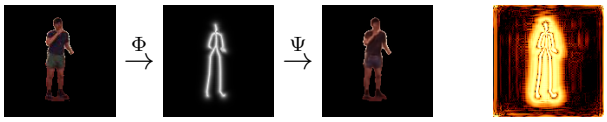


Figure 4. **Leaking appearance in the pose representation.** From left to right: input image $\boldsymbol{x}$, extracted skeleton image $\boldsymbol{y} = \Phi(\boldsymbol{x})$, and image reconstruction $\hat{\boldsymbol{x}} = \Psi(\Phi(\boldsymbol{x}))$. In principle, it should not be possible to reconstruct the full image from only the skeleton, but the function Φ can 'hide' the necessary information in a structured noise pattern, shown to the right as $\log \Phi(\boldsymbol{x})$.

First, we introduce a *dual representation of pose* as a vector of 2D keypoint coordinates and as a pictorial representation in the form of 'skeleton' image (section 3.1). We show that this dual representation provides a tight bottleneck that distils pose information effectively while making it possible to implement the auto-encoder (1) using powerful image-to-image translation networks.

Our second contribution is to introduce an *empirical prior* on the possible object poses (section 3.2). In this manner, we can constrain not just the individual pose samples $\boldsymbol{y}$, but their *distribution* $p(\boldsymbol{y})$ as well. In practice, the prior allows to use unpaired pose samples to improve accuracy and to learn an human-interpretable notion of pose that does not necessitate further learning to be used in applications.

3.1. Dual representation of pose & bottleneck

We consider a dual representation of the pose of an object as a vector of K 2D *keypoint coordinates* $\boldsymbol{p} = (p_1, \dots, p_K) \in \Omega^K$ and as an *image* $\boldsymbol{y} \in \mathbb{R}^\Omega$ containing a pictorial rendition of the pose as a skeleton (see fig. 2 for an illustration). Here the symbol $\Omega = \{1, \dots, H\} \times \{1, \dots, W\}$ denotes a grid of pixel coordinates.

Representing pose as a set of 2D keypoints provides a tight bottleneck that preserves geometry but discards appearance information. Representing pose as a skeleton image allows to implement the encoder and decoder networks as image translation networks. In particular, the image of the object $\boldsymbol{x}$ and of its skeleton $\boldsymbol{y}$ are *spatially aligned*, which makes it easier for a CNN to map between them.

Next, we show how to switch between the two representations of pose. We define the mapping $\boldsymbol{y} = \beta(\boldsymbol{p})$ from the coordinates $\boldsymbol{p}$ to the skeleton image $\boldsymbol{y}$ *analytically*. Let E be the set of keypoint pairs (i, j) connected by a skeleton edge and let $u \in \Omega$ be an image pixel. Then the skeleton image is given by:

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

The differentiable function $\boldsymbol{y} = \beta(\boldsymbol{p})$ defines a distance field from line segments that form the skeleton and applies an exponential fall off to generate an image. The visual effect is to produce a smooth line drawing of the skeleton. We also train an inverse function $\boldsymbol{p} = \eta(\boldsymbol{y})$, implementing it as a neural network regressor (see supplementary for details).

Given the two maps (η, β), we can use either representation of pose, as needed. In particular, by using the pictorial representation $\boldsymbol{y}$, the encoder/pose recogniser can be written as an image-to-image translation network $\Phi : \boldsymbol{x} \mapsto \boldsymbol{y}$ whose input $\boldsymbol{x} \in \mathbb{R}^{3\times H\times W}$ and output $\boldsymbol{y}$ are both images. The same is true for the conditional decoder $\Psi : (\boldsymbol{y}, \boldsymbol{x}') \mapsto \boldsymbol{x}$ of eq. (1).

While image-to-image translation is desirable architecturally, the downside of encoding pose as an image $\boldsymbol{y}$ is that it gives the encoder Φ an opportunity to 'cheat' and inject appearance information in the pose representation $\boldsymbol{y}$. We can prevent cheating by exploiting the coordinate representation of pose to filter out any hidden appearance information form $\boldsymbol{y}$. We do so by converting the pose image into keypoints and then back. This amounts to substituting $\boldsymbol{y} = \beta \circ \eta(\boldsymbol{y})$ in eq. (1), which yields the modified auto-encoding constraint:

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

3.2. Learning formulation & pose prior

Auto-encoding loss. In order to learn the auto-encoder (3), we use a dataset of N example pairs of video frames $\{(\boldsymbol{x}_i, \boldsymbol{x}'_i)\}_{i=1}^N$. Then the auto-encoding constraint (3) is enforced by optimizing a reconstruction loss. Here we use a *perceptual loss*:

$$\mathcal{L}_{\text{perc}} = \frac{1}{N}\sum_{i=1}^{N} \|\Gamma(\hat{\boldsymbol{x}}_i) - \Gamma(\boldsymbol{x}_i)\|_2^2, \tag{4}$$

where $\hat{\boldsymbol{x}}_i = \Psi(\beta \circ \eta \circ \Phi(\boldsymbol{x}_i), \boldsymbol{x}'_i)$ is the reconstructed image, Γ is a feature extractor. Instead of comparing pixels directly, the perceptual loss compares features extracted from a standard network such as VGG [5, 11, 15, 26], and leads to more robust training.

Pose prior. In addition to the N training image pairs $\{(\boldsymbol{x}_i, \boldsymbol{x}'_i)\}_{i=1}^N$, we also assume to have access to M sample poses $\{\bar{\boldsymbol{p}}_j\}_{j=1}^M$. Importantly, these sample poses are *unpaired*, in the sense that they are not annotations of the training images.

We use the unpaired pose samples to encourage the predicted poses $\boldsymbol{y}$ to be plausible. This is obtained by matching two distributions. The reference distribution $q(\boldsymbol{y})$ is given by the unpaired pose samples $\{\bar{\boldsymbol{y}}_j = \beta(\bar{\boldsymbol{p}}_j)\}_{j=1}^M$. The other distribution $p(\boldsymbol{y})$ is given by the pose samples $\{\boldsymbol{y}_i = \Phi(\boldsymbol{x}_i)\}_{i=1}^N$ predicted by the learned encoder network from the example video frames $\boldsymbol{x}_i$.

The goal is to match $p(\boldsymbol{y}) \approx q(\boldsymbol{y})$ in a distributional sense. This can be done by learning a discriminator network $D(\boldsymbol{y}) \in [0, 1]$ whose purpose is to discriminate between

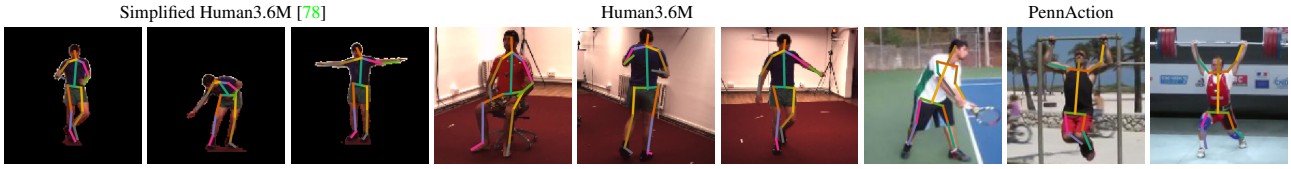


Figure 5. **Human pose predictions.** 2D keypoint predictions (visualised as connected limbs) on the simplified [78] (with no background), full Human3.6M [23], and PennAction [77] test sets. Proposed KeypointGAN directly predicts human landmarks in complex poses without any additional supervision. More samples are included in the supplementary.

the unpaired samples $\bar{\boldsymbol{y}}_j = \beta(\bar{\boldsymbol{p}}_j)$ and the predicted samples $\boldsymbol{y}_i = \Phi(\boldsymbol{x}_i)$. Samples are compared by means of the *difference adversarial loss* of [37]:

$$\mathcal{L}_{\text{disc}}(D) = \frac{1}{M}\sum_{j=1}^{M} D(\bar{\boldsymbol{y}}_j)^2 + \frac{1}{N}\sum_{i=1}^{N}(1 - D(\boldsymbol{y}_i))^2. \quad (5)$$

In addition to capturing plausible poses, the pose discriminator $D(\boldsymbol{y})$ *also* encourages the images $\boldsymbol{y}$ to be 'skeleton-like'. The effect is thus similar to the bottleneck introduced in section 3.1 and one may wonder if the discriminator makes the bottleneck redundant. The answer, as shown in sections 4 and 5, is negative: both are needed.

Overall learning formulation. Combining losses (4) and (5) yields the overall objective:

$$\mathcal{L}(\Phi, \Psi, D) = \lambda \mathcal{L}_{\text{disc}}(D, \Phi) + \mathcal{L}_{\text{perc}}(\Psi, \Phi), \quad (6)$$

where λ is a loss-balancing factor. The components of this model, *KeypointGAN*, and their relations are illustrated in fig. 2. Similar to any adversarial formulation, eq. (6) is minimized w.r.t. Φ, Ψ and maximised w.r.t. D.

Details. The functions Φ, Ψ, η and D are implemented as convolutional neural networks. The auto-encoder functions Φ and Ψ and the discriminator D are trained by optimizing the objective in eq. (6) (η is pre-trained using unpaired landmarks, for details see supplementary). Batches are formed by sampling random pairs of *video frames* $(\boldsymbol{x}_i, \boldsymbol{x}'_i)$ and unpaired pose $\bar{\boldsymbol{y}}_j$ samples. When sampling from *image datasets* (instead of videos), we generate image pairs as $(g_1(\boldsymbol{x}_i), g_2(\boldsymbol{x}_i))$ by applying random thin-plate-splines g_1, g_2 to training samples $\boldsymbol{x}_i$. All the networks are trained from scratch. Architectures and training details are in the supplementary.

4. Relation to image-to-image translation

Our method is related to unpaired image-to-image translation, of which CycleGAN [81] is perhaps the best example, but with two key differences: (a) it has a bottleneck (section 3.1) that prevents leaking appearance information into the pose representation $\boldsymbol{y}$, and (b) it reconstructs the image $\boldsymbol{x}$ conditioned on a second image $\boldsymbol{x}'$. We show in the experiments that these changes are critical for pose recognition performance, and conduct a further analysis here.

First, consider what happens if we drop both changes (a) and (b), thus making our formulation more similar to CycleGAN. In this case, eq. (1) reduces to $\boldsymbol{x} = \Psi(\Phi(\boldsymbol{x}))$. The trivial solution of setting both Φ and Ψ to the identity functions is only avoided due to the discriminator loss (5), which encourages $\boldsymbol{y} = \Phi(\boldsymbol{x})$ to look like a skeleton (rather than a copy of $\boldsymbol{x}$). In theory, then, this problem should be ill-posed as the pose $\boldsymbol{y}$ should not have sufficient information to recover the input image $\boldsymbol{x}$. However, the reconstructions from such a network still look reasonably good (see fig. 4). A closer look at logarithm of the generated skeleton $\boldsymbol{y}$, reveals that CycleGAN 'cheats' by leaking appearance information via subtle patterns in $\boldsymbol{y}$. By contrast, our bottleneck significantly limits leaking appearance in the pose image and thus its ability to reconstruct $\boldsymbol{x} = \Psi(\beta \circ \eta \circ \Phi(\boldsymbol{x}))$ from a single image; instead, reconstruction is achieved by injecting the missing appearance information via the auxiliary image $\boldsymbol{x}'$ using a conditional image decoder (eq. (3)).

5. Experiments

We evaluate our method, *KeypointGAN*, on the task of 2D landmark detection for human pose (section 5.1), faces (section 5.2), and cat heads (section 5.3) and outperform state-of-the-art methods (tables 1 to 3) on these tasks. We examine the relative contributions of components of our model in an ablation study (section 5.4). We study the effect of reducing the number of pose samples used in the empirical prior (section 5.5). Finally, we demonstrate image generation and manipulation conditioned on appearance and pose (section 5.6).

Evaluation. KeypointGAN directly outputs predictions for keypoints that are human-interpretable. In contrast, self-supervised methods [25, 35, 57, 58, 59, 71, 78] predict only *machine-interpretable* keypoints, as illustrated in fig. 1, and require at least some example images with paired keypoint annotations in order to learn to convert these landmarks to human-interpretable ones for benchmarking or for applications. We call this step *supervised post-processing*. KeypointGAN does not require this step, but we *also* include this result for a direct comparison with previous methods.

5.1. Human pose

Datasets. *Simplified Human3.6M* introduced by Zhang et al. [78] for evaluating unsupervised pose recognition, contains 6 activities in which human bodies are mostly upright; it comprises 800k training and 90k testing images. *Human3.6M* [23] is a large-scale dataset that contains 3.6M accurate 2D and 3D human pose annotations for 17 different activities, imaged under 4 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 [68]. *PennAction* [77] contains 2k challenging consumer videos of 15 sports categories. *MPI-INF-3DHP* [38] is a mocap dataset containing 8 subjects performing 8 activities in complex exercise poses. 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 skeleton prior respectively to ensure that the pose data does not contain labels corresponding to the training images. For the Human3.6M datasets we split the videos in half, while for PennAction we split in half the set of videos from each action category. We also evaluate the case when images and skeletons are sampled from different datasets and for this purpose we use the MPI-INF-3DHP mocap data.

Evaluation. We report 2D landmark detection performance on the simplified and original Human3.6M datasets. For Simplified Human3.6M, we follow the standard protocol of [78] and report the error for all 32 joints normalized by the image size. For Human3.6M, we instead report the mean error in pixels over 17 of the 32 joints [23]. To demonstrate learning from unpaired prior, we consider two settings for sourcing the images and the prior. In the first setting, we use *different* datasets for the two, and sample images from Human3.6M and poses from MPI-INF-3DHP. In the second setting, we use instead two *disjoint* parts of the *same* dataset Human3.6M for both images and poses. When using MPI-INF-3DHP dataset as the prior, we predict 28 joints, but use 17 joints that are common with Human3.6M for evaluation. We train KeypointGAN from scratch and compare its performance with both supervised and unsupervised methods.

Results. Table 1 reports the results on Simplified Human3.6M. As in previous self-supervised works [57, 78], we compare against the supervised baseline by Newell *et al*. [39]. Our model outperforms all the baselines [35, 57, 78] *without* the supervised post-processing used by the others.

Table 2 summarises our results on the original Human3.6M test set. Here we also compare against the supervised baseline [39] and the self-supervised method of [25]. Our model outperforms the baselines in this test too.

It may be surprising that KeypointGAN outperforms the supervised baseline. A possible reason is the limited number of supervised examples, which causes the supervised baseline to overfit. This can be noted by comparing the training / test errors: 14.61 / 19.52 for supervised hourglass and 13.79 / 14.46 for our method.

When poses are sampled from a different dataset (MPI-INF-3DHP) than the images (Human3.6M), the error is higher at 18.94 (but still better than the supervised alternative). This increase is due to the domain gap between the two datasets. Figure 5 shows some qualitative examples. Limitations of KeypointGAN are highlighted in fig. 6.

Method	all	wait	pose	greet	direct	discuss	walk
fully supervised							
Newell *et al*. [39]	**2.16**	**1.88**	**1.92**	**2.15**	**1.62**	**1.88**	**2.21**
self-supervised + supervised post-processing							
Thewlis *et al*. [57]	7.51	7.54	8.56	7.26	6.47	7.93	5.40
Zhang *et al*. [78]	4.14	5.01	4.61	4.76	4.45	4.91	4.61
Lorenz *et al*. [35]	2.79	—	—	—	—	—	—
self-supervised (no post-processing)							
KeypointGAN (ours)	**2.73**	**2.66**	**2.27**	**2.73**	**2.35**	**2.35**	**4.00**

Table 1. **Human landmark detection (Simplified H3.6M).** Comparison with state-of-the-art methods for human landmark detection on the Simplified Human3.6M dataset [78]. We report %-MSE normalised by image size for each activity.

Method	Human3.6M
fully supervised	
Newell *et al*. [39]	19.52
self-supervised + supervised post-processing	
Jakab & Gupta *et al*. [25]	19.12
self-supervised (no post-processing)	
KeypointGAN with *3DHP prior* (ours)	18.94
KeypointGAN with *H3.6M prior* (ours)	**14.46**

Table 2. **Human landmark detection (full H3.6M).** Comparison on Human3.6M test set with a supervised baseline Newell *et al*. [39], and a self-supervised method [25]. We report the MSE in pixels [23]. Results for each activity are in the supplementary.

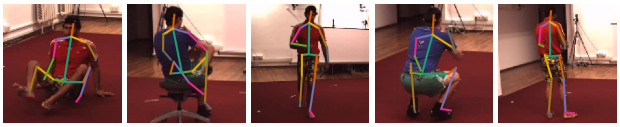


Figure 6. **Limitations. [1-2]** complex human poses like sitting are challenging to learn from a weak pose prior, **[3]** it could be difficult to disambiguate the sides due to bilateral symmetry, **[4-5]** occlusions are difficult to handle.

5.2. Human faces

Datasets. *VoxCeleb2 [10]* is a large-scale dataset consisting of 1M short clips of talking-head videos extracted from YouTube. *MultiPIE [54]* contains 68 labelled facial landmarks and 6k samples. We use this dataset as the only source

for the prior. *300-W [50]* is a challenging dataset of facial images obtained by combining multiple datasets [4, 45, 80] as described in [46, 57]. As in MultiPIE, 300-W contains 68 annotated facial landmarks. We use 300-W as our test dataset and follow the evaluation protocol in [46].

Results. As for human pose, we study a scenario where images and poses are sourced from a *different* datasets, using VoxCeleb2 and 300-W for the images, and MultiPIE (6k samples) for the poses (fig. 7). We train KeypointGAN from scratch using video frames from VoxCeleb2; then we fine-tune the model using our unsupervised method on the 300-W training images. We report performance on 300-W test set in table 3. KeypointGAN performs well even without any supervised fine-tuning on the target 300-W, and it already outperforms the unsupervised method of [58]. Adding supervised post-processing (on 300-W training set) as done in all self-supervised learning methods [57, 58, 59, 71], we outperform all except for [59] when they use their *HG* network that has 3 times more learnable parameters (4M vs 12M parameters). Interestingly we also outperform all supervised methods except [13, 72].

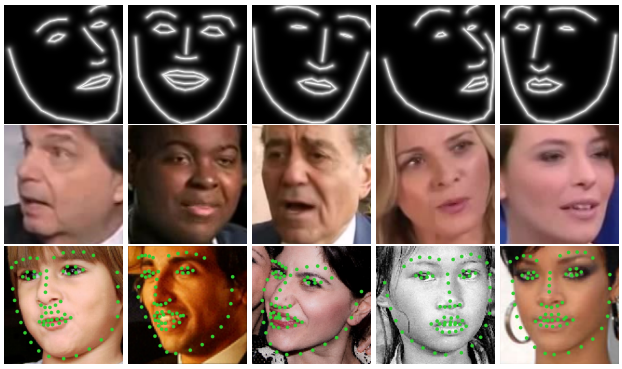


Figure 7. **Unpaired transfer.** We leverage approx. 6k landmarks from the MultiPIE dataset [54] as a prior **[top]** and unlabelled images from the the large-scale VoxCeleb2 [10] **[middle]** (1M clips, 6k identities) to train a detector that we test on the 300-W dataset [50] **[bottom]** (predictions in green) with state-of-the-art results (table 3). More qualitative results are in the supplementary.

5.3. Cat heads

Cat Head [76] dataset contains 9k images of cat heads each annotated with 7 landmarks. We use the same train and test split as [78]. We split the training set into two equally sized parts with no overlap. The first one is used to sample training images and the second one for the landmark prior. Our predictions are visualized in fig. 8.

5.4. Ablation study

As noted above, we can obtain our method by making the following changes to CycleGAN: (1) switching to a conditional image generator Ψ, (2) introducing the skeleton bottleneck $\beta \circ \eta$, and (3) removing the "second auto-encoder cycle" for the other domain (in our case the skeleton images). table 4 shows the effect of modifying CycleGAN in this manner on Simplified Human3.6M [78] for humans and on 300-W [50] for faces.

The baseline CycleGAN can be thought of as learning a mapping between images and skeletons via off-the-shelf image translation. Switching to a conditional image generator (1) does not improve the results because the model can still leak appearance information's pose. However, introducing the bottleneck (2) improves performance significantly for both humans (2.86% *vs.* 3.54% CycleGAN, a 20% error reduction) and faces (11.89% *vs.* 9.64% CycleGAN, a 19% error reduction). This also justifies the use of a conditional generator as the model fails to converge if the bottleneck is used without it. Removing the second cycle (3) leads to

Method	300-W
fully supervised	
LBF [46]	6.32
CFSS [82]	5.76
cGPRT [32]	5.71
DDN [75]	5.65
TCDCN [79]	5.54
RAR [72]	4.94
Wing Loss [13]	**4.04**
self-supervised + supervised post-processing	
Thewlis *et al.* [58]	9.30
Thewlis *et al.* [57]	7.97
Thewlis *et al.* [59] *SmallNet* †	5.75
Wiles *et al.* [71]	5.71
Jakab & Gupta *et al.* [25]	5.39
Thewlis *et al.* [59] *HourGlass* †	**4.65**
self-supervised	
KeypointGAN (ours w/o post-processing)	8.67
+ supervised post-processing	**5.12**

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 [79] (in %; ↓ is better) on the 300-W test set. †: [59] evaluate using two different networks: (1) *SmallNet* which we outperform, (2) *HourGlass* is not directly comparable due to much larger capacity (4M vs 12M parameters).

Figure 8. **Cat head landmarks.** Our predictions on Cat Head test set [76] consistently track landmarks across different views. More results are included in the supplementary.

further improvements, showing that this part is detrimental for our task.

Method	humans	faces
CycleGAN	3.54	11.89
+ conditional generator (1)	3.60	–
+ skeleton-bottleneck (2)	2.86	9.64
− 2nd cycle = **KeypointGAN** (ours) (3)	2.73	8.67
CycleGAN − 2nd cycle	3.39	11.36

Table 4. **Ablation study.** We start with the CycleGAN [81] model and sequentially augment it with — (1) conditional image generator (Ψ), (2) skeleton bottleneck ($\beta \circ \eta$), and (3) remove the second cycle-constraint resulting in our proposed KeypointGAN model. An auto-encoding model with a skeleton image as the intermediate representation (*i.e.* no keypoint bottleneck) and an adversarial loss is also reported (last row). We report 2D landmark detection error ($\downarrow$ is better) on the Simplified Human3.6M (section 5.1) for human pose, on the 300-W (section 5.2) for faces.

5.5. Unpaired sample efficiency

Table 5 demonstrates that KeypointGAN retains state-of-the-art performance even when we use only 50 unpaired landmark samples for the empirical prior. The experiment was done following the same protocols for training on face and human datasets as described previously.

# unpaired samples	humans	faces	
	no post-proc.	*no post-proc.*	*+ sup. post-proc.*
full dataset	2.73	8.67	5.12
5000	2.92 ± 0.05	–	–
500	3.30 ± 0.06	8.91 ± 0.15	5.22 ± 0.04
50	4.05 ± 0.02	8.92 ± 0.20	5.19 ± 0.06

Table 5. **Varying # of unpaired landmark samples.** We train KeypointGAN using varying numbers of samples for landmark prior. For faces, we sample the prior from MultiPIE dataset and evaluate on 300-W (section 5.2). For human pose, we sample the prior from the disjoint part of the Simplified Human3.6M training set and evaluate on the test set (section 5.1). We report the keypoint localisation error ($\pm\sigma$) (in %; $\downarrow$ is better). Full dataset has 6k unpaired samples for faces, and 400k for humans. Decreasing the number of unpaired landmark samples retains most of the performance.

5.6. Appearance and geometry factorization

The conditional image generator $\Psi : (\boldsymbol{y}^*, \boldsymbol{x}') \mapsto \hat{\boldsymbol{x}}$ of eq. (1) can also be used to produce novel images by combining pose and appearance from different images. Figure 9 shows that the model can be used to transfer the appearance of a human face identity on top of the pose of another. Though generating high quality images is not our primary goal, the ability to transfer appearance shows that KeypointGAN properly factorizes the latter from pose.

Figure 9. **Factorization of appearance and geometry.** *Reconstructed* image inherits appearance from the *style* image and geometry from the *target* image. **[left]:** human pose samples from Human3.6M. **[right]:** face samples from VoxCeleb2.

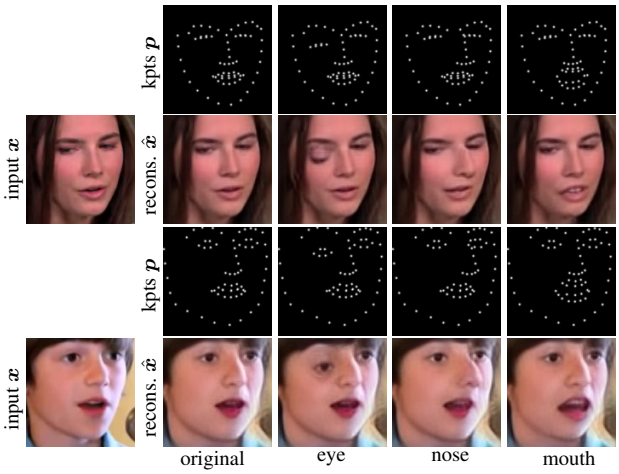


Figure 10. **Image editing using detected landmarks.** We show fine-grained control over the generated image by manipulating the coordinates of detected keypoints (*kpts*). The resulting changes are localised. Apart from demonstrating successful disentanglement of appearance and geometry, this also suggests that KeypointGAN assigns correct semantics to the detected landmarks.

This also demonstrates significant generalization over the training setting, as the system only learns from pairs of frames sampled from the same video and thus with same identity, but it can swap different identities. In fig. 10, we further leverage the disentanglement of geometry and appearance to manipulate a face by editing its keypoints.

6. Conclusion

We have shown that combining conditional image generation with a dual representation of pose with a tight geometric bottleneck can be used to learn to recognize the pose of complex objects such as humans without providing any labelled image to the system. In order to do so, KeypointGAN makes use of an unpaired pose prior, which also allows it to output

human-interpretable pose parameters. With this, we have achieved optimal landmark detection accuracy for methods that do not use labelled images for training.

Acknowledgements. We are grateful for the support of ERC 638009-IDIU, and the Clarendon Fund Scholarship. We would like to thank Triantafyllos Afouras, Relja Arandjelović, and Chuhan Zhang for helpful advice.

References

[1] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Proc. CVPR*, pages 1014–1021. IEEE, 2009. 2

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

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

[4] Peter N Belhumeur, David W Jacobs, David J Kriegman, and Neeraj Kumar. Localizing parts of faces using a consensus of exemplars. *TPAMI*, 35(12):2930–2940, 2013. 7

[5] Joan Bruna, Pablo Sprechmann, and Yann LeCun. Super-resolution with deep convolutional sufficient statistics. In *Proc. ICLR*, 2016. 4

[6] Adrian Bulat and Georgios Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *Proc. ECCV*, pages 717–732. Springer, 2016. 2

[7] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proc. CVPR*, pages 7291–7299, 2017. 2

[8] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *Proc. CVPR*, pages 4733–4742, 2016. 2

[9] Xianjie Chen and Alan L Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Proc. NIPS*, pages 1736–1744, 2014. 2

[10] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. *arXiv preprint arXiv:1806.05622*, 2018. 2, 6, 7

[11] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems*, pages 658–666, 2016. 4

[12] Pedro F Felzenszwalb and Daniel P Huttenlocher. Pictorial structures for object recognition. *International journal of computer vision*, 61(1):55–79, 2005. 2

[13] Zhen-Hua Feng, Josef Kittler, Muhammad Awais, Patrik Huber, and Xiao-Jun Wu. Wing loss for robust facial landmark localisation with convolutional neural networks. In *Proc. CVPR*, 2018. 7

[14] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015. 3

[15] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proc. CVPR*, pages 2414–2423, 2016. 4

[16] Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou. GANFIT: Generative adversarial network fitting for high fidelity 3D face reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[17] Zhenglin Geng, Chen Cao, and Sergey Tulyakov. 3D guided fine-grained face manipulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[18] Thomas Gerig, Andreas Morel-Forster, Clemens Blumer, Bernhard Egger, Marcel Luthi, Sandro Schönborn, and Thomas Vetter. Morphable face models - an open framework. In *Proc. Automatic Face & Gesture Recognition*, 2018. 2

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. NIPS*, pages 2672–2680, 2014. 3

[20] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Learning to read by spelling: Towards unsupervised text recognition. In *Proc. ICVGIP*, 2018. 3

[21] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017. 3

[22] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *Proc. ECCV*, pages 34–50. Springer, 2016. 2

[23] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *TPAMI*, 36(7):1325–1339, jul 2014. 2, 5, 6, 12

[24] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proc. CVPR*, 2017. 3

[25] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *Proc. NIPS*, 2018. 1, 2, 3, 5, 6, 7, 12, 19, 20, 21

[26] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proc. ECCV*, pages 694–711. Springer, 2016. 4

[27] Sam Johnson and Mark Everingham. Learning effective human pose estimation from inaccurate annotation. In *Proc. CVPR*, pages 1465–1472. IEEE, 2011. 2

[28] Angjoo Kanazawa, David W Jacobs, and Manmohan Chandraker. Warpnet: Weakly supervised matching for single-view reconstruction. In *Proc. CVPR*, pages 3253–3261, 2016. 2

[29] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proc. CVPR*, 2018. 2

[30] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 21

[31] Diederik P Kingma and Jimmy Ba. Adam: A method for

stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 19

[32] Donghoon Lee, Hyunsin Park, and Chang D Yoo. Face alignment using cascade gaussian process regression trees. In *Proc. CVPR*, 2015. 7

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proc. ECCV*, pages 740–755. Springer, 2014. 2

[34] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):248, 2015. 2

[35] Dominik Lorenz, Leonard Bereska, Timo Milbich, and Bjorn Ommer. Unsupervised part-based disentangling of object shape and appearance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10955–10964, 2019. 1, 3, 5, 6

[36] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013. 21

[37] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proc. ICCV*, pages 2794–2802, 2017. 5

[38] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian 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. 2, 6

[39] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Proc. ECCV*, 2016. 2, 6, 12

[40] Wanli Ouyang, Xiao Chu, and Xiaogang Wang. Multi-source deep learning for human pose estimation. In *Proc. CVPR*, pages 2329–2336, 2014. 2

[41] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *The IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2009. 2

[42] Tomas Pfister, James Charles, and Andrew Zisserman. Flowing convnets for human pose estimation in videos. In *Proc. CVPR*, pages 1913–1921, 2015. 2

[43] Leonid Pishchulin, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Poselet conditioned pictorial structures. In *Proc. CVPR*, pages 588–595, 2013. 2

[44] Varun Ramakrishna, Daniel Munoz, Martial Hebert, James Andrew Bagnell, and Yaser Sheikh. Pose machines: Articulated pose estimation via inference machines. In *Proc. ECCV*, pages 33–47. Springer, 2014. 2

[45] Deva Ramanan and Xiangxin Zhu. Face detection, pose estimation, and landmark localization in the wild. In *Proc. CVPR*, 2012. 7

[46] Shaoqing Ren, Xudong Cao, Yichen Wei, and Jian Sun. Face alignment at 3000 fps via regressing local binary features. In *Proc. CVPR*, 2014. 7

[47] Helge Rhodin, Mathieu Salzmann, and Pascal Fua. Unsupervised geometry-aware representation for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 750–767, 2018. 3

[48] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *Proc. CVPR*, volume 2, 2017. 2

[49] Matteo Ruggero Ronchi, Oisin Mac Aodha, Robert Eng, and Pietro Perona. It's all relative: Monocular 3d human pose estimation from weakly supervised data. In *BMVC*, 2018. 2

[50] Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: Database and results. *Image and vision computing*, 47:3–18, 2016. 2, 7

[51] Benjamin Sapp, Chris Jordan, and Ben Taskar. Adaptive pose priors for pictorial structures. In *Proc. CVPR*, pages 422–429, 2010. 2

[52] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D. Castillo, and David Jacobs. Sfsnet: Learning shape, refectance and illuminance of faces in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[53] Zhixin Shu, Mihir Sahasrabudhe, Alp Guler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos. Deforming autoencoders: Unsupervised disentangling of shape and appearance. In *Proceedings of the European Conference on Computer Vision*, 2018. 2, 3

[54] Terence Sim, Simon Baker, and Maan 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. 2, 6, 7

[55] Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. Finegan: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6490–6499, 2019. 3

[56] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proc. ECCV*, pages 712–729. Springer, 2018. 2

[57] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *Proc. ICCV*, 2017. 1, 2, 5, 6, 7

[58] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object frames by dense equivariant image labelling. In *Proc. NIPS*, 2017. 2, 5, 7

[59] James Thewlis, Samuel Albanie, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of landmarks by descriptor vector exchange. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6361–6371, 2019. 2, 5, 7

[60] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proc. CVPR*, pages 648–656, 2015. 2

[61] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph 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. 2

[62] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proc. CVPR*, pages 1653–1660, 2014. 2

[63] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1415–1424, 2017. 3

[64] Hsiao-Yu Fish Tung, Adam W Harley, William Seto, and Katerina Fragkiadaki. Adversarial inverse graphics networks: Learning 2d-to-3d lifting and image-to-image translation from unpaired supervision. In *Proc. ICCV*, volume 2, 2017. 2

[65] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proc. CVPR*, pages 4068–4076, 2015. 3

[66] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proc. CVPR*, 2017. 3

[67] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 21

[68] Ruben Villegas, Jimei Yang, Yuliang Zou, Sungryull Sohn, Xunyu Lin, and Honglak Lee. Learning to generate long-term future via hierarchical prediction. In *Proc. ICML*, 2017. 6

[69] Mengjiao Wang, Zhixin Shu, Shiyang Cheng, Yannis Panagakis, Dimitris Samaras, and Stefanos Zafeiriou. An adversarial neuro-tensorial approach for learning disentangled representations. *International Journal of Computer Vision*, 2019. 2

[70] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proc. CVPR*, pages 4724–4732, 2016. 2

[71] Olivia Wiles, A Koepke, and Andrew Zisserman. Self-supervised learning of a facial attribute embedding from video. In *Proc. BMVC*, 2018. 2, 3, 5, 7

[72] Shengtao Xiao, Jiashi Feng, Junliang Xing, Hanjiang Lai, Shuicheng Yan, and Ashraf Kassim. Robust facial landmark detection via recurrent attentive-refinement networks. In *Proc. ECCV*, 2016. 7

[73] Wei Yang, Wanli Ouyang, Xiaolong Wang, Jimmy Ren, Hongsheng Li, and Xiaogang Wang. 3d human pose estimation in the wild by adversarial learning. In *Proc. CVPR*, volume 1, 2018. 2

[74] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Proc. CVPR*, pages 1385–1392. IEEE, 2011. 2

[75] Xiang Yu, Feng Zhou, and Manmohan Chandraker. Deep deformation network for object landmark localization. In *Proc. ECCV*. Springer, 2016. 7

[76] Weiwei Zhang, Jian Sun, and Xiaoou Tang. Cat head detection-how to effectively exploit shape and texture features. In *European Conference on Computer Vision*, pages 802–816. Springer, 2008. 7

[77] Weiyu Zhang, Menglong Zhu, and Konstantinos 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. 5, 6

[78] Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee. Unsupervised discovery of object landmarks as structural representations. In *Proc. CVPR*, pages 2694–2703, 2018. 1, 2, 3, 5, 6, 7

[79] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Learning deep representation for face alignment with auxiliary attributes. *TPAMI*, 38(5):918–930, 2016. 7

[80] Erjin Zhou, Haoqiang Fan, Zhimin Cao, Yuning Jiang, and Qi 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. 7

[81] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. CVPR*, 2018. 2, 3, 5, 8, 19, 21

[82] Shizhan Zhu, Cheng Li, Chen Change Loy, and Xiaoou Tang. Face alignment by coarse-to-fine shape searching. In *Proc. CVPR*, 2015. 7

Appendix

This supplementary material provides further technical details, illustrations and analysis. We provide detailed quantitative evaluation on Human3.6M dataset (appendix A), extended versions of our qualitative results on factorization of appearance and geometry (appendix B), facial landmarks detection (appendix C), human pose estimation (appendix D), and cat head landmarks detection (appendix E). Finally, we give further implementation details in appendix F.

A. Human3.6M detailed results

We report the performance for each activity of the Human3.6M test set in table 6. We evaluated the performance on every 60th frame of the video sequences.

Method	all	wait	pose	greet	direct	discuss	walk	eat	phone	purchase	sit	sit down	smoke	take photo	walk dog	walk together
fully supervised																
Newell *et al*. [39]	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
self-supervised + supervised post-processing																
Jakab *et al*. [25]	19.12	16.63	15.01	16.68	14.73	15.69	17.74	16.53	23.27	17.35	24.66	33.14	20.31	20.96	**17.77**	16.31
self-supervised (no post-processing)																
Ours *3DHP prior*	18.94	15.33	14.37	16.08	15.90	17.24	14.51	17.30	19.66	17.39	22.79	30.84	18.50	24.21	23.77	16.16
Ours *H3.6M prior*	**14.46**	**11.40**	**10.39**	**11.85**	**11.26**	**13.72**	**11.85**	**12.02**	**14.42**	**12.90**	**17.01**	**25.71**	**14.35**	**18.67**	19.42	**11.90**

Table 6. **Human landmark detection (full H3.6M).** Comparison on Human3.6M test set with a supervised baseline Newell *et al*. [39], and a self-supervised method [25]. We report the MSE in pixels [23] for each activity. We highlight the minimum error across all models in bold.

B. Appearance and geometry factorization

Figure 11. **Factorization of appearance and geometry.** We supply different identities for *style* and *target* input images. *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 VoxCeleb2.

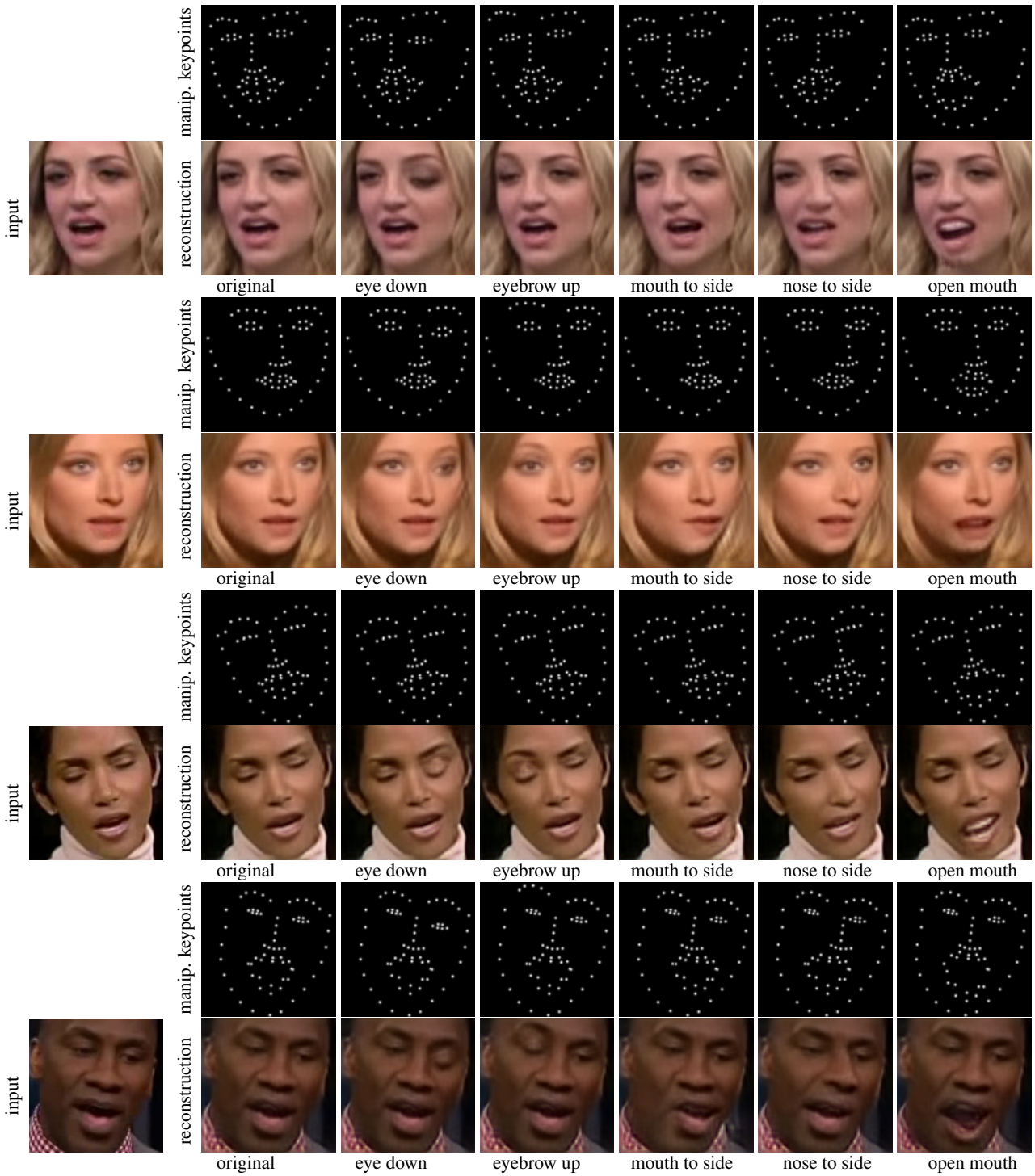


Figure 12. **Image editing using detected landmarks.** We show fine-grained control over the generated image by manipulating the coordinates of detected landmarks (*manip. keypoints*). For example, we pick landmarks corresponding to an eye and move them down [second column], or open the mouth [last column] (note, the generator fills in the teeth absent in the input images). The resulting changes are localized 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.

C. Facial landmarks detections

Figure 13. **Facial landmark detections on 300-W.** Randomly sampled predictions from 300-W test set. The model was trained with unlabelled images from VoxCeleb2 face videos dataset and unpaired landmarks sampled from MultiPIE dataset, hence shows significant generalization. Green markers denote our detections, blue correspond to the ground truth.

D. Human pose estimation

D.1. Pose detection on Human3.6M

Figure 14. **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.

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

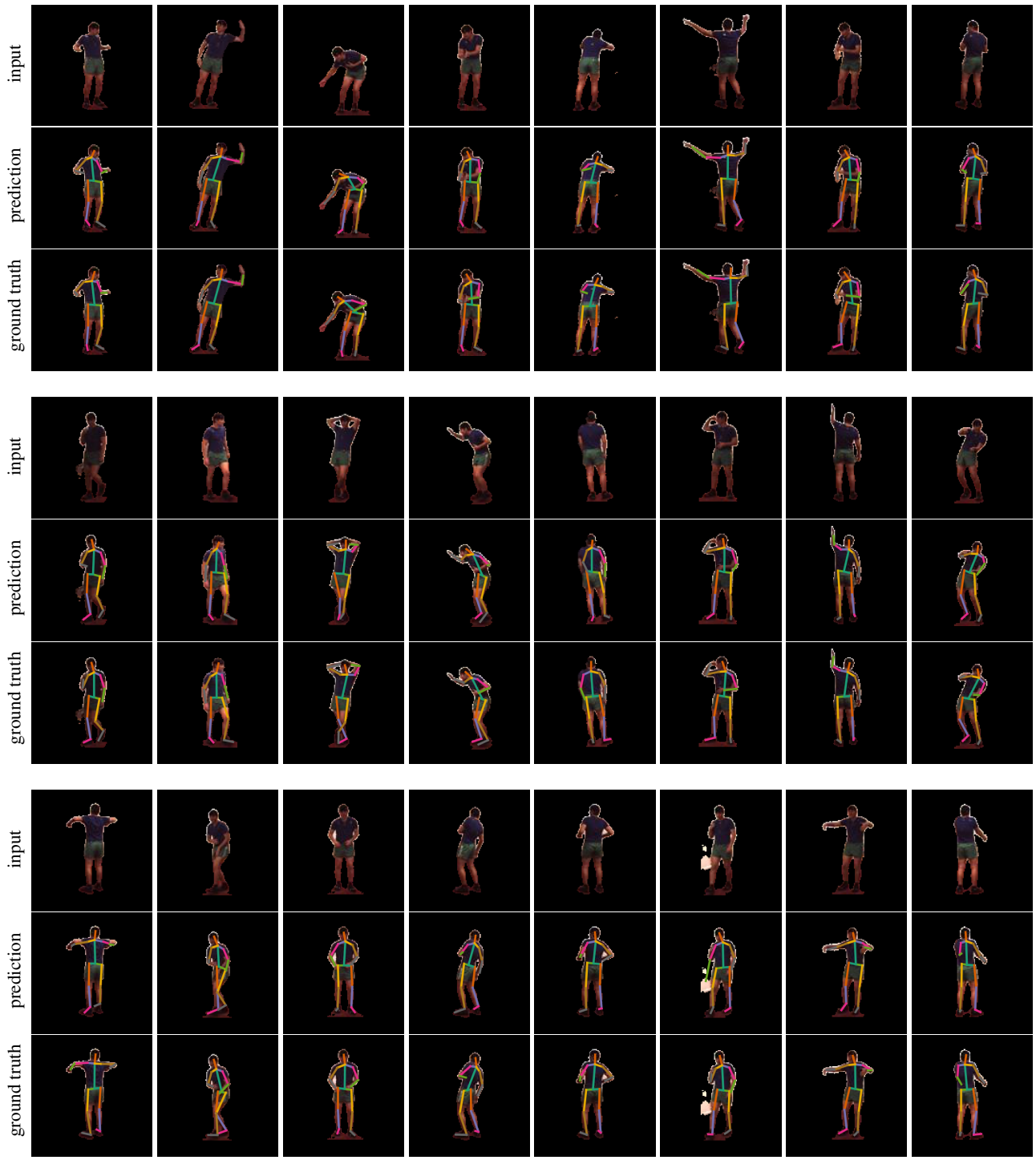


Figure 15. **Pose estimation on the Simplified Human3.6M.** Randomly sampled results from the Simplified Human3.6M test set. The model is trained with unpaired images and skeletons from Simplified Human3.6M.

E. Landmarks detection on Cat Heads

Figure 16. **Landmark detections on Cat Head.** Randomly sampled predictions on Cat Head test set.

F. Implementation details

F.1. Training details

The auto-encoder functions Φ and Ψ and the discriminator D are trained by optimizing the overall objective in eq. (5) of the main paper while setting $\lambda = 10$ (η is pre-trained using unpaired landmarks as detailed below). We use the Adam optimiser [31] with a learning rate of $2 \cdot 10^{-4}$, $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The batch size is set to 16 and the norm of the gradients is clipped to 1.0 for stability.

F.2. Pre-training the function η

The network η mapping the skeleton image $\boldsymbol{y}$ to its corresponding keypoint locations $\boldsymbol{p}$ is pre-trained before optimizing the overall objective (eq. (5) of the main paper). This is done by using the unpaired pose samples $\{\bar{\boldsymbol{p}}_j\}_{j=1}^M$ and by optimizing the loss $\frac{1}{M}\sum_{j=1}^M \mathcal{L}(\eta|\bar{\boldsymbol{p}}_j)$ where

$$\mathcal{L}(\eta|\bar{\boldsymbol{p}}) = \|\eta \circ \beta(\bar{\boldsymbol{p}}) - \bar{\boldsymbol{p}}\|^2 \tag{7}$$

is a simple ℓ^2 regression loss.

During the optimization of the overall objective, the function η is further fine-tuned by minimizing the same loss plus eq. (7) an additional term $\mathcal{L}(\eta|\boldsymbol{y}) = \lambda'\|\beta \circ \eta(\boldsymbol{y}) - \boldsymbol{y}\|^2$, where $\boldsymbol{y}$ is a reconstructed pose (see fig. 2 of the main paper). The latter ensures that network η works for poses that appear in the videos but not necessarily in the pose prior. The two terms are balanced by the coefficient λ'. After fine-tuning η, we noticed that it loses some of its ability to distinguish between frontal and dorsal views of human body (which is fairly ambiguous given only a skeleton image as input). We correct its predictions by using the pre-trained version of η at eq. (7) to determine the orientation of human body.

The function η is designed as a neural network that converts the skeleton image $\boldsymbol{y}$ into K heatmaps. The locations of keypoints are further obtained as in [25] by converting each heatmap into a 2D probability distribution. The expectation of this probability distribution corresponds to the location of the keypoints. The spatial coordinates are normalised to the $[-1, 1]$ range and we set $\gamma = \frac{1}{0.04}$ in eq. (2) of the main paper. The function is learned by minimizing the loss introduced above with $\lambda' = 0.1$.

F.3. Note on a second cycle constraint and discriminator

Standard CycleGAN [81] 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 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. Our ablation study shows that the additional cycle constraint and discriminator leads to worse performance, so we do not include them in our final version of the model.

F.4. Architectures

Figures 17 to 21 provide detailed descriptions of network architectures used in experiments.

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 17. **Image encoder Φ.** The network is based of the encoder and decoder network from [25]. Arrows on the side denote skip connections 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	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 [25].

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 [25]. 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 ($\tilde{\boldsymbol{y}}$ 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 [81]. LReLU stands for Leaky Rectified Linear Unit [36] that is used with 0.2 negative slope. Instance normalization [67] is used before every activation. 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.

Type	Kernel	Stride	Output ch.	Output size	Norm.	Activ.
Input ($\Psi(\tilde{\boldsymbol{y}}, \boldsymbol{x}')$ 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 [36]. We set the negative slope to 0.2. Every activation is preceded by instance normalization [67]. The architecture is loosely based on [30].

Chapter 5

KeypointDeformer: Unsupervised 3D Keypoint Discovery for Shape Control

KeypointDeformer: Unsupervised 3D Keypoint Discovery for Shape Control

Tomas Jakab[1,4*], Richard Tucker[4], Ameesh Makadia[4], Jiajun Wu[3], Noah Snavely[4], Angjoo Kanazawa[2,4]
[1]University of Oxford, [2]UC Berkeley, [3]Stanford University, [4]Google Research

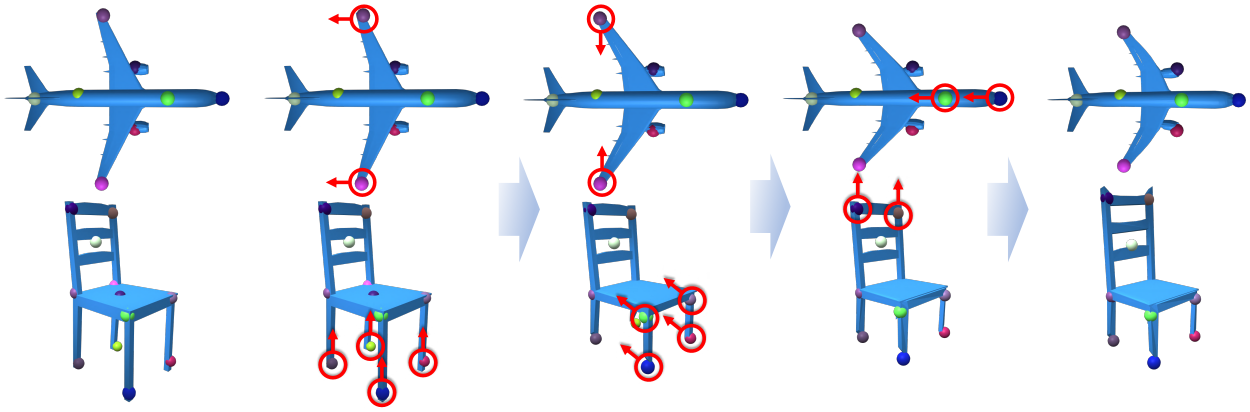


Figure 1: **Controlling shape deformation with unsupervised 3D keypoints.** We discover unsupervised 3D keypoints that allow for intuitive control of an object's shape. This figure shows individual steps of interactive control. The red arrows illustrate the direction in which the keypoints are manipulated. Note that the resulting deformations are localized and object parts are deformed in an intuitive manner—*e.g.* moving keypoints at the tip of the wings backward moves the wings backwards—all while preserving the details of the original shape.

Abstract

We introduce KeypointDeformer, a novel unsupervised method for shape control through automatically discovered 3D keypoints. We cast this as the problem of aligning a source 3D object to a target 3D object from the same object category. Our method analyzes the difference between the shapes of the two objects by comparing their latent representations. This latent representation is in the form of 3D keypoints that are learned in an unsupervised way. The difference between the 3D keypoints of the source and the target objects then informs the shape deformation algorithm that deforms the source object into the target object. The whole model is learned end-to-end and simultaneously discovers 3D keypoints while learning to use them for deforming object shapes. Our approach produces intuitive and semantically consistent control of shape deformations. Moreover, our discovered 3D keypoints are consistent across object category instances despite large shape variations. As our method is unsupervised, it can be readily deployed to new object categories without requiring annotations for 3D keypoints and deformations. Project page: http://tomasjakab.github.io/KeypointDeformer.

1. Introduction

Given the vast number of 3D shapes available on the Internet, providing users with intuitive and simple interfaces for semantically manipulating objects while preserving their key shape properties has a wide variety of applications in AI-assisted 3D content creation. In this paper, we propose to automatically discover intuitive and semantically meaningful control points for interactive editing, enabling detail-preserving shape deformation for object categories.

Specifically, we identify 3D keypoints as an intuitive and simple interface for shape editing. Keypoints are sparse 3D points that are semantically consistent across an object category. We propose a learning framework for unsupervised discovery of such keypoints and a deformation model that uses the keypoints to deform a shape while preserving local shape detail. We call our model *KeypointDeformer*.

Figure 1 describes the inference-time use case of KeypointDeformer. Given a novel shape, KeypointDeformer predicts 3D keypoints on the surface. If a user manipulates a keypoint on a chair leg upwards, the entire leg is deformed in the same direction (bottom). Our approach optionally enables the use of a categorical deformation prior on these

* Work done while interning at Google Research.

edits, such that if a user moves one side of an airplane wing backwards, the opposite side of the wing is deformed symmetrically in the same direction (top)—while if the user wishes to only move one side of the wing, our approach also allows this. Our framework enables stand-alone shape edits or shape alignment between two shapes, and can also synthesize novel variations of shapes for amplifying stock datasets.

While 3D keypoints may be a good proxy for shape editing, obtaining explicit supervision for keypoints and deformation models is not only expensive but also ill-defined. As such, we propose an unsupervised framework for jointly discovering the keypoints and the deformation model. To solve our problem, we devise two components that operate in concert: (1) a method for discovering and detecting keypoints, and (2) a deformation model that propagates keypoint displacements to the rest of the shape. To achieve these, we set up a proxy learning task where the goal is to align a source shape with a target shape, where the two can represent very different instances of a category. We also propose a simple yet effective keypoint regularizer that encourages learning of semantically consistent keypoints that are well-distributed, lie close to the object surface and implicitly preserve underlying shape symmetries. The result of our training approach is a deformation model that deforms a shape based on automatically discovered 3D control keypoints. Since the keypoints are low-dimensional, we can further learn a category prior on these keypoints, enabling semantic shape editing from sparse user inputs.

Overall, our method has following key benefits:

1. It gives users an intuitive and simple way to interactively control object shapes.
2. Both the keypoint prediction and deformation model are unsupervised.
3. We show that keypoints discovered by our method are better for shape control than other kinds of keypoints, including manually annotated ones.
4. Our unsupervised 3D keypoints are semantically consistent across object instances of the same category giving us sparse correspondences.

We evaluate the semantic consistency of our unsupervised 3D keypoints on standard benchmarks, and achieve state-of-the-art results among unsupervised methods. We also demonstrate the suitability of our keypoints for shape deformation. Finally, we provide qualitative results of user-guided interactive shape control, and include videos of interactive shape control on our project page.

2. Related Work

Shape deformation. Our approach is closely related to detail-preserving deformations studied in geometric modeling, including Laplacian-based shape editing [22], As-Rigid-As-Possible shape deformation [23], and cages [13]. While these approaches enable shape editing via many forms of user-specified constraints (e.g., points or sets in an optimization framework), a major challenge is that they rely purely on geometric properties and do not consider semantic attributes or category-specific shape priors for deformation. Such priors can be obtained from artists painting the object surface with stiffness properties [1] or learned from a set of meshes with known correspondence [20]. However, such supervisions are prohibitively expensive to obtain and are not applicable to novel shapes. Yumer *et al*. [33] address this issue in a data-driven framework that provides a set of sliders that control the attributes of a given shape. However, this approach requires a set of predefined attributes obtained from expert annotations. We propose an unsupervised approach, and provide users with direct semantic deformation handles in the form of keypoints. Furthermore, our formulation can incorporate a category-specific deformation basis on the discovered 3D keypoints, allowing for semantically consistent user edits from sparse keypoints edits (such that if one side of an airplane wing is extended, the other opposite side also extends).

Another related problem is deformation transfer [24], which transfers the deformation exhibited by a source mesh onto a target mesh via known correspondences between shapes. Recent approaches employ deep learning to implicitly learn the shape correspondences to align two shapes [31, 8, 29]. While we also use a shape alignment objective to train our framework, we make our intermediate control explicit in the form of keypoints, which allows for stand-alone shape editing. In contrast, prior approaches always require a target shape to express the desired deformation.

User-guided shape editing. Our approach is related to recent deep learning–based methods that learn generative models of shapes for interactive editing. Tulsiani *et al*. [28] learn to abstract shapes in terms of primitives, which can be used to edit the shape by transferring primitive deformations to the surface. However, shape editing is not their primary focus, and it is unclear how well the direct transfer of primitive transformations preserve local shape detail. Recent approaches take this idea further by learning a generative model of primitives in the form of set of point-based primitives [9], shape handles [5], or disconnected shape manifolds [19]. These methods enable interactive editing by searching for latent primitive representations that best match user edits. However, they require an involved user interface via sketching or directly manipulating the underlying set of primitives. Most critically, as the edits are based on generative models, these approaches may change the local details of the original shape. In contrast, we directly deform the source shape, leading to better preservation of shape detail.

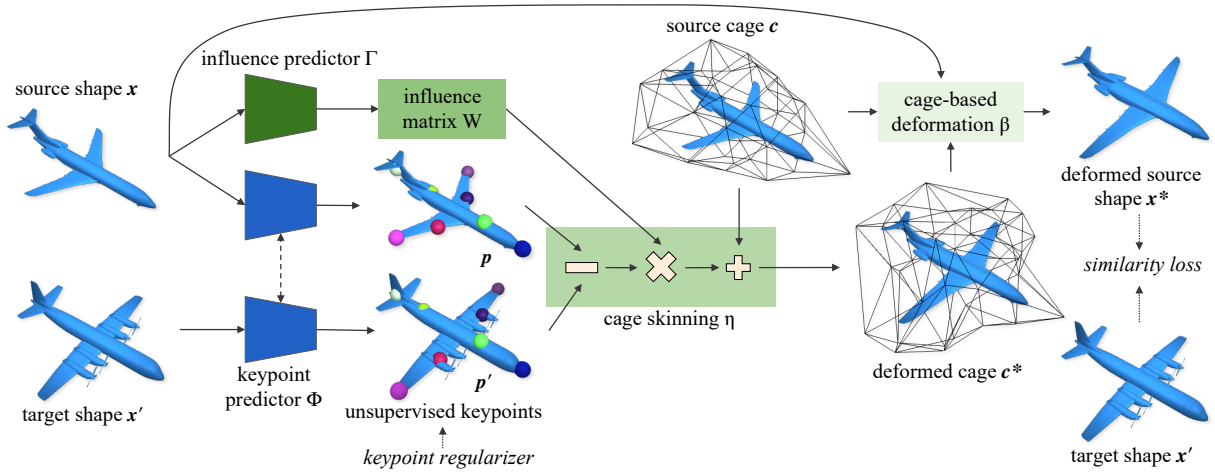


Figure 2: **Model.** Our model aligns the source shape $\boldsymbol{x}$ with the target shape $\boldsymbol{x}'$ using predicted unsupervised keypoints $\boldsymbol{p}$ and $\boldsymbol{p}'$. The unsupervised keypoints describe the objects pose and work as control points for the deformation. The model is trained end-to-end using a similarity loss between the deformed source shape $\boldsymbol{x}^*$ and the target shape $\boldsymbol{x}'$, as well as a keypoint regularization loss. During interactive shape manipulation at test time, a user can choose to input only the source shape $\boldsymbol{x}$ that the keypoint predictor Φ uses to estimate a set of unsupervised keypoints $\boldsymbol{p}$. The user can then manually control the keypoints $\boldsymbol{p}$ obtaining $\boldsymbol{p}'$ target keypoints that are fed into the deformation model to produce the deformed source shape $\boldsymbol{x}^*$ as demonstrated in Figure 1, Figure 9 and in the supplementary videos on our project page.

We qualitatively compare our approach to DualSDF [9] to illustrate this benefit.

Unsupervised keypoints. While the problem of unsupervised keypoint discovery is well studied in 2D [27, 34, 14, 10, 26, 11], this problem is relatively under-explored in 3D. Suwajanakorn *et al*. [25] detect 3D keypoints from a single image using 3D pose information as supervision. Here we focus on learning 3D keypoints on 3D shapes. Chen *et al*. [3] output a structured 3D representation to obtain sparse or dense shape correspondences. Closest to our approach in terms of 3D keypoint discovery is that of Fernandez *et al*. [4], which impose explicit symmetric constraints. In this work, we discover unsupervised keypoints for the purpose of shape control. While we focus on shape editing, our formulation results in state-of-the-art 3D keypoints for semantic consistency. Such unsupervised keypoints may be useful for robotics applications that use 3D keypoints as a latent representation for control [17, 6], and which currently require manually defined 3D keypoints as supervision.

3. Method

Our aim is to learn a keypoint predictor $\Phi : \boldsymbol{x} \rightarrow \boldsymbol{p}$ that maps a 3D object shape $\boldsymbol{x}$ to a sparse set of semantically consistent 3D keypoints $\boldsymbol{p}$. We also want to learn a conditional deformation model on keypoints $\Psi : (\boldsymbol{x}, \boldsymbol{p}, \boldsymbol{p}') \rightarrow \boldsymbol{x}'$ that deforms the shape $\boldsymbol{x}$ in accordance to the deformed control keypoints, where $\boldsymbol{p}$ describes the initial (source) keypoint locations and $\boldsymbol{p}'$ the target locations. Obtaining explicit supervision for keypoints and the deformation model is expensive and ill-defined. As such, we propose an unsupervised learning framework for training these functions. We do so by designing an auxiliary task of pair-wise shape alignment, where the key idea is to jointly learn keypoints and a deformation model that can bring two random shapes into alignment. Specifically, our model first predicts keypoint locations on the source and target shapes using a Siamese network. We then deform the source shape according to the correspondence provided by the discovered keypoints. In order to preserve local shape detail, we employ a cage-based deformation method, conditioned on keypoints. We devise a novel and highly effective, yet simple, keypoint regularization term that encourages keypoints to be well-distributed and lie close to the object surface. Figure 2 provides a schematic illustration of our framework.

3.1. Shape Deformation with Keypoints

We first predict keypoints from source and target meshes by representing each object as a point cloud $\boldsymbol{x} \in \mathbb{R}^{3\times N}$, uniformly sampled from the object mesh. The keypoint predictor Φ takes the shape as an input $\boldsymbol{x}$ and outputs an ordered set of 3D keypoints $\boldsymbol{p} = (p_1, \ldots, p_K) \in \mathbb{R}^{3\times K}$. The encoder is shared for both the source and target in a Siamese architecture. The shape deformation function Ψ takes the source shape $\boldsymbol{x}$ represented as a point cloud $\boldsymbol{x}$ as well as source keypoints $\boldsymbol{p}$ and target keypoints $\boldsymbol{p}'$. The keypoints $\boldsymbol{p}$ and $\boldsymbol{p}'$ are estimated by the keypoint predictor Φ. At test time, the user can input their own target keypoints $\boldsymbol{p}'$ for interactive shape deformation as illustrated in Figure 2.

In order to deform the object shape in a manner that pre-

serves its local shape detail, we use the recently introduced differentiable cage-based deformation algorithm [31]. Cages are a classical shape modeling method [13, 12, 15] that use a coarse enclosing mesh that is associated with the shape. Deforming the cage mesh results in an interpolated deformation of the enclosed shape. The cage-based deformation function $\beta : (\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{c}^*) \rightarrow \boldsymbol{x}^*$ takes a source control cage $\boldsymbol{c}$ and a deformed control cage $\boldsymbol{c}^*$, and deforms the input shape $\boldsymbol{x}$ that is in the form of a mesh or a point cloud. We automatically obtain the source cage $\boldsymbol{c}$ for each shape by starting with a spherical shape that is larger than the source shape $\boldsymbol{x}$ and iteratively pulling each of the cage vertices $\boldsymbol{c}_V$ towards the centre of the object until it is within a small distance from the object surface. The resulting cages are illustrated in Figure 2. While cages are a reliable method for shape-preserving deformation, modifying cages to achieve a desired deformation is not necessarily intuitive, particularly to novice users, because the cage vertices do not lie on the surface, do not have a coarse structure, and are not semantically consistent across different shapes. We propose keypoints as an intuitive handle to manipulate the cages.

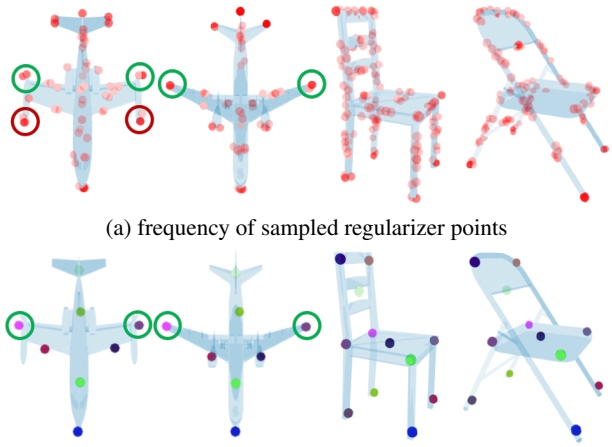
In order to control the object deformation using our discovered keypoints, we need to associate them with the cage vertices. We do so with a linear skinning function that takes the relative differences between the source and target keypoints $\Delta\boldsymbol{p} = \boldsymbol{p}' - \boldsymbol{p}$ and associates each of them with the source cage vertices $\boldsymbol{c}_V$ using an influence matrix $W \in \mathbb{R}^{C \times K}$ that we learn in an end-to-end manner, where C is the number of cage vertices and K is the number of discovered keypoints. The resulting deformed cage vertices $\boldsymbol{c}_V^*$ are then defined as

$$\boldsymbol{c}_V^* = \boldsymbol{c}_V + W\Delta\boldsymbol{p}. \tag{1}$$

In order to adjust for the fact that cages are unique to each shape, we represent the influence matrix as a function of the input shape $\boldsymbol{x}$. Specifically, the influence matrix is a composition $W(\boldsymbol{x}) = W_C + W_I(\boldsymbol{x})$ of a canonical W_C matrix that is shared with all instances of the object category and an instance specific offset W_I that is predicted from the source shape $\boldsymbol{x}$ using an influence predictor $W_I = \Gamma(\boldsymbol{x})$. We regularize the instance specific W_I matrix by minimizing its Frobenius norm to prevent overfitting of the resulting influence matrix W. We denoted this regularizer as $\mathcal{L}_{\text{inf}}$. Finally, we limit the matrix W to only influence at most M nearest cage vertices per each keypoint to encourage locality.

3.2. Losses and Regularizers

Our KeypointDeformer is trained end-to-end with stochastic gradient descent by minimizing a similarity loss between the source and target shape, as well as a keypoint regularization term and instance-specific influence matrix regularization term.



(a) frequency of sampled regularizer points

(b) unsupervised keypoints predictions

Figure 3: **Farthest Point Keypoint regularizer.** We use farthest point sampling with a random starting point to regularize the predicted keypoints. **(a)** illustrates the frequency of a given point being sampled by the farthest point sampling algorithm. Darker colours indicate higher probability of a point being sampled. The expected locations of sampled points provide good coverage and inherently follow the symmetry of the original shape. Also, a subset of them tend to be semantically stable across different object instances. Using expected sample locations as a prior for keypoint location works well as the keypoint predictor will learn to be robust to noise in these sampled points. This can be seen in the example of the airplane where the tips on the fuel tanks (shown in **red** circle) are ignored, and the keypoints are instead predicted **(b)** at the wingtip (shown in **green** circle) location that is more consistent across the dataset (most planes have wings, but many lack fuel tanks).

Similarity loss. Ideally, we would like to compute the similarity between the deformed source shape $\boldsymbol{x}$ and the target shape $\boldsymbol{x}'$ using known correspondences between the meshes. However, such correspondence is not available since we aim to train on generic collections of object category CAD models. We approximate the similarity loss by computing the Chamfer distance between the deformed source $\boldsymbol{x}^*$ and the target shape $\boldsymbol{x}'$ represented as point clouds. We denote this loss as $\mathcal{L}_{\text{sim}}$.

Farthest Point Keypoint regularizer. We propose a simple, yet highly effective keypoint regularizer $\mathcal{L}_{\text{kpt}}$ that encourages predicted keypoints $\boldsymbol{p}$ to be well-distributed, lie on the object surface, and preserve the symmetric structure of the underlying shape category. Specifically, we devise a Farthest Sampling Algorithm to sample an unordered set of points $\boldsymbol{q} = \{q_1, \ldots, q_J\} \in \mathbb{R}^{3 \times J}$ from the input shape $\boldsymbol{x}$ represented as a point cloud. The initial point for sampling is chosen at random, and hence each time we compute this regularization loss a different set of sampled points $\boldsymbol{q}$ is used. Given these stochastic farthest points, the regularizer minimizes the Chamfer distance between the predicted keypoints

$\boldsymbol{p}$ and sampled points $\boldsymbol{q}$. In other words, the regularizer encourages the keypoint predictor Φ to place the discovered keypoints $\boldsymbol{p}$ at the expectation of the sampled farthest points $\boldsymbol{q}$. Figure 3 illustrates the properties of the sampled regularizer points. The sampled points provide equally spaced coverage of the input object shape $\boldsymbol{x}$, are relatively stable across different instances, and preserve the symmetric structure of the original input shapes.

Another intuition behind this regularization is that we can consider the sampled farthest points $\boldsymbol{q}$ as a noisy prior over keypoint locations. This prior is not perfect—it may miss important points in some models, or place spurious points in others—but the neural network keypoint predictor will learn keypoints in a way that is robust to such noise, and instead, prefer to predict keypoints at consistent locations, as demonstrated in Figure 3.

Full objective. In summary, our full training objective is

$$\mathcal{L} = \mathcal{L}_{\text{sim}} + \alpha_{\text{kpt}}\mathcal{L}_{\text{kpt}} + \alpha_{\text{inf}}\mathcal{L}_{\text{inf}} \tag{2}$$

where α_{kpt} and α_{inf} are scalar loss coefficients. Our method is simple and does not require additional shape specific regularization for shape deformation, such as the point-to-surface distance, normal consistency, and symmetry losses employed in [31]. This is due to the fact that keypoints provide a low-dimensional correspondence between shapes and that cage deformations are a linear function of these keypoints, preventing extreme deformations that result in unwanted local shape deformations.

3.3. Categorical Shape Prior

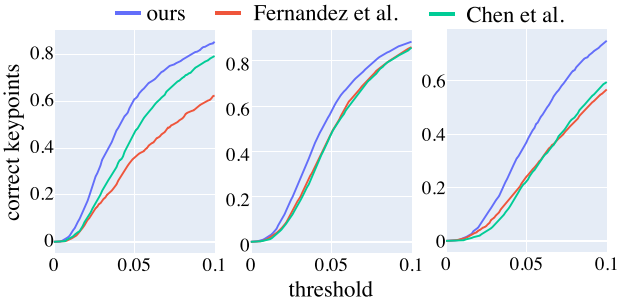
Since we represent an object shape as a set of semantically consistent keypoints, we can obtain a categorical shape prior by computing PCA on the keypoints predicted on the training set. This prior can be used to guide keypoint manipulation. For example, if user edits a single keypoint on an airplane wing, the remaining keypoints can be "synchronized" according to a prior by finding the PCA basis coefficients that best reconstruct the new position of the edited keypoint. The resulting reconstructed set of keypoints follow the prior defined by the data. This prior also allows sampling of novel shapes via sampling a new set of keypoints. This set of keypoints can be then used to deform the shape using our deformation model in order to, for instance, automatically augment libraries of stock 3D models.

4. Experiments

The main objectives of our experiments are to evaluate whether (1) our discovered keypoints are in general of good quality as keypoints (Section 4.2), (2) our discovered keypoints are better suited for shape deformation than other keypoints (Section 4.3), and (3) our method allows for intuitive shape control (Section 4.4). The supplementary material contains extended version of results and ablation studies.

	airplane	car	chair	motorbike	table
Chen *et al*. [3]	0.69	0.39	0.78	0.91	0.75
Fernandez *et al*. [4]	0.78	0.66	0.80	0.90	0.85
ours	**0.85**	**0.73**	**0.88**	**0.93**	**0.92**

Table 1: **Semantic part correspondence.** We report the average unsupervised keypoints correlation for each category. ↑ is better. Extended version with additional categories and detailed correlation tables can be found in the supplementary.



(a) airplane (b) car (c) chair

	airplane	car	chair
Chen *et al*. [3]	0.49	0.46	0.22
Fernandez *et al*. [4]	0.36	0.47	0.24
ours	**0.61**	**0.56**	**0.37**

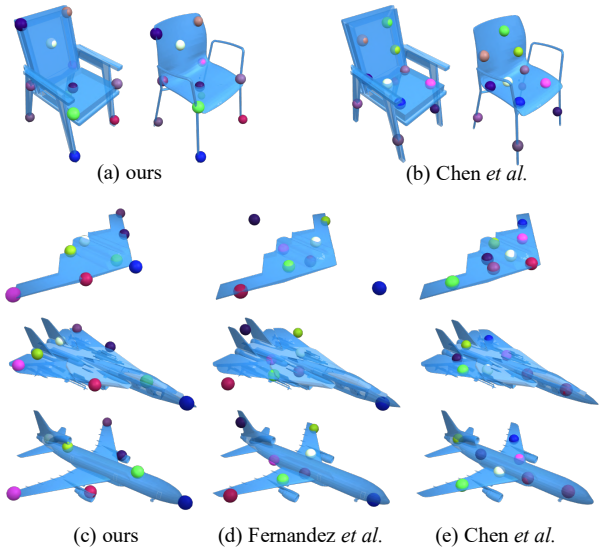
(d) Percentage of correct keypoints (PCK) @0.05

Figure 4: **Unsupervised 3D keypoints accuracy.** We measure the semantic consistency of keypoints following [27]. We train a linear regressor to predict manually annotated keypoints from unsupervised keypoints. The regressor accuracy on the test set estimates the semantic consistency of the underlying unsupervised keypoints. We show results in terms of PCK for airplane, car and chair category on the KeypointNet dataset [32].

4.1. Experimental Setup

Datasets. We train our KeypointDeformer using ShapeNet [2] following the standard training and testing split. We normalize all the shapes into a unit box. For evaluation, we use semantic part annotations for ShapeNet [30], as well as the KeypointNet [32] dataset, which contains semantic keypoint annotations for selected ShapeNet categories. Note that our method does not require any of these annotations for training. We also evaluate KeypointDeformer on real-world 3D scans of shoes from Google Scanned Objects dataset [7].

Implementation details. The keypoint predictor Φ and the influence predictor Γ are implemented as neural networks using a PointNet encoder and the whole model is optimized using the Adam optimizer. We use 1024 sampled points for the point cloud representation of shape $\boldsymbol{x}$. Unless otherwise mentioned, we predict 12 unsupervised keypoints for all



(a) ours (b) Chen *et al.* (c) ours (d) Fernandez *et al.* (e) Chen *et al.*

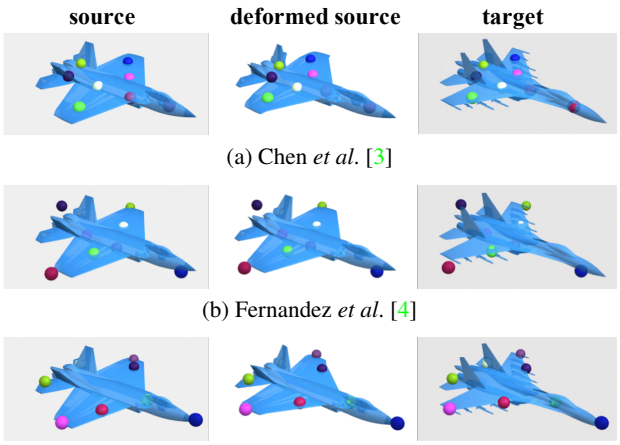
Figure 5: **Unsupervised 3D keypoints.** We compare our unsupervised 3D keypoints with Fernandez *et al.* [4] and Chen *et al.* [3]. Our keypoints are more semantically consistent despite large shape variations when compared to other methods. Keypoints obtained by Fernandez *et al.* [4] do not explain all the shapes well. Moreover, our keypoints are symmetrical without explicitly enforcing that in contrast with [4]. We show results on additional categories in the supplementary.

Figure 6: **Unsupervised 3D keypoints on real-world data.** We run our unsupervised keypoint detector on real-world scans of shoes [7]. The keypoints are semantically consistent across different shapes.

categories except for airplane and car where we use 8. The supplementary contains an ablation studying the effect of different number of unsupervised keypoints. We set the number of sampled farthest points $\boldsymbol{q}$ to the double of the number of keypoints. Detailed descriptions of network architectures and training details are in the supplemental material.

4.2. Semantic Consistency

We first demonstrate the quality of our unsupervised keypoints by evaluating their semantic consistency, *i.e.* whether they always correspond to the same semantic object parts or not. For instance, if a keypoint is predicted on the tip of the wing on one instance of an airplane, then that same keypoint should always correspond to the tip of the wing across



(a) Chen *et al.* [3]

(b) Fernandez *et al.* [4]

(c) ours

	Fernandez *et al.* [4]	Chen *et al.* [3]	annotations [32]	ours
CD	7.55	5.93	4.20	**3.02**

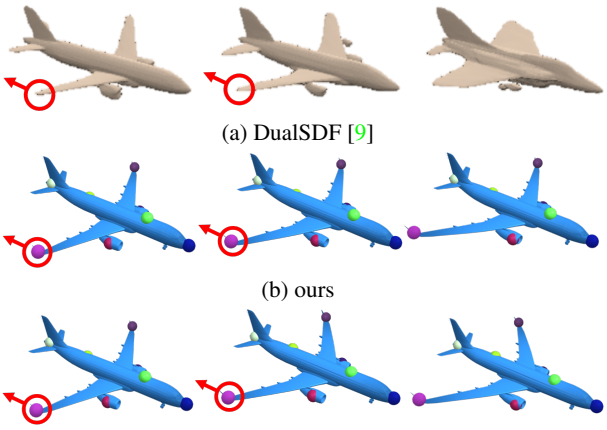
(d) Chamfer distance between deformed source and target

Figure 7: **Keypoints for shape deformation.** We replace our discovered keypoints in KeypointDeformer to compare with different keypoints detectors and manually annotated keypoints on keypoint-guided pairwise shape alignment for the airplane category. The degree of alignment is measured by the Chamfer distance between the deformed source and target shapes. Our discovered keypoints can align shapes better even when compared to manually selected keypoints from KeypointNet [32]. Keypoints from Fernandez *et al.* [4] and Chen *et al.* [3] fail to accurately align shapes as their keypoints are less precise. Data in the table are scaled by 10^3.

different instances. For this task we compare with recently introduced methods for unsupervised keypoint discovery from Fernandez *et al.* [4] and Chen *et al.* [3].

We evaluate semantic consistency using two protocols. First, we use an evaluation protocol of Fernandez *et al.* [4]. Since their evaluation is very coarse, we also follow an evaluation protocol for unsupervised keypoints established by Thewlis *et al.* [27].

The evaluation protocol of Fernandez *et al.* [4] employs the ShapeNet dataset with part annotations to measure the correlation between each keypoint and annotated semantic object parts across instances of the category. Each keypoint is associated with the nearest object part. This protocol has two limitations. First, a keypoint can be associated with an object part even if it lies far from the object (indicating a poor choice of keypoint). Second, this protocol does not account for boundary keypoints that are predicted just between two annotated object parts (which can still be high-quality, salient keypoints). To address these limitations, we propose a small modification to this protocol, in which we associate each keypoint with a given object part if it lies within its small



(a) DualSDF [9]

(b) ours

(c) ours with shape prior

Figure 8: **Comparison with DualSDF [9].** We move the wing tip in the direction of the red arrow. (a) DualSDF is a generative model and changing the position of the wing tip results in a change from an airliner to a jet fighter. In contrast, our method preserves the original structure of the mesh and allows for asymmetric manipulation when desired (b). Our method can also work in conjunction with a shape prior (Section 4.5) to achieve symmetrical manipulation (c).

neighborhood (0.05 from the object part when the object is normalized to unit box)—hence, a keypoint can be associated with multiple parts. For each keypoint, we compute its correlation with each object part. Since a keypoint can be associated with multiple parts, we consider only the most correlated part for a given keypoint in the final metric. The final metric then computes the average correlation over all the keypoints. We report semantic consistency results for ShapeNet categories in Table 1. Our keypoints show better average correlation when compared to Chen *et al*. [3] and Fernandez *et al*. [4].

Second, we adopt the standard unsupervised 2D keypoint evaluation protocol as in [27, 34, 10], since the semantic object parts are coarsely annotated (e.g., the airplane category comes with only 3 semantic parts). The objective of this protocol is to measure how predictive unsupervised keypoints are of semantic keypoints selected by humans. This is done by finding a linear mapping between the unsupervised keypoints and manually annotated ones. The linear mapping is established on the training set by fitting a linear regressor. The predictiveness of unsupervised keypoints is then measured in terms of this regressor's prediction error on the test set. We use the recent KeypointNet dataset [32], which contains semantic annotations on ShapeNet dataset. We report the performance in Figure 4. Our unsupervised keypoints are more predictive of manually annotated keypoints than other unsupervised keypoint. Figure 5 provides qualitative comparison of our unsupervised keypoints with those obtained by other methods.

Real world scans. We also demonstrate applicability of our unsupervised keypoint detector on real-world 3D scans of objects. We use the shoe category from Google Scanned Objects dataset [7]. We align the shapes using the automatical alignment method from [16]. We split the dataset into training and test sets with 219 and 36 samples respectively. We use the same hyper-parameters as done in experiments on ShapeNet. Figure 6 shows that our method learns semantically consistent 3D keypoints for shoes with largely different shapes.

4.3. Keypoints for Shape Deformation

To quantitatively demonstrate that controlled shape deformation is possible through unsupervised keypoints, we use the task of pairwise shape alignment, in which we deform a source shape into a target shape. In our case, the deformation is guided using keypoints. This task also evaluates that our discovered keypoints are more suitable for shape control than other keypoints. We modify our method by replacing our unsupervised keypoints with keypoints obtained from other methods. We then train our deformation model from scratch. We experiment with keypoints from [4], [3], and also manually annotated keypoints from [32]. Performance is evaluated by measuring the Chamfer distance between the deformed source shape and the target shape. We present results in Figure 7. The unsupervised keypoints obtained by other methods fail to capture the large variations in shapes in the dataset. Our keypoints, on the other hand, can follow the large changes in shapes. This ultimately leads to more accurate shape deformations.

4.4. Shape Control via Unsupervised 3D Keypoints

Our ultimate goal is to use automatically discovered keypoints to perform user-guided interactive shape deformation. Figure 9 shows interactive shape control using our unsupervised keypoints. Our method provides low-dimensional handles to control object shape. The control is intuitive as the deformation is semantically consistent, e.g., moving a keypoint on the leg of a chair or airplane wing results in movement of that object part in the same direction. Thus the user can easily edit shape meshes. Please refer to our project page for a demo video showcasing user-guided interactive shape control using keypoints.

The related work DualSDF [9] also allows for user-guided interactive shape deformation. However, the key distinction here is that DualSDF is a conditional generative model. Manipulating an object through its handle generates a new shape that respects the new position of the handle specified by the user, but the new generated shape can be very different from the original one. This aspect is illustrated in Figure 8, where DualSDF transforms an airliner to a jet fighter.

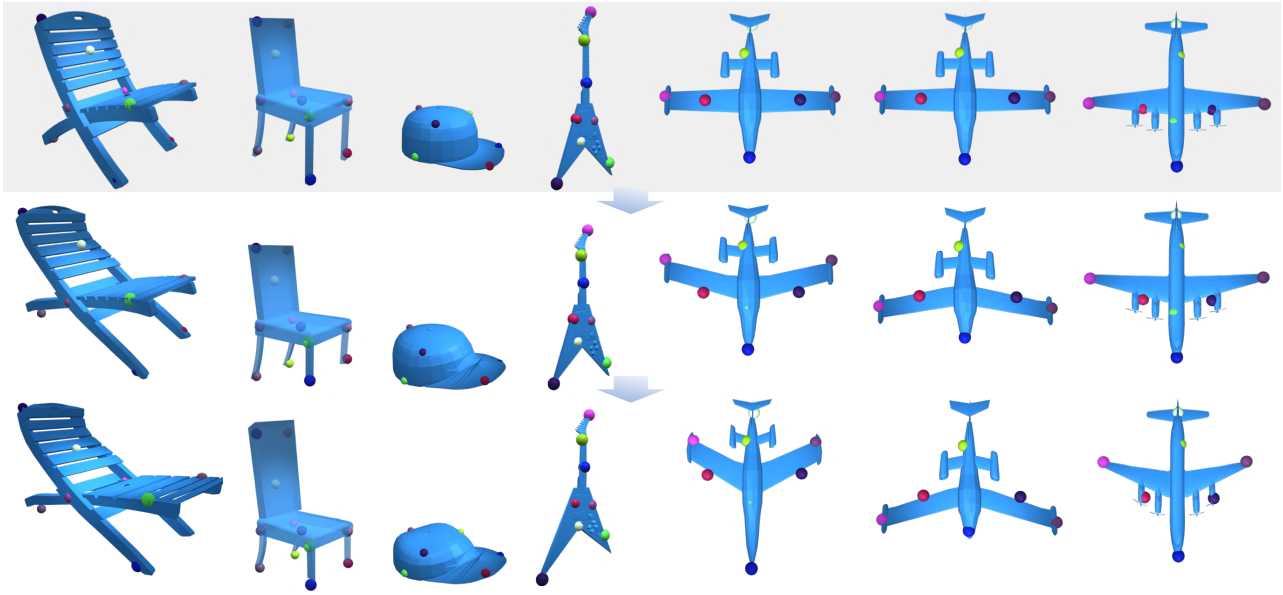


Figure 9: **Interactive shape control via 3D unsupervised keypoints.** We show iterative steps in user guided shape deformation using our discovered keypoint as handles. Top row shows initial state. Please refer to our project page for a demo video.

4.5. Categorical Shape Prior

Since our deformation model uses keypoints as its low-dimensional shape representation, we can compute categorical shape prior on them. We compute PCA on the set of predicted keypoints obtained from the training set. We set the number of basis to 8. As discussed in Section 4.5, we use the prior in two ways. First, we can use it during interactive shape control when the user manipulates only a single keypoint, to "synchronize" the rest of the keypoints according to the prior. This "synchronized" editing is used in Figure 8 where we drag only a single keypoint and the rest get automatically readjusted. Second, we can easily sample new deformations using sampled keypoints that we obtain by varying PCA basis coefficients. This can be applied to automatic dataset amplification as demonstrated Figure 10.

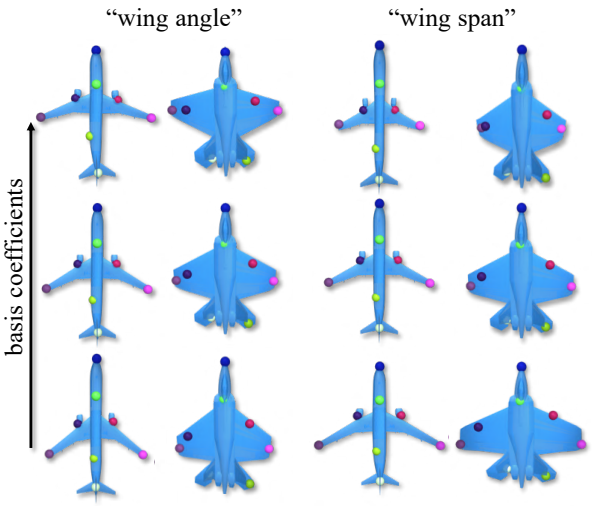


Figure 10: **Varying PCA basis coefficients for shape augmentation.** We sample new keypoints by varying its PCA basis coefficients. The sampled keypoints are used to deform the original shape obtaining a new set of shapes. The left two columns show results for a subspace that correlates with the wing angle. The right two columns show results for a subspace that correlates with the wing span.

5. Conclusion

We present a method for controlling the shape of 3D objects through automatically discovered semantic 3D keypoints and a deformation model learned jointly with the keypoints. The resulting KeypointDeformer model provides users with a simple interface for interactive shape control. One limitation of the method is that our approach assumes aligned shape collections. However, in our experiments with real scans, automatic alignment method was sufficient. Another limitation is that the keypoint representation does not allow modeling of individual object part rotations. In this work we focused on the task of shape control and keypoint prediction, however 3D keypoints has various usage in other applications such as robotics [17, 18]. It would be interesting to explore the applicability of our unsupervised 3D keypoints to other tasks in the future.

References

[1] Mario Botsch, Mark Pauly, Markus H Gross, and Leif Kobbelt. PriMo: coupled prisms for intuitive surface modeling. In *Symposium on Geometry Processing*, pages 11–20, 2006. 2

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 5, 12

[3] Nenglun Chen, Lingjie Liu, Zhiming Cui, Runnan Chen, Duygu Ceylan, Changhe Tu, and Wenping Wang. Unsupervised learning of intrinsic structural representation points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9121–9130, 2020. 3, 5, 6, 7, 13

[4] Clara Fernandez-Labrador, Ajad Chhatkuli, Danda Pani Paudel, Jose J Guerrero, Cédric Demonceaux, and Luc Van Gool. Unsupervised learning of category-specific symmetric 3d keypoints from point sets. *European Conference on Computer Vision (ECCV)*, 2020. 3, 5, 6, 7, 13

[5] Matheus Gadelha, Giorgio Gori, Duygu Ceylan, Radomir Mech, Nathan Carr, Tamy Boubekeur, Rui Wang, and Subhransu Maji. Learning generative models of shape handles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[6] Wei Gao and Russ Tedrake. kpam-sc: Generalizable manipulation planning using keypoint affordance and shape completion. *arXiv preprint arXiv:1909.06980*, 2019. 3

[7] GoogleResearch. Google scanned objects. https://fuel.ignitionrobotics.org/1.0/GoogleResearch/fuel/collections/Google%20Scanned%20Objects, September 2020. 5, 6, 7, 15

[8] Rana Hanocka, Noa Fish, Zhenhua Wang, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Alignet: Partial-shape agnostic alignment via unsupervised learning. *ACM Transactions on Graphics (TOG)*, 38(1):1–14, 2018. 2

[9] Zekun Hao, Hadar Averbuch-Elor, Noah Snavely, and Serge Belongie. Dualsdf: Semantic shape manipulation using a two-level representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7631–7641, 2020. 2, 3, 7

[10] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *Advances in neural information processing systems*, pages 4016–4027, 2018. 3, 7

[11] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Self-supervised learning of interpretable keypoints from unlabelled videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8787–8797, 2020. 3

[12] Pushkar Joshi, Mark Meyer, Tony DeRose, Brian Green, and Tom Sanocki. Harmonic coordinates for character articulation. *ACM Transactions on Graphics (TOG)*, 26(3):71–es, 2007. 4

[13] Tao Ju, Scott Schaefer, and Joe Warren. Mean value coordinates for closed triangular meshes. In *SIGGRAPH*, pages 561–566. 2005. 2, 4

[14] A Sophia Koepke, Olivia Wiles, and Andrew Zisserman. Self-supervised learning of a facial attribute embedding from video. In *BMVC*, page 302, 2018. 3

[15] Yaron Lipman, David Levin, and Daniel Cohen-Or. Green coordinates. *ACM Transactions on Graphics (TOG)*, 27(3):1–10, 2008. 4

[16] Ameesh Makadia and Kostas Daniilidis. Rotation recovery from spherical images without correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(7):1170–1175, 2006. 7

[17] Lucas Manuelli, Wei Gao, Peter Florence, and Russ Tedrake. kpam: Keypoint affordances for category-level robotic manipulation. *arXiv preprint arXiv:1903.06684*, 2019. 3, 8

[18] Lucas Manuelli, Yunzhu Li, Pete Florence, and Russ Tedrake. Keypoints into the future: Self-supervised correspondence in model-based reinforcement learning. *arXiv preprint arXiv:2009.05085*, 2020. 8

[19] Eloi Mehr, Ariane Jourdan, Nicolas Thome, Matthieu Cord, and Vincent Guitteny. Disconet: Shapes learning on disconnected manifolds for 3d editing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3474–3483, 2019. 2

[20] Tiberiu Popa, Dan Julius, and Alla Sheffer. Material-aware mesh deformations. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, pages 22–22. IEEE, 2006. 2

[21] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 14

[22] Olga Sorkine. Differential representations for mesh processing. *Computer Graphics Forum*, 25(4):789–807, 2006. 2

[23] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, pages 109–116, 2007. 2

[24] Robert W Sumner and Jovan Popović. Deformation transfer for triangle meshes. *ACM Transactions on graphics (TOG)*, 23(3):399–405, 2004. 2

[25] Supasorn Suwajanakorn, Noah Snavely, Jonathan J Tompson, and Mohammad Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. In *Advances in neural information processing systems*, pages 2059–2070, 2018. 3

[26] James Thewlis, Samuel Albanie, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of landmarks by descriptor vector exchange. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6361–6371, 2019. 3

[27] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *Proceedings of the IEEE international conference on computer vision*, pages 5916–5925, 2017. 3, 5, 6, 7

[28] Shubham Tulsiani, Hao Su, Leonidas J Guibas, Alexei A Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2635–2643, 2017. 2

[29] Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. 3dn: 3d deformation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1038–1046, 2019. 2

[30] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016. 5

[31] Wang Yifan, Noam Aigerman, Vladimir G Kim, Siddhartha Chaudhuri, and Olga Sorkine-Hornung. Neural cages for detail-preserving 3d deformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 75–83, 2020. 2, 4, 5, 14

[32] Yang You, Yujing Lou, Chengkun Li, Zhoujun Cheng, Liangwei Li, Lizhuang Ma, Cewu Lu, and Weiming Wang. Keypointnet: A large-scale 3d keypoint dataset aggregated from numerous human annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13647–13656, 2020. 5, 6, 7, 12

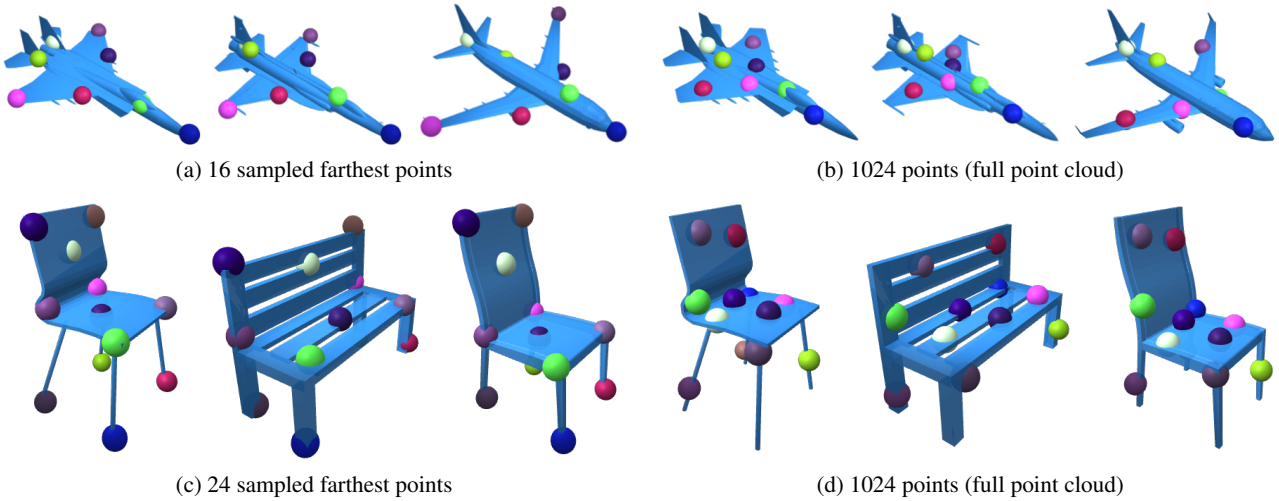
[33] Mehmet Ersin Yumer, Siddhartha Chaudhuri, Jessica K Hodgins, and Levent Burak Kara. Semantic shape editing using deformation handles. *ACM Transactions on Graphics (TOG)*, 34(4):1–12, 2015. 2

[34] Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee. Unsupervised discovery of object landmarks as structural representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2694–2703, 2018. 3, 7

Appendix

This supplemental document provides an ablation study (Appendix A), extensive results (Appendices B and C), and further implementation details (Appendix D). Please also refer to our video on our project pagethat demonstrates our method in action.

A. Ablations



(a) 16 sampled farthest points (b) 1024 points (full point cloud)

(c) 24 sampled farthest points (d) 1024 points (full point cloud)

Figure 11: **Farthest Point Keypoint regularizer ablation.** We investigate the influence of the number J of sampled farthest points $\boldsymbol{q}$ used for the keypoint regularizer (Section 3.2 of the main paper) on the quality of discovered unsupervised keypoints. We show unsupervised 3D keypoints trained using two versions of regularization. First, we set the number of sampled farthest points to double of the number of keypoints **(a, c)**. This is the setup that we use throughout the paper. Second, we set the number of sampled farthest points to the number of points in the point cloud representing the shape. This essentially results in a regularizer that is minimizing the Chamfer distance between the unsupervised keypoints and the object point cloud. Although the learned unsupervised keypoints have a good coverage **(b, d)** they are not as equally spaced and characteristic of the shape as **(a, c)**.

Varying number of regularizing points. We examine the importance of the number of sampled farthest points $\boldsymbol{q}$ on the quality of keypoint regularization (Section 3.2 of the main paper). Figure 11 shows the effect of different numbers of sampled farthest points on the discovered keypoints. Using a high number of sampled farthest points in the regularization fails to learn keypoints that are equally spaced and characteristic of the underlying shape.

Varying number of keypoints. We vary the number of unsupervised keypoints discovered by our method. Figure 12 shows that our keypoints remain semantically consistent for different numbers of discovered keypoints.

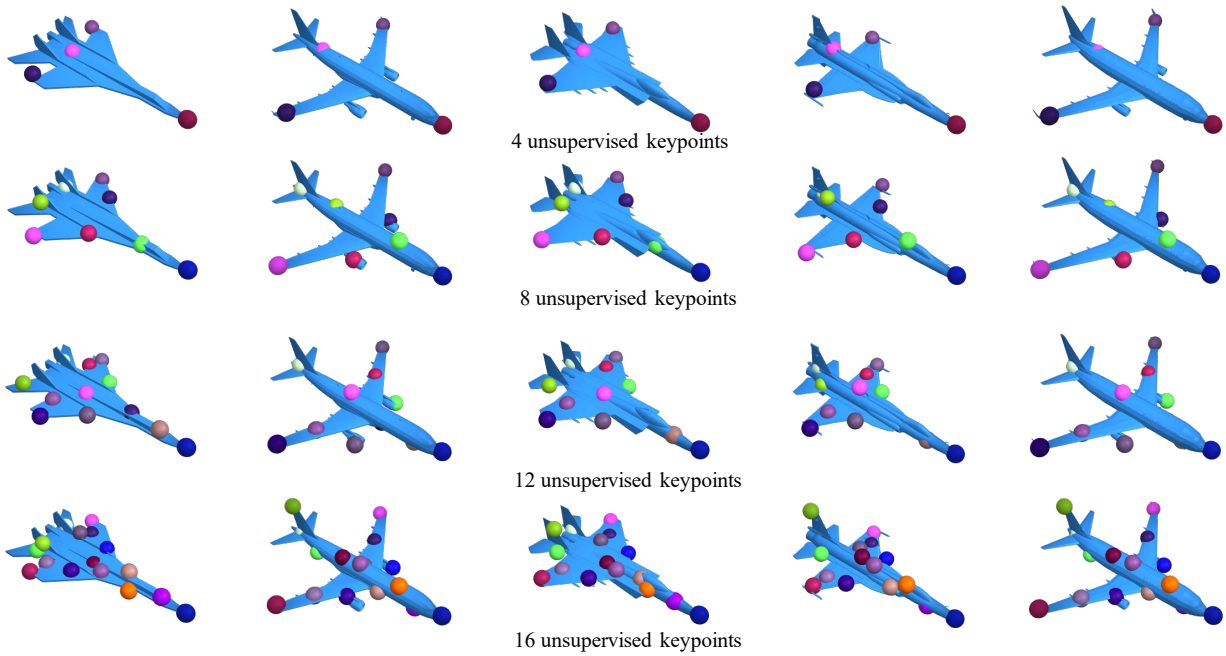
B. Shape Control via Unsupervised 3D Keypoints

We show user guided interactive shape control in our supplementary video on our project page. Figure 15 shows frames captured from user-guided interactive shape editing. Editing using our keypoints is fast and intuitive while preserving the character and details of the original shape.

C. Unsupervised 3D Keypoints

We show extended quantitative results for semantic part correspondence experiment and detailed correlation tables in Figure 13 for the ShapeNet Car category. Figures 16 to 20 show extensive *randomly* sampled qualitative test results for our unsupervised 3D keypoints.

http://tomasjakab.github.io/KeypointDeformer



(a) predicted unsupervised keypoints

# of unsupervised keypoints	4	8	12	16
PCK@0.05	0.56	0.61	0.71	0.71

(b) quantitative evaluation

Figure 12: **Varying number of keypoints.** The figure (a) shows the effect of different number of discovered keypoints (4, 8, 12, 16 from the top). The results are shown on randomly sampled results for ShapeNet Airplane category. Results in the table (b) are in terms of PCK@0.05 on airplanes from the KeypointNet dataset.

D. Implementation Details

Our model assumes that the shapes are aligned (in the same orientation). The initial cage is a 42-vertex icosphere. We limit the influence matrix W to influence at most M nearest cages vertices (Section 3.1) per each keypoint, with $M = \lfloor C/K \rfloor$, where C is the number of cage vertices and K is the number of discovered keypoints. We use a learning rate of 0.001. The scalar loss coefficients (Section 3.2) α_{kpt} and α_{inf} are set to 1.0 and 10^{-6} respectively. Figure 14 shows detailed description of network architectures used for the keypoint predictor Φ and the influence predictor Γ.

Datasets. KeypointNet [32] dataset contains semantic 3D keypoint annotations for ShapeNet dataset [2]. Some models in KeypointNet are missing full keypoint annotations, therefore we use a subset of annotated keypoints that are contained in at least 80% of the models. KeypointNet also does not follow the standard training and testing splits from ShapeNet. We resample the KeypointNet dataset splits to make it compatible with the original ShapeNet splits.

airplane

	kp1	kp2	kp3	kp4	kp5	kp6	kp7	kp8	best avg
body	0.30	0.29	0.35	0.17	0.01	0.01	0.57	0.73	
wing	0.09	0.49	0.01	0.90	0.98	0.98	0.02	0.49	
tail	0.10	0.00	0.00	0.01	0.00	0.00	0.82	0.01	
best	*0.30*	*0.49*	*0.35*	*0.90*	*0.98*	*0.98*	*0.82*	*0.73*	***0.69***

Chen *et al.*

	kp1	kp2	kp3	kp4	kp5	kp6	kp7	kp8	best avg
	0.02	0.89	0.00	0.89	0.77	0.08	0.00	0.09	
	0.00	0.10	0.62	0.00	0.00	0.89	0.62	0.88	
	0.65	0.00	0.00	0.02	0.00	0.01	0.00	0.01	
best	*0.65*	*0.89*	*0.62*	*0.89*	*0.77*	*0.89*	*0.62*	*0.88*	***0.78***

Fernandez *et al.*

	kp1	kp2	kp3	kp4	kp5	kp6	kp7	kp8	best avg
	0.01	0.02	0.08	0.03	0.92	0.87	0.68	0.52	
	0.77	0.96	0.76	0.95	0.00	0.01	0.02	0.03	
	0.00	0.00	0.00	0.01	0.00	0.00	0.12	0.86	
best	*0.77*	*0.96*	*0.76*	*0.95*	*0.92*	*0.87*	*0.68*	*0.86*	***0.85***

ours

car

	kp1	kp2	kp3	kp4	kp5	kp6	kp7	kp8	best avg
roof	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
wheels	0.16	0.37	0.01	0.01	0.07	0.00	0.04	0.00	
body	0.47	0.49	0.18	0.39	0.45	0.37	0.37	0.39	
best	*0.47*	*0.49*	*0.18*	*0.39*	*0.45*	*0.37*	*0.37*	*0.39*	***0.39***

Chen *et al.*

	kp1	kp2	kp3	kp4	kp5	kp6	kp7	kp8	best avg
	0.00	0.56	0.01	0.00	0.00	0.00	0.00	0.00	
	0.00	0.00	0.00	0.01	0.75	0.01	0.02	0.58	
	0.73	0.08	0.62	0.64	0.01	0.74	0.68	0.00	
best	*0.73*	*0.56*	*0.62*	*0.64*	*0.75*	*0.74*	*0.68*	*0.58*	***0.66***

Fernandez *et al.*

	kp1	kp2	kp3	kp4	kp5	kp6	kp7	kp8	best avg
	0.00	0.64	0.00	0.00	0.00	0.00	0.00	0.01	
	0.85	0.00	0.00	0.72	0.04	0.75	0.01	0.00	
	0.30	0.19	0.73	0.49	0.63	0.25	0.77	0.75	
best	*0.85*	*0.64*	*0.73*	*0.72*	*0.63*	*0.75*	*0.77*	*0.75*	***0.73***

ours

chair

	kp1	kp2	kp3	kp4	kp5	kp6	kp7	kp8	kp9	kp10	kp11	kp12	best avg
back	0.00	0.02	0.14	0.76	0.84	0.10	0.00	0.60	0.03	0.82	0.00	0.00	
seat	0.08	0.07	0.78	0.08	0.00	0.82	0.09	0.44	0.08	0.00	0.79	0.82	
legs	0.77	0.80	0.09	0.02	0.00	0.05	0.76	0.04	0.79	0.00	0.09	0.08	
best	*0.77*	*0.80*	*0.78*	*0.76*	*0.84*	*0.82*	*0.76*	*0.60*	*0.79*	*0.82*	*0.79*	*0.82*	***0.78***

Chen *et al.*

	kp1	kp2	kp3	kp4	kp5	kp6	kp7	kp8	kp9	kp10	kp11	kp12	best avg
	0.01	0.68	0.80	0.00	0.01	0.69	0.04	0.79	0.78	0.02	0.00	0.01	
	0.00	0.00	0.00	0.00	0.00	0.00	0.97	0.00	0.00	0.95	0.00	0.96	
	0.76	0.00	0.00	0.75	0.75	0.00	0.01	0.01	0.00	0.01	0.76	0.01	
best	*0.76*	*0.68*	*0.80*	*0.75*	*0.75*	*0.69*	*0.97*	*0.79*	*0.78*	*0.95*	*0.76*	*0.96*	***0.80***

Fernandez *et al.*

	kp1	kp2	kp3	kp4	kp5	kp6	kp7	kp8	kp9	kp10	kp11	kp12	best avg
back	0.00	0.74	0.00	0.73	0.00	0.01	0.01	0.88	0.01	0.94	0.96	0.01	
seat	0.86	0.61	0.02	0.67	0.01	0.88	0.01	0.01	0.02	0.00	0.00	0.91	
legs	0.01	0.30	0.94	0.31	0.93	0.52	0.88	0.00	0.91	0.01	0.01	0.48	
best	*0.86*	*0.74*	*0.94*	*0.73*	*0.93*	*0.88*	*0.88*	*0.88*	*0.91*	*0.94*	*0.96*	*0.91*	***0.88***

ours

(a) unsupervised keypoints correlation

	airplane	cap	car	chair	guitar	knife	laptop	motorbike	mug	skateboard	table	pistol	bag	rocket	earphone	lamp
Chen *et al.* [3]	0.69	0.24	0.39	0.78	0.97	0.94	0.95	0.91	0.50	0.89	0.75	0.78	0.35	0.56	0.30	0.50
Fernandez *et al.* [4]	0.78	0.45	0.66	0.80	0.93	0.92	0.85	0.90	0.78	0.92	0.85	0.60	0.72	0.61	0.24	0.40
ours	**0.85**	**0.71**	**0.73**	**0.88**	**0.99**	**0.96**	**0.96**	**0.93**	**0.94**	**0.96**	**0.92**	**0.91**	**0.85**	**0.90**	**0.72**	**0.53**

(b) average unsupervised keypoints correlation

Figure 13: **Semantic part correspondence.** We evaluate semantic part correspondence for the ShapeNet Car category. The tables (a) shows the frequency of each unsupervised keypoint **[kp*]** being associated with a given object part. Chen *et al.* [3] show worse performance in this task because that methods tends to predict keypoints inside the object far from the annotated object surface. We also report the average unsupervised keypoints correlation for each category (b). ↑ is better.

Operation	Kernel	Output channels	Output size	Activation
Input $\boldsymbol{x}$	-	3	1024	-
Conv 1D	1	64	1024	ReLU
Conv 1D	1	128	1024	ReLU
Conv 1D	1	256	1024	None
Max. pool	1024	256	1	-
Squeeze	-	256	-	-
Linear	-	256	-	LReLU
Linear	-	512	-	LReLU
Linear	-	256	-	LReLU
Linear	-	$3 \cdot K$	-	None
Reshape	-	3	K	-

(a) Keypoint predictor Φ

Operation	Kernel	Output channels	Output size	Activation
Input $\boldsymbol{x}$	-	3	1024	-
Conv 1D	1	64	1024	ReLU
Conv 1D	1	128	1024	ReLU
Conv 1D	1	256	1024	None
Max. pool	1024	256	1	-
Squeeze	-	256	-	-
Linear	-	$C \cdot K$	-	LReLU
Linear	-	$C \cdot K$	-	LReLU
Linear	-	$C \cdot K$	-	None
Reshape	-	C	K	-

(b) Influence predictor Γ

Figure 14: **Network architectures.** The network architectures are based on a PointNet encoder [21, 31]. K is the number of discovered keypoints, C is the number of cage vertices. LReLU stands for Leaky ReLU with 0.1 negative slope.

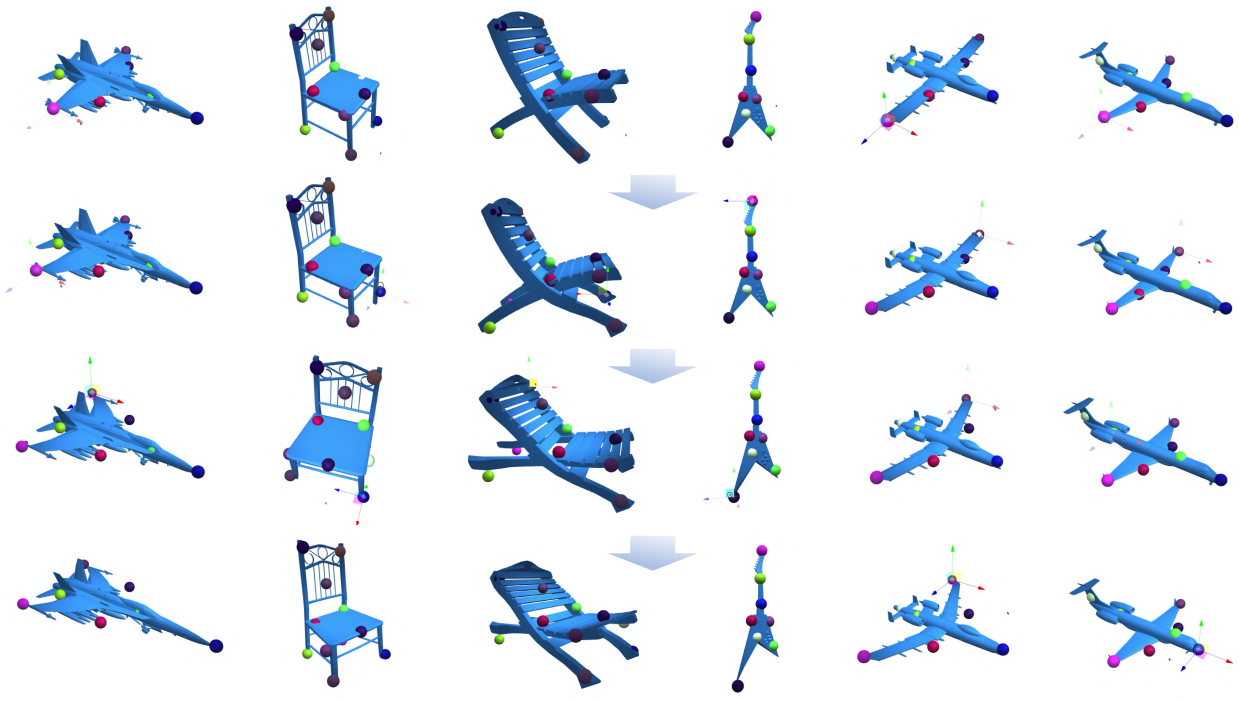


Figure 15: **Shape control via unsupervised 3D keypoints.** We show steps from shape editing using our unsupervised 3D keypoints. Please refer to our supplementary video on our project page to see the editing in action.

Figure 16: **Unsupervised 3D keypoints on real-world 3D scans.** Randomly sampled results with 8 unsupervised keypoints on real-world 3D scans of shoes from Google Scanned Objects dataset [7].

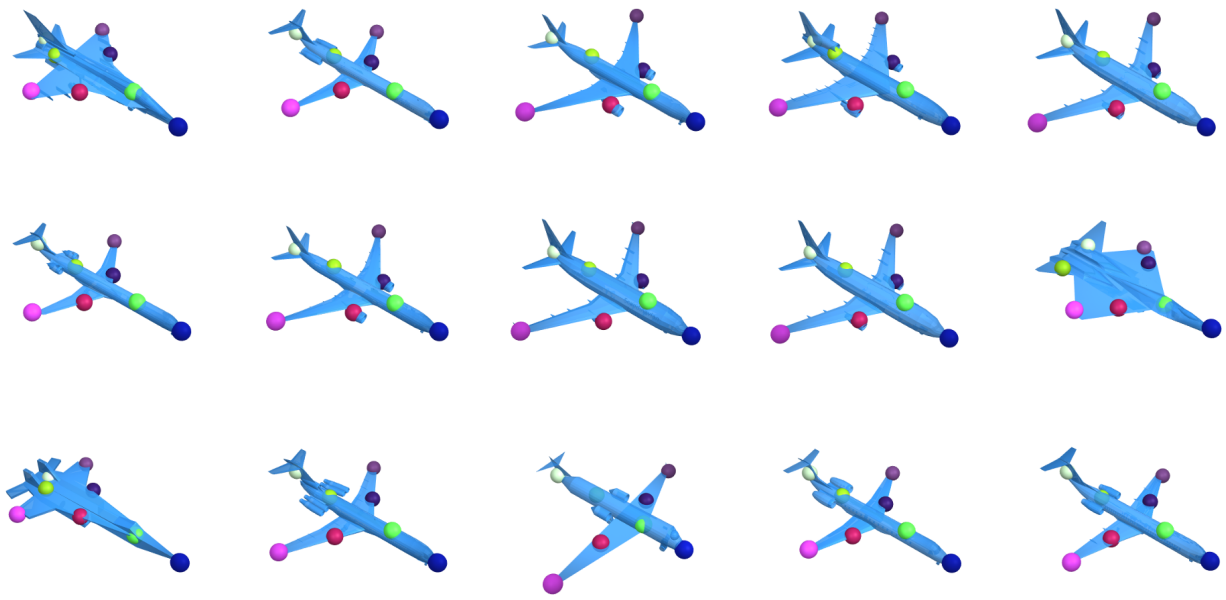


Figure 17: **Unsupervised 3D keypoints.** Randomly sampled results with 8 unsupervised keypoints for ShapeNet Airplane category.

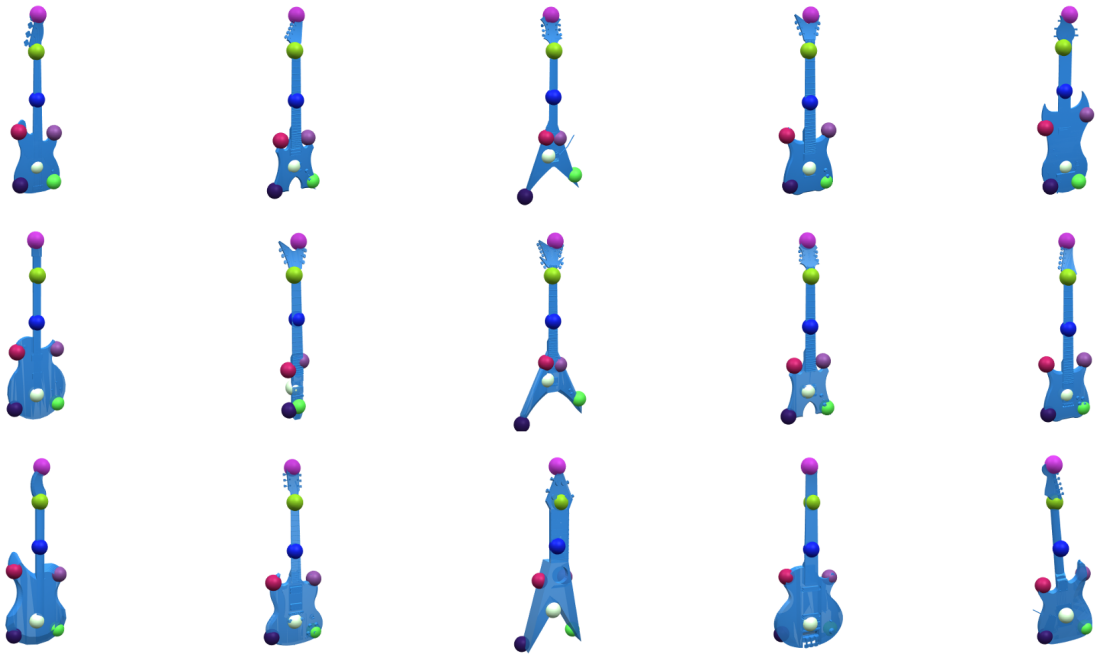


Figure 18: **Unsupervised 3D keypoints.** Randomly sampled results with 8 unsupervised keypoints for ShapeNet Guitar category.

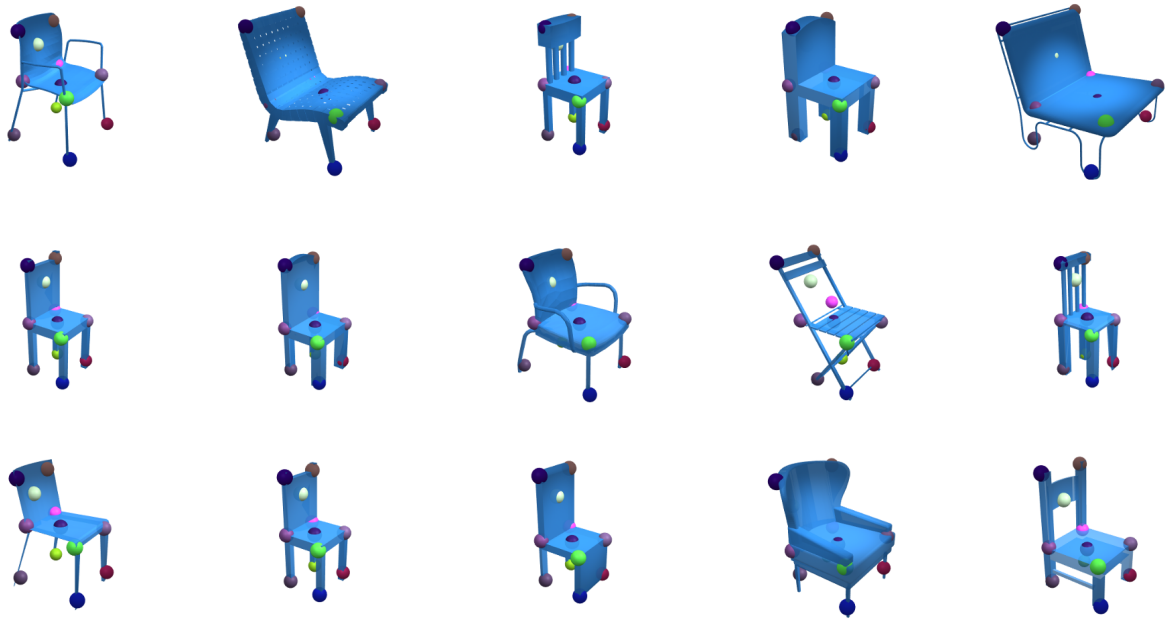


Figure 19: **Unsupervised 3D keypoints.** Randomly sampled results with 12 unsupervised keypoints for ShapeNet Chair category.

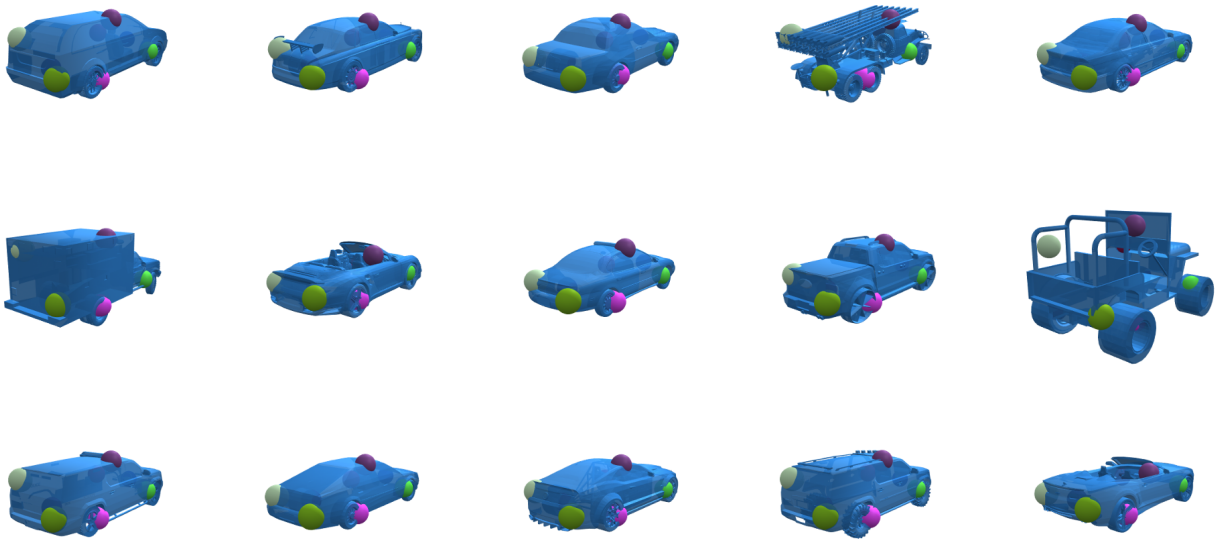


Figure 20: **Unsupervised 3D keypoints.** Randomly sampled results with 8 unsupervised keypoints for ShapeNet Car category.

Chapter 6

DOVE: Learning Deformable 3D Objects by Watching Videos

DOVE: Learning Deformable 3D Objects by Watching Videos

Shangzhe Wu* **Tomas Jakab*** **Christian Rupprecht** **Andrea Vedaldi**

Visual Geometry Group, University of Oxford
{szwu, tomj, chrisr, vedaldi}@robots.ox.ac.uk
dove3d.github.io

Abstract

Learning deformable 3D objects from 2D images is an extremely ill-posed problem. Existing methods rely on explicit supervision to establish multi-view correspondences, such as template shape models and keypoint annotations, which restricts their applicability on objects "in the wild". In this paper, we propose to use monocular videos, which naturally provide correspondences across time, allowing us to learn 3D shapes of deformable object categories without explicit keypoints or template shapes. Specifically, we present *DOVE*, which learns to predict 3D canonical shape, deformation, viewpoint and texture from a single 2D image of a bird, given a bird video collection as well as automatically obtained silhouettes and optical flows as training data. Our method reconstructs temporally consistent 3D shape and deformation, which allows us to animate and re-render the bird from arbitrary viewpoints from a single image.

1 Introduction

Recently, with the adoption of machine learning, reconstructing 3D shapes from 2D images has advanced considerably. While this task traditionally requires establishing correspondences between multiple views [13], learning-based approaches have demonstrated the possibility of inferring 3D shapes from a single image, by learning priors for a specific object category [3, 9, 57, 12, 23, 10]. However, most of these methods rely on ground-truth 3D data [3, 9, 58, 57, 12, 23, 33, 43, 48, 10] or shape models [31, 19, 68, 49, 69] as supervision for training, and thus only work well on very specific categories, such as faces, human bodies and synthetic objects, where such data are available.

Recent unsupervised and weakly-supervised methods have shown promising reconstruction results on a wider variety of objects, by replacing the 3D ground-truth with weaker forms of supervision, such as 2D keypoints [20], 3D viewpoints [20, 39] and category-specific template shapes [25, 69, 11]. These additional sources of information provide strong supervisory signals for establishing correspondences between different instances, allowing the models to be trained on single-view image collections. Although these annotations are relatively cheaper compared to 3D ground-truth, it is still prohibitively difficult to collect such annotations for all object categories in the world at large scale.

In this paper, we aim to go one step further and leverage an even cheaper source of data to learn correspondences for reconstructing deformable 3D objects, which is "in-the-wild" monocular videos that are readily available online. Videos naturally provide rich correspondences across time as different views of the objects are observed. If the objects are static, existing methods [62, 29, 17] are able to learn 3D shapes based on similar principals as Structure-from-Motion (SfM) [5, 13]. However, many moving objects "in the wild"—such as animals—are not static, but rather highly deformable. Modeling these articulated shapes from monocular videos alone is very challenging due to the lack of multi-view consistency.

*Equal contribution.

Preprint. Under review.

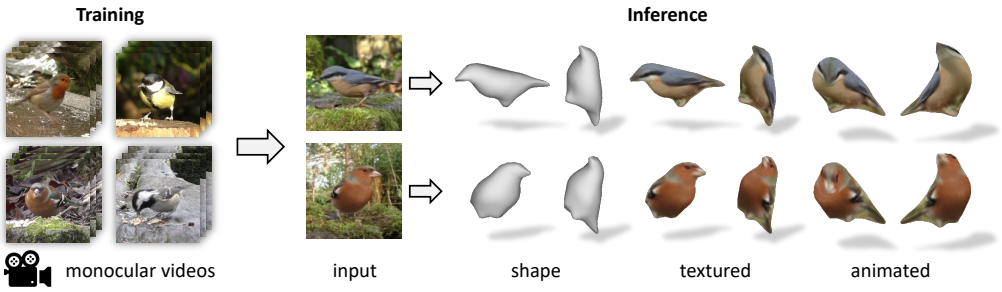


Figure 1: **DOVE - Deformable Objects from VidEos.** Given a collection of video clips of an object category as training data, we learn a model that is able to predict a textured, articulated 3D mesh of the object from a single input image.

In this work, we propose a method that automatically discovers and leverages correspondences in monocular videos for category-specific 3D reconstruction of deformable objects. Specifically, our model is trained on a collection of short video clips capturing objects of the same category, as illustrated in Fig. 1. To simplify the problem, here we assume videos with a static background, which are easy to obtain simply using a stationary camera (as is often done to capture wild animals), and rely on the object itself as the only source of motion, which is the exact opposite of [29]. Similar to previous methods [20, 11, 27], we also obtain 2D object masks using an off-the-shelf segmentation model, and assume bilateral symmetry in the shape and texture (but in the *canonical* pose rather than individual frames as in [20] due to asymmetric articulation), which is true for most animals. However, unlike these existing approaches, our method does not require explicit geometric supervision, such as keypoints, viewpoint or template shapes, and instead relies on the temporal information inherent in videos alone for learning 3D shapes.

To do this, we adopt a *photo-geometric auto-encoding* approach that disentangles shape, texture and pose of an object from each frame, with the objective of reconstructing the original image as well as the 2D object mask. Although on single frames this is an ill-posed problem, temporal consistency between consecutive frames in the videos provides powerful cues that make the problem tractable. First, the identity of the object remains the same in the video clip. By enforcing a constant base shape and texture, we can factor out the motion of the object with an articulation model and establish correspondences across different poses. As a result of this decomposition, we can also animate the objects with arbitrary poses, as shown in Fig. 1. Second, with videos, methods such as optical flow can be used to estimate 2D correspondences between frames automatically, which also provide geometric cues for recovering 3D shape and pose.

In summary, we present a method that learns to reconstruct deformable 3D object categories from "in-the-wild" monocular videos. By exploiting temporal consistency in videos, our method eliminates the need for explicit geometric supervision required by existing methods, such as keypoints, viewpoint or template shapes, which could potentially enable us to model a much wider range of objects where such annotations are difficult to obtain. Furthermore, our model produces temporally consistent reconstructions, whereas state-of-the-art methods produce inconsistent results as they are trained on independent, single images.

2 Related Work

Learning-based Reconstruction. A primary motivation for introducing machine learning in 3D reconstruction is to enable reconstruction from single views, which necessitates learning suitable shape priors. In particular, we focus the discussion on unsupervised and weakly-supervised methods that do not require explicit 3D ground-truth for training. Early unsupervised examples include monocular depth predictors trained from egocentric videos of rigid scenes, such as [8] and SfMLearner [67].

Other authors instead explored learning to reconstruct full 3D meshes of objects rather than depth maps [23, 20, 30, 22, 57, 42, 15, 11, 27, 28, 60]. Many of these works consider object categories and learn from still images only, generally using masks and additional sources of supervision or prior assumption: CMR [20] uses 2D keypoint annotations (in addition to masks) to initialize shape

Table 1: **Related Work.** Flow indicates optical flow. [1]coarse template shape and [2]camera estimated from keypoints using SfM, [3]outputs texture flow, [4]shape bases initialized from CMR. *LASR and an online adaptation version of VMR have to be optimized on test sequences.

	Supervision						Output				
Method	Template	View	Keypoints	Mask	Flow	Video	3D	2.5D	Deform	View	Texture
VMR* [28]				✓		✓	✓		✓	✓	(✓)[3]
LASR* [63]				✓	✓	✓	✓		✓	✓	✓
Unsup3D [59]								✓		✓	✓
CSM [25]	✓			✓						✓	
CMR [20]	(✓)[1]	(✓)[2]	✓	✓			✓			✓	(✓)[3]
U-CMR [11]	✓			✓			✓			✓	✓
UMR [27]				✓			✓			✓	(✓)[3]
VMR [28]	(✓)[4]	(✓)[2]	✓	✓			✓		✓	✓	(✓)[3]
Ours				✓	✓	✓	✓		✓	✓	✓

and viewpoint using SfM. U-CMR [11] extends CMR so that keypoint annotations are not required, but they require a template shape of the category beforehand. They also advocate for extensive view sampling (camera multiplex) to address the challenge of discovering the object viewpoint. UMR [27] replaces supervised keypoints with unsupervised part segmentations from SCOPS [18]. VMR [28] extends CMR and removes the symmetric shape assumption by constraining the base shape deformation with an As-Rigid-As-Possible (ARAP) [51] regularizer. Unsup3D [59] learns objects from still images, but only with limited viewpoint variation. CSM [25] and articulated CSM [26] learn to pose an externally-provided (articulated) 3D template of an object category to images while only using masks as supervision. We summarize the key differences in Table 1.

Adversarial learning can supplant the lack of multiple views of the same object. The idea is that the learned 3D model can be used to generate arbitrary views of an object, such that the discriminator network can provide supervisory signal to learn the geometry by telling whether the generated views are plausible or not. Examples include the works of [24, 2], PlatonicGAN [16], HoloGAN [37], BlockGAN [38], Shelf-supervised learning [64] and GRAF [50]. This type of approach does not require information about specific viewpoints, but does require to know the distribution of viewpoints in the training data (as the discriminator is sensitive to that). Overall, promising results can be achieved on synthetic data as well as a few real-life object categories, but general methods usually recover only very coarse 3D shapes or 3D feature volumes that are difficult to extract.

Reconstruction from Multiple Views and Videos. Most works using multiple views and videos focus on reconstructing individual instances of an object. Classic 3D reconstruction uses SfM pipelines from multiple views of a rigid scene, with pipelines such as KinectFusion [35] and DynamicFusion [36] integrating depth sensors for 3D reconstruction of dense static and deformable surfaces. NeRF [34] and its deformable extensions [44, 7, 55, 46, 40] such as D-NeRF [45] synthesize novel views from densely sampled multi-views of a static or mildly dynamic scene using a Neural Radiance Field, from which explicit 3D geometry can be further extracted. A more recent work, LASR [63], optimizes a single 3D deformable model on an individual video sequence.

Prior works on learning 3D categories form videos mostly focus on reconstruction of human bodies or faces [56, 1, 4, 21, 65, 6, 54]. However, they all require a shape prior, such as a parametric shape model [31]. [41] and [17] consider turn-table like videos to learn to reconstruct rigid object categories. VMR [28] introduces a test-time adaptation on individual videos that enforces temporal consistency of the shape and texture predictions obtained from a model pre-trained on object category images (apart from extending CMR as explained above). In contrast, our method learns a 3D shape model of a deformable object category from videos, which allows for single image reconstruction at inference.

3 Method

Our goal is to learn 3D shapes for a deformable object category from a collection of videos clips. Specifically, given a dataset of short video clips of the objects captured with stationary cameras, we would like to train a reconstruction model that takes as input a single image of an object and predicts the 3D shape, texture and articulated 3D pose of it. Fig. 2 gives an overview of the training pipeline.

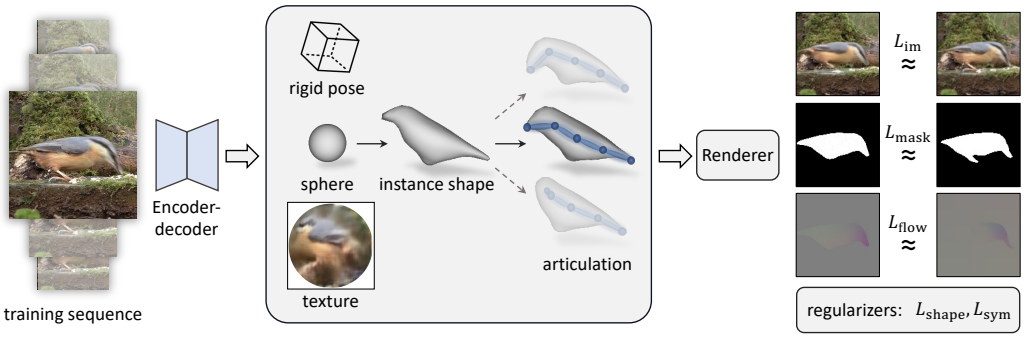


Figure 2: **Training Pipeline.** Form a single frame of a video, we predict the 3D pose, shape and texture of the object. The shape is further disentangled into category shape, instance shape and deformation using bone skinning. Using a differentiable rendering step, we can train the model end-to-end by reconstructing the image and by enforcing temporal consistencies.

3.1 Photo-Geometric Auto-Encoding

Our reconstruction model takes in a single frame $I \in \mathbb{R}^{3\times H\times W}$ from a video sequence and predicts the articulated 3D shape, texture and rigid pose of the object using three networks, denoted by f_S, f_T and f_P respectively. This information is then recombined to generate (render) an image of the object, which can be compared to the input video frames for supervision.

The shape is given by a triangulated mesh with fixed connectivity and variable vertex positions $V \in \mathbb{R}^{3\times K}$. We distinguish instance-specific shape variations from frame-specific articulations by obtaining V in two steps, described next.

Instance-Specific Deformation. The first step is to reconstruct the shape V_{ins} of a specific video object instance in a canonical 'rest pose'. This accounts for the fact that different object instances (e.g., different birds) have similar but different shapes. The shape is given by:

$$V_{\text{ins}} = V_{\text{cat}} + \Delta V_{\text{ins}}, \tag{1}$$

where V_{cat} is a learned category-specific template and ΔV_{ins} is the instance-specific shape variation.

We also assume that the object has a bilateral symmetry [59]. Vertices in this instance mesh are thus coupled in symmetric pairs by a permutation matrix $\Pi \in \{0,1\}^{K\times K}$ (such that $\Pi^2 = I$). Furthermore, the bilateral symmetry plane of the object is mapped to the yz-plane in canonical space (an arbitrary choice). With this, a mesh V is symmetric if, and only if, $V\Pi = \text{diag}(-1,1,1)V$. In practice, we impose this constraint on both V_{cat} and ΔV_{ins} storing only half of the vertices and inferring the other half using this equation.

Articulation. In order to account for the effect of articulation, the object's rest shape V_{ins} is further deformed by means of a *posing function* $V = g(V_{\text{ins}}, \xi)$ where ξ are the pose parameters. We model this function by means of a small number of virtual 'bones'. Let $\bar{V}_i$ be the homogeneous coordinates of a particular mesh vertex relative to a bone $b \in \{1,\dots,B\}$ to which it is rigidly attached, $\pi(b)$ the *parent* of bone b in the kinematic tree, and $\mathbf{J}_b$ the location of the joint between the two bones expressed relative to the parent and let ξ_b be the parameters of the joint rotation. Then, we can express the vertex relative to the parent bone as $g_b(\xi)\bar{V}_i$ where:

$$g_b(\xi) = \begin{bmatrix} R_{\xi_b} & \mathbf{J}_b \\ 0 & 1 \end{bmatrix}.$$

Applying this construction recursively along the kinematic tree, we obtain the overall bone transformation $G_b = G_{\pi(b)} \circ g_b$, where the base case G_1 is the pose of the root joint w.r.t. the world. With this, the location of the vertex in the world is given by $V_i = G_b(\xi)\bar{V}_i$.

By inverting these relations, given the mesh V_{ins} at rest pose ξ_0, we can express its vertices relative each bone as $\bar{V}_i = G_b(\xi_0)^{-1}[V_{\text{ins}}]_i$. As commonly done [31], we allow each vertex i to associate softly to one or more bones b via weights w_{bi}, obtaining the smoothed posing (or *skinning*) function:

$$V_i = \left(\sum_{b=1}^{B} w_{ki} G_b(\xi) G_b(\xi_0)^{-1}\right) [V_{\text{ins}}]_i. \tag{2}$$

Initialization. Unlike prior work such as [11], we do not assume that a category-specific, 3D template is available. Instead, we simply take a sphere as a base mesh V_{base} and optimize a set of vertex deformations $\Delta V_{\text{cat}} \in \mathbb{R}^{3\times K}$ as trainable parameters. The category shape is then obtained by deforming the base shape as $V_{\text{cat}} = V_{\text{base}} + \Delta V_{\text{cat}}$. We also require to define the bone structure. Given the shape corresponding to the rest pose (after one epoch of training in a pose-free manner), we define a fixed number of bones inside the mesh that we set to lie on two line segments going from the two most extreme point of the mesh to the center. We then divide each line segment into equally-sized parts that define the origin and the length of each bone.

Shape and Pose Predictors. The final shape of the object V in each frame is thus expressed as:

$$V = g(V_{\text{ins}}, \xi) = g(V_{\text{base}} + \Delta V_{\text{cat}} + \Delta V_{\text{ins}}, \xi), \tag{3}$$

where V_{base} is fixed (a sphere), the template ΔV_{cat} is the same for all objects and treated as a vector of learnable parameters and the other quantities are predicted by networks from each input frame I. We predict the rigid pose of the object by means of a *rigid pose network* f_P as $(\xi_1, \mathbf{J}_1) = f_P(I|\theta)$. Recall that parameter ξ_1 corresponds to the root bone rotation, and $\mathbf{J}_1$ to its translation w.r.t. the world reference frame; together, they thus express the rigid component of the object pose. The instance-specific deformation and articulation parameters, ΔV_{ins} and ξ, are instead predicted by a *shape network*, also from the input frame: $(\Delta V_{\text{ins}}, \xi_{2:B}) = f_S(I|\theta)$.

Appearance Model and Differentiable Rendering. We represent the texture of the object using a texture map $T \in \mathbb{R}^{3\times H_T \times W_T}$. The vertices of the base mesh V_{base} are assigned to fixed texture uv-coordinates and the texture inherits the symmetry of the base mesh. Formally, this means that the mesh vertices $V_{\text{base}} \in \mathbb{R}^{3\times K}$ are mapped to uv-coordinates $U \in \mathbb{R}^{2\times K}$ such that $U\Pi = U$.

The texture is also predicted from the video frame I by a *texture network* f_T as $T = f_T(I|\theta)$. Given the posed mesh V and the texture T, we can then *render* an image $\hat{I} = \mathcal{R}(V, T)$ of the object using standard perspective-correct texture mapping with barycentric coordinates. We also render the 2D mask of the object as $\hat{M} = \mathcal{R}(V)$. Specifically, we use the PyTorch3D differentiable mesh renderer [47]. We overlay the rendered mesh to the background of the object. Since the camera is fixed, the latter can easily be extracted from frames where the object is not detected, which are then averaged (an alternative to restricting the photometric loss described below to the masked pixels).

3.2 Learning from Videos

Our goal is to learn the reconstruction model from a collection of video sequences $\mathcal{S} = \{S_i\}_{i=1}^{|\mathcal{S}|}$, where each sequence S_i consists of frames $S_i = (I_{it}, M_{it})_{t=1}^{|S_i|}$ cropped around object instances. Here, i denotes the sequence index and t the frame index (time). These sequences are obtained by pre-processing the videos using an off-the-shelf instance segmentation technique (Mask R-CNN [14]).

Temporal Consistency. One important property of the frames in a video track is that they show the *same* object instances and thus share the same instance-specific shape $V_{\text{ins},i}$ and texture T_i. Recall, however, that different shapes $V_{\text{ins},i}$ and textures T_i are predicted from individual frames I_{it} by networks f_S and f_T. We encourage these predictions to be consistent (constant) for a sequence by randomly replacing them with their sequence-wise averages $\Delta V_{\text{ins},i} = \frac{1}{|S_i|}\sum_{t=1}^{|S_i|} \Delta V_{\text{ins},it}$ and $T_i = \frac{1}{|S_i|}\sum_{t=1}^{|S_i|} T_{it}$ for rendering. The idea is that inconsistent textures will lead to blurry averages and thus higher reconstruction error. This encourages temporally consistent textures.

Rendering Losses. Given a training image I_{it} and a corresponding mask M_{it}, we first consider the image reconstruction loss:

$$L_{\text{im},it} = \|\hat{I}_{it} - I_{it}\|_1, \;\; \text{where} \;\; \hat{I}_{it} = \mathcal{R}(V_{it}, T_i) \;\; \text{and} \;\; V_{it} = g(V_{\text{base}} + \Delta V_{\text{cat}} + \Delta V_{\text{ins},i}, \xi_{it}). \tag{4}$$

As explained above, the instance shape $\Delta V_{\text{ins},i}$ and texture T_i are sequence averages, independent of the time index t. We also define a similar loss for the rendered mask:

$$L_{\text{mask},it} = \lambda_{\text{m}}\|\hat{M}_{it} - M_{it}\|_2^2 + \lambda_{\text{dt}}\|\,\text{dt}(M_{it}) \odot \hat{M}_{it}\|_1, \;\; \text{where} \;\; \hat{M}_{it} = \mathcal{R}(V_{it}), \tag{5}$$

$\text{dt}(\cdot)$ is the distance transform of the mask, $\odot$ denotes Hadamard product, and λ_{m} and λ_{dt} are the balancing weights. Additionally, we also use a perceptual loss [66] between image and reconstruction that helps to recover sharper images and textures.

Optical Flow. Another advantage of videos is that 2D correspondences between consecutive video frames can be estimated accurately and robustly using optical flows and can be used as a supervisory signal. To this end, let $F_{it} \in \mathbb{R}^{H\times W\times 2}$ be the optical flow image between frames t and $t+1$ (we use the off-the-shelf RAFT model [52]). We define an *optical flow loss*:

$$L_{\text{flow},it} = \|\hat{F}_{it} - F_{it}\|_2^2, \tag{6}$$

where $\hat{F}_{it}$ is the flow image derived from the predicted geometry. Computing the flow for the mesh vertices is easy, but this needs to be interpolated for all image pixels. We do this by 'recycling' the differentiable renderer. First, we compute the 2D displacement for the shape vertices projected to image plane $\delta_{it} = \pi(V_{i,t+1} - V_{it}) \in \mathbb{R}^{2\times K}$. We can then use the renderer to render an optical flow image by simply treating these 2D displacement vectors as vertex colors: $\hat{F}_{it} = \mathcal{R}(V_{it}, \delta_{it})$.

3.3 Additional Regularization and Overall Objective

Posing Symmetry. We have assumed that objects have a bilateral symmetry, which is easily enforced in canonical rest pose. However, we can also enforce that the *space of poses* is symmetric too [53].[2] To this end, let ηI denote an image flipped horizontally. Furthermore, overload the same operator so that its action on a pose $\eta\xi$ is to flip the pose w.r.t. the horizontal axis (formally $G_b(\eta\xi) = \text{diag}(-1,1,1)G_{\Pi_b}(\xi)\,\text{diag}(-1,1,1)$ where Π_b is the symmetric counterpart of bone b). Then, we minimize the loss:

$$L_{\text{sym},it} = \|f_P(\eta I_{it}) - \eta(f_P(I_{it}))\|_2^2. \tag{7}$$

In our experiments, since the deformation of the birds is rather simple, we only impose this constraint on the rigid pose (*i.e.* $b = 1$).

Geometric Smoothing. We also use smoothing losses for the 3D geometry that are typical in the computer graphics literature. To prevent the instance-specific shapes from unreasonable deviations from the category prior, we add an As-Rigid-As-Possible (ARAP) regularizer [51] between the prior shape and the instance shapes:

$$L_{\text{ins},i} = \sum_{k=1}^{K} \min_{R\in SO(3)} \sum_{j\in\mathcal{N}_k} w_{jk}\|([V_{\text{ins},i}]_j - [V_{\text{ins},i}]_k) - R([V_{\text{cat},i}]_j - [V_{\text{cat},i}]_k)\|^2, \tag{8}$$

where $\mathcal{N}_k$ is the fan of vertices around vertex k. ARAP thus encourages triangle fans to deform rigidly. We also add a Laplacian smoothing loss and a normal consistency loss on the instance shapes:

$$L_{\text{Lap},i} = \sum_{k=1}^{K} \left\| [V_{\text{ins},i}]_k - \frac{1}{|\mathcal{N}_k|}\sum_{j\in\mathcal{N}_k} [V_{\text{ins},i}]_j \right\|^2, \quad L_{\text{nrm},i} = \sum_{f\in\mathcal{F}}\sum_{f'\in\mathcal{N}_f} \left(1 - \frac{\mathbf{n}_{if}\cdot\mathbf{n}_{if'}}{\|\mathbf{n}_{if}\|\|\mathbf{n}_{if'}\|}\right), \tag{9}$$

where $\mathbf{n}_{if}$ is the normal of face f in $V_{\text{ins},i}$ and $\mathcal{N}_f$ are the faces adjacent to f in the mesh. We sum the three geometric losses in the shape regularizer $L_{\text{shape},i} = \lambda_{\text{ins}}L_{\text{ins},i} + \lambda_{\text{Lap}}L_{\text{Lap},i} + \lambda_{\text{nrm}}L_{\text{nrm},i}$.

Overall Learning Objective. The final loss is the weighted sum of individual terms:

$$L = \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\left(L_{\text{shape},i} + \frac{1}{|\mathcal{S}_i|}\sum_{t=1}^{|S_i|}(L_{\text{im},it} + L_{\text{mask},it} + \lambda_{\text{flow}}L_{\text{flow},it} + \lambda_{\text{sym}}L_{\text{sym},it})\right). \tag{10}$$

4 Experiments

4.1 Dataset and Implementation Details

Dataset. We extract a large number of short video clips of birds from YouTube, searching for "bird videos for cats".[3] Mask R-CNN [14] is then used to detect and segment bird instances and the videos are automatically split into short clips, each containing a single bird. The images and the masks are cropped around the birds and resized to 128×128 for training. We also run the

[2] Note the individual poses are not symmetric, but we can find a symmetric counterpart for any given pose.

[3] All clips are extracted from an 8h video by Paul Dinning available at https://youtu.be/xbs7FT7dXYc.

Figure 3: **Qualitative Examples.** We show multiple views of the reconstructed mesh together with a textured view and animated version of the bird that we obtained by rotating the learned bones. We find that the model is able to recover the shape well even when seen from novel viewpoints. The animation is able to generate believable poses.

Table 2: **Mask Reprojection IoU.** We measure the shape reconstruction quality and temporal consistency via the mean intersection over union (mIoU) of reprojected masks. We take the predicted shape at frame t and reproject it with the camera pose of frame $t + \Delta t$. The higher the better.

frame offset	$\Delta t = 0$	$\Delta t = 5$	$\Delta t = 10$
CMR [20] (finetuned)	0.786 ± 0.131	0.759 ± 0.135	0.749 ± 0.135
U-CMR [11] (finetuned)	0.799 ± 0.059	0.786 ± 0.067	0.782 ± 0.069
VMR [28] (finetuned)	0.821 ± 0.068	0.793 ± 0.080	0.780 ± 0.083
UMR [27] (finetuned)	0.852 ± 0.042	0.818 ± 0.067	0.805 ± 0.070
UMR [27] (from scratch)	0.839 ± 0.042	0.803 ± 0.067	0.789 ± 0.071
Ours (articulation fixed)	0.871 ± 0.065	0.835 ± 0.083	0.820 ± 0.086
Ours (articulation transferred)	0.871 ± 0.065	0.859 ± 0.068	0.853 ± 0.068

off-the-shelf RAFT model [52] on the full frames to estimate optical flow between consecutive frames, and account for the cropping and resizing to obtain the correct optical flow for the crops. With this procedure, we collected $2,269$ sequences with paired image, mask and flow, each containing 16 to a few hundred frames, totaling $170,755$ frames. We randomly split them into $2,097$ training and 172 testing sequences.

Implementation Details. Our reconstruction model is implemented using three neural networks (f_S, f_P, f_T) as well as a set of of trainable parameters for the categorical prior shape ΔV_{cat}. The shape network f_S and the rigid pose network f_P are simple encoders with downsampling convolutional layers that take in an image and predict vertex deformations ΔV_{ins}, skinning parameters $\xi_{2:B}$, and rigid pose ξ_1 and $\mathbf{J}_1$ as flattened vectors. The texture network f_T is an encoder-decoder that predicts the texture map T from an image. We use Adam optimizers with a learning rate of 0.0001 for all networks, and a learning rate 0.01 for the category shape parameters ΔV_{cat}. All images resized to 128×128. We use a symmetric ico-sphere with 642 vertices and $1,280$ triangles as the initial mesh. For each training iteration, we randomly sample 8 consecutive frames from 8 sequences. The models are trained from scratch for 10 epochs with which takes roughly 3 days on one NVIDIA RTX-6000 GPU. During the first epoch we disable the instance shape and frame-specific deformation to learn a reasonable prior shape first. All details are included in the supplementary material.

4.2 Qualitative Results

Figure 3 shows single frame reconstruction results obtained from our model. Note that videos are no longer required during inference. Despite not requiring any explicit 3D, viewpoint or keypoint supervision, our model learns to reconstruct accurate 3D shapes from only monocular training videos. We can animate the reconstructed 3D bird with our skinning model by adjusting the bone rotations.

4.3 Comparisons with State-of-the-Art Methods

We compare our model with a number of state-of-the-art learning-based reconstruction methods, including CMR [20], U-CMR [11], UMR [27] and VMR [28]. CMR requires 2D keypoint annotations for initializing the 3D shape and viewpoints and also for the training loss. U-CMR removes keypoint supervision but requires a 3D template shape to begin with, and UMR replaces that with self-supervised part segmentation maps using SCOPS [18] which requires supervised ImageNet

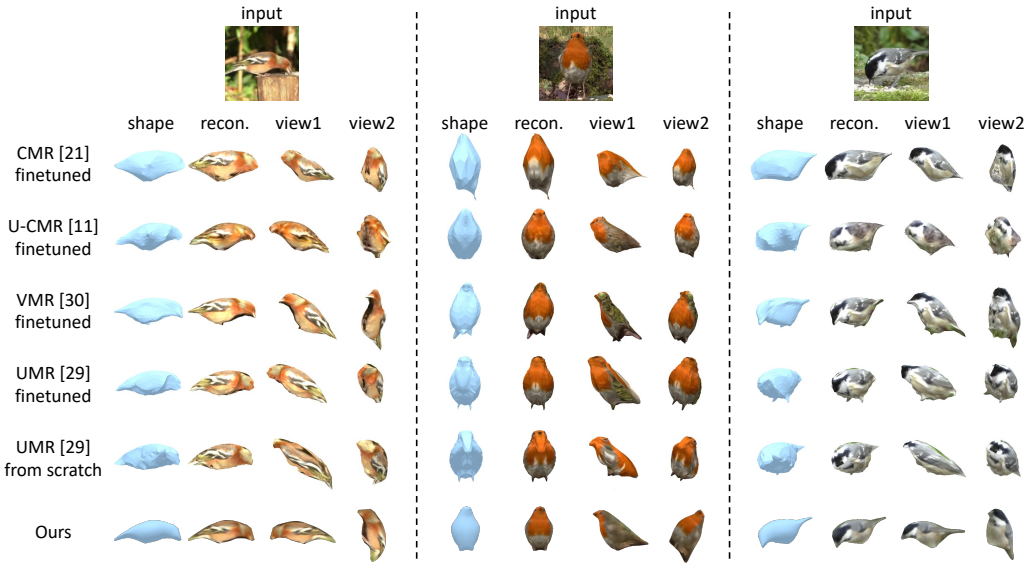
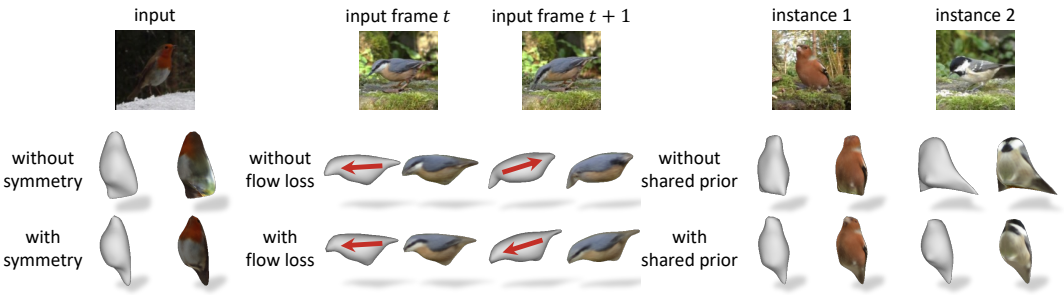


Figure 4: **Visual Comparison.** We compare our reconstructions to state-of-the-art methods by finetuning their models on our dataset. Our method consistently reconstructs reasonable 3D shapes, whereas others produce poor shapes for certain input poses.



(a) Ablation on symmetry (b) Ablation on flow loss (c) Ablation on learned category prior

Figure 5: **Ablation.** We train our model without some of the key components: a) symmetry assumption, b) optical flow and c) a learned base shape across sequences. All parts fulfill a specific role in the reconstruction pipeline that regularize the learned 3D shape and pose.

pretraining. VMR [28] allows for deformations but it requires the same level of supervision as CMR. VMR also introduces an optional test-time optimization procedure that adapts their pre-trained model on individual test videos to improve temporal consistency. The authors also present a self-supervised variant of VMR that uses only segmentation masks for supervision but produces worse results, and we were not able to reproduce it. For evaluation, we consider the first and the second (in Section 6.4) versions of VMR. We finetuned the trained models for all four methods on our dataset. We also trained UMR from scratch with SCOPS predictions obtained from a pre-trained SCOPS model.

Quantitative Comparisons. We evaluate the methods on our video test set. Since we do not have ground-truth 3D shape for direct evaluation, we measure reconstruction quality via a mask reprojection accuracy from one frame to another, using the object masks predicted by Mask R-CNN [14] as the pseudo ground-truth. For each test sequence, we predict the shape at frame t and render the object mask from the pose at frame $t + \Delta t$ with an offset Δt of 0, 5 and 10 frames. We then compute the mean Intersection over Union (mIoU) between the rendered masks and the ground-truth masks over all frames. This metric not only measures the accuracy of the predicted shape, but also its consistency over time. Table 2 summarizes the results, which suggests that our model achieves both better shape reconstruction and temporal consistency. We also compute the metrics on our model with frame-specific deformations predicted at frame $t + \Delta t$ applied to the shape predicted at frame t. This further improves the mask reprojection IoU, which confirms that our model learns correct

frame-specific deformations. Other methods overfit their projected shape to a single frame, resulting in a larger decrease in reprojection accuracy with increasing frame offset Δt.

Qualitative Comparisons. Fig. 4 shows a qualitative comparison of different methods. Among previous methods, CMR produces the most robust reconstructions as it relies on keypoint supervision, but still performs poorly for challenging poses, such as frontal views. Our method reconstructs accurate shape and pose, despite not using keypoint or template supervision. Moreover, the reconstructions obtained by our method are more consistent temporally compared to other methods. We refer the reader to the animations on our project website for more results. Note that our model is trained on 128×128 images, whereas other methods train on 256×256 images and, except U-CMR, sample the texture from the input image, hence the difference in the texture quality.

4.4 Ablation and Analysis

Symmetry. To understand the effect of symmetry, we ablate the model by training it without the symmetry constraint on the rest-pose shape and on the texture. Figure 5a shows an example of the comparison with our full model. The model without symmetry produces unrealistic shapes indicating that symmetry is a useful prior, even when learning deformable shapes.

Optical Flow. We analyze the effect of the optical flow loss by comparing the results with and without the flow loss. An example is shown in Fig. 5b. Without the flow loss, the model produces inconsistent reconstructions in ambiguous poses and tends to confuse head and tail of the birds. The optical flow helps resolve such ambiguities and improves temporal consistency.

Prior Shape. We train another model without the learned category prior shape, predicting individual shapes for each bird. The resulting reconstructions are inconsistent across different instances, shown in Fig. 5c. This suggests that the full model is able to leverage shape prior of the whole category to predict the instance shapes, which is a major benefit of learning in a reconstruction pipeline.

4.5 Limitations

In this section we identify several weaknesses of our model. Due to the nature of the videos—seeds in front of the camera—the vast majority of clips show the birds from front and side positions. This results in degraded performance when the bird is facing away from the camera and the pose predictions rarely exceed $\pm 90\,\mathrm{deg}$.

In terms of supervision, we still require segmentation masks obtained from the off-the shelf model and their quality affects the fidelity of our reconstructions. Thus, similar to comparable methods, our reconstructions do not capture fine details such as legs and the beak. The texture prediction sometimes results in low quality reconstructions especially when the input image is affected by motion blur.

In view of extending this method to other animal categories such as dogs or horses, birds have a rather simple geometry and deformation structure, which might not generalize well.

5 Conclusions

We have presented a method to learn articulated 3D representations of deformable objects from monocular videos without explicit geometric supervision, such as keypoints, viewpoint or template shapes. The resulting 3D meshes are temporally consistent and can be animated. The method can be trained from YouTube videos and only needs off-the-shelf object detection and optical flow models for data preprocessing. In the future, we hope to extend the presented approach to more categories, such as horses and cats, as training data is readily available for those as well.

Broader Impact Our work focuses on 3D reconstruction of deformable objects from monocular videos. We expect this work to be most useful for object categories that do not have sophisticated 3D ground-truth annotations and 3D shape models. This is mainly the case for animals, as shown for birds here, and thus the work can potentially be of use in behavioral research of animals in the wild. Our dataset does not contain any humans, only birds in nature background and thus does not violate personal privacy of individuals. The dataset has some bias as according to the author, the video clips were recorded in the UK and thus only depict bird species that appear there. Overall, we expect this work to impact mostly the research community with very little impact on society in the short term.

Acknowledgments

We thank Xueting Li for sharing the code for VMR with us. Shangzhe Wu is supported by Facebook Research. Tomas Jakab is supported by Clarendon Scholarship. Christian Rupprecht is supported by Innovate UK (project 71653) on behalf of UK Research and Innovation (UKRI) and by the European Research Council (ERC) IDIU-638009. Andrea Vedaldi is supported by European Research Council (ERC) IDIU-638009.

References

[1] Anurag Arnab, Carl Doersch, and Andrew Zisserman. Exploiting temporal context for 3d human pose estimation in the wild. In *CVPR*, pages 3395–3404, 2019.

[2] Ching-Hang Chen, Ambrish Tyagi, Amit Agrawal, Dylan Drover, Rohith MV, Stefan Stojanov, and James M. Rehg. Unsupervised 3d pose estimation with geometric self-supervision. In *CVPR*, 2019.

[3] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, 2016.

[4] Carl Doersch and Andrew Zisserman. Sim2real transfer learning for 3d human pose estimation: motion to the rescue. *NeurIPS*, 32:12949–12961, 2019.

[5] Olivier Faugeras and Quang-Tuan Luong. *The Geometry of Multiple Images*. MIT Press, 2001.

[6] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *ECCV*, pages 534–551, 2018.

[7] Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *CVPR*, 2021.

[8] Ravi Garg, B. G. Vijay Kumar, Gustavo Carneiro, and Ian D. Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016.

[9] Rohit Girdhar, David Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *ECCV*, 2016.

[10] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *ICCV*, 2019.

[11] Shubham Goel, Angjoo Kanazawa, and Jitendra Malik. Shape and viewpoint without keypoints. In *ECCV*, 2020.

[12] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *CVPR*, 2018.

[13] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, 2017.

[15] Paul Henderson and Vittorio Ferrari. Learning single-image 3d reconstruction by generative modelling of shape, pose and shading. *IJCV*, pages 1–20, 2019.

[16] Philipp Henzler, Niloy J. Mitra, and Tobias Ritschel. Escaping plato's cave using adversarial training: 3D shape from unstructured 2D image collections. In *ICCV*, 2019.

[17] Philipp Henzler, Jeremy Reizenstein, Patrick Labatut, Roman Shapovalov, Tobias Ritschel, Andrea Vedaldi, and David Novotny. Unsupervised learning of 3d object categories from videos in the wild. In *CVPR*, 2021.

[18] Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz. SCOPS: self-supervised co-part segmentation. In *CVPR*, 2019.

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

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

[21] Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In *CVPR*, pages 5614–5623, 2019.

[22] Hiroharu Kato and Tatsuya Harada. Learning view priors for single-view 3d reconstruction. In *CVPR*, pages 9778–9787, 2019.

[23] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *CVPR*, 2018.

[24] Yasunori Kudo, Keisuke Ogaki, Yusuke Matsui, and Yuri Odagiri. Unsupervised adversarial learning of 3D human pose from 2D joint locations. *arXiv preprint arXiv:1803.08244*, 2018.

[25] Nilesh Kulkarni, Abhinav Gupta, and Shubham Tulsiani. Canonical surface mapping via geometric cycle consistency. In *ICCV*, 2019.

[26] Nilesh Kulkarni, Abhinav Gupta, David F. Fouhey, and Shubham Tulsiani. Articulation-aware canonical surface mapping. In *CVPR*, pages 449–458, 2020.

[27] Xueting Li, Sifei Liu, Kihwan Kim, Shalini De Mello, Varun Jampani, Ming-Hsuan Yang, and Jan Kautz. Self-supervised single-view 3D reconstruction via semantic consistency. In *ECCV*, 2020.

[28] Xueting Li, Sifei Liu, Shalini De Mello, Kihwan Kim, Xiaolong Wang, Ming-Hsuan Yang, and Jan Kautz. Online adaptation for consistent mesh reconstruction in the wild. In *NeurIPS*, 2020.

[29] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T. Freeman. Learning the depths of moving people by watching frozen people. In *CVPR*, 2019.

[30] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *ICCV*, 2019.

[31] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: a skinned multi-person linear model. *ACM TOG*, 2015.

[32] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, 2013.

[33] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019.

[34] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[35] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew W. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *Proc. ISMAR*, 2011.

[36] Richard A. Newcombe, Dieter Fox, and Steven M. Seitz. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *CVPR*, 2015.

[37] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *ICCV*, 2019.

[38] Thu Nguyen-Phuoc, Christian Richardt, Long Mai, Yong-Liang Yang, and Niloy J. Mitra. Blockgan: Learning 3d object-aware scene representations from unlabelled images. In *NeurIPS*, 2020.

[39] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *CVPR*, 2020.

[40] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. *arXiv preprint arXiv:2104.03110*, 2021.

[41] David Novotný, Diane Larlus, and Andrea Vedaldi. Learning 3D object categories by looking around them. In *ICCV*, 2017.

[42] Junyi Pan, Xiaoguang Han, Weikai Chen, Jiapeng Tang, and Kui Jia. Deep mesh reconstruction from single rgb images via topology modification networks. In *ICCV*, pages 9964–9973, 2019.

[43] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, June 2019.

[44] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo-Martin Brualla. Deformable neural radiance fields. *arXiv preprint arXiv:2011.12948*, 2020.

[45] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *CVPR*, 2020.

[46] Amit Raj, Michael Zollhoefer, Tomas Simon, Jason Saragih, Shunsuke Saito, James Hays, and Stephen Lombardi. Pva: Pixel-aligned volumetric avatars. *arXiv preprint arXiv:2101.02697*, 2020.

[47] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020.

[48] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *ICCV*, 2019.

[49] Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael Black. Learning to regress 3D face shape and expression from an image without 3D supervision. In *CVPR*, 2019.

[50] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *NeurIPS*, 2020.

[51] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, pages 109–116, 2007.

[52] Zachary Teed and Jia Deng. RAFT: recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020.

[53] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Modelling and unsupervised learning of symmetric deformable object categories. In *NeurIPS*, 2018.

[54] Luan Tran and Xiaoming Liu. On learning 3d face morphable model from in-the-wild images. *IEEE TPAMI*, 43(1):157–171, 2019.

[55] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. *arXiv preprint arXiv:2012.12247*, 2020.

[56] Hsiao-Yu Tung, Hsiao-Wei Tung, Ersin Yumer, and Katerina Fragkiadaki. Self-supervised learning of motion capture. In *NIPS*, 2017.

[57] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV*, 2018.

[58] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *NeurIPS*, pages 82–90, 2016.

[59] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3D objects from images in the wild. In *CVPR*, 2020.

[60] Shangzhe Wu, Ameesh Makadia, Jiajun Wu, Noah Snavely, Richard Tucker, and Angjoo Kanazawa. De-rendering the world's revolutionary artefacts. In *CVPR*, 2021.

[61] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018.

[62] Christopher Xie, Keunhong Park, Ricardo Martin-Brualla, and Matthew Brown. Fig-nerf: Figure-ground neural radiance fields for 3d object category modelling. *arXiv preprint arXiv:2104.08418*, 2021.

[63] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Huiwen Chang, Deva Ramanan, William T. Freeman, and Ce Liu. LASR: Learning articulated shape reconstruction from a monocular video. In *CVPR*, 2021.

[64] Yufei Ye, Shubham Tulsiani, and Abhinav Gupta. Shelf-supervised mesh prediction in the wild. In *CVPR*, 2021.

[65] Jason Y Zhang, Panna Felsen, Angjoo Kanazawa, and Jitendra Malik. Predicting 3d human dynamics from video. In *ICCV*, pages 7114–7123, 2019.

[66] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[67] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017.

[68] Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *CVPR*, 2017.

[69] Silvia Zuffi, Angjoo Kanazawa, Tanja Berger-Wolf, and Michael J. Black. Three-d safari: Learning to estimate zebra pose, shape, and texture from images "in the wild". In *ICCV*, 2019.

6 Supplementary Materials

6.1 Model Details

Shape and Texture. For the base shape, we use an ico-sphere with 642 vertices and 1,280 faces. Since the texture is symmetric, we compute the texture coordinates of both half spheres by unwrapping a half sphere onto the plane as illustrated in Fig. 6. Mathematically, this is computed as:

$$u = \frac{2 \arccos |x|}{\pi} \frac{z}{\sqrt{z^2 + y^2}}, \quad v = \frac{2 \arccos |x|}{\pi} \frac{y}{\sqrt{z^2 + y^2}}. \tag{11}$$

Rigid Pose. We represent the rigid rotation of the object using Euler angles about the xy-axes, with azimuth ranging from $-180°$ to $180°$ and elevation ranging from $-90°$ to $90°$, and disable the rotation about z-axis as the birds do not roll significantly on the ground. As a result, the body roll movement is factored into the articulation model, as illustrated by the 4-th principal component in the learned articulation model in Fig. 8. The translations in xyz-axes are capped at a range roughly corresponding to half of the image size.

Skinning Model. As described in the main paper, we estimate the bone structure using the two most extreme points of the mesh at its rest pose. The one with positive z coordinate is selected as the head end and the other one the tail end, ensuring consistent orientation of the bone structure. The rotation of individual bones is represented using Euler angles about the xyz-axes in its local coordinate frame, where the center of rotation is specified by the joint location.

Recall Eq. (2) of the main paper where we use skinning weights w_{bi} that associate each mesh vertex with the bones. We softly assign each mesh vertex with the bones based on its distances to the bones. The weights are defined as the inverse of the distance between a vertex $[V_{\text{ins}}]_i$ and a bone b at their rest pose. We normalize this distance for each vertex over all the bones with softmax function:

$$w_{bi} = \frac{e^{d_{bi}}}{\sum_{k=1}^{B} e^{d_{ki}}}, \quad d_{bi} = 1/(\min_{r \in [0,1]} \|[V_{\text{ins}}]_i - r\mathbf{s}_{b1} - (1-r)\mathbf{s}_{b2}\|_2^2 + \epsilon), \tag{12}$$

where $(\mathbf{s}_{b1}, \mathbf{s}_{b2})$ is the line segment defining the bone b at its rest pose and ϵ is a small number to avoid division by zero.

6.2 Network Architectures

All the network architectures are described in in Table 3 and Section 6.4. Abbreviations of the components are defined as follows:

- Conv$(c_{in}, c_{out}, k, s, p)$: 2D convolution with c_{in} input channels, c_{out} output channels, kernel size k, stride s and padding p
- Deconv$(c_{in}, c_{out}, k, s, p)$: 2D deconvolution with c_{in} input channels, c_{out} output channels, kernel size k, stride s and padding p
- Upsample(s): 2D nearest-neighbor upsampling with a scale factor s.
- GN(n): group normalization [61] with n groups
- LReLU(p): leaky ReLU [32] with a slope p

6.3 Training Details

The model is trained for 10 epochs, which takes three days on one NVIDIA RTX-6000 GPU. For each epoch, we iterate through the training set by densely sampling 8 short sequences in a batch for each iteration. The number of iterations in each epoch is therefore roughly the total number of frames in the training set, which is approximately 170k. Each sequence contains 8 consecutive frames.

During the first epoch, we disable instance-specific deformation and frame-specific articulation, and train the category-specific prior shape together with texture and rigid pose, in order to learn a prior shape over the entire category as well as rough texture and rigid pose estimations. After the second epoch, we decrease the weights of the shape regularizers to 1/5, in order for the model to refine the

Table 3: Architecture of the shape f_S and pose network f_P. The network follows a convolutional encoder structure. n is the number of parameters predicted by each network.

Encoder	Output size
Conv(3, 64, 4, 2, 1) + LReLU(0.2)	64×64
Conv(64, 128, 4, 2, 1) + LReLU(0.2)	32×32
Conv(128, 256, 4, 2, 1) + LReLU(0.2)	16×16
Conv(256, 512, 4, 2, 1) + LReLU(0.2)	8×8
Conv(512, 512, 4, 2, 1) + LReLU(0.2)	4×4
Conv(512, 128, 4, 1, 0) + ReLU	1×1
Conv(128, n, 1, 1, 0) $\rightarrow$ output	1×1

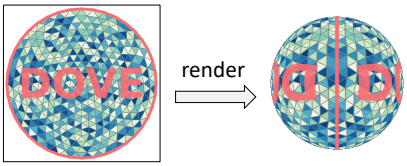


Figure 6: **Texture Mapping.** The texture is mapped from a circle in the texture to both, left and right side, of the initial sphere.

Table 4: Training details and hyper-parameter settings.

Parameter	Value/Range
Optimizer	Adam
Learning rate (f_S, f_P, f_T)	1×10^{-4}
Learning rate (ΔV_{ref})	1×10^{-2}
Number of iterations	170k
Number of sequences per batch	8
Number of frames per sequence	8
Loss weight λ_{m}	2
Loss weight λ_{dt}	0.5
Loss weight λ_{ins}	20
Loss weight λ_{Lap}	0.5
Loss weight λ_{nrm}	0.5
Loss weight λ_{flow}	100
Loss weight λ_{sym}	0.05
Input image size	128×128
Texture image size	128×128
Field of view (FOV)	25°
Camera location	$(0, 0, 10)$
Initial mesh center	$(0, 0, 0)$
Elevation angles	$(-90^\circ, 90^\circ)$
Azimuth angles	$(-180^\circ, 180^\circ)$
Number of bones	6

shape details. To better enforce temporal consistency, we randomly replace both the instance-specific shape and texture with the predictions averaged across the training sequence with a 50% probability.

We use Adam optimizers with a learning rate of 0.0001 for the networks and 0.01 for the trainable category shape parameters. The learning rates are decayed by a factor of 0.7 after each epoch starting from the third epoch (after the full deformable model is trained for one epoch).

6.4 Additional Results

Please refer to the **supplementary video** at dove3d.github.io for more qualitative results.

Shape Reconstruction Evaluation. As in Table 2 of the main paper, we evaluate the shape reconstruction quality and temporal consistency by re-projecting the shape of an object at frame t_0 to a future pose at frame $t_0 + \Delta t$. Figure 7 plots the mean Intersection over Union (mIoU) between the shape reprojections and the pseudo ground truth masks for various Δt. As the objects move over time, this estimates the consistency of the shape reconstructions across multiple views. We find that our model with articulation is able to correctly model the shape and deformation of the object over time. Moreover, the performance improvement gained by transferring the articulations also shows our model learns correct frame-specific articulations.

PCA Analysis on Learned Articulation Space. We analyze the learned articulated shape space using Principal Component Analysis (PCA). Figure 8 visualizes the first 6 principal components. The model learns meaningful articulations capturing typical bird movements.

Additional Reconstruction Results. Fig. 9 shows more bird reconstructions from various viewpoints. The model is robust against various input images, including frontal views and blurry images.

Texture Swapping and Animation. Our model reconstructs the birds in the canonical pose, where the shape and texture of different birds are aligned in the canonical representation. This allows us to easily edit the texture, for example swapping the texture with another bird, as shown in Fig. 10.

Moreover, with the learned articulation model, we can also easily animate the reconstructed birds in 3D, by rotating the bones, also illustrated Fig. 10.

Additional Qualitative Comparison. Fig. 11 provides a few more examples comparing the reconstruction results of our model and several state-of-the-art methods. Our model learns more accurate 3D shapes, despite not requiring explicit geometric supervision from keypoints, viewpiont or template shapes. More comparisons on entire video sequences are provided in the supplementary video.

Table 5: Architecture of the texture network f_T. The network follows an encoder-decoder structure.

Encoder	Output size
Conv(3, 64, 4, 2, 1) + GN(16) + LReLU(0.2)	64×64
Conv(64, 128, 4, 2, 1) + GN(32) + LReLU(0.2)	32×32
Conv(128, 256, 4, 2, 1) + GN(64) + LReLU(0.2)	16×16
Conv(256, 512, 4, 2, 1) + LReLU(0.2)	8×8
Conv(512, 512, 4, 2, 1) + LReLU(0.2)	4×4
Conv(512, 512, 4, 1, 0) + ReLU	1×1
Decoder	**Output size**
Deconv(512, 512, 4, 1, 0) + ReLU	4×4
Upsample(2) + Conv(512, 512, 3, 1, 1) + GN(128) + ReLU	8×8
Upsample(2) + Conv(512, 256, 3, 1, 1) + GN(64) + ReLU	16×16
Upsample(2) + Conv(256, 128, 3, 1, 1) + GN(32) + ReLU	32×32
Upsample(2) + Conv(128, 64, 3, 1, 1) + GN(16) + ReLU	64×64
Upsample(2) + Conv(64, 64, 3, 1, 1) + GN(16) + ReLU	128×128
Conv(64, 3, 5, 1, 2) + Sigmoid $\rightarrow$ output T	128×128

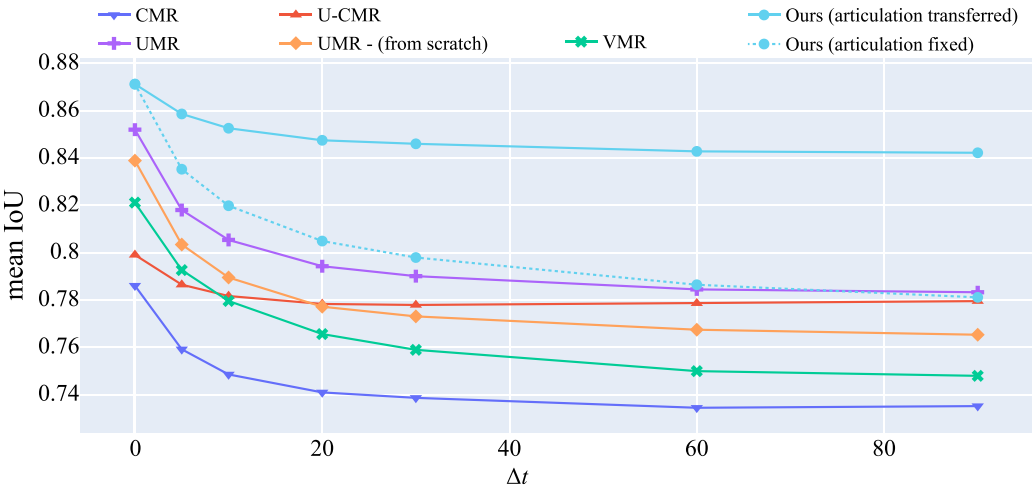


Figure 7: **Mask Reprojection IoU.** We project the shape predicted on frame t_0 to the pose predicted from a future frame $t_0 + \Delta t$ and measure the IoU with the ground truth mask at $t_0 + \Delta t$. Our method produces accurate shapes that are correctly projected to various poses in different frames, whereas others methods produce incorrect shapes indicated by significantly lower mask reprojection IoUs.

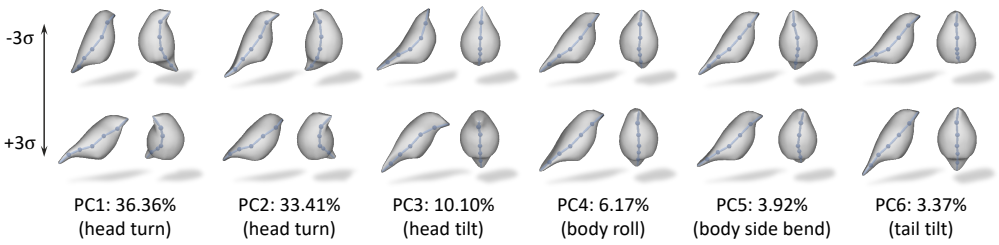


Figure 8: **PCA on Learned Articulations.** Our model learns meaningful articulations with the bone-based skinning model, including typical head and tail movements of birds.

Figure 9: **Additional Results.** We show the reconstructed birds from various viewpoints.

Figure 10: **Texture Swapping and Animation.** Top: since our model learns a canonical representation for all objects, we can easily swap the texture across different instances. Bottom: we can also easily animate the 3D birds using our learned articulation model.

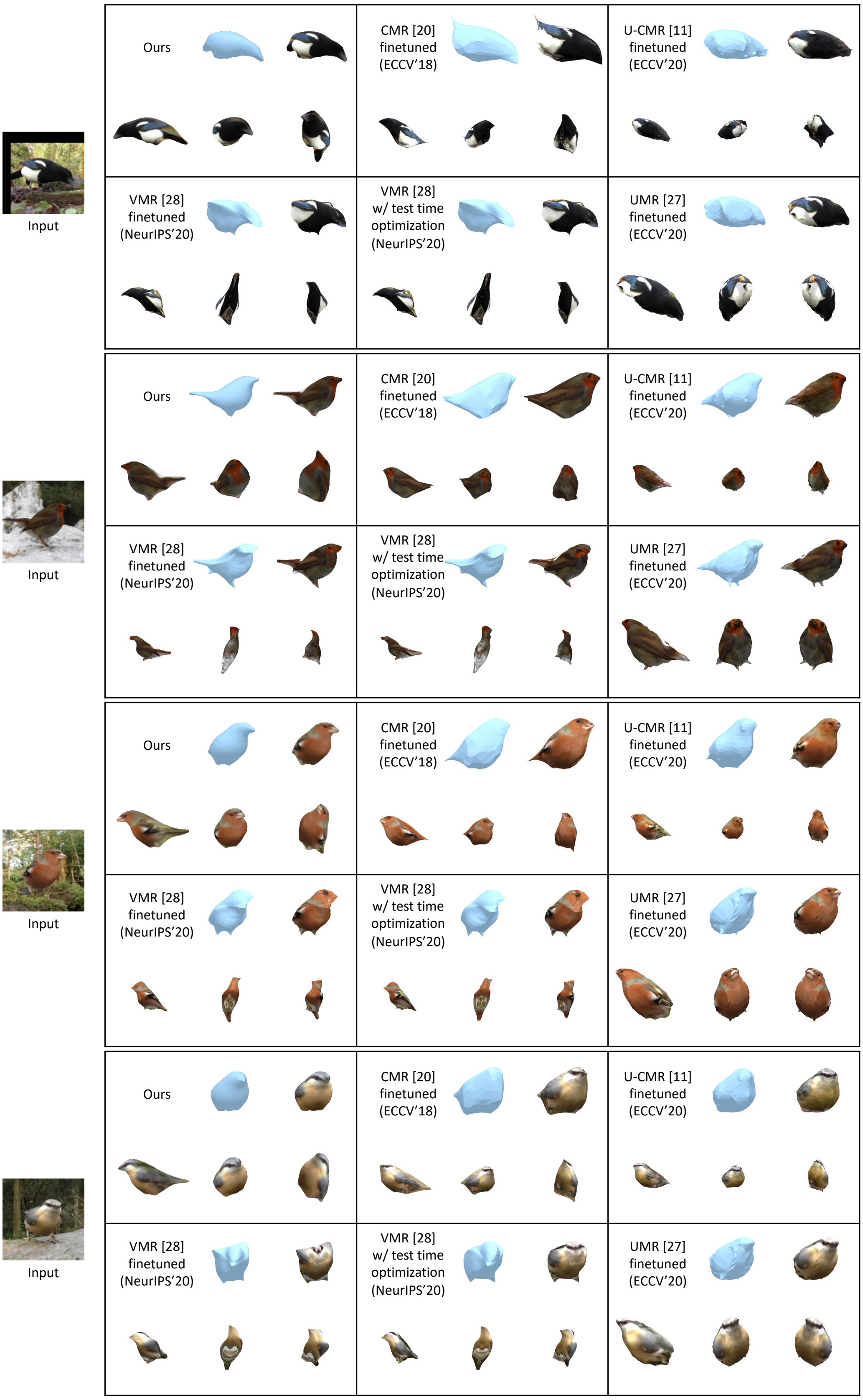


Figure 11: **Comparisons.** Our model learns more plausible 3D shapes compared to SOTA methods.

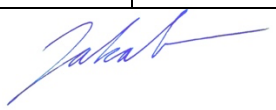
Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	DOVE: Learning Deformable 3D Objects by Watching Videos	
Publication Status	☐ Published ☑ Submitted for Publication	☐ Accepted for Publication ☐ Unpublished and unsubmitted work written in a manuscript style
Publication Details	Wu, S., Jakab, T., Rupprecht, C. and Vedaldi, A., 2021. DOVE: Learning Deformable 3D Objects by Watching Videos. *arXiv preprint arXiv:2107.10844*.	

Student Confirmation

Student Name:	Tomas Jakab		
Contribution to the Paper	• Development of the method and experiments • Optical flow pre-processing • Implementation of sampling-based learning pipeline • Implementation of deformation models – ARAP, skinning • Evaluation and comparison with baseline methods – CMR, UCMR, UMR, VMR • Manuscript • Visualizations, interactive demo • Supplemental material		
Signature 		Date	17. 09. 2021

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Andrea Vedaldi		
This is a faithful representation of Tomas' contributions.		
Signature 	Date	17 September 2021

This completed form should be included in the thesis, at the end of the relevant chapter.

Chapter 7

Summary and Impact

In this thesis, we have demonstrated how a machine can learn structural object representations using self-supervised learning. We achieved that by extending the autoencoding framework with engineered bottlenecks and regularizers designed to distill structural representations from images and 3D shapes. Overall we have presented four methods that span 2D and 3D object landmarks estimation, object deformation and category mesh reconstruction.

Chapter 3 introduces a method for self-supervised 2D object landmark estimation that achieved the state-of-the-art results for facial landmark estimation. The method was widely adopted and inspired many applications in image animation [105, 104], video prediction [82, 62], control [65, 82, 123, 8], robotic manipulation [94], object part and keypoints estimation [77, 76, 17, 101], and image-to-image translation [128].

Building on the findings from the previous work, chapter 4 presents a method that learns human-interpretable 2D object landmarks from unlabeled images and unpaired landmark prior. It does not require any supervised post-processing to align the predictions with annotations as other self-supervised methods. It achieves state-of-the-art performance for human pose estimation among methods that do not require labelled images for training.

Chapter 5 presents a method for shape control through automatically discovered 3D object landmarks that is intuitive and semantically consistent. The method also outperforms existing self-supervised approaches for 3D landmark estimation for 3D shapes.

Finally, in chapter 6 we introduce a method that estimates articulated 3D shapes from a single image by learning from monocular videos of deformable objects. The method does not need manual annotations in the form of keypoints, viewpoints or template shapes as required by most of the comparable approaches. Instead, it uti-

lizes temporal consistency presented in videos and optical flow estimated by generic algorithms.

Bibliography

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014.

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

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

[4] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017.

[5] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018.

[6] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4733–4742, 2016.

[7] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[8] Boyuan Chen, Pieter Abbeel, and Deepak Pathak. Unsupervised learning of visual 3d keypoints for control. In *International Conference on Machine Learning*, pages 1539–1549. PMLR, 2021.

[9] Ching-Hang Chen, Ambrish Tyagi, Amit Agrawal, Dylan Drover, Rohith Mv, Stefan Stojanov, and James M Rehg. Unsupervised 3d pose estimation with geometric self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5714–5724, 2019.

[10] Hui Chen and Bir Bhanu. 3d free-form object recognition in range images using local surface patches. *Pattern Recognition Letters*, 28(10):1252–1262, 2007.

[11] Nenglun Chen, Lingjie Liu, Zhiming Cui, Runnan Chen, Duygu Ceylan, Changhe Tu, and Wenping Wang. Unsupervised learning of intrinsic structural representation points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9121–9130, 2020.

[12] Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. *Advances in Neural Information Processing Systems*, 32:9609–9619, 2019.

[13] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[14] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018.

[15] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

[16] Chitra Dorai and Anil K. Jain. Cosmos-a representation scheme for 3d free-form objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1115–1130, 1997.

[17] Aysegul Dundar, Kevin Shih, Animesh Garg, Robert Pottorff, Andrew Tao, and Bryan Catanzaro. Unsupervised disentanglement of pose, appearance and background from images and videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[18] Olivier Faugeras and Quang-Tuan Luong. *The Geometry of Multiple Images*. MIT Press, 2001.

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

[20] Clara Fernandez-Labrador, Ajad Chhatkuli, Danda Pani Paudel, Jose J Guerrero, Cédric Demonceaux, and Luc Van Gool. Unsupervised learning of category-specific symmetric 3d keypoints from point sets. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pages 546–563. Springer, 2020.

[21] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3636–3645, 2017.

[22] Matheus Gadelha, Giorgio Gori, Duygu Ceylan, Radomir Mech, Nathan Carr, Tamy Boubekeur, Rui Wang, and Subhransu Maji. Learning generative models of shape handles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

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

[24] Ravi Garg, B. G. Vijay Kumar, Gustavo Carneiro, and Ian D. Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[25] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[26] Rohit Girdhar, David Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[27] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[28] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 270–279, 2017.

[29] Shubham Goel, Angjoo Kanazawa, and Jitendra Malik. Shape and viewpoint without keypoints. In *European Conference on Computer Vision*, pages 88–104. Springer, 2020.

[30] Shubham Goel, Angjoo Kanazawa, and Jitendra Malik. Shape and viewpoint without keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[31] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, 2014.

[32] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[33] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. Unsupervised cycle-consistent deformation for shape matching. In *Computer Graphics Forum*, volume 38, pages 123–133. Wiley Online Library, 2019.

[34] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Learning to read by spelling: Towards unsupervised text recognition. In *Proc. ICVGIP*, 2018.

[35] Rana Hanocka, Noa Fish, Zhenhua Wang, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Alignet: Partial-shape agnostic alignment via unsupervised learning. *ACM Transactions on Graphics (TOG)*, 38(1):1–14, 2018.

[36] Zekun Hao, Hadar Averbuch-Elor, Noah Snavely, and Serge Belongie. Dualsdf: Semantic shape manipulation using a two-level representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7631–7641, 2020.

[37] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

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

[39] Paul Henderson and Vittorio Ferrari. Learning single-image 3d reconstruction by generative modelling of shape, pose and shading. *IJCV*, pages 1–20, 2019.

[40] Philipp Henzler, Niloy Mitra, and Tobias Ritschel. Escaping plato's cave using adversarial training: 3d shape from unstructured 2d image collections. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[41] Philipp Henzler, Jeremy Reizenstein, Patrick Labatut, Roman Shapovalov, Tobias Ritschel, Andrea Vedaldi, and David Novotny. Unsupervised learning of 3d object categories from videos in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4700–4709, 2021.

[42] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[43] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017.

[44] Haibin Huang, Evangelos Kalogerakis, Siddhartha Chaudhuri, Duygu Ceylan, Vladimir G Kim, and Ersin Yumer. Learning local shape descriptors from part correspondences with multiview convolutional networks. *ACM Transactions on Graphics (TOG)*, 37(1):1–14, 2017.

[45] Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz. Scops: Self-supervised co-part segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 869–878, 2019.

[46] Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz. SCOPS: self-supervised co-part segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[47] Eldar Insafutdinov and Alexey Dosovitskiy. Unsupervised learning of shape and pose with differentiable point clouds. In *Advances in neural information processing systems*, 2018.

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

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

[50] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[51] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *Advances in neural information processing systems*, 2018.

[52] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Self-supervised learning of interpretable keypoints from unlabelled videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8787–8797, 2020.

[53] Tomas Jakab, Richard Tucker, Ameesh Makadia, Jiajun Wu, Noah Snavely, and Angjoo Kanazawa. Keypointdeformer: Unsupervised 3d keypoint discovery for shape control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[54] Sam Johnson and Mark Everingham. Learning effective human pose estimation from inaccurate annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1465–1472. IEEE, 2011.

[55] Tao Ju, Scott Schaefer, and Joe Warren. Mean value coordinates for closed triangular meshes. In *SIGGRAPH*, pages 561–566. 2005.

[56] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[57] Angjoo Kanazawa, David W Jacobs, and Manmohan Chandraker. Warpnet: Weakly supervised matching for single-view reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3253–3261, 2016.

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

[59] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[60] Hiroharu Kato and Tatsuya Harada. Learning view priors for single-view 3d reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9778–9787, 2019.

[61] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[62] Yunji Kim, Seonghyeon Nam, In Cho, and Seon Joo Kim. Unsupervised keypoint learning for guiding class-conditional video prediction. *Advances in neural information processing systems*, 32, 2019.

[63] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[64] Yasunori Kudo, Keisuke Ogaki, Yusuke Matsui, and Yuri Odagiri. Unsupervised adversarial learning of 3D human pose from 2D joint locations. *arXiv preprint arXiv:1803.08244*, 2018.

[65] Tejas Kulkarni, Ankush Gupta, Catalin Ionescu, Sebastian Borgeaud, Malcolm Reynolds, Andrew Zisserman, and Volodymyr Mnih. Unsupervised learning of object keypoints for perception and control. *arXiv preprint arXiv:1906.11883*, 2019.

[66] G. Larsson, M. Maire, and G. Shakhnarovich. Learning representations for automatic colorization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[67] Quoc V. Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S. Corrado, Jeff Dean, and Andrew Y. Ng. Building high-level features using large scale unsupervised learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 507—-514, 2012.

[68] John P Lewis, Matt Cordner, and Nickson Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 165–172, 2000.

[69] Jiaxin Li and Gim Hee Lee. Usip: Unsupervised stable interest point detection from 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 361–370, 2019.

[70] Xueting Li, Sifei Liu, Shalini De Mello, Kihwan Kim, Xiaolong Wang, Ming-Hsuan Yang, and Jan Kautz. Online adaptation for consistent mesh reconstruction in the wild. *Advances in Neural Information Processing Systems*, 33, 2020.

[71] Xueting Li, Sifei Liu, Kihwan Kim, Shalini De Mello, Varun Jampani, Ming-Hsuan Yang, and Jan Kautz. Self-supervised single-view 3d reconstruction via semantic consistency. In *European Conference on Computer Vision*, pages 677–693. Springer, 2020.

[72] Xueting Li, Sifei Liu, Shalini De Mello, Kihwan Kim, Xiaolong Wang, Ming-Hsuan Yang, and Jan Kautz. Online adaptation for consistent mesh reconstruction in the wild. In *Advances in neural information processing systems*, 2020.

[73] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T. Freeman. Learning the depths of moving people by watching frozen people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[74] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014.

[75] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[76] Shilong Liu, Lei Zhang, Xiao Yang, Hang Su, and Jun Zhu. Unsupervised part segmentation through disentangling appearance and shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8355–8364, 2021.

[77] Dominik Lorenz, Leonard Bereska, Timo Milbich, and Bjorn Ommer. Unsupervised part-based disentangling of object shape and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10955–10964, 2019.

[78] Nadia Magnenat-Thalmann, Richard Laperrire, and Daniel Thalmann. Joint-dependent local deformations for hand animation and object grasping. In *In Proceedings on Graphics interface'88*. Citeseer, 1988.

[79] Dimitrios Mallis, Enrique Sanchez, Matthew Bell, and Georgios Tzimiropoulos. Unsupervised learning of object landmarks via self-training correspondence. *Advances in Neural Information Processing Systems*, 33, 2020.

[80] Alexander Mathis, Pranav Mamidanna, Kevin M. Cury, Taiga Abe, Venkatesh N. Murthy, Mackenzie W. Mathis, and Matthias Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, 2018.

[81] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[82] Matthias Minderer, Chen Sun, Ruben Villegas, Forrester Cole, Kevin Murphy, and Honglak Lee. Unsupervised learning of object structure and dynamics from videos. *arXiv preprint arXiv:1906.07889*, 2019.

[83] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016.

[84] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[85] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[86] Thu Nguyen-Phuoc, Christian Richardt, Long Mai, Yong-Liang Yang, and Niloy J. Mitra. Blockgan: Learning 3d object-aware scene representations from unlabelled images. In *Advances in neural information processing systems*, 2020.

[87] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

[88] David Novotný, Diane Larlus, and Andrea Vedaldi. Learning 3D object categories by looking around them. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[89] David Novotny, Nikhila Ravi, Benjamin Graham, Natalia Neverova, and Andrea Vedaldi. C3dpo: Canonical 3d pose networks for non-rigid structure from motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7688–7697, 2019.

[90] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[91] Junyi Pan, Xiaoguang Han, Weikai Chen, Jiapeng Tang, and Kui Jia. Deep mesh reconstruction from single rgb images via topology modification networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9964–9973, 2019.

[92] Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[93] Tomas Pfister, James Charles, and Andrew Zisserman. Flowing convnets for human pose estimation in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1913–1921, 2015.

[94] En Yen Puang, Keng Peng Tee, and Wei Jing. Kovis: Keypoint-based visual servoing with zero-shot sim-to-real transfer for robotics manipulation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7527–7533. IEEE, 2020.

[95] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021.

[96] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020.

[97] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *International Conference on Computer Vision*, 2021.

[98] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 2017.

[99] Matteo Ruggero Ronchi, Oisin Mac Aodha, Robert Eng, and Pietro Perona. It's all relative: Monocular 3d human pose estimation from weakly supervised data. In *Proceedings of the British Machine Vision Conference*, 2018.

[100] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *Proceedings third international conference on 3-D digital imaging and modeling*, pages 145–152. IEEE, 2001.

[101] Enrique Sanchez and Georgios Tzimiropoulos. Object landmark discovery through unsupervised adaptation. *Advances in Neural Information Processing Systems*, 32, 2019.

[102] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *Advances in neural information processing systems*, 2020.

[103] Zhixin Shu, Mihir Sahasrabudhe, Riza Alp Guler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos. Deforming autoencoders: Unsupervised disentangling of shape and appearance. In *Proceedings of the European conference on computer vision (ECCV)*, pages 650–665, 2018.

[104] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2377–2386, 2019.

[105] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. *Advances in Neural Information Processing Systems*, 32, 2019.

[106] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[107] Olga Sorkine. Differential representations for mesh processing. *Computer Graphics Forum*, 25(4):789–807, 2006.

[108] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, pages 109–116, 2007.

[109] Supasorn Suwajanakorn, Noah Snavely, Jonathan Tompson, and Mohammad Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. In *Advances in neural information processing systems*, 2018.

[110] James Thewlis, Samuel Albanie, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of landmarks by descriptor vector exchange. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6361–6371, 2019.

[111] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object frames by dense equivariant image labelling. *Advances in neural information processing systems*, 30, 2017.

[112] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *Proceedings of the IEEE international conference on computer vision*, pages 5916–5925, 2017.

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

[114] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, 2014.

[115] Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Multi-view consistency as supervisory signal for learning shape and pose prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2897–2905, 2018.

[116] Shubham Tulsiani, Hao Su, Leonidas J Guibas, Alexei A Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2635–2643, 2017.

[117] Hsiao-Yu Fish Tung, Adam W Harley, William Seto, and Katerina Fragkiadaki. Adversarial inverse graphics networks: Learning 2d-to-3d lifting and image-to-image translation from unpaired supervision. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, 2017.

[118] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4068–4076, 2015.

[119] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[120] Ruben Villegas, Jimei Yang, Yuliang Zou, Sungryull Sohn, Xunyu Lin, and Honglak Lee. Learning to generate long-term future via hierarchical prediction. In *Proceedings of the International Conference on Machine Learning*, 2017.

[121] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[122] Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. 3dn: 3d deformation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1038–1046, 2019.

[123] Xudong Wang, Long Lian, and Stella X Yu. Unsupervised visual attention and invariance for reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6677–6687, 2021.

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

[125] O. Wiles, A. S. Koepke, and A. Zisserman. Self-supervised learning of a facial attribute embedding from video. In *Proceedings of the British Machine Vision Conference*, 2018.

[126] Shangzhe Wu, Tomas Jakab, Christian Rupprecht, and Andrea Vedaldi. DOVE: Learning deformable 3d objects by watching videos. *arXiv preprint arXiv:2107.10844*, 2021.

[127] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3D objects from images in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[128] Wayne Wu, Kaidi Cao, Cheng Li, Chen Qian, and Chen Change Loy. Transgaga: Geometry-aware unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8012–8021, 2019.

[129] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Huiwen Chang, Deva Ramanan, William T. Freeman, and Ce Liu. LASR: Learning articulated shape reconstruction from a monocular video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[130] Wei Yang, Wanli Ouyang, Xiaolong Wang, Jimmy Ren, Hongsheng Li, and Xiaogang Wang. 3d human pose estimation in the wild by adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, 2018.

[131] Yufei Ye, Shubham Tulsiani, and Abhinav Gupta. Shelf-supervised mesh prediction in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[132] Wang Yifan, Noam Aigerman, Vladimir G Kim, Siddhartha Chaudhuri, and Olga Sorkine-Hornung. Neural cages for detail-preserving 3d deformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 75–83, 2020.

[133] Mehmet Ersin Yumer, Siddhartha Chaudhuri, Jessica K Hodgins, and Levent Burak Kara. Semantic shape editing using deformation handles. *ACM Transactions on Graphics (TOG)*, 34(4):1–12, 2015.

[134] Andrei Zaharescu, Edmond Boyer, Kiran Varanasi, and Radu Horaud. Surface feature detection and description with applications to mesh matching. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 373–380. IEEE, 2009.

[135] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[136] Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee. Unsupervised discovery of object landmarks as structural representations. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2694–2703, 2018.

[137] Yuxuan Zhang, Wenzheng Chen, Huan Ling, Jun Gao, Yinan Zhang, Antonio Torralba, and Sanja Fidler. Image gans meet differentiable rendering for inverse graphics and interpretable 3d neural rendering. *arXiv preprint arXiv:2010.09125*, 2020.

[138] Yu Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 689–696. IEEE, 2009.

[139] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[140] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[141] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.