Supplementary Material
Measuring Generalisation to Unseen Viewpoints, Articulations, Shapes and Objects for 3D Hand Pose Estimation under Hand-Object Interaction

1 Introduction

In this document, we provide the reader with more analyses of the evaluated approaches in the main paper and the experiments conducted specifically for some participating methods.

To give more insights to the readers, Fig. 1 shows different level of errors for a given test image on the mean joint errors (MJE) in mm when projected on the 2D image. We also provide an overview of the approaches used in the main paper in Tables 1, 2 and 3. Later, we present qualitative results from the approaches on each task in Figures 2, 3 and 4.

Furthermore, this document provides more experimental results in Section 2, conducted by the participated approaches based on different backbone architectures and similarly, Section 3 shows experimental evaluation on the ensembling techniques in pre-processing, post-processing and methodological level.

Frame success rate analysis with additional approaches are also presented in Section 4 and followed by the analysis in Section 5 on joint success rates.

Section 6 shows analysis of the methods based on the structured test set for seen/unseen viewpoints, articulations, shapes and objects. Please refer to Fig. 6, 7 and 8 of the main paper for the error visualizations. These accuracies are also more meaningful when we refer to the training and test data distribution in Fig. 4 of the main paper.

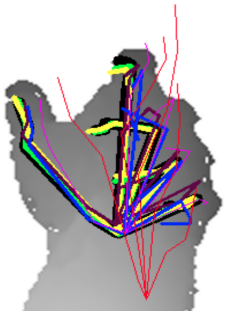


Fig. 1: Visualization of **ground-truth** hand pose and poses with varying level of MJEs, $< 5mm$, $< 10mm$, $< 20mm$, $< 30mm$, $< 40mm$, $< 60mm$. More specifically, MJE (mm) of the visualized poses are 1.75, 6.88, 13.94, 15.32, 35.67, 52.15, respectively. Best viewed in color.

Table 1: Task 1 - Methods' Overview

Username	Description	Input	Pre-processing	Post-processing	Synthetic Data	Backbone	Loss	Optimizer
Rokid [22]	2D CNN joint regression	Depth 224×224	Initial pose est. to crop	✗	570K Synthetic + Mixed Synthetic	EfficientNet-b0 [16]	Wing [1]	Adamax
A2J [18]	2D CNN, offset + depth regression with anchor points and weighting	Depth 384×384	Bbox crop	Scale+rotation, 10 backbone models ensemble	✗	ResNet-152	Smooth L1	Adam
AWR [5]	2D CNN, dense direction & offset rep. Learnable adaptive weighting	Depth 256 × 256 segm. 128 × 128 pose est.	Bbox crop ESPNet-v2 [7] for binary segm. iter. refinement of CoM	Ensemble from 5 models	✗	ResNet-50&101 SRN [9] HRNet [14]	Smooth L1	Adam
NTIS [8]	3D CNN Deeper V2V-PoseNet [8] Weighted sub-voxel prediction	Voxels 88×88 × 88	Multi-scale CoM refinement hand cropping	Models from 6 training epochs N confident sub-voxel pred. Truncated SVD refinement	✗	V2V-PoseNet	L2	RMSProp
Strawberryfg	Integral Pose Regression [15] 3D supervision voxels +volume rendering	Depth image 256 × 256 3D point proj. Multi-layer depth Voxels	Coarse-to-fine hand cropping by thresholding	✗	✗	ResNet-50	L1	RMSProb
BT [6]	3D supervision with cloud reconst. Permutation invariant [6] Point-to-pose + point-to-latent voting.	Point cloud 512 3D vectors	View correction [6]	✗	✗	ResPel [6] for feat. extract FoldingNet [20] for reconstruction	L2 Chamfer and EMD KL constraint	Adam

Table 2: Task 2 - Methods' Overview

Username	Description	Input	Pre-processing	Post-processing	Synthetic Data	Backbone	Loss	Optimizer
NTIS [8]	3D CNN Deeper V2V-PoseNet [8] Weighted sub-voxel prediction	Voxels 88×88 × 88	Multi-scale com-ref-net for hand cropping	Models from 6 training epochs N sub-voxel pred., Truncated SVD and temporal smoothing refinement	✗	V2V-PoseNet	L2	RMSProp
A2J [18]	2D CNN offset and depth regression with anchor points and weighting	Depth 256×256	Bbox crop	Ensemble predictions from 3 training epochs	✗	SEResNet-101 [4]	Smooth L1	Adam
CrazyHand	2D CNN tree-like branch structure regression with hand morphology	Depth 128×128	Iterative CoM	✗	✗	ResNet-50	L2	-
BT [6]	Differentiable Mano [12] layer Permutation invariant [6] Point-to-pose+ point-to-latent voting	Point cloud 512 3D points	View correction [6]	✗	32K synthetic + random objects from HO-3D [2]	ResPel [6]	L2 pose L2 Mano vertex KL constraint	Adam

Table 3: Task 3 - Methods' Overview

Username	Description	Input	Pre-processing	Post-processing	Synthetic Data	Backbone	Loss	Optimizer
ETH_NVIDIA [13]	2D CNN, 2D location + relative depth Heatmap-regression + an MLP for denoising absolute root depth	RGB 128×128	Bbox crop	✗	✗	ResNet-50	L1	SGD
NLE [11]	2D hand proposals + classification of multiple anchor poses + regression of 2D-3D keypoint offsets *w.r.t.* the anchors	RGB 640×480	✗	Ensemble multiple pose proposals and ensemble over rotated images	✗	ResNet-101	Smooth L1 for reg. Log loss for classif. RPN [10] for localization loss	SGD
BT [19]	Multi-modal input with latent space alignment [19] Differentiable Mano [12] layer	RGB 256 × 256 Point cloud - 356	Bbox cropping	✗	100K synthetic + random objects from HO-3D [2]	EncoderCloud: ResPEL [6] EncoderRGB: ResNet-18 DecoderMano: 6 fully-connected	L2 pose, L2 Mano vert. Chamfer, Normal and Edge length for mesh KL constraint	Adam

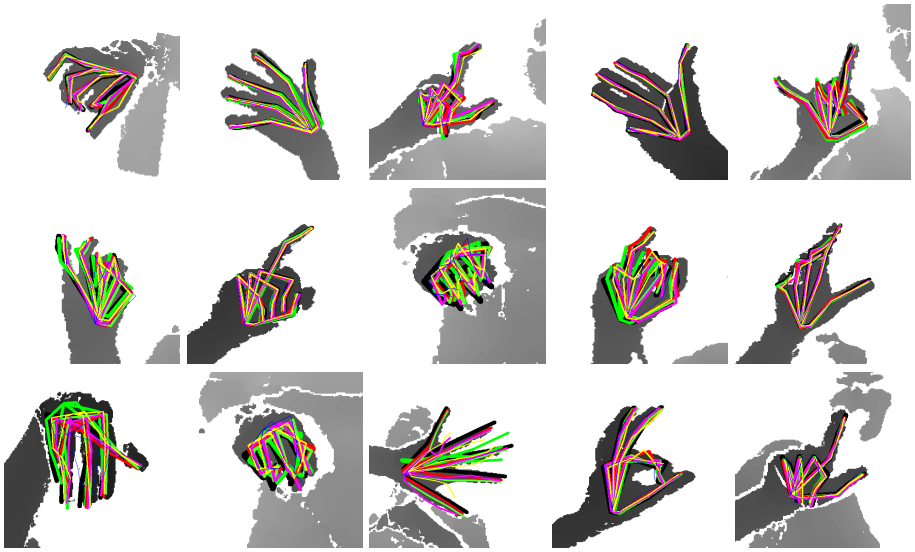


Fig. 2: Task 1 - Visualization of the **ground-truth** annotations and estimations of ***Rokid***, ***A2J***, ***AWR***, ***NTIS***, ***Strawberryfg***, ***BT***.

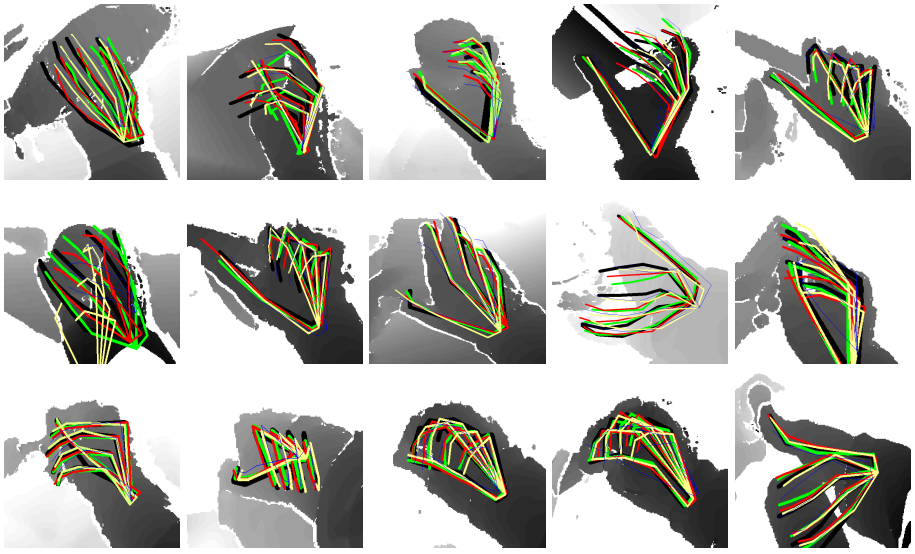


Fig. 3: Task 2 - Visualization of the **ground-truth** annotations and estimations of ***NTIS***, ***A2J***, ***CrazyHand***, ***BT***.

(a) Extrapolation (b) Shape (c) Object (d) Interpolation

Fig. 4: Task 3 - Visualization of the **ground-truth** annotations and estimations of ***ETH_NVIDIA***, ***NLE***, ***BT***. Each column shows different examples used in our evaluation criteria.

2 Experiments with Different Backbone Architectures

While Residual Network (ResNet) [3] backbones are well adopted by many approaches and ResNet-50 or ResNet-101 architectures obtain better results compared to other backbone models as reported in experiments of *AWR* and *NLE*. However, most approaches adopt ensembling predictions from models trained with different backbone architectures and this improves the performance as showed in Table 4 and Table 5.

Table 4: Extrapolation MJE obtained with different backbone architectures in *AWR* experiments. 'center1' denotes using thresholds to compute hand center, 'center2 + original' denotes using semantic segmentation network to compute hand center and extract hand region from original depth images, 'center2 + segmented' denotes using semantic segmentation network to compute hand center while extract hand region from network's output mask.

Backbone	Extrapolation MJE (mm)
Resnet50 (center1)	20.70
Resnet50 (center2 + original)	14.89
Resnet50 (center2 + segmented)	14.75
Resnet101 (center2 + original)	14.57
Resnet101 (center2 + segmented)	14.44
HRnet48	17.23
SRN	16.00
SRN_multi_size_ensemble	15.20
HRNet_Resnet50_shape_ensemble	14.68
model_ensemble	**13.67**

Table 4 shows the experiments for impact of different network backbones and different ways of obtaining the hand center by *AWR*. Changing the way of attaining hand center from 'center1' to 'center2 + original' yields an improvement of $5.81mm$, 'center2 + segmented' further improves by $0.14mm$. The best result is obtained with a backbone of ResNet-101, $14.44mm$.

At the final stage, multiple models are ensembled (model_ensemble in Table 4) including ResNet-101 (center2+segmented), ResNet-101 (center2+original), ResNet-50 (center2+original), SRN_multi_size_ensemble and HRNet_Resnet50-_shape_ensemble. Since ESPNetv2 [7] sacrifices accuracy for speed to some extent, the segmentation results are not accurate enough and may contain wrists or lack part of the fingers, therefore cropping hand regions from original depth images sometimes yields better performance.

Among the ensembled networks, SRN [9] is a stacked regression network which is robust to self-occlusion and when depth values are missing. It performs the best for Shape extrapolation, but is sensitive to the cube size that are used when cropping hand region. The mean error of a single-stage SRN with cube size

200mm already reaches 16mm. Ensembling SRN with cube size 180mm, 200mm and 220mm, the results of SRN_multi_size_ensemble is 15.20mm.

SRN performs the best on the shape evaluation axis. For example, single SRN can achieve 12.32mm and SRN_multi_size_ensemble can achieve 11.85mm.

HRNet-48 makes a major success in human pose estimation, but we do not get desired results after applying it. The mean error of single HRNet-48 is 17.23mm. Although it converges faster and has relatively lower loss than ResNet-50 and ResNet-101 in the training stage, it performs worse during inference. HRNet-48 predicts well on some of the shapes. Therefore, the depth images are divided into 20 categories according to the proportion of hand pixels over all pixels. The prediction error in training set is used to compute the weight of each category, which is used to weight the test set results. The weighted results depicted with HRNet_Resnet50_shape_ensemble reaches mean error of 14.68mm.

The model_ensemble refers to ensembling predictions of five models including ResNet-101 (14.44mm), ResNet-101_noseg (14.57mm), ResNet-50_noseg (14.89mm), HRNet_Resnet50_shape_ensemble (14.68mm), SRN_multi_size_ensemble (15.20mm). Among them, the first four models are based on adaptive weighting regression (AWR) network with different backbones.

Table 5: Impact of different network architectures, in *NLE* experiments. No color jittering is applied during training in these experiments. MJE (mm) metric is used. Please note that for this experiment while ResNet-50 and ResNet-152 backbones results are obtained with 10 different anchor poses while the rest use 5 different anchor poses in *NLE*' settings for Pose Proposal Integration (PPI).

Backbone	Extrapolation	Interpolation	Object	Shape
ResNet-50	34.63	5.63	23.22	**17.79**
ResNet-101	**32.56**	4.49	**18.68**	18.50
ResNet-152	37.56	4.24	20.11	18.58
ResNext-50	33.88	4.99	25.67	19.70
ResNext-101	38.09	**3.83**	21.65	20.93

Table 5 shows comparison of different residual based backbones. Deeper backbones can obtain lower errors on Interpolation however, the method obtains higher errors on Extrapolation criteria and ResNet-101 a medium depth seems to be a reasonable choice in most cases in *NLE* experiments. While errors on different evaluation criteria with ResNext based arthictectures tend to vary a lot, ResNet based backbones are more solid.

Components of V2V-PoseNet architecture include: Volumetric Basic Block, Volumetric Residual Block, and Volumetric Downsampling and Upsampling Block. *NTIS* uses the same individual blocks as in V2V-PoseNet [8] but with a wider architecture. *NTIS*' experiment, see Table 6 shows that quadrupling the number of kernels in individual blocks provides the best results.

Table 6: Impact of widening the architecture used in V2V-PoseNet [8] in *NTIS* experiments. The number of kernels in each block in V2V-PoseNet architecture is quadrupled (wider).

Architecture V2V-PoseNet [8]	Extrapolation MJE (mm)
Original	38.33
Wider	**36.36**

3 Impact of Ensembling Techniques

In this section, we provide the experiments to show the importance of ensembling techniques. These techniques include ensembling in data pre-processing, methodological ensembles and ensembles as post-processing.

NLE' experiments on methodological and post-processing ensembling techniques. *NLE* adopts an approach based on LCR-Net++[11] where poses in the training set are clustered to obtain anchor poses and during inference, the test samples are first classified to these anchors and the final hand pose estimation is regressed from the anchor poses. Table 7 shows the impact of using different number of anchor poses. Shape extrapolation axis is heavily affected with the number anchor poses. While the number of obtained anchor poses from the training set increases from 1 to 50, the shape extrapolation error decreases from 21.08mm to 16.55mm. On the other hand, the number of anchor poses does not seem to have an observable impact on the other criteria, however; this can be because of the size of Task 3 test set and also because of the low hand pose variances in Task 3.

Table 7: Impact of number of anchor poses, in *NLE* experiments, obtained with k-means clustering for Pose Proposal Integration (PPI). No color jittering is applied during training in these experiments. ResNet-101 backbone architecture and MJE (mm) metric is used.

#Anchor poses	Extrapolation	Interpolation	Object	Shape
1	37.68	**3.99**	28.69	21.08
5	**32.56**	4.49	18.68	18.50
10	37.57	4.35	19.38	18.33
20	34.67	4.38	21.10	16.94
50	35.64	4.86	**17.84**	**16.55**

NLE's experiments later show the impact of learning and inferencing both 2D and 3D pose, and the impact of pose proposal integration [11] (PPI) compared to non-maximum suppression approach to obtain the poses. Learning to estimate 2D pose of a hand significantly impacts the extrapolation capability especially in Object axis. We believe this is because the objects occlude the hands and 2D information can be better obtained and help to guide estimation of the 3D hand

Table 8: Importance of pose proposal integration [11] (PPI) compared to non-max suppression (NMS), and of joint 2D-3D regression in *NLE* experiments (ResNet-50 backbone and 5 anchor poses are used). MJE (mm) metric is used.

2D-3D Estimation	Post.	Extrapolation	Interpolation	Object	Shape
3D only	NMS	38.59	8.48	37.31	18.78
2D+3D	NMS	38.08	7.60	28.45	18.73
2D+3D	PPI	**34.63**	**5.63**	**23.22**	**17.79**

poses. Later the pose proposal with 5 anchor poses brings a significant boost for extrapolation capabilities of the method.

Table 9: Importance of rotation data augmentation in *NLE* experiments, conducted with a ResNet-101 backbone architecture and 5 anchor poses. MJE (mm) metric is used.

#Test Rot.	Extrapolation	Interpolation	Object	Shape
1	29.55	4.85	18.09	17.35
4	**28.83**	4.63	**18.06**	16.77
12	29.19	**4.06**	18.39	**15.79**

NLE adopts another ensembling technique in the post-processing stage where test images are rotated by uniformly covering the space and the predictions obtained from each rotated test sample is ensembled. Experiments of *NLE* show that rotation as a post-processing ensemble technique helps significantly on shape extrapolation as well as interpolation axis and has minor impacts on other extrapolation criteria. Table 9 shows the impact of different number of rotation ensembles.

Strawberryfg ensembling as data pre-processing and orientation refinement per limb. Strawberryfg makes use of different input types obtained from the depth input image and their combinations to use them in their approach. Different input types include 3D joints projection, multi-layer depth and voxel representations and a list of input types and their combinations adopted to train different models are listed in Table 10. The impact of each mentioned model is reported in Table 11. The model used with different combination of different input types obtained from the depth images has no significant impact on evaluation criteria. We believe that this is because each different input type has different characteristics for the model to learn from and it's hard for the model to adapt to each type. Maybe a kind of adaptive weighting technique as adopted by some other approaches participated in the challenge can help in this case. However, as ensembling results of different models is proven to be helpful with all the approaches adopted the technique seems to be helpful in this case as well. 'Combined' model as depicted in Table 11 obtains the best results for all evaluation criteria. *Strawberryfg*' experiment report

to have 10.6% on articulation, 10% on interpolation, 8.4% on viewpoint, 7.2% on extrapolation, 6.2% on shape criteria improvements with ensembling of 4 models.

Table 12 using *Strawberryfg* shows the impact of patch orientation refinement networks adopted for each limb of a hand to show the impact. Orientation refinement brings a significant impact with $1mm$ lower error on all evaluation criteria.

Table 10: Input data types for four different models used in *Strawberryfg* experiments.

Model Id	Input Type			
	Depth Image	3D Points	Projection Multi-layer Depth	Depth Voxel
1	✓	✗	✗	✗
2	✓	✓	✓	✗
3	✓	✓	✗	✓
4	✓	✓	✓	✓

Table 11: MJE (mm) obtained in *Strawberryfg* experiments by using different models trained with different input types, see Table 10. 'Combined' model refers to ensembling predictions from all 4 models.

Model Id	Extrapolation	Viewpoint	Articulation	Shape	Interpolation
1	20.99	14.70	8.42	14.85	9.35
2	21.39	15.34	8.25	15.21	9.17
3	21.02	16.12	8.52	15.30	9.61
4	21.19	15.78	8.36	15.23	9.32
Combined	**19.63**	**14.16**	**7.50**	**14.21**	**8.42**

Table 12: Impact of local patch refinement and volume rendering supervision adopted by *Strawberryfg* . Model 4 with 4 different inputs are used in this evaluation, see Table 10.

Model Id - Type	Extrapolation	Viewpoint	Articulation	Shape	Interpolation
4 - w/o refinement & volume rendering	22.56	16.77	9.20	15.83	10.15
4 - w/ refinement & volume rendering	**21.19**	**15.78**	**8.36**	**15.23**	**9.32**

A2J ensembling in post-processing. At inference stage, *A2J* applies rotation and scale augmentations. More specifically, *A2J* rotates the test samples with $-90°/45°/90°$, and scales with factor 1/1.25/1.5. Then these predictions are averaged. Several backbone models are trained, including ResNet-50/101/152,

SE-ResNet-50/101, DenseNet-169/201, EfficientNet-B5/B6/B7. Input image sizes are $256 \times 256/288 \times 288/320 \times 320/384 \times 384$. The best single model is ResNet-152 with input size 384×384, it achieves $14.74mm$ on the extrapolation axis. Finally, these predictions are ensembled with weights to obtain a final error of $13.74mm$ on the extrapolation axis.

NTIS ensembling in post-processing with confident joint locations, Truncated SVDs and temporal smoothing. NTIS adopts a post-processing technique for refinement of hand poses where several inverse transformations of predicted joint positions are applied; in detail, *NTIS* uses truncated singular value decomposition transformations (Truncated SVDs; 9 for Task 1 and 5 for Task 2) with number of components $n \in 10, 15, 20, 25, 30, 35, 40, 45, 50$ obtained from the training ground-truth hand pose labels and prepares nine refined pose candidates. These candidates are combined together as final estimation that is collected as weighted linear combination of pose candidates with weights $w \in 0.1, 0.1, 0.2, 0.2, 0.4, 0.8, 1.0, 1.8/4.7$. Table 13 shows the impact of ensembling confident joint predictions and refinement stage with Truncated SVDs.

Table 13: Impact of refinement with Truncated SVDs in *NTIS* experiments on Task 1. Improvement is $\tilde{1}\%$. $N = 100$ most confident joint locations are ensembled for this experiment. Results reported in MJE (mm) metric.

SVD refinement	Extrapolation
w/	**15.81**
w/o	15.98

Since Task 2 is based on sequences and test samples are provided in order, *NTIS* applies temporal smoothing on the predictions from each frame and provides experimental results in Table 14 with different context sizes for smoothing. While temporal smoothing helps to decrease the extrapolation error, large context sizes do not impact much on the error.

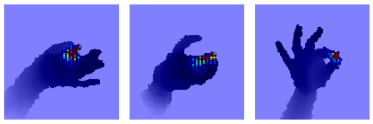
Table 14: Impact of temporal smoothing and the context size (k) for smoothing in *NTIS* experiments on Task 2 using exact same V2V-PoseNet [8] architecture.

Smoothing Context Size (k)	Extrapolation MJE (mm)
0	39.76
3	38.32
5	38.31
7	38.33

AWR methodological ensembling with AWR operation. Fig. 5 shows the impact of learnable adaptive weighting regression (AWR) approach on the probability

maps of the target joints. When the target joint is visible and easy to distinguish, the weight distribution of AWR tends to focus more on pixels around it as standard detection-based methods do, which helps to make full use of local evidence. When depth values around the target joint are missing, the weight distribution spreads out to capture information of adjacent joint. Later, Table 15 shows the impact of the AWR operation on two other datasets, NYU [17] and HANDS'17 [21].

(a) w/o AWR



(b) w/ AWR

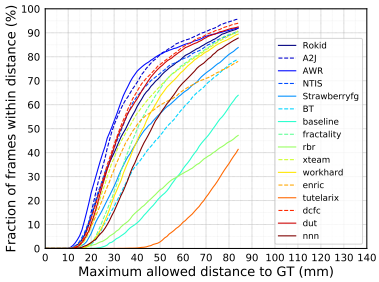
Fig. 5: Impact of AWR operation on the target joints' probability maps.

Table 15: *AWR* experiments for w/o adaptive weighting on NYU [17] and HANDS'17 [21] datasets. Results reported in MJE (mm) metric.

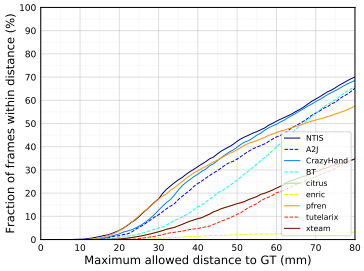
Dataset	w/o AWR	w/ AWR
NYU	7.87	**7.48**
HANDS'17	7.98	**7.48**

4 Frame Success Rates for All Participated Users in the Challenge

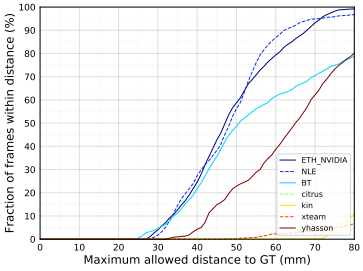
Fig. 6 shows the analysis of all participated users in the challenge's tasks. We analysed the selected methods (6 for Task 1, 4 for Task 2 and 3 for Task 3) based on their methodological variances and results in the main paper and this supplementary document, however the challenge have received 16 submissions for Task 1, 9 submissions for Task 2 and 7 submissions for Task 3 to be evaluated from different users in the submission system.



(a) Task 1 - Extrapolation



(b) Task 2 - Extrapolation

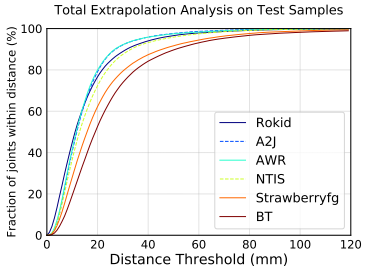


(c) Task3 - Extrapolation

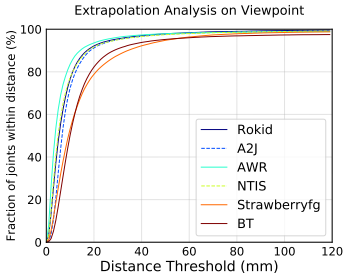
Fig. 6: All participated methods' accuracy analysis on different evaluation axis where each frames' error is estimated by considering the maximum error of all joints in that frame.

5 Joint Success Rates of the Analysed Approaches

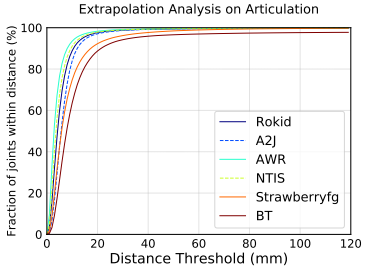
In this section, we provide success rate analyses for each of three tasks based on all joints in the test set. Please note the difference of the figures below compared to the success rate analysis based on frames as showed in Fig. 7, Fig. 8 and Fig. 9 in the main paper. Comparing the joint based analysis and the frame based analysis, we can note that all methods have different error variance for different joints and therefore the approaches tend to obtain higher accuracies based on considering each joint independently. Readers can find the related discussion in the main paper Section 5.



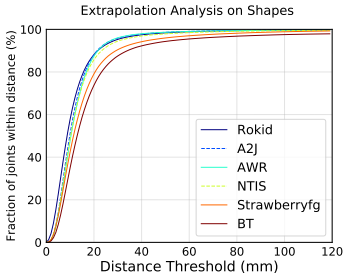
(a) Extrapolation



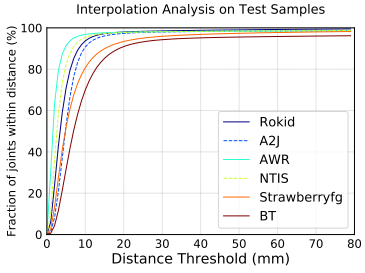
(b) Viewpoint



(c) Articulation

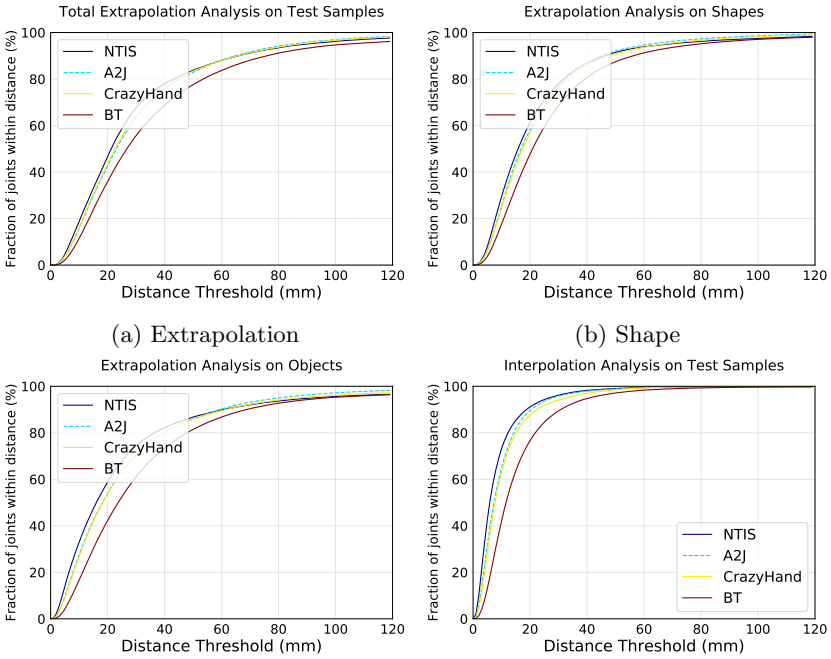


(d) Shape



(e) Interpolation

Fig. 7: Task 1 - Success rate analysis on different evaluation axis where each joints' error in the set is evaluated for measuring the accuracy.



(a) Extrapolation (b) Shape

(c) Object (d) Interpolation

Fig. 8: Task 2 - Success rate analysis on different evaluation axis where each joints' error in the set is evaluated for measuring the accuracy.

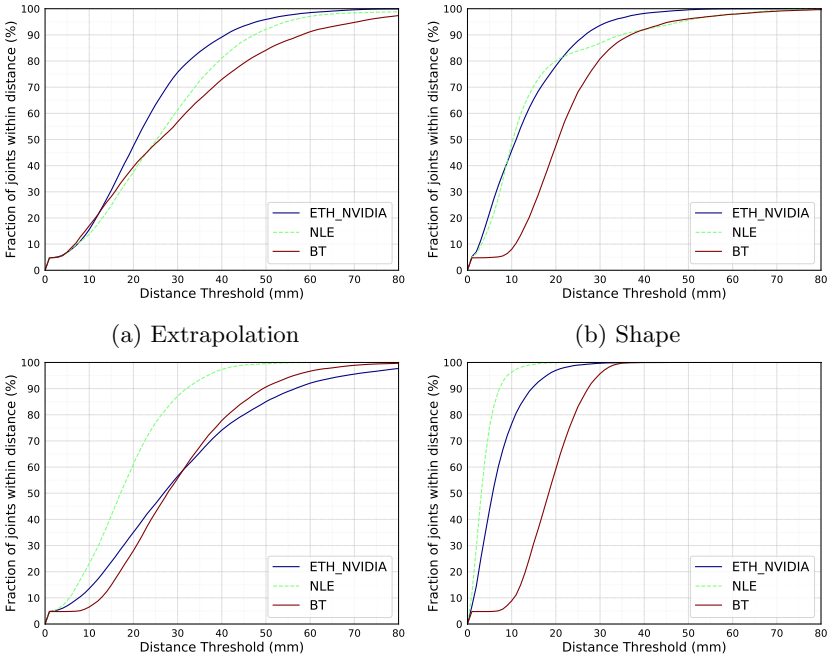
6 Evaluation Criteria Analysis

In this section, we provide more analysis and discussions on the generalisation power of the methods that could not be discussed in the paper due to space requirements.

Fig. 6 (f-i) of the main paper shows the average errors obtained on the different evaluation axis based on if the evaluation criterion has seen in the training set or not. Overall, while unseen shapes and viewpoints are harder to extrapolate in most of the cases, some unseen articulations are easier to extrapolate than some seen articulations which are hard to estimate the hand pose from.

Viewpoint extrapolation. As claimed in the main paper, the approaches tend to have larger errors on extreme angles like $[-180, -150]$ or $[150, 180]$ for azimuth viewpoint or similarly in elevation viewpoint and it's harder to extrapolate to unseen viewpoints in the training. While the approach by *Rokid* fills those unseen gaps with the generated synthetic data, other approaches mostly rely on their ensemble-based methodologies or their 3D properties. Please see the main paper, Section 5, for their properties.

Both Fig. 6 (g) for azimuth angles and (h) for elevation angles show the analysis for the viewpoints. Most of the extrapolation intervals (except the edges



(a) Extrapolation (b) Shape

(c) Object (d) Interpolation

Fig. 9: Task 3 - Success rate analysis on different evaluation axis where each joints' error in the set is evaluated for measuring the accuracy.

since both edges used to evaluate extrapolation) show distributions similar to a Gaussian which is expected since the mid-intervals are most far away viewpoints from a seen viewpoint from the training set. While both elevation and azimuth extrapolation errors are always higher than the interpolation error obtained with the corresponding methods, however the azimuth extrapolation tends to be varying more than the elevation extrapolation for some angles.

Articulation extrapolation. Fig. 6 (i) shows the average errors for 32 articulation clusters. 16 of those clusters have already seen in the training set while 16 have never seen and only available in the test set. While the samples that fall into some clusters, (*e.g.* 16, 18, 19, 20 and 31) tend to be harder to estimate most of the time, however some articulations without depending on seen (*e.g.* 1, 7, 8, 17) or unseen are hard to estimate as well because of the type of the articulation. Fig. 13 shows the example frames for the 32 clusters.

Shape extrapolation. Fig. 6 (f) shows average errors obtained for different shape types seen/unseen. All approaches have higher errors on unseen hand shapes (2, 3, 4, 5, 9) compared to errors obtained on shapes (1, 6, 7, 8, 10) seen in the training set.

Fig. 8 (c, f) of the main main paper shows the MJE analysis based on seen/unseen shapes (c) and objects (f). A list of objects that appear in the task

test set is given in Fig. 11. Although shape 'S5' refers to an unseen shape, all methods can extrapolate to this shape better than some other seen shapes in the training set. This can be explained with 'S5' being similar to some other shapes and it has the lowest number of frames (easy examples) compared to number of test frames from other shapes in the test set, see Fig. 3 (bottom right) in the main paper submission for the distributions of the training and test set. A similar aspect has been observed in [21] where different hand shape analysis has been provided, see Fig. 12. However, all methods tend to have higher errors on the frames from another unseen test shape 'S3' as expected.

Object extrapolation. Poses for hands with unseen objects, 'O3' power drill and 'O6' mug, are harder to extrapolate by most methods since their shapes are quite different than the other seen objects in the training set. Please note that seen 'O2' object has the lowest number of frames in the test set. Some example frames for the listed objects are showed in Fig. 10.

7 Dataset Details

(a) O1 - Cracker box (b) O2 - Picher base (c) O3 - Power drill

(d) O4 - Sugar box (e) O5 - Mustard bottle (f) O6 - Mug

Fig. 10: Example frames for the objects appear in Task 3, HO-3D [2] dataset.

Object Id	Object Name	Seen in the Training Set
O1	cracker box	✓
O2	pitcher base	✓
O3	power drill	✗
O4	sugar box	✓
O5	mustard bottle	✓
O6	mug	✗

(c) Object List

Fig. 11: Task 3 - Mean joint error analysis for the Shapes and Objects criteria on the corresponding test sets for the shapes and objects interpolation and extrapolation. Transparent and solid colors represent seen and unseen, respectively.

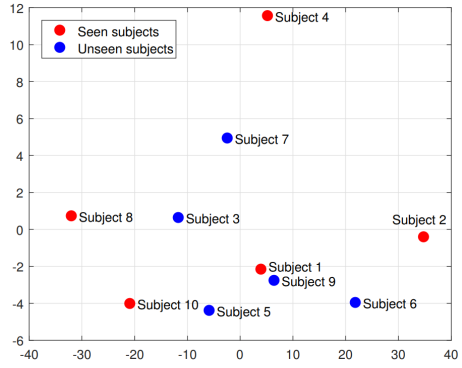


Fig. 12: Visualization of different hand shape distributions, appear in [21], by using the first two principal components of the hand shape parameters. Figure is taken from [21].

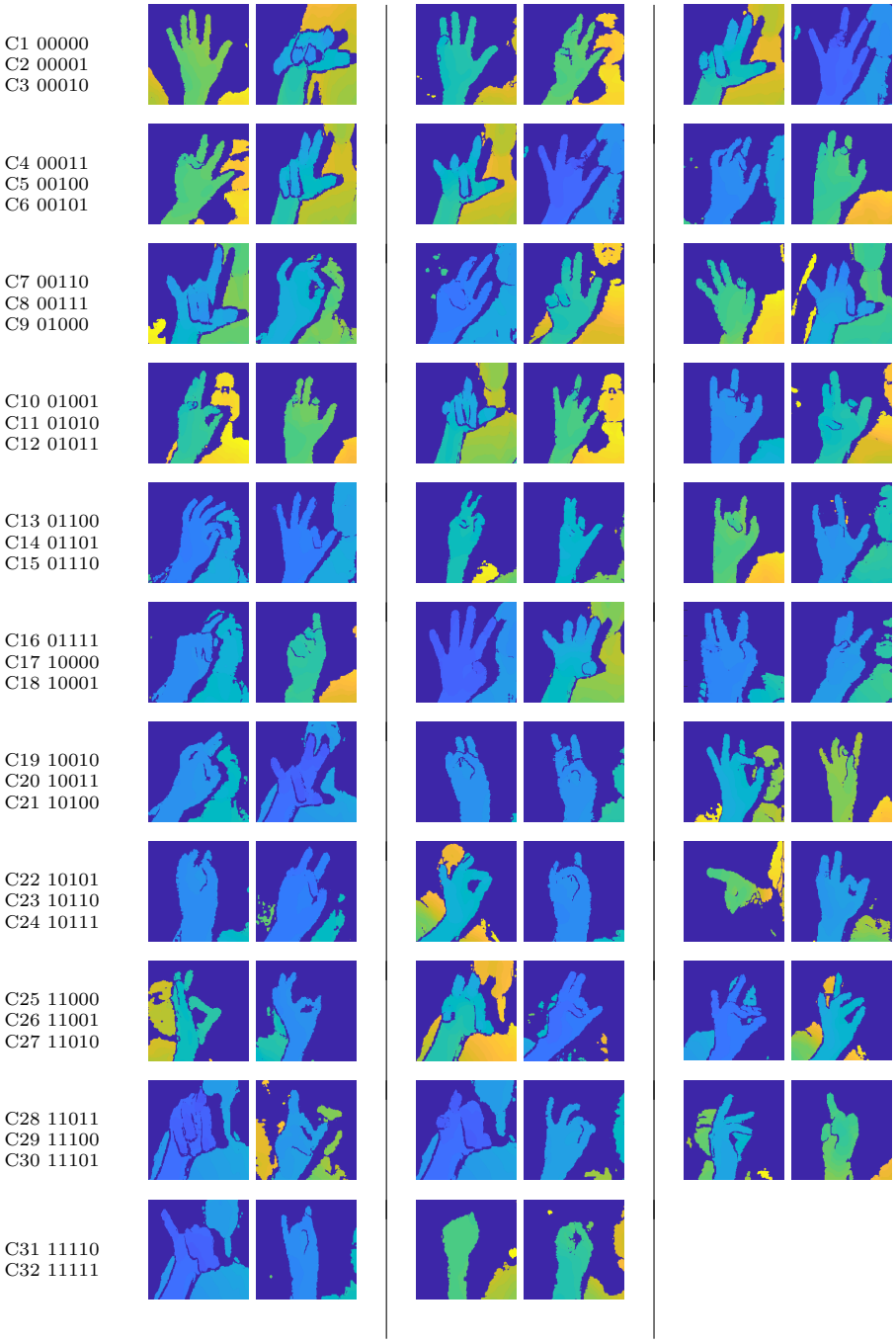


Fig. 13: Examples frames for 32 articulation clusters used in the evaluations. Each row shows cluster ids and their respective binary representations for two example images of three clusters. Each binary representation is constructed from thumb to pinky fingers with 0 representing closed and 1 representing open fingers.

References

1. 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 *CVPR*, 2018.
2. Shreyas Hampali, Markus Oberweger, Mahdi Rad, and Vincent Lepetit. HO-3D: A multi-user, multi-object dataset for joint 3D hand-object pose estimation. In *arXiv preprint arXiv:1907.01481v1*, 2019.
3. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
4. Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks. In *CVPR*, 2018.
5. Weiting Huang, Pengfei Ren, Jingyu Wang, Qi Qi, and Haifeng Sun. AWR: Adaptive weighting regression for 3D hand pose. In *AAAI*, 2020.
6. Shile Li and Dongheui Lee. Point-to-pose voting based hand pose estimation using residual permutation equivariant layer. In *CVPR*, 2019.
7. Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda Shapiro, and Hannaneh Hajishirzi. ESPNet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In *ECCV*, 2018.
8. Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2V-PoseNet: Voxel-to-voxel prediction network for accurate 3D hand and human pose estimation from a single depth map. In *CVPR*, 2018.
9. Pengfei Ren, Haifeng Sun, Qi Qi, Jingyu Wang, and Weiting Huang. SRN: Stacked regression network for real-time 3D hand pose estimation. In *BMVC*, 2019.
10. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.
11. Grégory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. LCR-Net++: Multi-person 2D and 3D pose detection in natural images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(5):1146–1161, 2019.
12. Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied Hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 36(6):245:1–245:17, 2017.
13. Adrian Spurr, Umar Iqbal, Pavlo Molchanov, Otmar Hilliges, and Jan Kautz. Weakly supervised 3D hand pose estimation via biomechanical constraints. *arXiv preprint arXiv:2003.09282*, 2020.
14. Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.
15. Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *ECCV*, 2018.
16. Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019.
17. Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *TOG*, 2014.
18. Fu Xiong, Boshen Zhang, Yang Xiao, Zhiguo Cao, Taidong Yu, Joey Tianyi Zhou, and Junsong Yuan. A2J: Anchor-to-joint regression network for 3D articulated pose estimation from a single depth image. In *ICCV*, 2019.
19. Linlin Yang, Shile Li, Dongheui Lee, and Angela Yao. Aligning latent spaces for 3D hand pose estimation. In *ICCV*, 2019.
20. Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. FoldingNet: Point cloud auto-encoder via deep grid deformation. In *CVPR*, 2018.
21. S. Yuan, Q. Ye, G. Garcia-Hernando, and T-K. Kim. The 2017 hands in the million challenge on 3d hand pose estimation. *arXiv preprint arXiv:1707.02237*, 2017.

22. Zhaohui Zhang, Shipeng Xie, Mingxiu Chen, and Haichao Zhu. HandAugment: A simple data augmentation method for depth-based 3D hand pose estimation. *arXiv preprint arXiv:2001.00702*, 2020.