Perceiving 3D Human-Object Spatial Arrangements from a Single Image in the Wild (Supplementary Materials)

Jason Y. Zhang[1]*, Sam Pepose[2]*, Hanbyul Joo[2],
Deva Ramanan[1,3], Jitendra Malik[2,4], and Angjoo Kanazawa[4]

[1] Carnegie Mellon University
[2] Facebook AI Research
[3] Argo AI
[4] UC Berkeley

1 Supplemental Material

In this section, we describe implementation details in Sec. 1.1 and the mesh processing pipeline in Sec. 1.2. We also include more qualitative results in Sec. 1.3 and describe a few failure modes in Fig. 11.

1.1 Implementation details

We represent rotations for the object poses using the 6-DoF rotation representation introduced in [6]. We optimize the the occlusion aware silhouette loss using the ADAM optimizer [3] with learning rate 1e-3 for 100 iterations. We compute the edge maps $E(M)$ using $\text{MaxPool}(M) - M$ with a filter size of 7. Since the occlusion-aware silhouette loss is susceptible to getting stuck in local minima, we initialize with 10,000 randomly generated rotations and select the pose that produces the lowest loss value. For some categories (bicycle, bench, motorcycle), we found it beneficial to bias the sampling toward upright poses (elevation between -30 and 30 degrees, azimuth between 0 and 360 degrees).

We jointly optimize the 3D spatial arrangement loss (**??**) using ADAM with learning rate 1e-3 for 400 iterations. The trainable parameters are intrinsic scale $s^i \in \mathbb{R}$ for the i-th human and intrinsic scale $s^j \in \mathbb{R}$, rotation $\mathcal{R}^j \in SO(3)$, and translation $\mathbf{t}^j \in \mathbb{R}^3$ for the j-th object instance. The loss weights λ_i are tuned qualitatively on the COCO-2017 val set. We initialized the optimization with the human poses estimated using [2] and the best object pose estimated in Sec. **??** per object instance. To improve computational speed, we downsample the SMPL human meshes to 502 vertices and 1000 faces when computing losses.

A list of interaction parts pairs can be found in Tab. 1, and an enumeration of the sizes of the 3D bounding boxes used to compute the interaction losses can be found in Tab. 2.

1.2 Pre-processing Mesh Models

In Fig. 1, we show all mesh instances that we built for each 3D category. To better cover the shape variation within the an object category, we use multiple

Category	Part Pairs (Object Part, Human Part)
Bat	(Handle, L Palm), (Handle, R Palm)
Bench	(Seat, Butt), (Seat Back, Back)
Bicycle	(Seat, Butt), (Handlebars, L Palm), (Handlebars, R Hand)
Laptop	(Laptop, L Palm), (Laptop, R Palm)
Motorcycle	(Seat, Butt), (Handlebars, L Palm), (Handlebars, R Palm)
Skateboard	(Skateboard, L Foot), (Skateboard, R Foot)
Surfboard	(Surfboard, L Foot), (Surfboard, R Foot), (Surfboard, L Palm) (Surfboard, R Palm)
Tennis Racket	(Handle, L Palm), (Handle, R Palm)

Table 1: **List of Parts Pairs used per category.** Each parts pair consists of a part of an object and a part of the human body. These parts pairs are used to assign human-object interactions.

mesh models for a few object categories (e.g., motorcycle, bench, and laptop). All the meshes are pre-processed to be watertight and are simplified with a low number of faces and uniform face size, to make the the optimization more efficient. For the pre-processing, we first fill in the holes of the raw mesh models (e.g., the holes in the wheels or tennis racket) to make the projection of the 3D models consistent with the silhouettes obtained by the instance segmentation algorithm [1,4]. Then, we perform a TSDF fusion approach [5] that converts the raw meshes to watertight and simplified meshes. Finally, we reduce the number of mesh vertices using MeshLab[1].

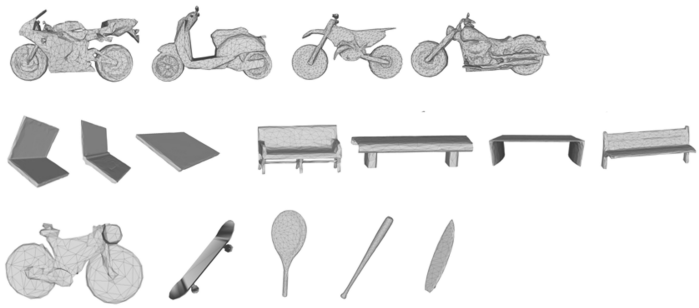


Fig. 1: **Mesh models from various 3D object categories.** Here, we show all the mesh models that we used. **First row:** Motorcycle. **Third Row:** Laptop, Bench. **Fourth Row:** Bicycle, Skateboard, Tennis Racket, Baseball Bat, Surfboard.

[1] http://www.meshlab.net/

Category	XY (Coarse)	XY (Fine)	Z Depth
Bat	0.5	2.5	5
Bench	0.3	0.5	10
Bicycle	0	0.7	4
Laptop	0.2	0	2.5
Motorcycle	0	0.7	5
Skateboard	0	0.5	10
Surfboard	0.8	0.2	50
Tennis Racket	0.4	2	5

Table 2: **Size of 3D Bounding Boxes.** To determine whether to apply the coarse interaction loss, we take the bounding box of the object and the bounding box of the person and expand each by the coarse expansion factor (Column 2). If the expanded bounding boxes overlap and the difference in the depths of the person and object is less than the depth threshold (Column 4), then we consider the person and object to be interacting. To determine whether to apply the fine interaction loss, we similarly take the bounding boxes corresponding to the object part and person part, expand the bounding boxes by the fine expansion factor (Column 3), and check for overlap. If the expanded bounding boxes overlap and the difference in depths of the parts is less than the depth threshold, then we consider the person part and object part to be interacting.

1.3 More Qualitative Results

We show results on a large number of COCO images (test set) for each category evaluated in the main paper: baseball bats (Fig. 3), benches (Fig. 4), bicycles (Fig. 5), laptops (Fig. 6), motorcycles (Fig. 7), skateboards (Fig. 8), surfboards (Fig. 9), and tennis rackets (Fig. 10).

Fig. 2: **Screenshot from our comparison evaluation test interface.** Annotators were asked to evaluate which 3D arrangement looks more accurate, in this case, without and with the depth ordering loss for a picture of a person with a surfboard. Clockwise from top-left: original image, image with rendered projection, top-down view, frontal view.

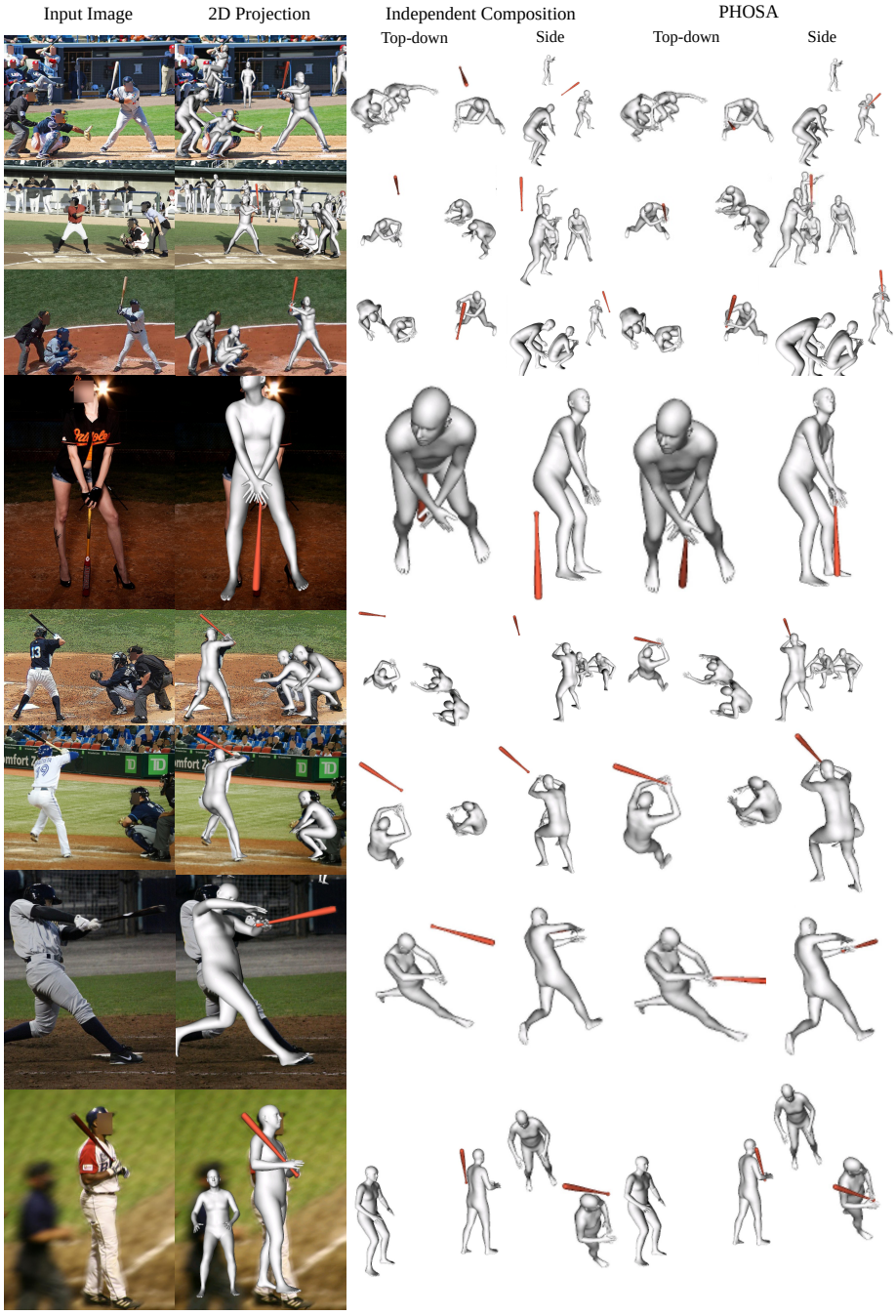


Fig. 3: **Our output on COCO images with baseball bats.**

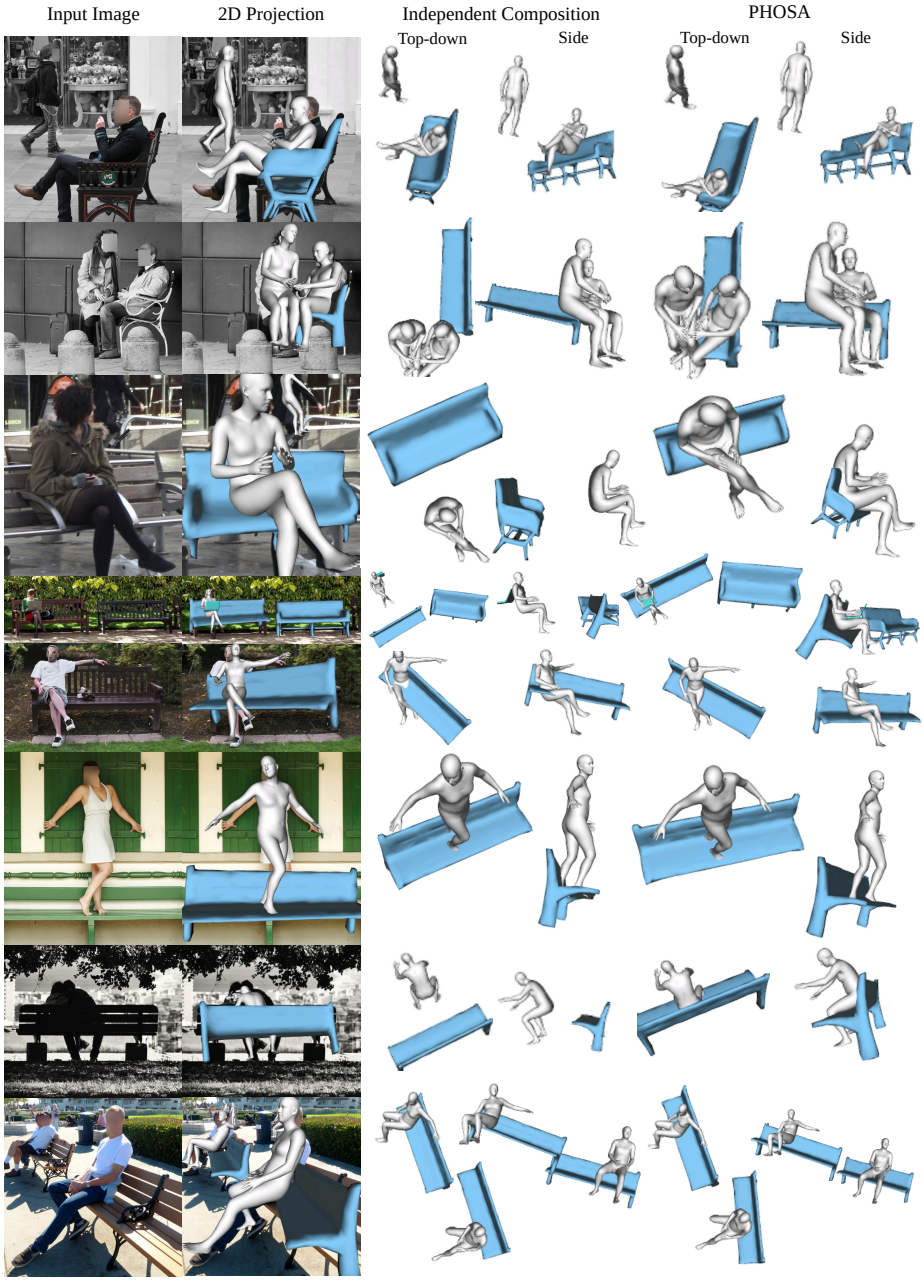


Fig. 4: **Our output on COCO images with benches.**

Fig. 5: **Our output on COCO images with bicycles.**

Fig. 6: **Our output on COCO images with laptops.**

Fig. 7: **Our output on COCO images with motorcycles.**

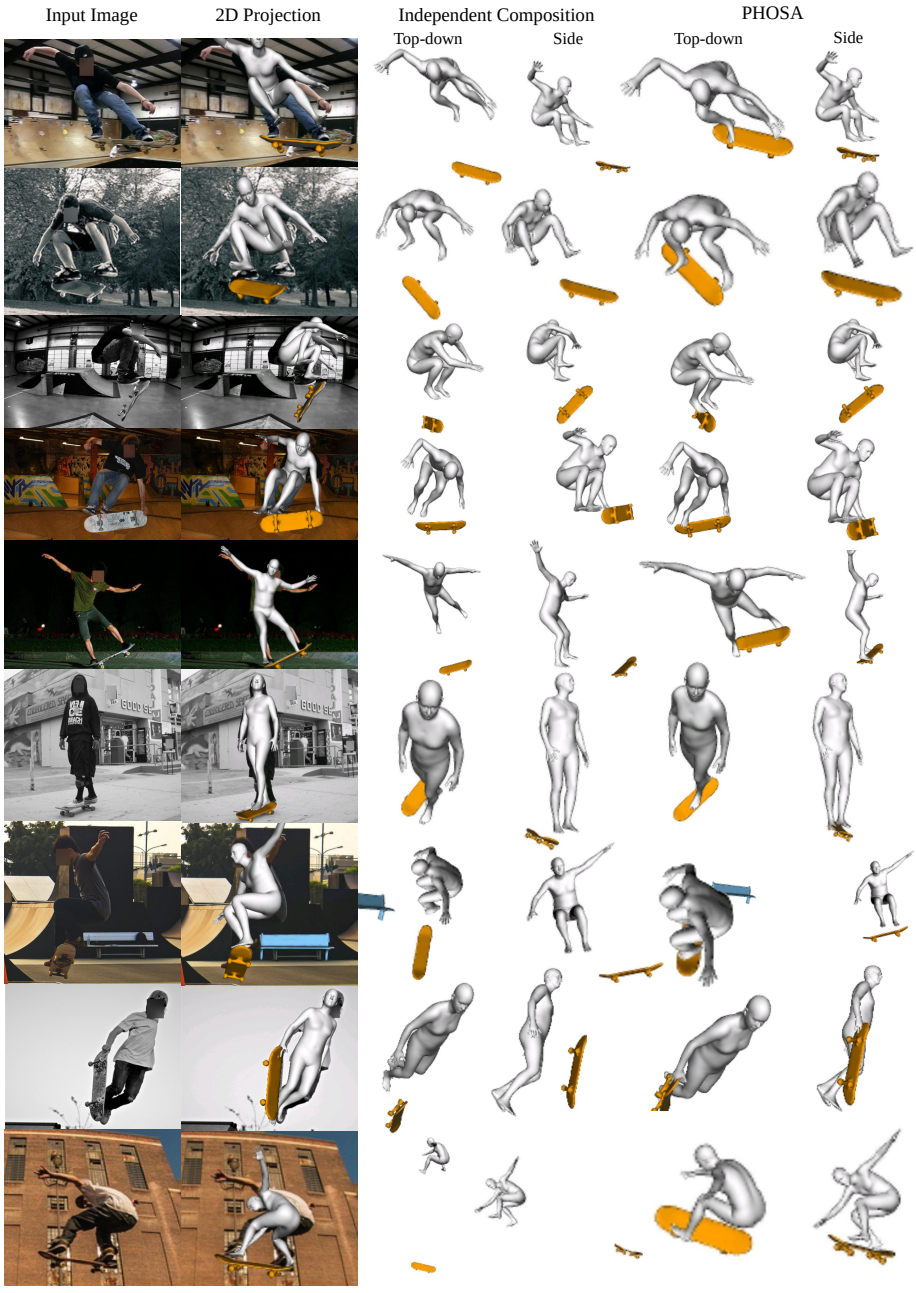


Fig. 8: **Our output on COCO images with skateboards.**

Fig. 9: **Our output on COCO images with surfboards.**

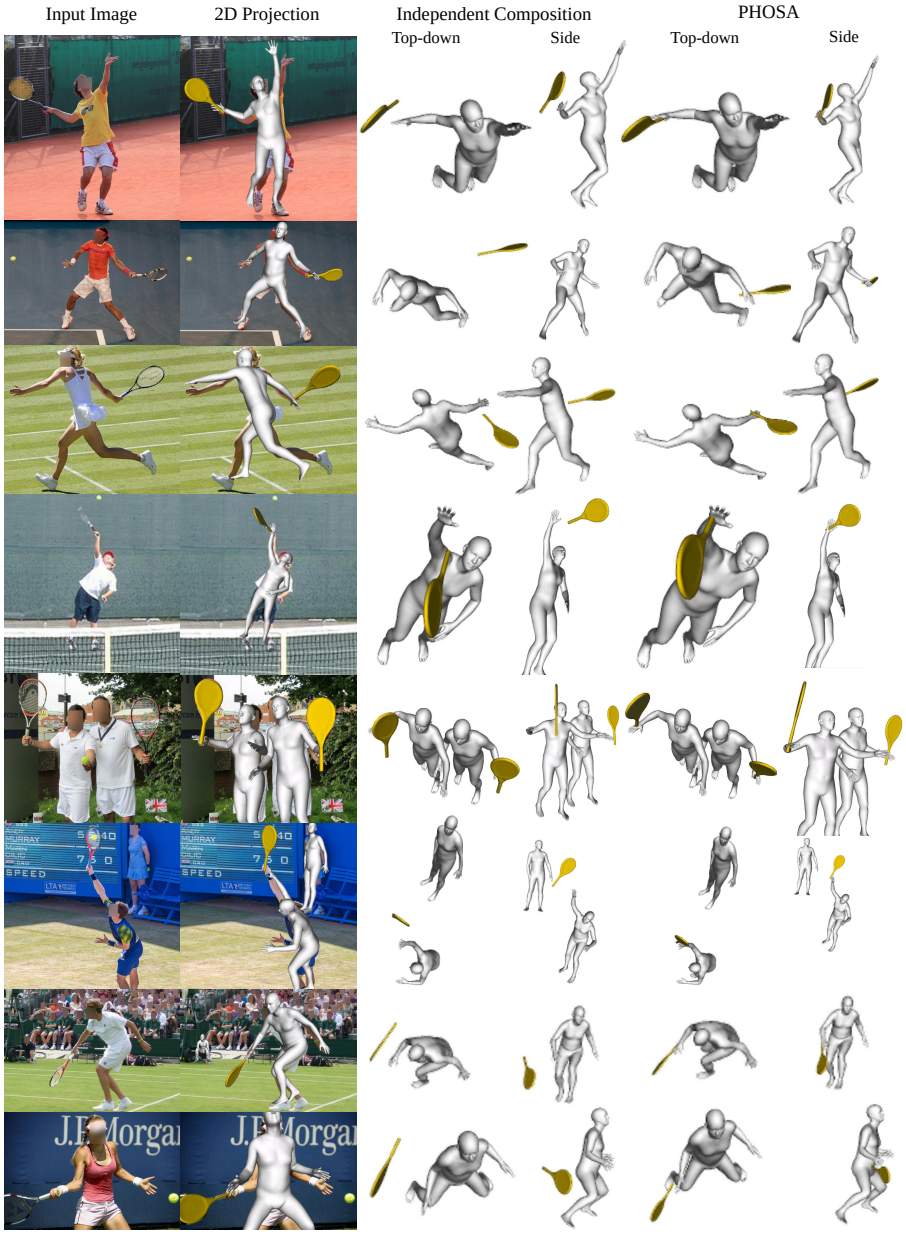
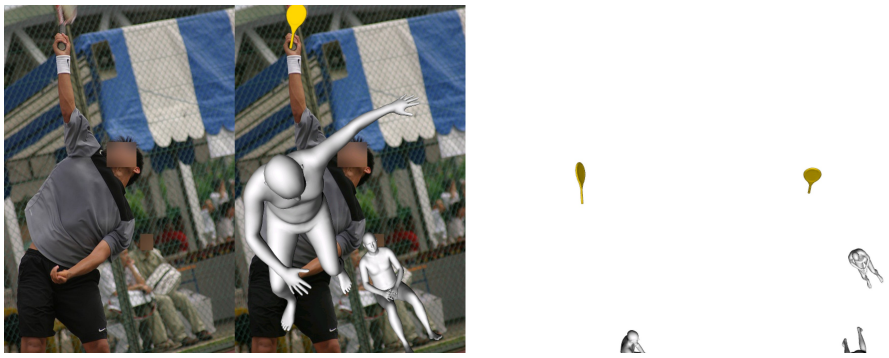
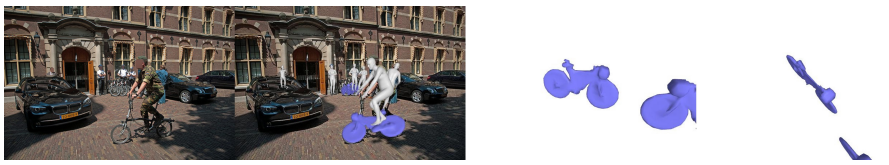


Fig. 10: **Our output on COCO images with tennis rackets.**



(a) **Human pose failure.** Our human pose estimator sometimes incorrectly estimates the pose of the person, such as in this challenging snapshot of a tennis player performing a volley. In such cases, it can be difficult to reason properly about human-object interaction since the hand is misplaced and far from the tennis racket.

(b) **Object pose failure.** The predicted masks are sometimes unreliable for estimating the pose of the object. In such cases, it difficult to recover a plausible scene reconstruction.



(c) **Incorrect reasoning about interaction due to scale.** The interaction loss requires a reasonable scale initialization. Sometimes, objects in the real world can fall outside the expected scale distribution, such as in the case of this small bicycle.

Fig. 11: **Failure modes.** In this figure, we describe a few failure modes of our method.

References

1. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV (2017)
2. Joo, H., Neverova, N., Vedaldi, A.: Exemplar fine-tuning for 3d human pose fitting towards in-the-wild 3d human pose estimation. arXiv preprint arXiv:2004.03686 (2020)
3. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
4. Kirillov, A., Wu, Y., He, K., Girshick, R.: Pointrend: Image segmentation as rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9799–9808 (2020)
5. Stutz, D., Geiger, A.: Learning 3d shape completion under weak supervision. IJCV pp. 1–20 (2018)
6. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. In: CVPR. pp. 5745–5753 (2019)