Supplementary Material for 3D Hand Pose Estimation in Everyday Egocentric Images

Aditya Prakash, Ruisen Tu, Matthew Chang, and Saurabh Gupta

University of Illinois Urbana-Champaign
{adityap9,ruisent2,mc48,saurabhg}@illinois.edu
https://bit.ly/WildHands

In this document, we first describe issues with the use of crops in hand pose estimation (Sec. A) and provide additional implementation details about the architecture (Sec. B.1) & training protocol (Sec. B.2). We then present additional analysis of results (Sec. C) and describe our Epic-HandKps dataset (Sec. D). Next, we provide qualitative comparisons (Sec. E) of our WildHands model with FrankMocap [24] (Fig. E) & HaMeR [20] (Fig. D) on Epic-HandKps and with ArcticNet [7] on ARCTIC (Fig. G). We also show failure cases of WildHands on both Epic-HandKps (Fig. F) & ARCTIC (Fig. H) and more visualizations of perspective distortion induced ambiguity in hands (Fig. I).

A Issues with Use of Crops in Current Hand Pose Estimation Methods

As noted in the main paper, the current practice of using crops around hands as input to the network suffers from 3 potential issues.

In exocentric views, as the camera can be arbitrarily placed with respect to the hand, the hand location in the image doesn't carry any signal. However, in an egocentric setting as the camera is mounted on the head, the 2D location of the hand in the image is indicative of the hand pose. Thus, it may be useful to also use the location of the hand in the image as an additional input into the network.

The second issue comes from the observation that the 3D that best explains the 2D appearance while assuming that the hand is directly in front of the camera may not explain the same 2D appearance at another location in the camera's field of view. We explain this further in Section A.1 and Figure A.

The third issue is a manifestation of ambiguity in predicting 3D from 2D image crops as described in past work [21]. At a high-level, [21] note that the same 2D image pattern at different locations in the image can correspond to different underlying 3D shape. This poses a problem when training a neural network that consumes the 2D crop as input to predict the 3D shape, from the same input the network is expected to predict different 3D outputs. We analyze this ambiguity in Section A.2 and Figure B.

Our use of KPE provides the neural network with information about the location that a crop comes from and circumvents these afore mentioned issues.

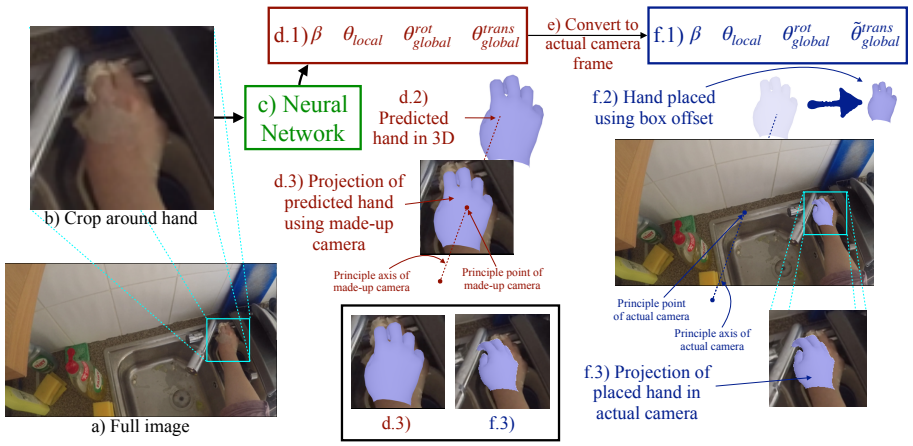

Fig. A: Given an image (shown in (a)) with a bounding box around a hand to predict 3D pose for, FrankMocap [24] and HaMeR [20] feed the crop (shown in (b)) to a neural network (shown in (c)). The neural network outputs the 3D shape, articulation and global pose, denoted by $\beta, \theta_{\text{local}}, \theta^{\text{trans}}_{\text{global}}, \theta^{\text{rot}}_{\text{global}}$ in (d.1). As the neural network is trained with 2D reprojection losses that assume a *made up camera* with the principal point being at the center of the crop and a fixed focal length, these predictions (shown in (d.2)) when projected according to this made-up camera, conform well with the 2D image as shown in (d.3). FrankMocap and HaMeR then convert these predictions to the actual camera frame as shown in (e) by adjusting the translation part of global pose to $\tilde{\theta}^{\text{trans}}_{\text{global}}$, such that the projection of the root joint shifts by the same amount as the crop's shift in the image. **As the visualizations in (f.3) show, the 3D projection doesn't conform to the appearance of the 2D crop anymore!** This is because when the hand shifts in the camera's field of view it incurs perspective distortion causing it to project differently from when it is viewed head on. Note the differences in the projected hand shape in (d.3) and (f.3). The 3D that correctly explains the crop when viewed head on, necessarily can't explain the appearance of the crop where it actually is in the image.

A.1 3D that explains a 2D crop at a given location in image doesn't explain the 2D of the same crop at another location in the image

FrankMocap [24] and HaMeR [20] uses crops around hand as input to the neural network. The neural network resizes the crop to a fixed size and uses it to predict the 3D shape β, local 3D pose θ_{local} and global 3D pose θ_{global}. Here, we break the global pose into a translation and rotation component $\theta^{\text{trans}}_{\text{global}}$ and $\theta^{\text{rot}}_{\text{global}}$.

The network is trained with losses that measure similarity in local and global 3D shape, 3D keypoints and *2D projections of 3D joints in the 2D image*. For the 2D keypoint reprojection loss, the model assumes a made up camera matrix with a principal point at the center of the crop and a fixed focal length of 5000. Thus, the model learns 3D shapes and articulation such that the 3D keypoints when projected to 2D using this made-up camera, they line up with the ground truth 2D keypoints.

At inference time, FrankMocap [24] and HaMeR [20] then adjusts the prediction to account for the location and scale of the hand crop and the focal length of the camera (if known). Specifically, they adjust the *translation of the root joint*, *i.e.* $\theta^{\text{trans}}_{\text{global}}$ to say $\tilde{\theta}^{\text{trans}}_{\text{global}}$. It rescales the Z coordinate by inverse of the scale of the actual box, and modifies the X and Y coordinates such that after adjustment the root joint reprojects to the shifted box. Figure A shows the process.

The adjustments made by FrankMocap and HaMeR merely place the root joint at the correct location. They fail to account for the perspective distortion that varies across the image. The 3D hand that accurately conforms to the 2D keypoints when placed at the center of the field of view (d.3), will necessarily not conform to the shifted version of these keypoints (f.3). While this effect is always present, it is particularly severe for hands in egocentric images due to large field of view and hands being close to the camera.

A.2 Ambiguity in Absolute and Root-relative 3D Shape due to Cropping

Depending on the location in the camera's field of view, the same object looks different due to perspective distortion. Prakash *et al.* [21] note that this distortion creates ambiguity in perceiving shape from image crops, *i.e.* the same crop could correspond to different underlying absolute and root-relative 3D shapes depending on its location. Note that 3D from a single image is already an under-constrained task. Working with image crops, without keeping track of where the crop came from, further exaggerates this ambiguity. Recent work [16] makes a similar observation about ambiguity in absolute pose for human pose. In this section, we check if using crops around hands in egocentric images also lead to an additional ambiguity in absolute and root-relative 3D hand shape.

To answer this question, we analyze right hands in the ARCTIC [7] dataset since it contains 3D annotations for hand joints at different locations in the image. Consider the following distances between a pair of hands:

– **Pixel distance between hand crops,** measured as the pixel distance between the centroid of the 2D hand keypoints from the two hands.

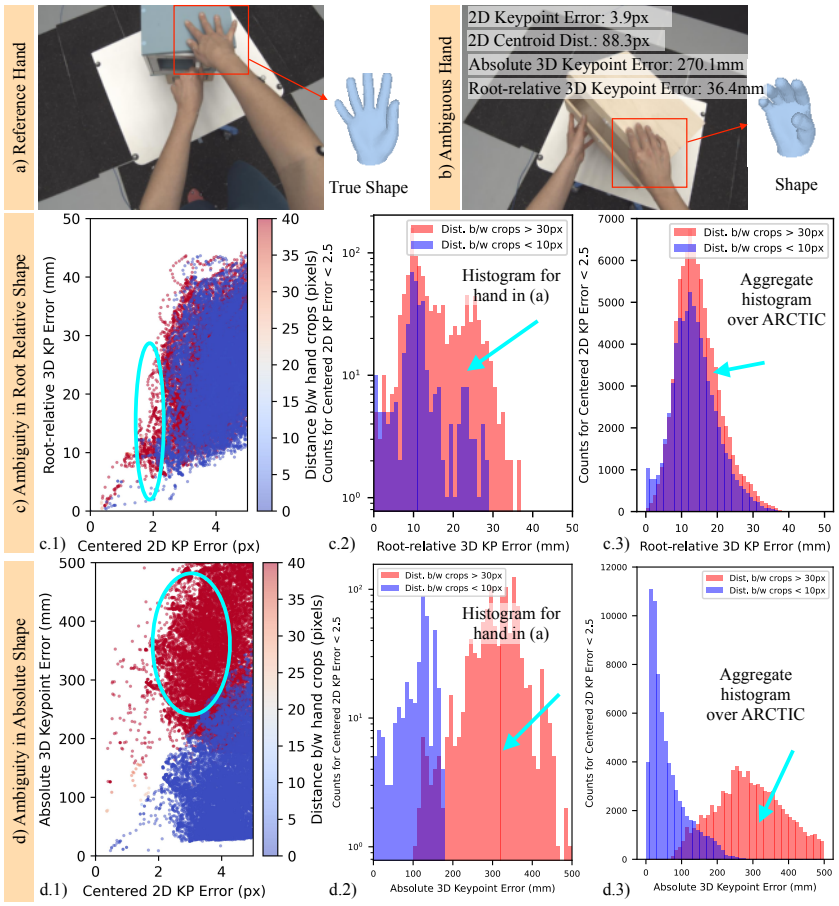

Fig. B: As compared to the reference hand in (a), the hand shown in (b) has a similar 2D shape (as measured by 2D keypoint error) but a very different 3D shape (as measured using root-relative 3D keypoint error). Fingers appear short due to being foreshortened in (a) but being further bent in (b). Thus, perspective distortion leads to "the same crop corresponding to different underlying 3D shape." Section A.2 discusses that this ambiguity (2D shape being similar but 3D shape being different) arises from ignoring the crop's location in the image. As we let crops go farther away, we start finding more such ambiguity (red points in b.1) than if we restrict to close-by crops (blue points in b.1), as pointed by the cyan circle. The histogram shown in (b.2) visualizes this in a different way to highlight (cyan arrow) the many more ambiguous cases in crops far away (red region) than in crops close by (blue region). (c.1) and (c.2) presents the same analysis but for *absolute 3D shape*, and finds much more ambiguity in absolute 3D shape than in root-relative 3D shape. (b.3 and c.3) presents this histogram aggregated over 200 randomly chosen reference hands from the ARCTIC dataset. Ambiguity is not just present in the reference hand in (a), but exists across the dataset. **KPE mitigates this ambiguity. Control experiments in Table 4 show improvements in relative and absolute pose across H2O, Assembly, Ego-Exo4D, and Epic-HandKps datasets.** In line with histograms in this figure, there is a larger improvement in absolute pose than in root relative pose.

- **Centered 2D keypoint error,** measures the average pairwise pixel distance between 2D keypoints from the 2 hands, *after bringing their centroids to the origin.*
- **Absolute 3D keypoint error,** as the residual between 3D keypoints to measure 3D shape mismatch between hands,
- **Root relative 3D keypoint error,** as the residual between the 3D keypoints *after aligning the root joints in location and orientation.*

Figure B (c.1) and Figure B (c.2) plots these distances between a reference hand (shown in Figure B (a)) and all other hands in ARCTIC. X-axis plots the centered 2D keypoint error (in pixels), Y-axis plots the 3D keypoint error between hands (absolute in a.1 and root-relative in a.2, in mm). The color of the point denotes the pixel distance between the hand crops.

When constraining the crop to be around the reference hand location (blue points in the plots) the 3D error is small, *i.e.* there is only a small amount of ambiguity. However, when we let the crop be anywhere in the image (the red points), even though the 2D keypoints within the crop are very similar, the 3D hand pose is quite different. The additional height of the red points (denoted by the cyan circles), denotes the *additional* ambiguity induced by ignoring the 2D location of the hand crop in the visual field on top of the ambiguity of making 3D inferences from 2D images (depicted by the blue points). Both absolute and root-relative 3D shape exhibit ambiguity.

Figure B (c.2) and Figure B (d.2) plots the distribution of the points in Figure B (c.1) and Figure B (d.1). Note again the additional ambiguity when crops can be anywhere in the image (red regions) *vs*. when the crops are close to the reference hand (blue regions) as pointed by the cyan arrows.

Figure B (c.3) and Figure B (d.3) aggregate these histograms over 200 randomly sampled hands from the ARCTIC dataset. Ambiguity is not just present in the reference hand in Figure B (a), but exists across the dataset.

Figure B (b) shows an example hand that has high 3D shape error but low centered 2D keypoint error. Fingers appear short due to being foreshortened in Figure B (a) but being further bent in Figure B (b).

To account for the larger number of crops present at arbitrary locations in the image compared to crops that are close to the reference hand, the distance in the plots in Figure B were computed after a PnP alignment [15] between the 2D keypoints for the reference hand and all 3D hands in ARCTIC. For each hand in ARCTIC, we do this alignment two ways. We do one alignment the default way, where PnP searches for the 3D rotation and translation that best aligns a hand to the 2D keypoints of the reference hand. We do another alignment that additionally allows for an arbitrary 2D shift in the hand keypoints for the reference hands. This balances the data and thus factors out differences due to spatially varying density of hands across the image.

Figure I shows qualitative examples for the hand shape and the corresponding location in the field of view after the PnP alignment to the reference 2D keypoints with appropriate shifts. It also shows similar histograms and visualizations without

the PnP alignment. Figure J and Figure K, show similar plots and visualization for another 2 hands from ARCTIC.

B Implementation Details

B.1 Architecture

We build on top of the ArcticNet-SF [7] and FrankMocap [24] models. Here, we provide additional details about the encoder, intrinsics-aware positional encoding (KPE [21]), decoder, bounding box predictor used in the architecture.

Encoder: Our models uses hand crops as input (resized to 224×224 resolution), which are processed by a ResNet50 [12] backbone to get $7 \times 7 \times 2048$ feature maps. The left and right hand crops as processed separately but the parameters are shared. We also use global image features in our model, computed by average pooling the $7 \times 7 \times 2048$ feature map to get a 2048-dimensional vector.

Incorporating KPE encoding: We add the intrinsics-aware positional encoding (KPE [21]) to the $7 \times 7 \times 2048$ feature map. KPE comprises sinusoidal encoding of the angles θ_x and θ_y (Sec. 4.1 in the main paper), resulting in $5 * 4 * K$ dimensional sparse encoding (4 for corners and 1 for center pixel) and $H \times W \times 4 * K$ resolution dense encoding, where K is the number of frequency components (set to 4). For the sparse KPE variant, we broadcast it to 7×7 resolution whereas for the dense KPE variant, we interpolate it to 7×7 resolution and concatenate to the feature map. This concatenated feature is passed to a 3 convolutional layers (with 1024, 512, 256 channels respectively, each with kernel size of 3×3 and ReLU [18] non-linearity) to get a $3 \times 3 \times 256$ feature map. This is flattened to 2304-dimensional vector and passed through a 1-layer MLP to get a 2048-dimensional feature vector. We do not use batchnorm [13] here since we want to preserve the spatial information in the KPE encoding whereas normalization would deteriorate it.

Decoder: It consists of an iterative architecture, similar to decoder in HMR [14]. The inputs are the 2048-dimensional feature vector and initial MANO [23] (shape β, articulation θ_{local} and global pose θ_{global}, all initialized as 0-vectors) & weak perspective camera parameters (initialized from the 2048-dimensional feature vector). Each of these parameters are predicted using a separate decoder head. The rotation parameters θ_{local}, θ_{global} are predicted in matrix form and converted to axis-angle representation to feed to MANO model. Each decoder is a 3-layer MLP with the 2 intermediate layers having 1024 channels and the output layer having the same number of channels as the predicted parameter. The output of each decoder is added to the initial parameters to get the updated parameters. This process is repeated for 3 iterations. The output of the last iteration is used for the final prediction.

For the auxiliary supervision used to train our model, we also predict hand segmentation masks and grasp labels. The hand parameters β, θ_{local}, and θ_{global} are passed to a differentiable MANO layer [10, 23] to get the hand vertices. These vertices are used to differentiably render a soft segmentation mask using SoftRasterizer [17, 22]. The grasp classifier head on θ_{local}, θ_{global} & β (predicted

by WildHands) as input and is implemented as a 4-layer MLP (with 1024, 1024, 512, 128 channels and ReLU non-linearity after each). This MLP predicts logits for the 8 different grasp classes defined in [2].

Bounding box predictor: Since our model takes hand crops as input, we need to predict the bounding box of the hand in the image. On ARCTIC, we train a bounding box predictor on the ARCTIC training set by finetuning a MaskRCNN [11] model. For Epic-HandKps, we use the recently released hand detector from [1]. During training, we use the ground truth bounding box for the hand crop (with small perturbation), estimated using the 2D keypoints and scaled by a fixed value of 1.5 to provide additional context around the hand. All the ablations use ground truth bounding box for the hand crop.

B.2 Training

We use different sources of supervision to train our model, depending on the dataset. We train WildHands jointly on multiple datasets: (1) Arctic [7]: using 3D supervision on β, θ_{local}, θ_{global}, 3D hand keypoints, 2D projections of 3D keypoints in the image and translation of root joint w.r.t. camera, (2) AssemblyHands [19]: using supervision on 3D hand keypoints and 2D projections of 3D keypoints in the image (it does not represent hands using MANO), (3) Epic-Kitchens [3]: using segmentation masks and grasp labels, and (4) Ego4D [8]: using segmentation masks and grasp labels. Following [7], we use L2 loss for β, θ_{local}, θ_{global}, 3D keypoints and 2D keypoints. L1 loss is used for segmentation masks and cross-entropy for grasp lables. The loss weights used for each terms are: 5.0 for 2D keypoints, 5.0 for 3D keypoints, 10.0 for θ_{global}, 10.0 for θ_{local}, 0.001 for β, 1.0 for translation, 10.0 for segmentation and 0.1 for grasp loss.

We use the training sets of ARCTIC (187K images), AssemblyHands (360K), VISOR split (30K) of Epic-Kitchens & 45K images from Ego4D to train different models used in the experiments. The validation is done on a separate set of 2D keypoints annotations collected on 250 hand crops from Epic-Kitchens train set. All models are trained for 100 epochs with a learning rate of 1e-5. We also adopt early stopping to terminate the training if the validation loss does not improve for 5 checkpoints. The checkpoints are saved every 5 epochs for models trained without AssemblyHands & every 1 epoch for models trained with AssemblyHands. For multi-gpu training, we use DDP strategy in pytorch lightning.

C Additional Analysis

In the main paper, we report results in the zero-shot generalization setting on AssemblyHands, H2O, Ego-Exo4D and Epic-HandKps. Here, we provide further details on Ego-Exo4D evaluation, HaMeR [21] experiments with models training in different settings and non-zero-shot results of WildHands on ARCTIC, AssemblyHands and Epic-HandKps. We use the same metrics as the experiments in the main paper.

EgoExo4D evaluation: We use 3D hands annotations from the validation split of Ego-Exo4D dataset. These 3D annotations are computed by running 3D triangulation with the 2D hand keypoints annotations in different views. These views are obtained from 1 egocentric camera and 4 exocentric cameras in the scene. Since these exocentric cameras are often far from the user, the hands appear quite small in the image due to which the 2D keypoints are often inaccurate. This leads to very noisy 3D annotations even though RANSAC is used to reduce the effect of outliers and post-processing is done to filter annotations. We do not report MRRPE for Ego-Exo4D results due to large noise in wrist annotations. Note that the egocentric image in Ego-Exo4D contains large amounts of radial distortion due to the use of fisheye lens, so we remove the distortions in raw sensor data using the official code[1], before providing as input to the model.

C.1 HaMeR Experiments

Comparisons with HaMeR [20]: In Tab.6 of the main paper, we compare to the recently released HaMeR model. HaMeR is trained in 2 different settings: (a) with 7 lab + 3 wild datasets (5%), (b) also adding the recent HInt [20] dataset containing in-the-wild images from New Days subset of Hands23 [2], VISOR [4] and Ego4D [9] with 2D keypoint annotations. Here, we report results of HaMeR trained in both settings in Tab. A. WildHands outperforms FrankMocap across all metrics and beats HaMeR on 3 of 6 metrics. We expect scaling up the backbone and datasets used to train WildHands can lead to even stronger performance.

Table A: Systems comparison. We compare with publicly released models: FrankMocap [24] and concurrent work HaMeR [20]. FrankMocap uses a ResNet-50 backbone and was trained on 6 lab datasets. HaMeR uses a ViT-H [6] backbone and was trained on 12 lab+in-the-wild datasets across nearly 3M images.

	H2O		**Assembly**		**Ego-Exo4D**	**Epic-HandKps**
	MPJPE	**MRRPE**	**MPJPE**	**MRRPE**	**MPJPE**	**L2 Error**
FrankMocap [24] (ResNet-50, 6 lab)	58.51	-	97.59	-	175.91	13.33
HaMeR [20] (ViT-H, 7 lab + 3 wild)	25.99	148.51	**44.49**	335.63	110.96	**4.47**
HaMeR [20] (ViT-H, 7 lab + 3 wild + HInt)	**23.82**	147.87	45.49	334.52	116.46	4.56
WildHands (ResNet-50, 2 lab + 1 wild)	31.08	**49.49**	80.40	**148.12**	**55.84**	7.20

Epic-HandKps results for HaMeR: We also report HaMeR results on Epic-HandKps for both variants, trained with and without HInt, in Tab. A. We compute the L2 metric by transforming the 3D hand mesh prediction from crop frame to the camera coordinate frame and projecting in the full image using ground truth intrinsics. We observe that training HaMeR with HInt leads to slight improvement in most settings. Our WildHands models outperforms HaMeR on in-the-wild settings, *i.e.*, Ego-Exo4D and Epic-HandKps.

[1] https://github.com/EGO4D/ego-exo4d-egopose/tree/main/handpose/data_preparation

Note that another way of evaluating 2D pose for HaMeR is to use the 2D hand predictions in the crop frame and rescale & shift them to the full image. This leads to a 2D pose error of 3.24 for HaMeR and 3.52 for HaMeR trained with HInt on Epic-HandKps. This is much better than the 2D projections of 3D hand mesh in the full image. This is due to the made up camera used by HaMeR, which assumes a fixed focal length of 5000 during training. Egocentric images generally operate in a much smaller focal length resulting in large difference between the results. WildHands uses the ground truth intrinsics as input to the model and is able to predict much better 3D hands in egocentric images.

C.2 Additional Ablations

Here, we show results with models trained on the same datasets used for evaluation. Specifically, we show improvements due to crops & KPE (Tab. B), auxiliary supervision (Tab. C), effect of scaling up data (Tab. D) & 3D evaluation results on AssemblyHands (Tab. E). The trends are consistent with the zero-shot experiments. One notable difference is that we do not see much benefits on ARCTIC and AssemblyHands in 3D metrics when evaluating WildHands trained on these datasets. This is likely due to the use of strong 3D ground truth during training, thus auxiliary supervision from out-of-domain EPIC not being very useful. We provide further ablations on the models reported in the main paper below.

Using predicted hand bounding box: As mentioned in Sec. B.1, we finetune a MaskRCNN model to predict bounding box on ARCTIC and use hand predictor from [2] on Epic-HandKps. When using predicted bounding box instead, we see a drop in performance on both ARCTIC and Epic-HandKps (Tab. F). This is expected since the predicted bounding box is not always accurate. We see similar trend for FrankMocap [24] as well.

Ignoring camera intrinsics: The KPE [21] encoding captures the location of the hand crop in the camera's field of view. However, camera intrinsics may not always be available in the wild. So, we also explore a variant of our model which ignores camera intrinsics and uses only the crop location w.r.t. to the image center (Tab. F). We expect the intrinsics to matter more in multiple datasets setting involving images captured from different cameras, so we remove intrinsics from the model trained jointly on ARCTIC, AssemblyHands & Epic-Kitchens. While we notice a slight dip in performance, it is significantly better than ArcticNet-SF [7] or FrankMocap [24].

Removing global features from grasp classifier: Our model uses a classifier to predict the grasp type from the estimated hand pose parameters. While hand pose is indicative of grasp type, hand poses might also occur in thin air, without actually grasping the object. So, we verify if using global image information could be useful in this case. From the results in Tab. F, we see that removing global image features improves L2 error on Epic-HandKps. This could be because the pseudo ground truth is generated using the predictions of a recent model [2] which might already be trained to distinguish between hand poses in thin air and grasping poses. However, we see a drop in performance on ARCTIC when

removing global features. This is likely due to the model exploiting information from surrounding objects since all the 11 objects are scene during training.
Transformer variant: We modify the architecture of WildHands to use transformers [5, 25] instead of convolutions. This increases the capacity our the model by 4 times. We notice a significant improvement on ARCTIC (Tab. F) but not on Epic-HandKps. In fact, the performance decreases. This could be due to large capacity leading to overfitting on ARCTIC since varition in ARCTIC is limited. Another could be that strong supervision is not available on in-the-wild data used for training which could be causing the transformer to overfit to weird signals in the data. This needs more analysis to better understand the effect of transformer in our model.
Object loss in ArcticNet-SF: The default ArcticNet-SF [7] model predicts both the hand pose and the object pose from a single image. Since we focus only on hand pose, we also examine the effect of removing the object loss. We notice that MPJPE worsens but MRRPE improves. It is hard to identify the reason but could be due to the model overfitting to weird signals in objects since all 11 objects are seen during training. The performance on Epic-HandKps decreases, which also indicates some extent of overfitting to ARCTIC objects.

D Epic-HandKps

Since 3D hand annotations are difficult to collect for in-the-wild images, we instead collect 2D annotations for the 21 hand joints and use it to evaluate the 2D projections of the predicted 3D keypoints. We refer to this dataset as Epic-HandKps. We sample 5K images from the validation set of VISOR split of Epic-Kitchens and get the 21 joints annotated via Scale AI. We use the same joint convention as ARCTIC [7]. We crop the images around the hand using the segmentation masks in VISOR and provide the crops to annotators for labeling. Note that most of these images do not have all the 21 keypoints visible. Following ARCTIC, we only consider images with atleast 3 visible joints for evaluation. Moreover, since the models in our experiments required hand crops as input, we only evaluate on those images for which hand bounding box is predicted by the recently released hand detector model from [2]. This leaves us with 4724 hand annotations, with 2697 right hands and 2027 left hands. We show sample annotations in Fig. C.

E Visualizations

Qualitative comparisons: We provide visualizations of the predicted hand pose from WildHands & FrankMocap in Fig. E and WildHands & HaMeR in Fig. D. We show projection of the predicted hand in the image and rendering of the hand mesh from 2 more views. Our WildHands model is able to predict better hand poses from a single image than FrankMocap [24] in challenging occlusion scenarios involving dexterous interactions, *e.g.* stirring a ladle, grasping objects, rolling dough, pick & place, sprinkling toppings. We also observe improvements

Fig. C: Epic-HandKps annotations. We collect 2D joint annotations (shown in blue) for 5K in-the-wild egocentric images from Epic-Kitchens [3] dataset. We show few annotations here. We also have the label for the joint corresponding to each keypoint. Note the large variation in dexterous poses of hands interactiong with objects.

over HaMeR: 2D projections of the 3D hand predicted by WildHands align better with the hand in the image and the scale of the predicted hand is more accurate. These are consistent with quantitative improvements in Tab. A.

We provide similar visualizations on ARCTIC in Fig. G and compare with ArcticNet-SF [7]. Our WildHands model predicts better hand poses in scenes involving interaction with articulated objects.

Failure Cases: On Epic-HandKps (Fig. F), we observe that images in which the fingers are barely visible, *e.g.* when kneading a dough in top row, are quite challenging. Moreover, our model is sometimes unable to predict wide palm poses, *e.g.* grasps in bottom row. On ARCTIC (Fig. H), we observe similar failure cases, *i.e.* when fingers are barely visible (top row), wide palm poses (bottom row).

Fig. D: Qualitative comparison with HaMeR on Epic-HandKps. Our WildHands model is able to predict better hand poses from a single image than HaMeR [20] in challenging occlusion scenarios involving dexterous interactions, *e.g.* images of hands grasping the objects from Epic-HandKps. Our predictions are better aligned with the hands in the image and HaMeR sometimes predicts large size hands. We show projection of the predicted hand in the image and rendering of the hand mesh from 2 more views.

Fig. E: Visualizations on Epic-HandKps. Our WildHands model is able to predict better hand poses from a single image than FrankMocap [24] in challenging occlusion scenarios involving dexterous interactions, *e.g.* images of hands grasping the objects from Epic-HandKps. We show projection of the predicted hand in the image and rendering of the hand mesh from 2 more views.

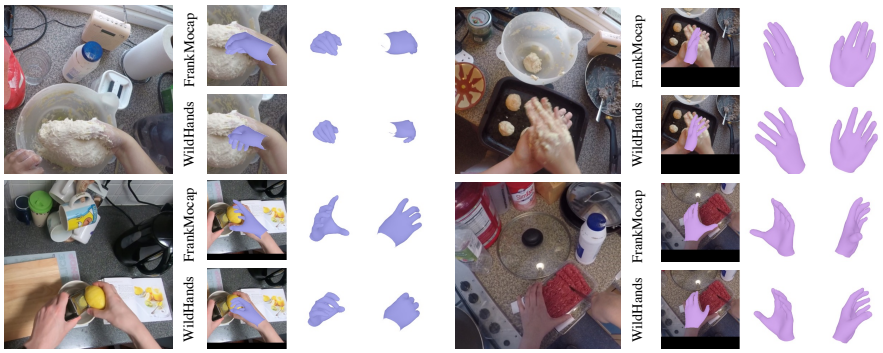

Fig. F: Failure cases on Epic-HandKps. We observe that images in which the fingers are barely visible, *e.g.* when kneading a dough in top row, are quite challenging. Moreover, our model is sometimes unable to predict wide palm poses, *e.g.* grasps in bottom row.

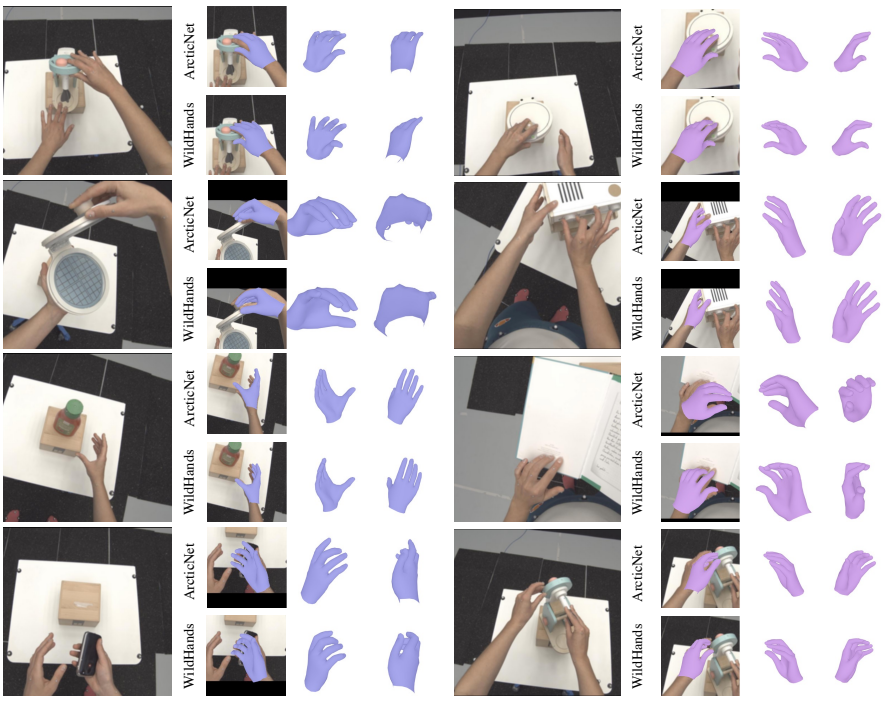

Fig. G: Visualizations on ARCTIC. Our WildHands model is able to predict better hand poses from a single image than ArcticNet-SF [7] in scenes involving interaction with articulated objects. We show projection of the predicted hand in the image and rendering of the hand mesh from 2 additional views.

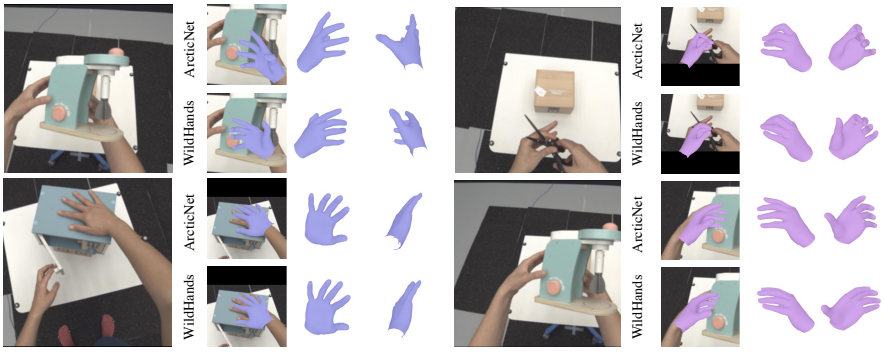

Fig. H: Failure cases on ARCTIC. We observe similar failure cases as Fig. F, *i.e.* when fingers are barely visible (top row), wide palm poses (bottom row).

Table B: Role of Cropping and KPE Encoding. Switching to crops from full image input (as used in ArcticNet-SF [7]), we see a small improvement in root relative shape metric (MPJPE) but a huge degradation in global pose metrics (MRRPE). Adding in the KPE encoding, either sparse or dense, maintains the root relative shape metric, but improves the global pose metric significantly, while also improving the Epic-HandKps metrics. To maximally isolate the impact of cropping and KPE encodings, these comparisons are done without any auxiliary supervision.

Method	**network input**	**KPE encoding**	**ARCTIC val** MPJPE ↓	MRRPE ↓	**Epic-HandKps** L2 error ↓
ArcticNet-SF [7]	image	n/a	22.60	32.71	35.07
WildHands (no aux.)	crop	-	21.40	58.24	34.12
WildHands (no aux.)	crop	dense	**21.13**	30.20	19.99
WildHands (no aux.)	crop	sparse	21.50	**28.70**	**17.07**

Table C: Role of auxiliary supervision. Both grasp and segmentation auxiliary supervision contribute to the final performance, with segmentation providing a bigger gain as compared to grasp labels.

Method	**ARCTIC val** MPJPE ↓	MRRPE ↓	**Epic-HandKps** L2 error ↓
WildHands	**20.57**	28.81	**6.68**
− grasp	21.64	30.52	9.27
− segmentation	20.99	29.75	13.60
− grasp − segmentation	21.50	**28.70**	17.07

Table D: Effect of Scaling-up Data. While additional datasets on top of the ARCTIC data do not improve metrics on ARCTIC, adding either Assembly (3D supervision) or EPIC (with auxiliary supervision) improves metrics on Epic-HandKps, with the auxiliary supervision from the in domain EPIC data helping more.

Training Datasets	**ARCTIC val** MPJPE ↓	MRRPE ↓	**Epic-HandKps** L2 error ↓
ARCTIC	21.50	**28.70**	17.07
ARCTIC + Assembly	23.20	30.00	11.05
ARCTIC + EPIC (aux)	**20.57**	28.81	6.68
ARCTIC + Assembly + EPIC (aux)	21.76	29.66	**6.56**

Table E: Results on AssemblyHands. Using crops and KPE encodings also improve performance when training and testing on the AssmeblyHands dataset.

Method	network input	KPE encoding	Assembly val MPJPE ↓	Assembly val MRRPE ↓
ArcticNet-SF	image	n/a	23.96	58.21
WildHands (no aux.)	crop	n/a	22.55	53.92
WildHands (no aux.)	crop	sparse	22.24	35.58
WildHands (no aux., also trained on ARCTIC)	crop	sparse	**19.72**	**28.13**

Table F: Ablations. We explore variants of WildHands which use predicted bounding box instead of ground truth bounding box (GT hand box), ignoring camera intrinsics & using only crop location w.r.t. to image center, removing global image features from the grasp classifier (grasp-glb) and using transformer [5,25] instead of convolutional architecture. We also consider ArcticNet-SF [7] trained without object loss and FrankMocap [24] evaluation with predicted hand bounding box instead of GT hand box.

Method	ARCTIC val MPJPE ↓	ARCTIC val MRRPE ↓	Epic-HandKps L2 error ↓
WildHands	20.57	28.81	6.68
− GT hand box	21.02	30.31	7.28
− camera intrx	20.65	31.09	7.47
− segmentation	20.99	29.76	13.60
− segmentation − grasp-glb	21.90	29.87	12.41
WildHands (no aux.)	21.50	28.70	17.07
+ Transformer	19.50	27.11	20.36
ArcticNet-SF [7]	22.60	32.71	35.07
− no object loss	23.25	31.50	33.61
FrankMocap [24]	53.99	N/A	13.33
− GT hand box	57.00	N/A	14.57

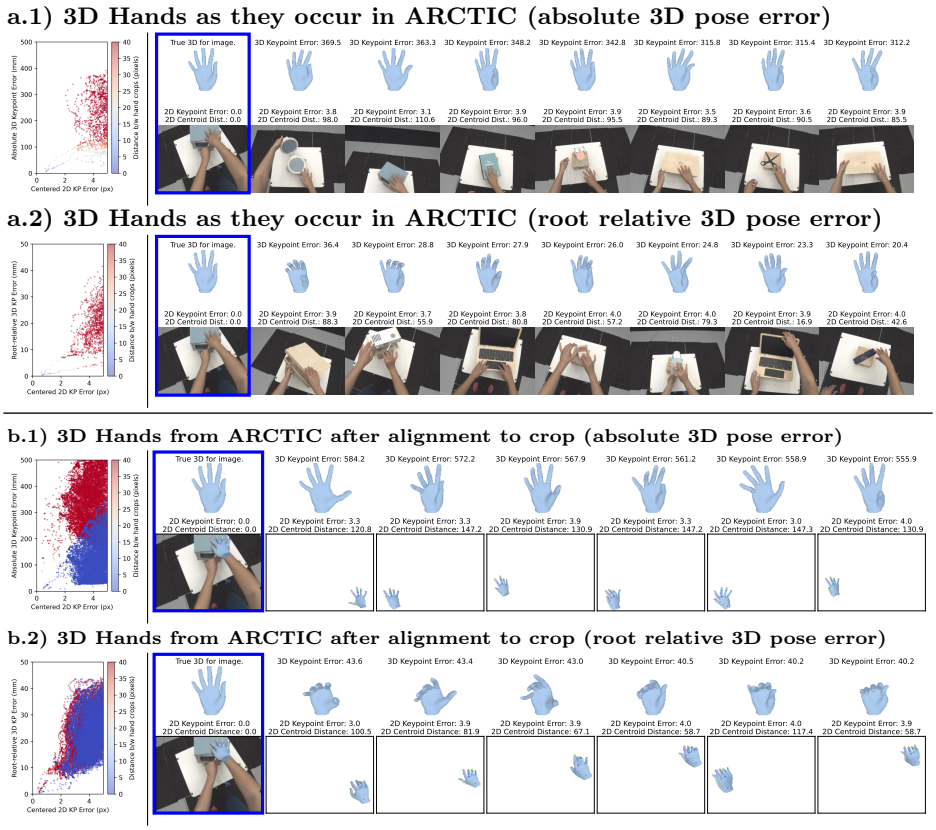

Fig. I: Qualitative visualizations for hands with high 3D shape error (absolute in (a.1) and root-relative in (a.2)) but low 2D keypoint error (after centering). The right hand in blue box is the reference hand w.r.t. which we measure 2D and 3D keypoint errors. **(b.1 and b.2)** show examples for the 3D hand shape and the corresponding location in the field of view such that the centered 2D projection of the hand keypoints match the reference in the blue box. Across all 4 settings, note the diversity in 3D hand shape and pose but the similarity in centered 2D hand pose. Scatter plots on the left are same as those in Fig. 3 in the main paper. Fig. J and Fig. K provide 2 more examples.

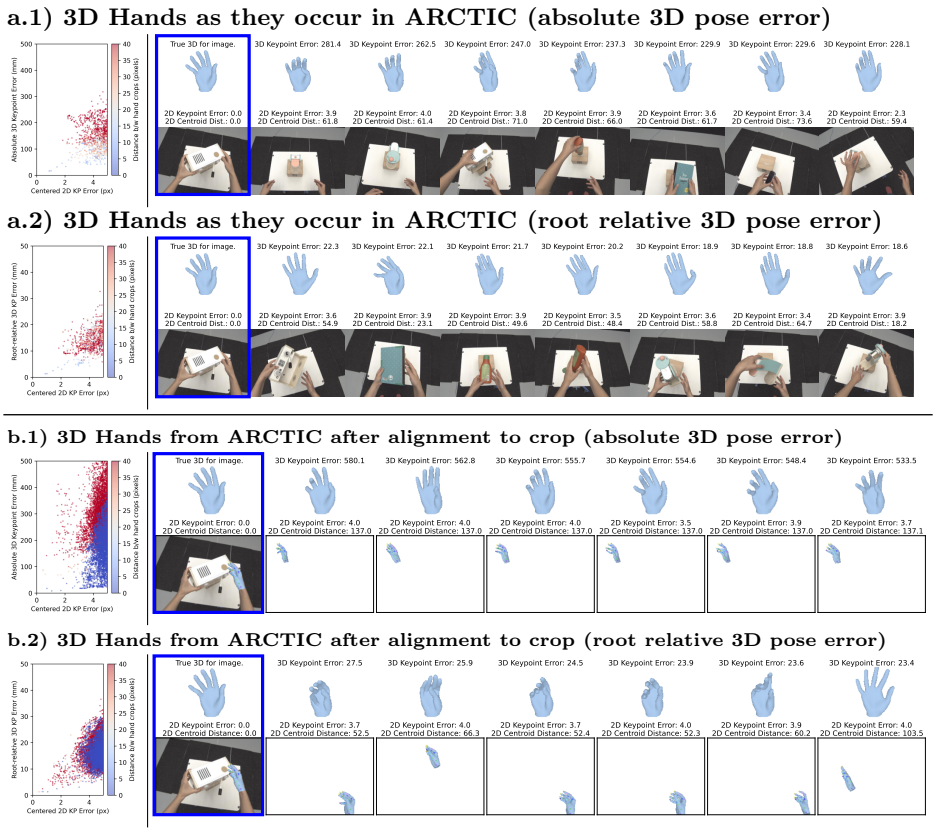

Fig. J: Another example similar to Fig. I. **Qualitative visualizations for hands with high 3D shape error (absolute in (a.1) and root-relative in (a.2)) but low 2D keypoint error (after centering).** The right hand in blue box is the reference hand w.r.t. which we measure 2D and 3D keypoint errors. **(b.1 and b.2)** show examples for the 3D hand shape and the corresponding location in the field of view such that the centered 2D projection of the hand keypoints match the reference in the blue box. Across all 4 settings, note the diversity in 3D hand shape and pose but the similarity in centered 2D hand pose.

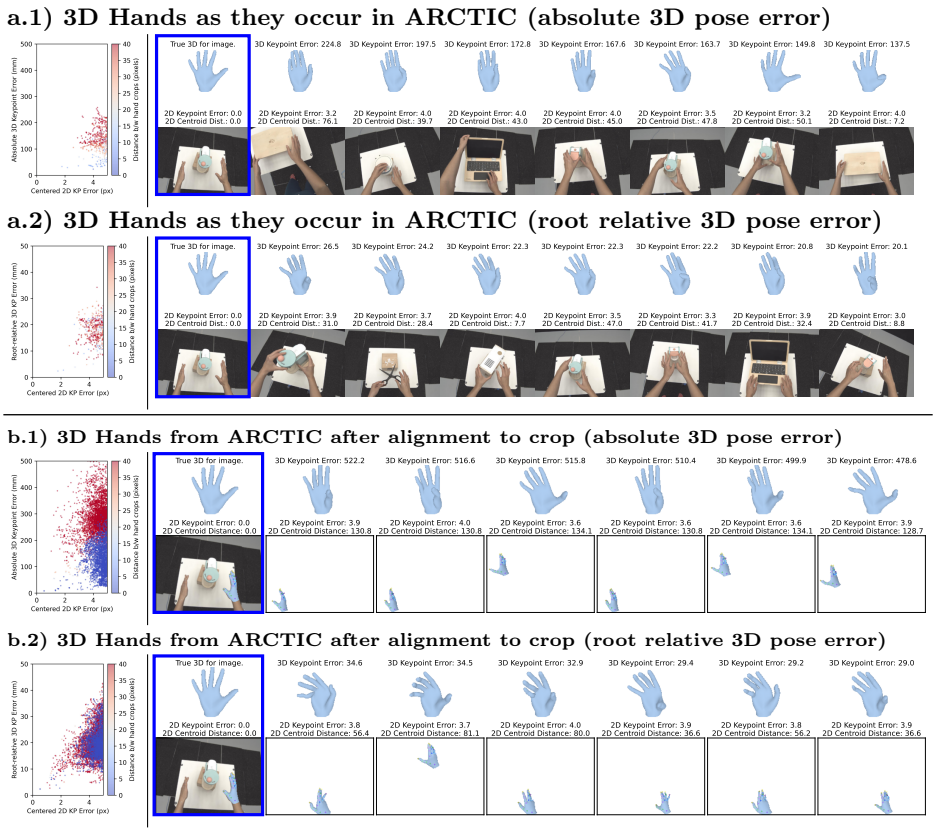

Fig. K: Another example similar to Fig. I. **Qualitative visualizations for hands with high 3D shape error (absolute in (a.1) and root-relative in (a.2)) but low 2D keypoint error (after centering).** The right hand in blue box is the reference hand w.r.t. which we measure 2D and 3D keypoint errors. **(b.1 and b.2)** show examples for the 3D hand shape and the corresponding location in the field of view such that the centered 2D projection of the hand keypoints match the reference in the blue box. Across all 4 settings, note the diversity in 3D hand shape and pose but the similarity in centered 2D hand pose.

References

1. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
2. Cheng, T., Shan, D., Hassen, A.S., Higgins, R.E.L., Fouhey, D.: Towards a richer 2d understanding of hands at scale. In: Advances in Neural Information Processing Systems (NeurIPS) (2023)
3. Damen, D., Doughty, H., Farinella, G.M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., Wray, M.: The epic-kitchens dataset: Collection, challenges and baselines. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2020)
4. Darkhalil, A., Shan, D., Zhu, B., Ma, J., Kar, A., Higgins, R., Fidler, S., Fouhey, D., Damen, D.: Epic-kitchens visor benchmark: Video segmentations and object relations. In: NeurIPS Track on Datasets and Benchmarks (2022)
5. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: Proceedings of the International Conference on Learning Representations (ICLR) (2021)
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
7. Fan, Z., Taheri, O., Tzionas, D., Kocabas, M., Kaufmann, M., Black, M.J., Hilliges, O.: ARCTIC: A dataset for dexterous bimanual hand-object manipulation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
8. Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., et al.: Ego4d: Around the world in 3,000 hours of egocentric video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
9. Grauman, K., Westbury, A., Torresani, L., Kitani, K., Malik, J., Afouras, T., Ashutosh, K., Baiyya, V., Bansal, S., Boote, B., et al.: Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. arXiv preprint arXiv:2311.18259 (2023)
10. Hasson, Y., Varol, G., Tzionas, D., Kalevatykh, I., Black, M.J., Laptev, I., Schmid, C.: Learning joint reconstruction of hands and manipulated objects. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
11. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
13. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Bach, F.R., Blei, D.M. (eds.) Proceedings of the International Conference on Machine Learning (ICML) (2015)
14. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-end recovery of human shape and pose. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)

15. Lepetit, V., Moreno-Noguer, F., Fua, P.: EPnP: An Accurate O(n) Solution to the PnP Problem. International Journal of Computer Vision (IJCV) (2009)
16. Li, Z., Liu, J., Zhang, Z., Xu, S., Yan, Y.: Cliff: Carrying location information in full frames into human pose and shape estimation. In: European Conference on Computer Vision. pp. 590–606. Springer (2022)
17. Liu, S., Li, T., Chen, W., Li, H.: A general differentiable mesh renderer for image-based 3d reasoning. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2020)
18. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the International Conference on Machine Learning (ICML) (2010)
19. Ohkawa, T., He, K., Sener, F., Hodan, T., Tran, L., Keskin, C.: Assemblyhands: Towards egocentric activity understanding via 3d hand pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12999–13008 (2023)
20. Pavlakos, G., Shan, D., Radosavovic, I., Kanazawa, A., Fouhey, D., Malik, J.: Reconstructing hands in 3d with transformers. arXiv preprint arXiv:2312.05251 (2023)
21. Prakash, A., Gupta, A., Gupta, S.: Mitigating perspective distortion-induced shape ambiguity in image crops. arXiv **2312.06594** (2023)
22. Ravi, N., Reizenstein, J., Novotny, D., Gordon, T., Lo, W.Y., Johnson, J., Gkioxari, G.: Accelerating 3d deep learning with pytorch3d. arXiv:2007.08501 (2020)
23. Romero, J., Tzionas, D., Black, M.J.: Embodied hands: Modeling and capturing hands and bodies together. ACM Transactions on Graphics (ToG) (2017)
24. Rong, Y., Shiratori, T., Joo, H.: Frankmocap: Fast monocular 3D hand and body motion capture by regression and integration. Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV Workshops) (2021)
25. Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Advances in Neural Information Processing Systems (NeurIPS) (2017)