Supplementary Material - RGBT-Dog: A Parametric Model and Pose Prior For Canine Body Analysis Data Creation

Jake Deane[1] Sinéad Kearney[1] Kwang In Kim[2] Darren Cosker[1]
University of Bath[1] POSTECH[2]
jcdeane21@gmail.com, sineadkearney08@gmail.com, kimkin@postech.ac.kr, dpcc22@bath.ac.uk

This supplementary material contains additional material to support the main paper. This material is presented in the following sections:

1. Section 1 covers additional details of our RGBT-Dog model including the shape and texture PCA spaces.
2. Section 2 covers additional details for generating data and some additional data examples.
3. Section 3 covers some additional material regarding our experiments utilising our synthetic labels in the main paper including training details and model architectures.
4. Section 4 covers additional results from our experiments including the full tables.
5. Section 5 covers some additional results from applying models trained with our data to other animals.

Some material is repeated from the main paper for the readers convenience.

1. Additional RGBT-Dog model details

Details of shape parameters. Like many parametric models we utilise a PCA shape space to build meshes, this parameter governs the actual shape of the mesh not the way the mesh is posed; in other words this parameter governs the non posed appearance of the mesh. As in Kearney *et al*. [5] we build a PCA shape space from photogrammetry scans of 11 dogs in a similar manner to [5]. The body shape is defined by a vector of size $\mathbb{R}^{3N}$ (where $N = 2426$) vertices like in SMAL/SMPL. To build our PCA shape space we fist normalise the poses as was done in SMPL [8] to ensure all these dog shapes have the same poses. It should be noted that these scans were cleaned by a digital artist to remove any artefacts from the scanning process. Using these scans we can employ principal components analysis over the meshes, allowing us to build a mean shape, $\overline{V}$ and eigenvectors $E_V = [e_{V_1}, \cdots, e_{V_{10}}]$, the first 10 principal components (i.e. the orthonormal shape displacements) each of size 2426×3. Thus our PCA shape model can be summarised as $\{\overline{V}, E_V\}$. Given this PCA space we can generate a new shape for the dog mesh via $V' = E_V\beta + \overline{V}$ producing a new body shape $V' \in \mathbb{R}^{2426\times3}$ of a dog in the normalised (i.e. standardised) pose.

Details of texture parameters. Our PCA texture space is generated from the 12 UV scans (Fig. 1). Each texture map is originally represented as a multi-dimensional array: $\{T_i\}_{i=1}^{12} \subset \mathbb{R}^{f\times d\times d\times d\times 3}$ where $f = 4,848$ is the number of mesh faces and $d = 4$ is the resolution of the texture, and we convert it to a vector of size $f \times d \times d \times d \times 3$. Each element of T_i is normalized into the range $[0, 1]$. Applying PCA to $\{T_i\}$, we obtain the eigenvector matrix $E = \{\mathbf{e}_1, \dots, \mathbf{e}_{12}\}^\top$ with normalized eigenvectors $\{\mathbf{e}_i\}_{i=1}^{12}$ of the covariance matrix of $\{T_i\}$. Given this model, a new texture T' can be generated using the first 11 principal components of E by

$$T' = \tau(E\psi + \overline{T}) \tag{1}$$

where $\overline{T}$ is the mean texture, ψ is a randomly sampled vector, and τ is a threshold operator confining the outputs to be in the range $[0, 1]$. Some of the principal components of our PCA shape space can be seen visualised on the left of Fig. 2.

Details of RGBT-Dog: Our RGBT-Dog model, M, uses standard vertex-based linear blend skinning to generate a mesh m with $N = 2,426$ vertices and $K = 43$ joints: A mesh is defined by a function $Q(\beta, \psi, \theta, r, t, W)$

$$m = Q(\beta, \psi, \theta, r, t, W). \tag{2}$$

with shape $\beta \in \mathbb{R}^{10}$ and texture $\psi \in \mathbb{R}^{11}$ parameters used to explore the respective PCA spaces generating mesh shapes and texture. $\theta \in \mathbb{R}^{43\times3}$ are the pose parameters presented in Euler axis angle. Unlike in SMPL-X we do not include the root rotation in the pose: root rotation $r \in \mathbb{R}^3$ and translation $t \in \mathbb{R}^3$ are used to determine dogs' position and orientation in 3D space. This setting enables us to avoid providing these parameters for the camera. Blend weights $W \in \mathbb{R}^{N\times K}$ are used with a standard linear blend skinning

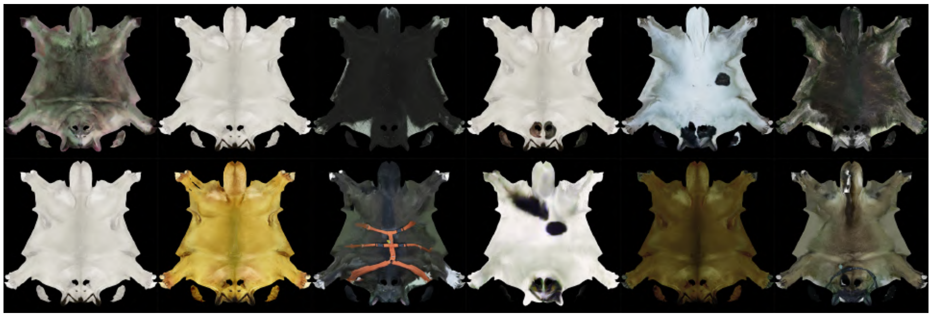

Figure 1. Dog textures as UV maps produced from cleaned photogrammetary scans

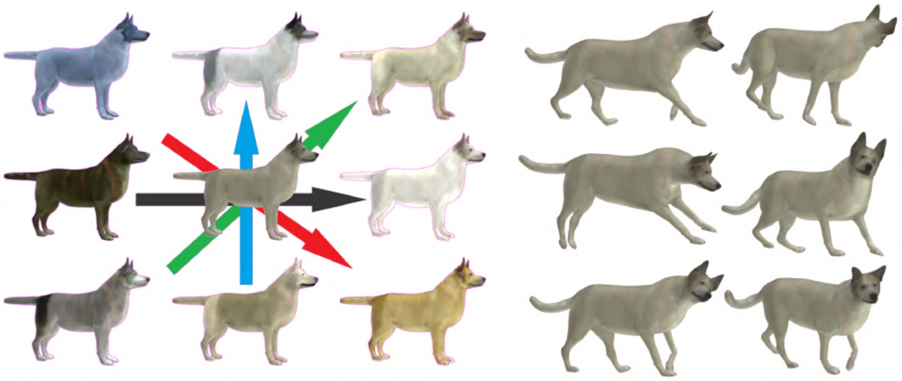

Figure 2. Left: First four principal components of the texture PCA space, displayed on a generic dog mesh. The mean texture is in the middle, with the red, green, blue and grey arrows representing the first, second, third and fourth components respectively. Each component shows ±2 standard deviations. Right: Motion from [5] applied to the dog with mean texture.

function to skin the mesh via the deformed vertices. This mesh is then placed in 3D space using the root rotation r and translation t. Finally, the texture map generated from our texture parameter ψ is applied to the canine mesh creating the final textured, articulated canine mesh m.

Below we describe the mesh creation process in greater detail with regards to the effect of the shape, pose and texture parameters.

Shape blend shaping: We start with a template body shape in the 'rest' pose, a canine equivalent of the T-pose for human body models. This template shape, $\overline{V} \in \mathbb{R}^{3N}$ is the average of the shapes for our photogrammetry scans, i.e. the average of the meshes. Using our shape PCA space $E_V \in \mathbb{R}^{2426\times3\times10}$ we can generate a new shape V'. We do this by multiplying our shape parameter $\beta \in \mathbb{R}^{10}$ with the eigenvectors for the PCA space E_V to create vertex displacements $E_V\beta \in \mathbb{R}^{2426\times3}$ which we add to the mean shape $\overline{V}$ resulting in a new set of verticies $V' \in \mathbb{R}^{2426\times3}$. This linear blend shaping process is summarised in Eq. (3).

$$V' = E_V\beta + \overline{V} \tag{3}$$

Joint locations: Next we obtain the 3D locations of the joints of the canine models skeleton from the displaced vertices V'. This is because, much like in SMPL, different canine shapes have different joint locations; to avoid artifacts when the mesh model is posed we require accurate 3D locations of the joints in the rest pose. To accomplish this a sparse joint regressor matrix $J \in \mathbb{R}^{43\times2426}$ which regresses initial 3D joint locations for this altered mesh $j_n \in \mathbb{R}^{43\times3} = JV'$ as the weighted average of the neighbouring vertices. An illustration of the joint regressor is shown in Fig. 3 where the 3D joints of a canine mesh are determined from the vertices of the mesh.

Applying pose: We apply pose in a simplified manner compared to SMPL/SMPL-X. In such work the use of pose blend shapes are used to represent muscle based deformations whenever any joints are rotated. The final joint positions are obtained from applying a joint regressor to the posed vertices with pose blend shapes (after posing). However this can lead to inconsistencies between the joint locations obtained from the regressor and those obtained from the pose parameter via forward kinematics. As a motive of our model is the creation of data for training machine learning models, we employ a simplified version for RGBT-Dog where no pose blend shapes are used to ensure consistency between the joints obtained from the regressor (see below)

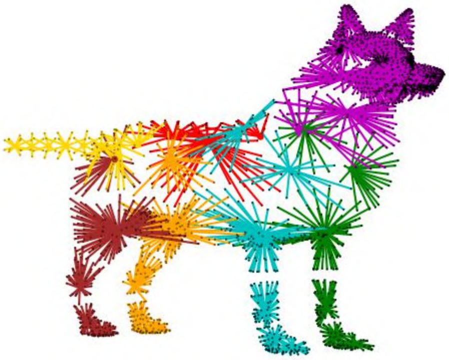

Figure 3. An illustration of the skeleton and joint regressor J for RGBT-Dog. Due to the orientation of the mesh, some keypoints are less visible than others.

and those from the pose parameter via forward kinematics.

We have the joints of the shaped mesh in neutral pose, j_n as discussed above. Using the pose parameter, θ, we can use forward kinematics to calculate the joint positions in the new pose, K_{3D} from the neutral pose joints, j_n. From this operation we also obtain the relative global transformation of each joint in the form, $A \in \mathbb{R}^{43\times4\times4}$. From this and a set of skinning weights, $W \in \mathbb{R}^{2426\times43}$ and the use of the generalised matrix product, we obtain the final vertex transformation for each vertex, $G \in \mathbb{R}^{2426\times4\times4}$. These are used to shape the vertices of the mesh with respect to the locations of the joints determined by the pose parameter, θ as neighbouring vertices are influenced differently by the same joint transformation. G is then applied to V' in a homogeneous transformation to create the final mesh $m = G \cdot V^h$, where $V^h \in \mathbb{R}^{2426\times4}$ is the homogeneous form of V', taking the in-homogeneous form to obtain the mesh $m \in \mathbb{R}^{2426\times3}$. We then use our root translation parameter, t to alter the 3D global position of the mesh and associated keypoints. This is simply accomplished via

$$m = m + t \tag{4}$$

$$K_{3D} = K_{3D} + t \tag{5}$$

This produces our mesh m however it is not yet textured. We can produce a new texture, T' using our texture PCA space comprised of a mean texture $\overline{T}$ and eigenvectors E_T. We can then apply this texture to our mesh m using generic mapping functions creating a textured, posed articulated mesh m. This entire process is, mathematically summarised by Eq. (2) for readers convenience as in the main paper, where $m \in \mathbb{R}^{2426\times3}$ is the canine mesh and M is the RGBT-Dog model.

This mesh can then be rendered to obtain the image, I, silhouette map, S (by providing a completely white texture map and rendering on a black background) and the 2D keypoints, K_{2D}, in addition to K_{3D}. However we still need to generate the part-segmentation map.

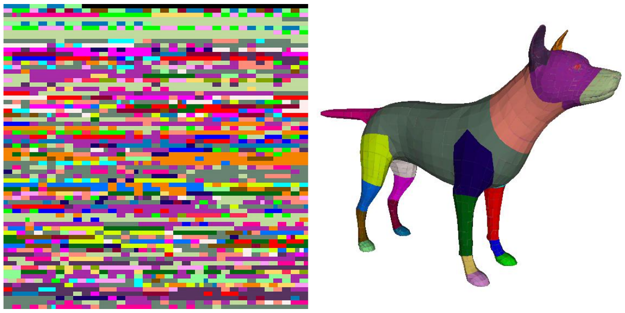

Figure 4. Left: Texture map for part-segmentation. Right: A rendered dog mesh, textured with the part-segmentation texture map.

1.1. Part-segmentation map generation details

In this section we provide additional details on the construction of the part-segmentation maps for the data. We take the mesh m and apply a part-segmentation texture map where each coordinate in the map (visualised in the left of Fig. 4) corresponds to a face on the mesh. The application of this texture map to a mesh can be seen in Fig. 4. This textured mesh can then be rendered to create our part-segmentation map. It should be noted that the maps for the Synth images and PGT images differ slightly: the maps for synthetic images contain labels for the eyes which we can see on the right of Fig. 4. These labels are absent on the synthetic part-segmentation maps obtained from fitting to real images (i.e. the PGT part-segmentation maps); this is because it is difficult to obtain an accurate fit for these labels as there are no eye keypoints provided in [2] and indeed there are no such keypoints provided by RGBT-Dog. As such we would have difficulty obtaining accurate labels for our PGT maps and thus the eye label is overwritten with the head label in the UV map which is then applied to the mesh. As a result, our PGT maps are made up of 25 labels as opposed to the Synth data's 26 (as Synth does not have an unknown label): As noted in the main paper, for the purposes of training our convolutional neural network based models, the eye labels for our Synth data part-segmentation maps are overwritten with the head label to create parity between the two datasets during training.

It should also be noted that in our generation of the segmentation maps during the fitting for the PGT images we take advantage of the ground truth silhouette maps provided by [2] in order to make the most accurate maps possible. To do this we utilise the ground truth data (keypoints and silhouettes) provided by Biggs *et al*. [2]. RGBT-Dog produces an initial rendered part-segmentation map. For this map, if a ground truth key point for a body part is unlabelled we pro-

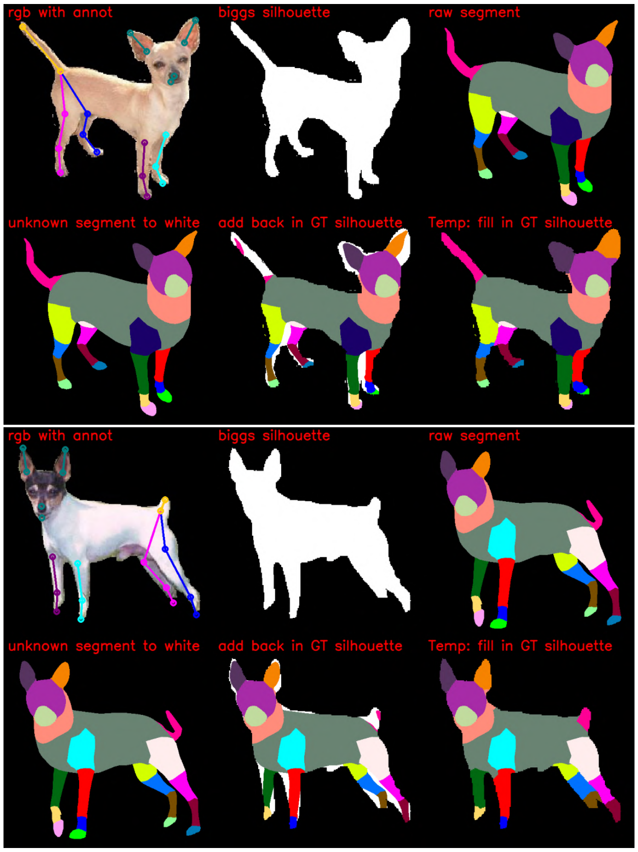

Figure 5. Part-segmentation map refinement process. From left to right, starting at top left: (1) The RGB image of [2] with ground truth keypoints annotated, (2) the corresponding silhouette map, (3) the original part-segmentation map produced by RGBT-Dog, (4) if a joint is not labelled the corresponding body part is given the unknown label, (5) add back in the ground truth silhouette, (6) remaining pixels at the border of the silhouette filled in based on the value of its nearest valid neighbour creating the final map.

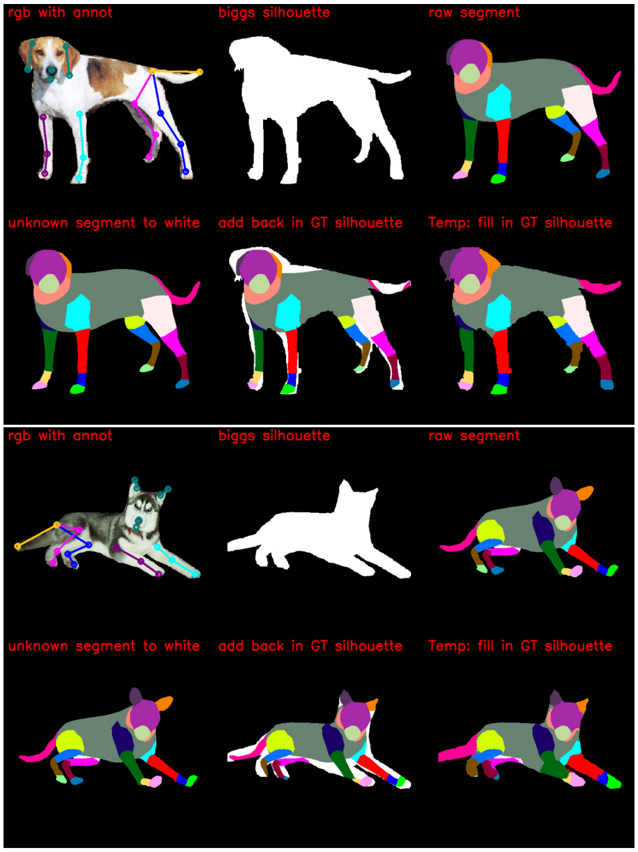

Figure 6. Part-segmentation map refinement process. From left to right, starting at top left: (1) The RGB image of [2] with ground truth keypoints annotated, (2) the corresponding silhouette map, (3) the original part-segmentation map produced by RGBT-Dog, (4) if a joint is not labelled the corresponding body part is given the unknown label, (5) add back in the ground truth silhouette, (6) remaining pixels at the border of the silhouette filled in based on the value of its nearest valid neighbour creating the final map.

vide a white label, the *unknown* label to the corresponding body part as we cannot be sure of this body part in an automated procedure. We can then refine this part-segmentation map using the ground truth silhouette map provided by [2]. This is done via adding the rendered part-segmentation map back to the silhouette then filling in the remaining silhouette with the value of its nearest neighbouring pixels. This process is visualised in Figs. 5 to 7.

2. Additional data examples

In this section we provide additional examples of our Synth and PGT datasets. As in the main paper we present the 2D keypoints and part-segmentation maps for both the PGT data and Synth data. The PGT examples can be found in Figs. 8 to 10; in these instances we can clearly see instances of the unknown (white) label. In Figs. 11 to 13 we provide examples from Synth. As can be seen in these two sets of figures, the Synth dataset contains eyes labels as note above but these are absent for PGT data. We can also see many instances of the unknown label in Figs. 8 to 10.

3. Experiment details

In this section we present some additional details regarding the models and training procedures used to validate the data we have created for canine body analysis. Some material is repeated from the main paper for the readers convenience.

We use the stacked hourglass model [9] for 2D pose estimation and part-segmentation. For training the keypoint estimation models, we employ the mean squared error between the predicted keypoint heatmaps and the ground-truth heatmaps as in [9] and other papers where heatmaps have

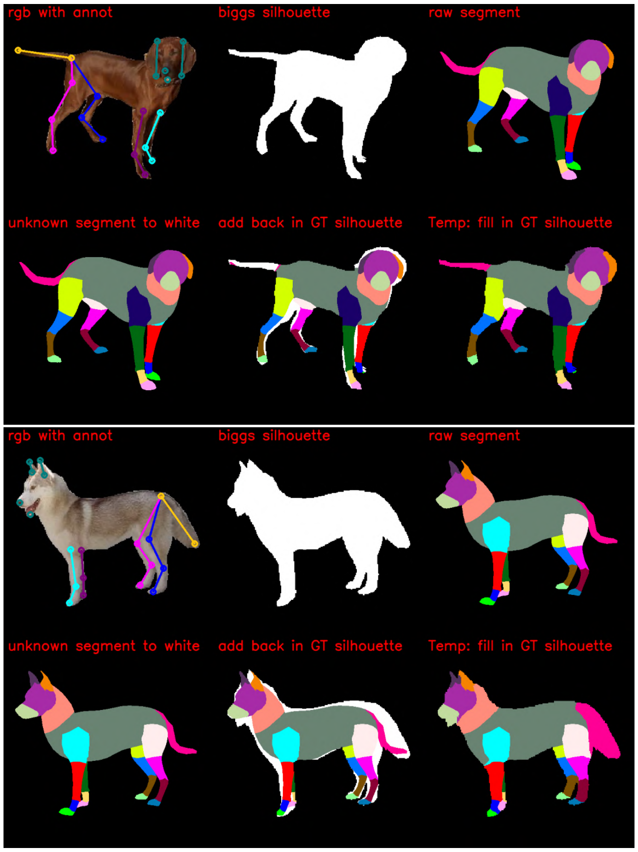

Figure 7. Part-segmentation map refinement process. From left to right, starting at top left: (1) The RGB image of [2] with ground truth keypoints annotated, (2) the corresponding silhouette map, (3) the original part-segmentation map produced by RGBT-Dog, (4) if a joint is not labelled the corresponding body part is given the unknown label, (5) add back in the ground truth silhouette, (6) remaining pixels at the border of the silhouette filled in based on the value of its nearest valid neighbour creating the final map.

been used (e.g. [10]. For the part-segmentation, the softmax cross-entropy between the ground-truth and predicted segmentation maps was used. For part-segmentation, we assigned a weight of zero to the 'unknown'[1] label in their loss evaluation to prevent our model from learning the unknown label.

For both tasks, across all datasets we trained for five epochs with a batch size of ten and a learning rate of 0.001. For data augmentation, we employed random horizontal and vertical flipping, Gaussian blur, hue saturation and random noise using ImgAug [4].

Regarding part-segmentation, the Synth maps possess labels for the eyes (as can be seen in the rendered mesh of Fig. 4) whereas the maps for the PGT do not. In order to enforce parity with respect to the number of data labels we fold the eye labels of the Synth dataset into the head label during the data loading (i.e. these three parts are given the same label) as mentioned above. This results in the maps seen in Figs. 11 to 13.

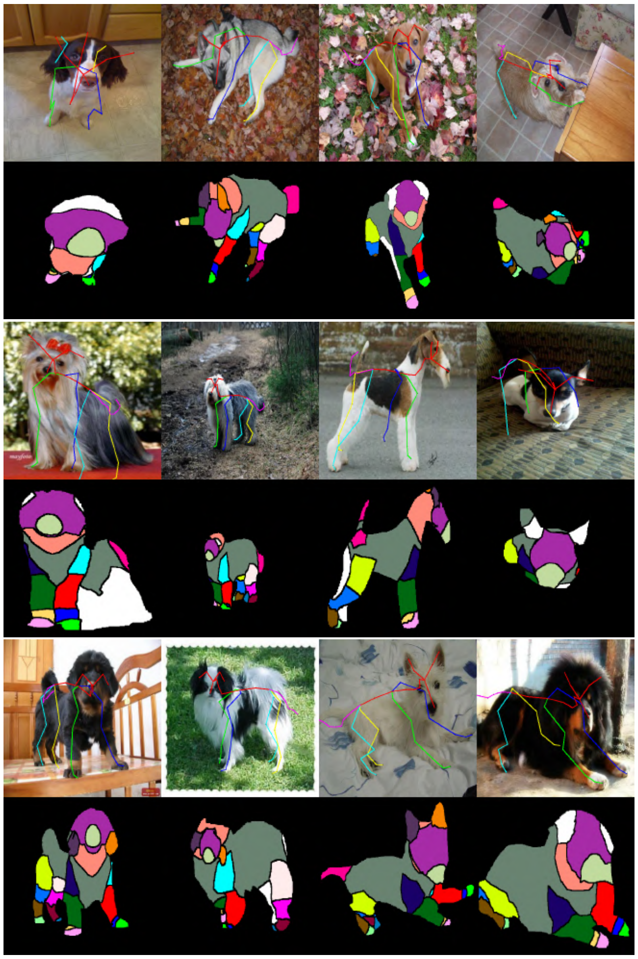

Figure 8. Examples of our PGT data: Top, RGB images with 2D pose overlaid. Bottom, corresponding Part-segmentation map.

4. Additional results for dogs

In this section, we present the full versions of the results tables found in the main paper.

4.1. Pose estimation

Our full pose estimation results can be found in Tab. 1; this table allows us to analyse the results at a lower level as we can compare individual joints. As is to be expected, our results for **Synth** under-perform compared to **Mixed/PGT** especially for poorer performing keypoints such as Tail End. As is to be expected keypoints that are often unobscured such as the head, nose and neck keypoints show significantly better results compared to more easily obscured

[1] See above for details regarding this label

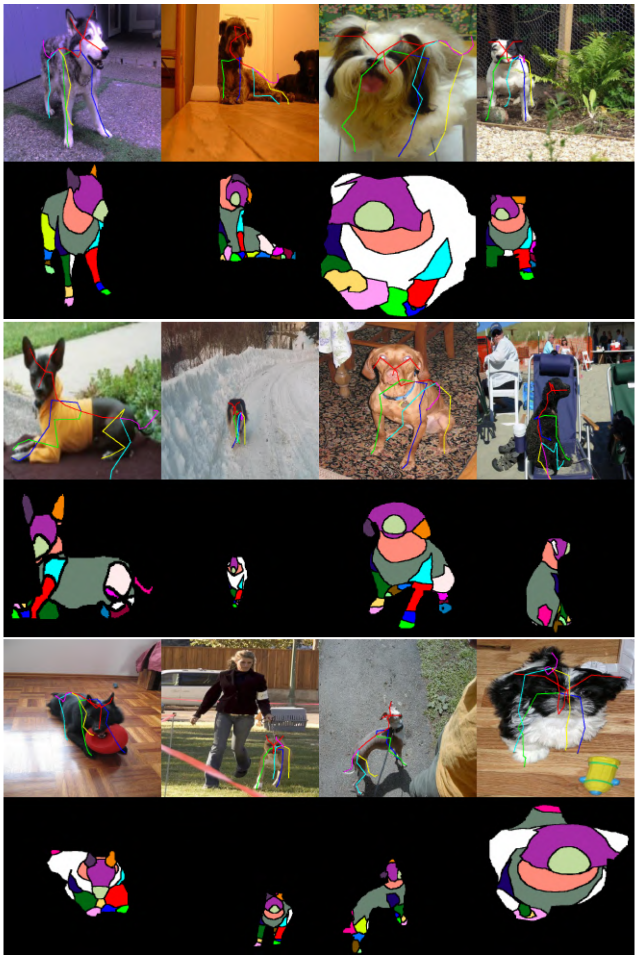

Figure 9. Examples of our PGT data: Top, RGB images with 2D pose overlaid. Bottom, corresponding Part-segmentation map.

keypoints such as those of the tail or upper limbs. Potentially performance for more easily obscured limbs could be improved through the use of loss weightings.

We provide some additional visualisations of our results in Figs. 14 and 15. As we noted in the main paper and highlighted by Tab. 1, our trained models notably struggle with the tail and ear keypoints. This is to be expected to some degree, such points possess far greater degrees of freedom compared to other points. Figures 14 and 15 also illustrate another notable feature of our results: occlusion of limbs. As shown, limb keypoints can be easily mistaken for one another by the trained model resulting in incorrect predictions particularly for those towards the end of the kinematic chain.

4.2. Part-segmentation

In Tab. 2 we present the full part-segmentation results, expanding on the summarised results presented in the main paper. As noted in the main paper, when training on

	PGT	Mix	Synth
Root	49.60	54.40	27.60
Spine 1	56.20	60.80	32.60
Spine 2	71.60	72.80	40.60
L.Shoulder	61.40	65.60	32.60
L.Arm	52.60	65.00	25.80
L.Forearm	55.60	63.80	26.80
L.Wrist	58.20	59.60	20.80
L.Hand	58.80	52.20	21.40
L.Finger	54.00	50.00	20.20
R.Shoulder	63.00	68.40	31.60
R.Arm	55.20	59.00	29.40
R.Forearm	58.20	57.60	23.80
R.Wrist	53.80	56.00	22.40
R.Hand	50.40	50.40	23.80
R.Finger	49.40	51.00	24.20
Neck 1	74.40	76.20	38.40
Neck 2	74.20	76.80	36.00
Neck 3	78.00	81.80	39.00
Neck 4	80.60	84.20	40.00
Head	83.60	86.80	45.00
Nose	82.80	81.60	31.00
L.Ear	72.00	64.60	28.80
L.Ear End	54.60	51.80	20.40
R.Ear	70.80	72.60	32.20
R.Ear End	52.20	45.40	21.80
L.Leg	39.20	43.60	21.60
L.Lower Leg	42.20	43.40	17.40
L.Ankle	34.60	40.20	15.80
L.Foot	35.20	39.00	14.20
L.Toe	34.80	37.20	13.60
R.Leg	45.00	48.80	22.40
R.Lower Leg	43.60	46.20	20.00
R.Ankle	39.20	38.40	13.20
R.Foot	37.20	34.80	14.00
R.Toe	36.40	34.60	11.80
Tail Base	47.00	54.80	26.40
Tail 1	46.40	53.20	24.60
Tail 2	45.00	48.80	21.80
Tail 3	45.20	46.20	16.80
Tail 4	39.00	39.60	14.40
Tail 5	35.60	33.80	13.00
Tail 6	30.00	29.00	11.80
Tail End	27.60	30.00	11.00
Average	52.89	54.65	24.19

Table 1. Full results for our pose estimation experiments. Results are presented as percentage of correct keypoints (PCK) as in the main paper.

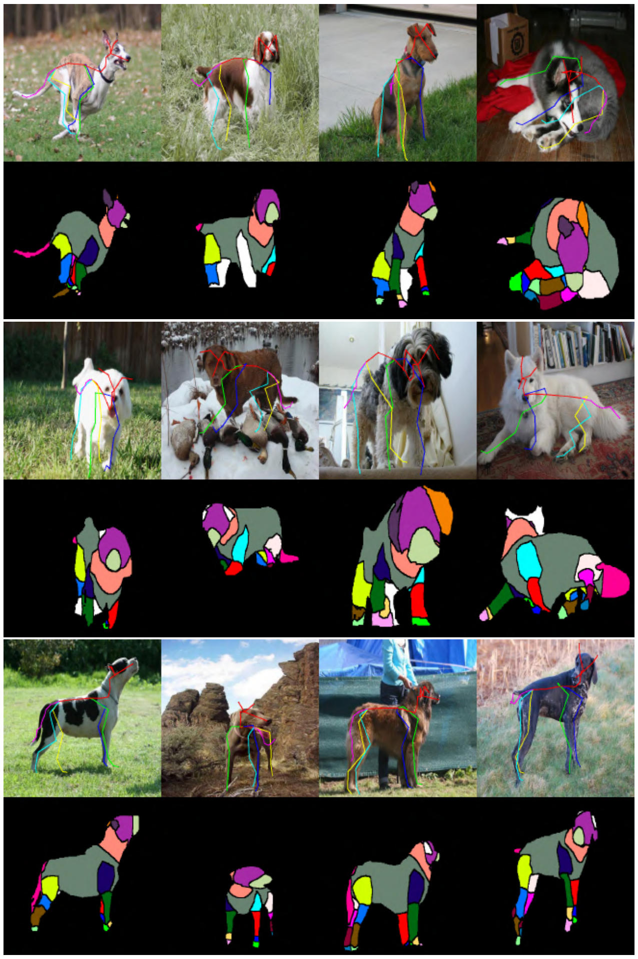

Figure 10. Examples of our PGT data: Top, RGB images with 2D pose overlaid. Bottom, corresponding Part-segmentation map.

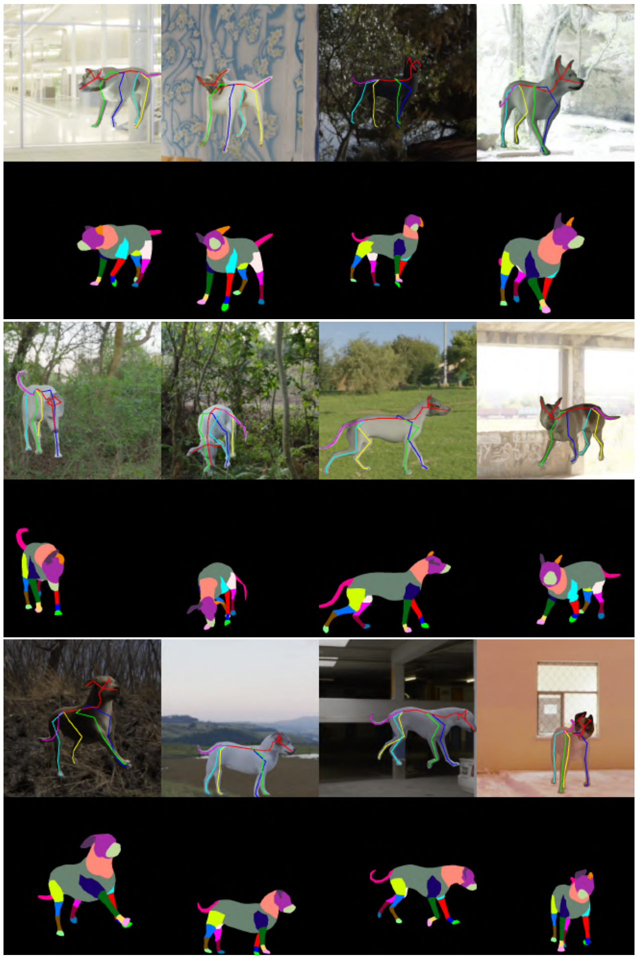

Figure 11. Examples of our Synth data: Top, RGB images with 2D pose overlaid. Bottom, corresponding Part-segmentation map.

the Synth dataset alone, our stacked hourglass models deliver worse results when comparing to the Mixed and PGT datasets. This is expected from the domain gap. Closing this gap between training images allows for a model to more easily adapt to real images as the difference in pose is less of a concern. Regardless, these results still fall behind those of our supervised learning experiments for the PGT and Mixed datasets. Regardless of dataset there are some commonalities across the results: Notably the tail, feet, hands and ears are some of the worst performing labels though this is understandable; the tail is a very malleable body part and easily obscured while body parts such as the feet are frequently obscured not to mention small in comparison to large parts such as the neck and torso (which display far better results).

In Figs. 16 to 18 we display some additional results for part-segmentation. As noted in the main paper, we are able to achieve fairly accurate results there is still room for improvement. This is understandable; despite our ability to generate canines with a variety of shapes, textures and poses we cannot cover all of the variety of appearance of canines unlike what Wood *et al*. [11] were able to achieve with their diverse face model. Our Synth data lacks features that are present in real images, namely competitive stimuli such as other animals or humans which can also make any adaptation difficult. In these figures we can see that, as noted in the main paper, our model can fix inaccuracies in the ground truth (be they the unknown label or incorrect labels) and is able to achieve overall accurate labels for large body parts. However as we move towards smaller body parts, performance worsens (as backed up by Tab. 2); as seen in the figures; smaller body parts such as the tail and paws become harder to predict. Similarly, accurate labelling of the ears and nose becomes more difficult as well. The former issue is potentially a result of the small size of the paws (and similar parts) and the large degree of flexibility of the tail. The latter is potentially a result of the shape space of our RGBT-Dog model, and as a results the shapes of the dogs of our Synth data do not posses the full range of body shapes seen

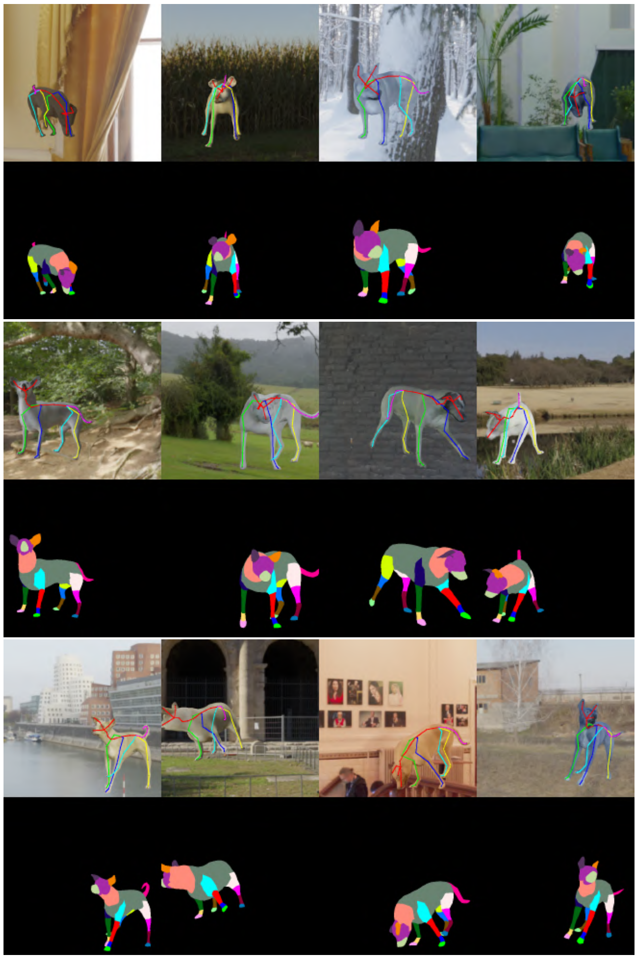

Figure 12. Examples of our Synth data: Top, RGB images with 2D pose overlaid. Bottom, corresponding Part-segmentation map.

in real images. This issue could be resolved by expanding the shape and texture PCA spaces of RGBT-Dog via additional photogrammetry scans.

4.3. Model fitting

In Fig. 19 we provide some additional examples of the visualisations of fitting RGBT-Dog to the images of Biggs *et al*. [2]. From these images, one an see that RGBT-Dog is able to recapture the skeleton and shape of canines with impressive, though not perfect, accuracy. One notable failing is that RGBT-Dog is unable to recreate the tongue as seen in rows 3 and 4 in Fig. 19.

5. Generalising to other animals

In this section we present some additional results from applying our trained stacked hourglass network to other animals. For both experiments, the stacked hourglass model trained on **Mixed** data is used as it performed the best across both tasks.

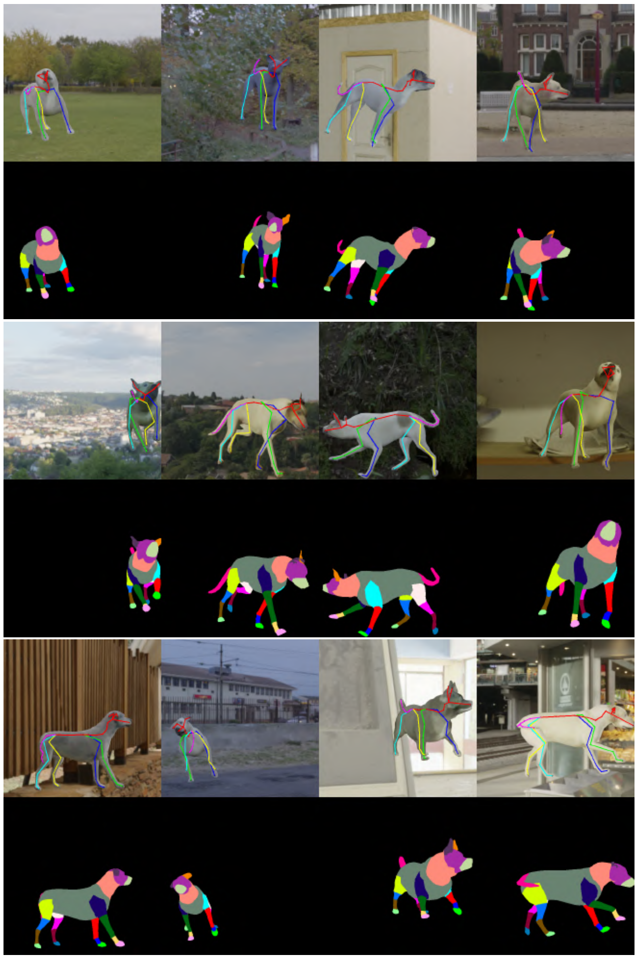

Figure 13. Examples of our Synth data: Top, RGB images with 2D pose overlaid. Bottom, corresponding Part-segmentation map.

We present some additional qualitative results for part-segmentation in Figs. 20 and 21. Understandably, given the wide variety of animals we are evaluating these results vary in terms of performance. Unsurprisingly the results for wolves, foxes and hyenas deliver the best results as they are members of the canine family and thus there is virtually no domain gap between the training data and these images. We can see particularly strong results in general for animals such as bears, deer, raccoon, pigs, horses and surprisingly rhinoceroses. For many subjects however the trained model while able to accurately recognise many body parts (e.g. legs and torso) is unable to label certain species specific features such as horns or antlers though this is expected; our model is not trained to recognise such features.

We can also see a similar issue for species where the body shape differs significantly from dogs. For example the rat and hamster images in Fig. 21 show that we can find features such as the head, neck and torso but struggle with parts such as the limbs and tail as such points are either very

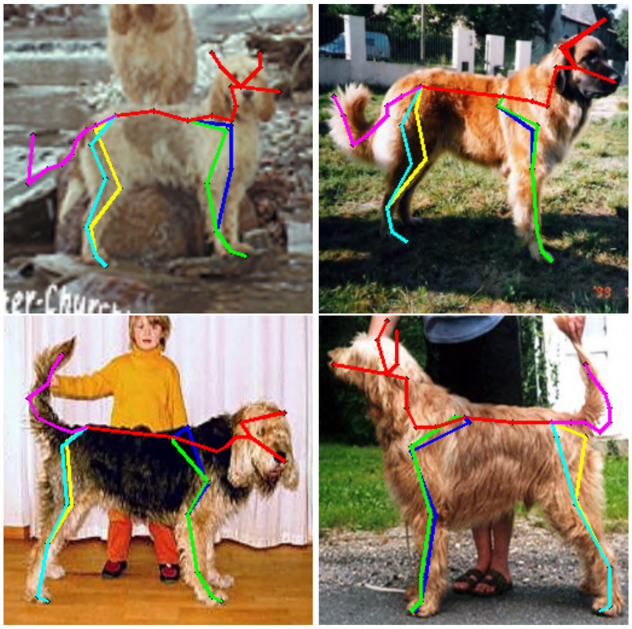

Figure 14. Predicted keypoints from hourglass trained with mixed data on images from [2]. One can see that accurately predicting keypoints for tail and ears is challenging.

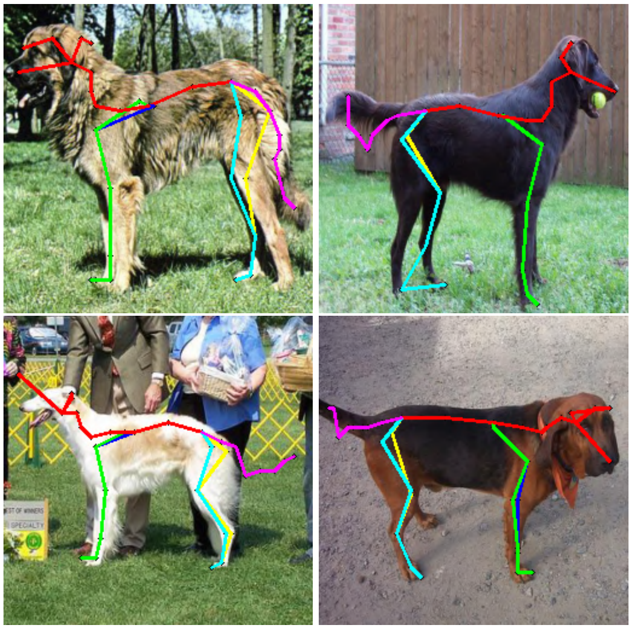

Figure 15. Predicted keypoints from hourglass trained with mixed data on images from [2]. One can see that accurately predicting keypoints for tail and ears is challenging.

small or occluded by objects such as grass blades. The presence of additional stimuli such as other animals (Fig. 21) or humans (Fig. 20) can also lead to worse performance compared to when the animal is on its own as shown by the

	PGT	Mix	Synth
Background	85.66	86.36	73.58
Torso	47.75	50.03	23.67
L.Hand	16.24	22.84	3.31
L.Wrist	11.95	12.89	3.06
L.Forearm	36.31	38.08	3.52
L.Arm	27.36	28.91	1.31
R.Hand	25.39	29.78	5.08
R.Wrist	17.03	16.71	2.46
R.Forearm	37.27	37.28	7.10
R.Arm	21.33	23.54	5.74
L.Foot	9.07	12.66	1.97
L.Ankle	26.50	21.80	1.86
L.LowerLeg	22.10	21.08	3.61
L.Leg	35.77	35.91	2.86
R.Foot	15.99	15.43	2.15
R.Ankle	24.06	15.62	3.46
R.LowerLeg	19.88	20.68	4.26
R.Leg	35.74	31.76	7.87
Tail	29.70	30.40	3.93
Neck	31.51	33.86	9.05
Head	57.50	57.06	17.20
L.Ear	23.69	28.60	2.26
R.Ear	24.66	26.73	2.91
Nose	37.77	42.95	11.23
Mean	30.01	30.87	8.48

Table 2. IOU results. Results are given in percentage terms.

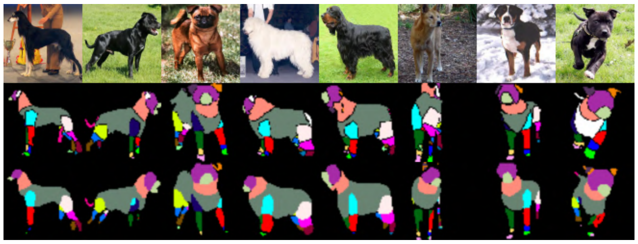

Figure 16. From top to bottom: Images from [6], PGT part segmentation maps, part-segmentation maps predicted by stacked hourglass model trained on mixed data.

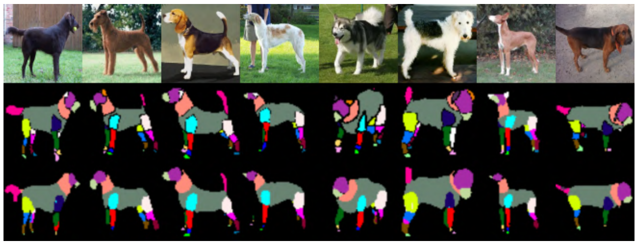

Figure 17. From top to bottom: Images from [6], PGT part segmentation maps, part-segmentation maps predicted by stacked hourglass model trained on mixed data.

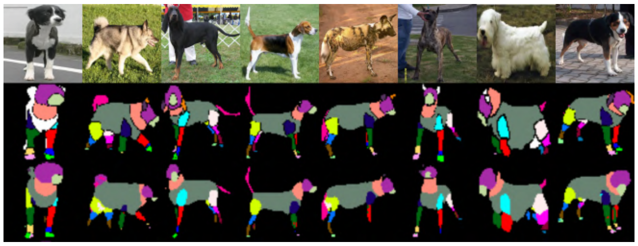

Figure 18. From top to bottom: Images from [6], PGT part segmentation maps, part-segmentation maps predicted by stacked hourglass model trained on mixed data.

results for the camel images in Fig. 20.

Interestingly we can also see that our model has learned semi-accurate labels for seals, lizards and crocodiles who have a notable appearance domain gap from dogs. Understandably there are still some errors in these predictions but nonetheless this does provide encouragement for future work as no such data exists for many of these subjects. Future work could utilise a small subset of animal specific data using transfer learning from our data to refine for species specific part-segmentation. As noted in the main paper all animal images utilised for this non canine evaluation come from the Kaggle Animal Image Dataset [1] and OpenImagesV6 [7].

For pose estimation we make use of the dataset of [12] where we employ the Euclidean distance between shared, visible keypoints as the keypoints of [12] are only a subset of RGBT-Dog's. These can be found in Tab. 3 measured in terms of pixels where we compare to the results for dogs. The results on dogs come from our own test data where the results were calculated on the shared RGBT-Dog/AP10K keypoints for the sake of parity.

One thing that should be noted about our quantitative results for pose estimation is that the dataset of [12] provides many instances of extreme closeups and obscuration. As a result, due to our **PGT** and **Synth** data lacking such instances, our model struggles to generalise to these instances.

From Tab. 3 we can see that our model generalises better to certain animals. Interestingly we see better results for herd/paddock animals like horses, buffaloes and deer compared to animals such as beavers or rats. This is likely because knowledge regarding a dogs skeleton is more transferable due to similarity in body pose and these animals are often seen in fields with little obscuration. The poor performance for animals such as gorillas and monkeys is expected; these animals possess very different skeletal structures to dogs leading to poor generalisation. We also see that the performance for wolves and chihuahuas is worse then we would expect though these images contain many close ups and occlusions which our model struggles to generalise to. Animals such as jaguars and leopards also display poor results due to such issues. A selection of these results are visualised in Figs. 22 to 24 where we see the effect of occlusion and different body shapes on results.

Animal	Euclidean distance	Animal	Euclidean distance
Antelope	9.19	King Cheetah	14.82
Argali Sheep	11.74	Leopard	18.73
Bison	11.79	Lion	13.41
Buffalo	11.00	Panther	13.12
Cow	13.76	Snow Leopard	15.61
Sheep	10.93	Tiger	16.79
Chihuahua	12.14	Giraffe	14.48
Fox	10.62	Hippo	12.50
Wolf	12.53	Chimpanzee	16.42
Beaver	16.77	Gorilla	19.27
Alouatta	14.45	Rabbit	12.48
Monkey	13.82	Skunk	14.91
Noisy Night Monkey	14.76	Mouse	13.69
Spider Monkey	16.91	Rat	15.20
Uakar	15.49	Otter	14.87
Deer	11.16	Weasel	11.54
Moose	12.70	Raccoon	12.74
Hamster	15.71	Rhino	13.07
Elephant	13.95	Marmot	13.89
Horse	11.39	Squirrel	14.05
Zebra	14.42	Pig	14.88
Bobcat	13.90	Black Bear	10.77
Cat	14.13	Brown Bear	11.64
Cheetah	14.49	Panda	16.32
Jaguar	20.95	Polar Bear	9.93
All	13.70	**Dog**	**8.82**

Table 3. Average error in Euclidean distance from our Mixed data trained model for animals of Yu *et al*. [12]. Dog value is obtained from our own test data using the same keypoints as [12].

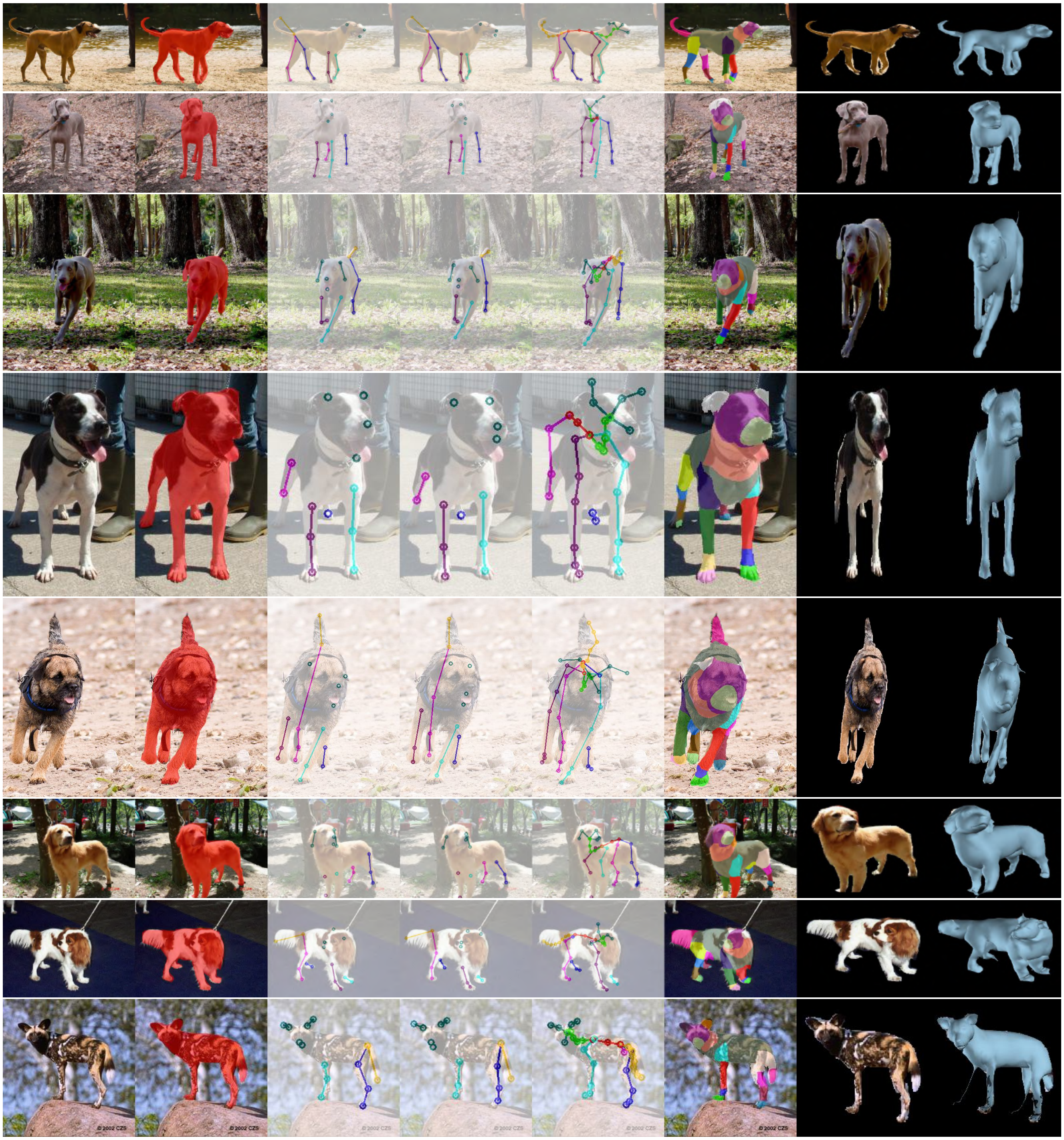

Figure 19. Left to Right: Image [6], Silhouette from [3], Keypoints from [3], Keypoints from SMALR [13], Keypoints from RGBT-Dog, part-segmentation Map from RGBT-Dog, canine image with background masked out,canine with RGBT-Dog rendered onto it.

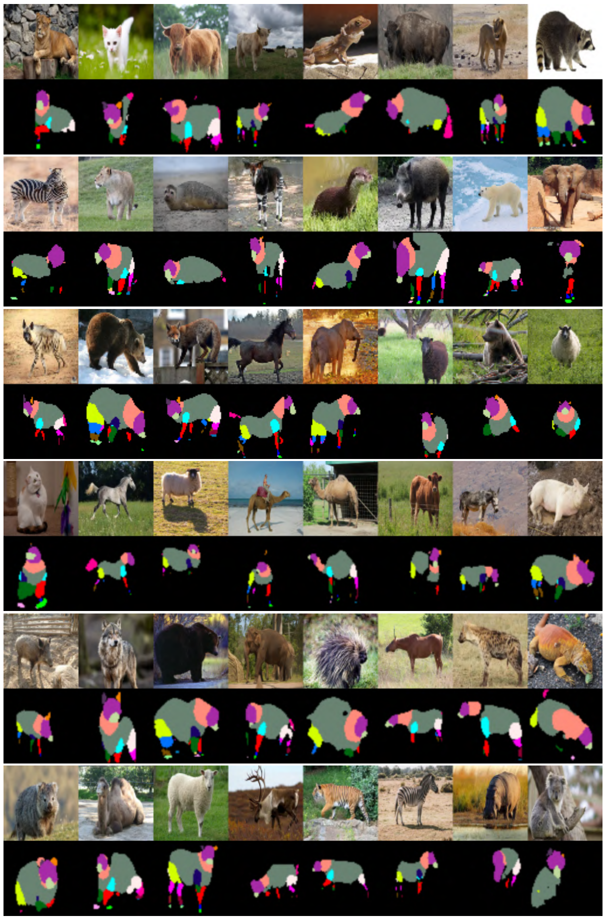

Figure 20. Result from applying stacked hourglass model trained on our Mixed data to images of other animals.

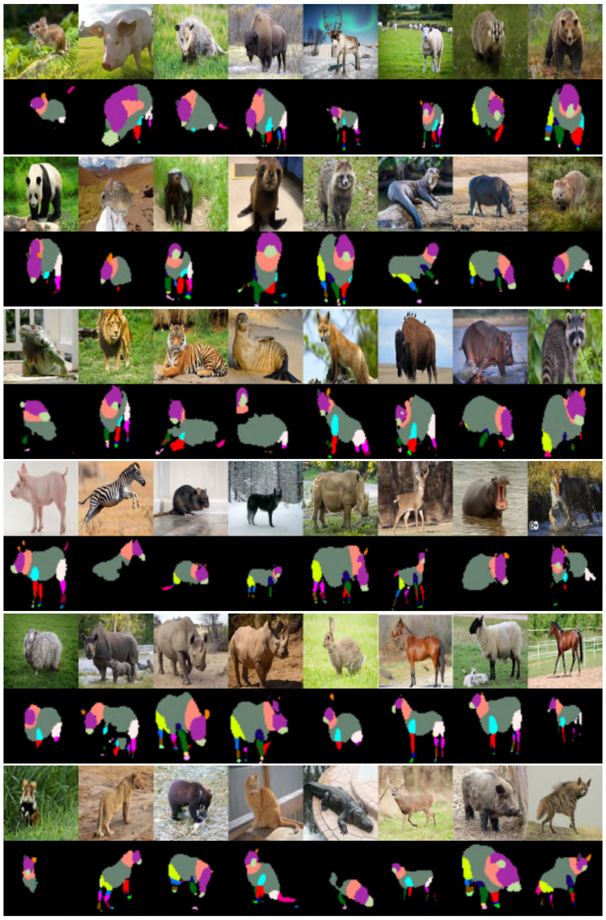

Figure 21. Further results from applying stacked hourglass model trained on our Mixed data to images of other animals.

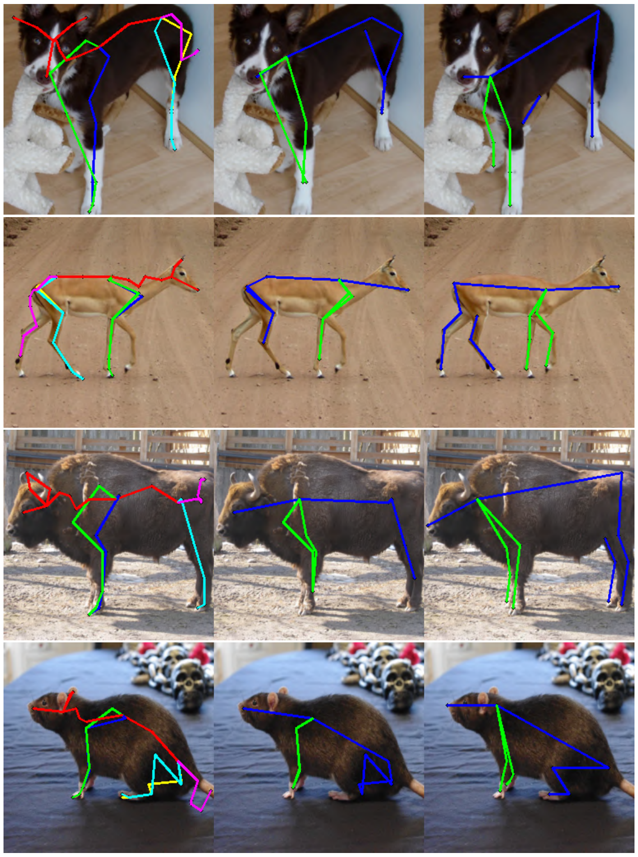

Figure 22. From left to right: AP-10K images [12] with predicted RGBT-Dog keypoints; predicted RGBT-Dog keypoints shared with AP-10K; ground truth provided by [12].

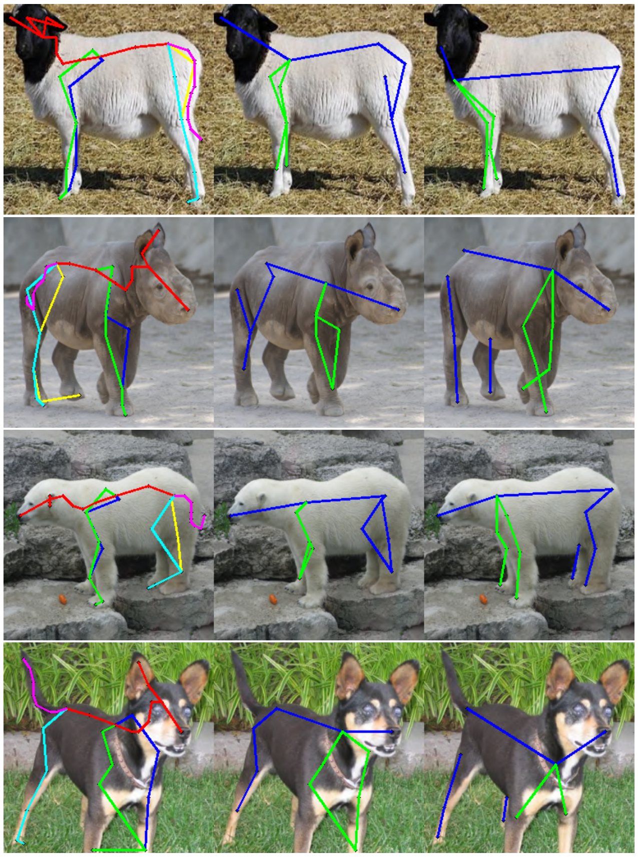

Figure 23. From left to right: AP-10K images [12] with predicted RGBT-Dog keypoints; predicted RGBT-Dog keypoints shared with AP-10K; ground truth provided by [12].

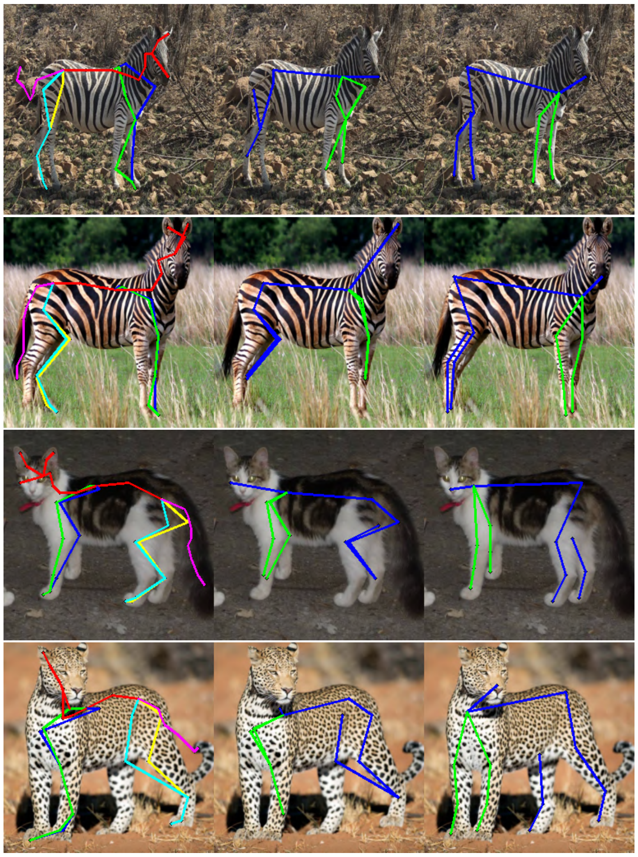

Figure 24. From left to right: AP-10K images [12] with predicted RGBT-Dog keypoints; predicted RGBT-Dog keypoints shared with AP-10K; ground truth provided by [12].

References

[1] Sourav Banerjee. Animal Image Dataset (90 Different Animals) (version 3), 2021. 10

[2] Benjamin Biggs, Oliver Boyne, James Charles, Andrew Fitzgibbon, and Roberto Cipolla. Who left the dogs out? 3D animal reconstruction with expectation maximization in the loop. In *European Conference on Computer Vision (ECCV)*, pages 9498–9507, July 2020. arXiv: 2007.11110. 3, 4, 5, 8, 9

[3] Benjamin Biggs, Thomas Roddick, Andrew Fitzgibbon, and Roberto Cipolla. Creatures great and SMAL: Recovering the shape and motion of animals from video. In *Asian Conference on Computer Vision (ACCV)*, Nov. 2018. arXiv: 1811.05804. 11

[4] Alexander Jung, Kentaro Wada, Jon Crall, Satoshi Tanaka, Jake Graving, Christoph Reinders, Sarthak Yadav, Joy Banerjee, Gabor Vecsei, Adam Kraft, Zheng Rui, Jirka Borovec, Christian Vellentin, Semen Zhydenko, Killian Pfeiffer, Ben Cook, Ismael Fernandez, Francois-Michel De Rainville, Chi-Hung Weng, Abner Ayala-Acevedo, Rapheal Meudec, Mathias Laporte, et al. imgaug, 2020. 5

[5] Sinead Kearney, Wenbin Li, Martin Parsons, Kwang In Kim, and Darren Cosker. RGBD-Dog: Predicting canine pose from RGBD sensors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. arXiv: 2004.07788. 1, 2

[6] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford Dogs. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, June 2011. 9, 10, 11

[7] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The Open Images Dataset V4. *International Journal of Computer Vision*, 128(7):1956–1981, July 2020. 10

[8] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: a skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6):1–16, Oct. 2015. 1

[9] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked Hourglass Networks for Human Pose Estimation. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, pages 483–499, Cham, 2016. Springer International Publishing. 4

[10] Nadine Rüegg, Silvia Zuffi, Konrad Schindler, and Michael J. Black. BARC: Learning To regress 3D dog shape from images by exploiting breed information. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3876–3884, 2022. 5

[11] Erroll Wood, Tadas Baltrusaitis, Charlie Hewitt, Sebastian Dziadzio, Thomas J. Cashman, and Jamie Shotton. Fake it till you make it: face analysis in the wild using synthetic data alone. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3661–3671, Montreal, QC, Canada, Oct. 2021. IEEE. 7

[12] Hang Yu, Yufei Xu, Jing Zhang, Wei Zhao, Ziyu Guan, and Dacheng Tao. AP-10K: A Benchmark for Animal Pose Estimation in the Wild. *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. arXiv: 2108.12617. 10, 13, 14

[13] Silvia Zuffi, Angjoo Kanazawa, and Michael J Black. Lions and tigers and bears: Capturing non-rigid, 3D, articulated shape from images. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5524–5532, 2018. 11