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

Anil Armagan[1], Guillermo Garcia-Hernando[1,2], Seungryul Baek[1,20], Shreyas Hampali[3], Mahdi Rad[3], Zhaohui Zhang[4], Shipeng Xie[4], MingXiu Chen[4], Boshen Zhang[5], Fu Xiong[6], Yang Xiao[5], Zhiguo Cao[5], Junsong Yuan[7], Pengfei Ren[8], Weiting Huang[8], Haifeng Sun[8], Marek Hrúz[9], Jakub Kanis[9], Zdeněk Krňoul[9], Qingfu Wan[10], Shile Li[11], Linlin Yang[12], Dongheui Lee[11], Angela Yao[13], Weiguo Zhou[14], Sijia Mei[14], Yunhui Liu[15], Adrian Spurr[16], Umar Iqbal[17], Pavlo Molchanov[17], Philippe Weinzaepfel[18], Romain Brégier[18], Grégory Rogez[18], Vincent Lepetit[3,19], and Tae-Kyun Kim[1,21]

Abstract. We study how well different types of approaches generalise in the task of 3D hand pose estimation under single hand scenarios and hand-object interaction. We show that the accuracy of state-of-the-art methods can drop, and that they fail mostly on poses absent from the training set. Unfortunately, since the space of hand poses is highly dimensional, it is inherently not feasible to cover the whole space densely, despite recent efforts in collecting large-scale training datasets. This sampling problem is even more severe when hands are interacting with objects and/or inputs are RGB rather than depth images, as RGB images also vary with lighting conditions and colors. To address these issues, we designed a public challenge (HANDS'19) to evaluate the abilities of current 3D hand pose estimators (HPEs) to interpolate and extrapolate the poses of a training set. More exactly, HANDS'19 is designed (a) to evaluate the influence of both depth and color modalities on 3D hand pose estimation, under the presence or absence of objects; (b) to assess the generalisation abilities *w.r.t.* four main axes: shapes, articulations, viewpoints, and objects; (c) to explore the use of a synthetic hand models to fill the gaps of current datasets. Through the challenge, the overall accuracy has dramatically improved over the baseline, especially on extrapolation tasks, from 27mm to 13mm mean joint error. Our analyses highlight the impacts of: Data pre-processing, ensemble approaches, the use of a parametric 3D hand model (MANO), and different HPE methods/backbones.

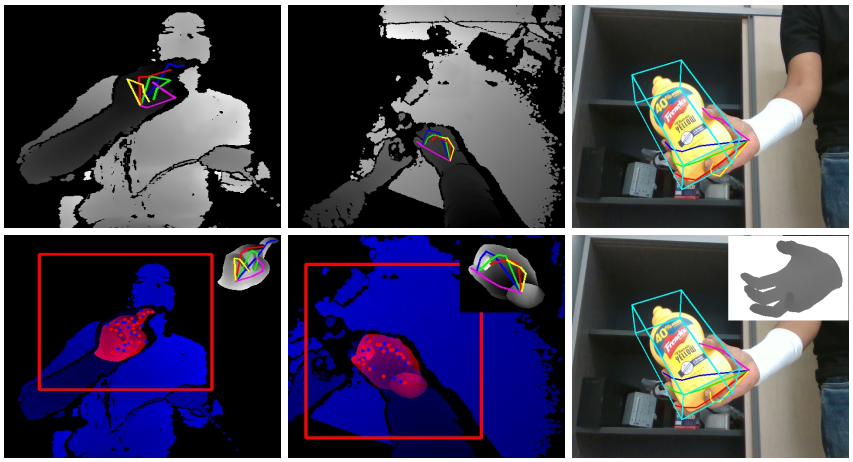
[1]Imperial College London, [2]Niantic, Inc., [3]Graz Uni. of Technology, [4]Rokid Corp. Ltd., [5]HUST, [6]Megvii Research Nanjing, [7]SUNY Buffalo, [8]BUPT,[9]Uni. of West Bohemia, [10]Fudan Uni., [11]TUM, [12]Uni. of Bonn, [13]NUS, [14]Harbin Inst. of Technology, [15]CUHK, [16]ETH Zurich, [17]NVIDIA Research, [18]NAVER LABS Europe, [19]ENPC ParisTech, [20]UNIST, [21]KAIST.

Challenge webpage: https://sites.google.com/view/hands2019/challenge

1 Introduction

3D hand pose estimation is crucial to many applications including natural user-interaction in AR/VR, robotics, teleoperation, and healthcare. The recent successes primarily come from large-scale training sets [48], deep convolutional neural networks [11,25], and fast optimisation for model fitting [17,26]. State-of-the-art methods now deliver satisfactory performance for viewpoints seen at training time and single hand scenarios. However, as we will show, these methods substantially drop accuracy when applied to egocentric viewpoints for example, and in the presence of significant foreground occlusions. These cases are not well represented on the training sets of existing benchmarks [6,23,24]. The challenges become even more severe when we consider RGB images and hand-object interaction scenarios. These issues are well aligned with the observations from the former public challenge HANDS'17 [47]: The state-of-the-art methods dropped accuracy from frontal to egocentric views, and from open to closure hand postures. The average accuracy was also significantly lower under hand-object interaction [6].

Given the difficulty to interpolate and extrapolate poses from the training set, one may opt for creating even larger training sets. Unfortunately, an inherent challenge in 3D hand pose estimation is the very high dimensionality of the problem, as hand poses, hand shapes and camera viewpoints have a large number of degrees-of-freedom that can vary independently. This complexity increases even more when we consider the case of a hand manipulating an object. Despite the recent availability of large-scale datasets [48], and the development of complex calibrated multi-view camera systems to help the annotation or synthetic



(a) Task 1 (b) Task 2 (c) Task 3

Fig. 1. Frames from the three tasks of our challenge. For each task, we show the input depth or RGB image with the ground-truth hand skeleton (top) and a rendering of the fitted 3D hand model as well as a depth rendering of the model (bottom). The ground-truth and estimated joint locations are shown in blue and red respectively.

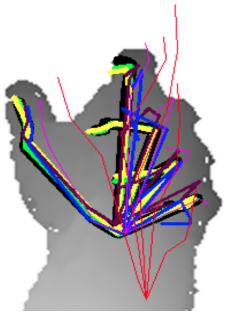


Fig. 2. Visualization of **ground-truth** hand pose and poses with varying level of MJEs, $< 5mm$, $< 10mm$, $< 20mm$, $< 30mm$, $< 40mm$, $< 60mm$. MJE (mm) of the visualized poses are 1.75, 6.88, 13.94, 15.32, 35.67, 52.15, respectively. Best viewed in color.

data [15,32,52], capturing a training set that covers completely the domain of the problem remains extremely challenging.

In this work, we therefore study in depth the ability of current methods to interpolate and extrapolate the training set, and how this ability can be improved. To evaluate this ability, we consider the three tasks depicted in Fig. 1, which vary the input (depth and RGB images) or the camera viewpoints, and introduce the possible manipulation of an object by the hand. We carefully designed training and testing sets in order to evaluate the generalisation performance to unseen viewpoints, articulations, and shapes of the submitted methods.

HANDS'19 fostered dramatic accuracy improvement compared to a provided baseline, which is a ResNet-50 [11]-based 3D joint regressor trained on our training set, from $\mathbf{27}\boldsymbol{mm}$ to $\mathbf{13}\boldsymbol{mm}$. Please see Fig. 2 for visualization of varying level of hand pose errors. This paper provides an in-depth analysis of the different factors that made this improvement possible.

2 HANDS 2019 Challenge Overview

The challenge consists of three different tasks, in which the goal is to predict the 3D locations of the hand joints given an image. For training, images, hand pose annotations, and a 3D parametric hand model [30] for synthesizing data are provided. For inference, only the images and bounding boxes of the hands are given to the participants. These tasks are defined as follows:

Task 1: Depth-Based 3D Hand Pose Estimation: This task builds on BigHand2.2M [48] dataset, as for the HANDS 2017 challenge [46]. No objects appear in this task. Hands appear in both third person and egocentric viewpoints.

Task 2: Depth-Based 3D Hand Pose Estimation while Interacting with Objects: This task builds on the F-PHAB dataset [6]. The subject manipulates objects with their hand, as captured from an egocentric viewpoint. Some object models are provided by [6].

Task 3: RGB-Based 3D Hand Pose Estimation while Interacting with Objects: This task builds on the HO-3D [9] dataset. The subject manipulates objects with their hand, as captured from a third person viewpoint. The objects are used from the YCB dataset [42]. The ground truth wrist position of the test images is also provided in this task.

The BigHand2.2M [48] and F-PHAB [6] datasets have been used by 116 and 123 unique institutions to date. HANDS'19 received 80 requests to access the datasets with the designed partitions, and 17, 10 and 9 participants have evaluated their methods on Task 1, Task 2 and Task 3, respectively.

3 Evaluation Criteria

We evaluate the generalisation capability of HPEs in terms of four "axes": Viewpoint, Articulation, Shape, and Object. For each axis, frames within a dataset are automatically annotated by using the ground-truth 3D joint locations and the object information to annotate each frame in each axis. The annotation distribution of the dataset for each axis are used are used to create a training and a test set. Using the frame annotations on each axis, the sets are sampled in a structured way to have the test frames that are similar to the frames in the training data (for interpolation) and also the test frames where axes' annotations are never seen in the training data (for extrapolation). More details on the dataset are given in Section 4. To measure the generalisation of HPEs, six evaluation criteria are further defined with the four main axes:

Viewpoint, **Articulation**, **Shape** and **Object** are respectively used for measuring the extrapolation performance of HPEs on the frames with articulation cluster, viewpoint angle, hand shape and object type (axis annotations) that are not present in the training set. **Extrapolation** is used to measure the performance on the frames with axis annotations that do not overlap/present in the training set. Lastly, **Interpolation** is defined to measure the performance on the frames with the axis annotations present in the training set.

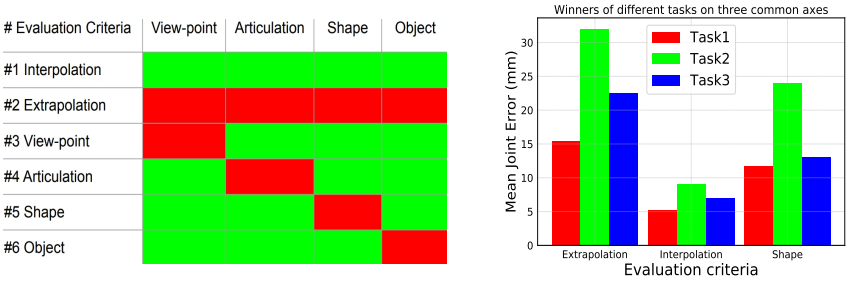


Fig. 3. Left: The six evaluation criteria used in the challenge. For each axis (Viewpoint, Articulation, Shape, Object), we indicate if hand poses in an evaluation criterion are also available (green) in the training set or not (red). Right: MJE comparison of the best methods for the Extrapolation, Interpolation and Shape criteria on each task.

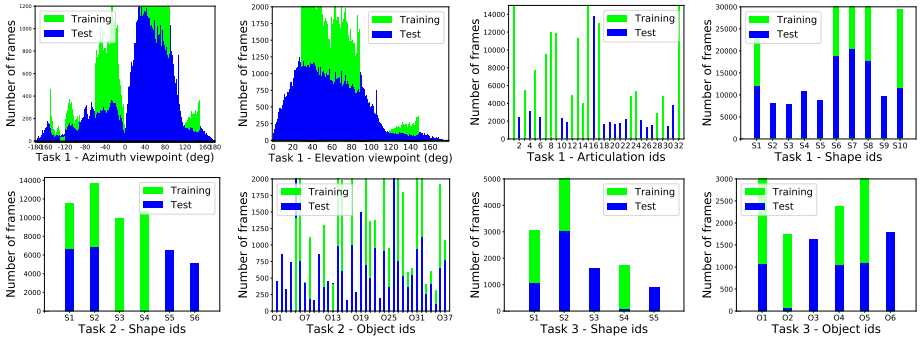


Fig. 4. Distributions of the training and test datasets for Task 1 (top), Task 2 (bottom left), and Task 3 (bottom right). The splits are used to evaluate the extrapolation power of the approaches and decided based on the viewpoints, the articulation clusters of the hand pose, the hand shape, and the type of the object present.

The challenge uses the mean joint error (MJE) [26] as the main evaluation metric. Results are ranked according to the **Extrapolation** criterion which measures the total extrapolation power of the approaches with MJE on all axes. We also consider success rates based on maximum allowed distance errors for each frame and each joint for further analysis.

Fig. 3 (left) summarises the six evaluation strategies, and Fig. 3 (right) shows the accuracies obtained by the best approaches, measured for the three evaluation criteria that could be evaluated for all three tasks. Articulation and viewpoint criteria are only considered for Task 1 since the joint angles are mostly fixed during object interaction and hence the Articulation criteria is not as meaningful as in Task 1 for the other tasks. The Viewpoint criteria is not meaningful for Task 2 which is for egocentric views since the task's dataset constrains the relative palm-camera angle to a small range. For Task 3, the data scarcity is not helping to sample enough diverse viewpoints. The extrapolation errors tend to be three times larger than the interpolation errors while the shape is a bottleneck among the other attributes. Lower errors on Task 3 compared to Task 2 are likely due to the fact that the ground truth wrist position is provided for Task 3.

4 Datasets

Given a task, the training set is the same and the test frames used to evaluate each criterion can be different or overlapped. The number of training frames are 175K, 45K and 10K for Task 1, 2 and 3 respectively. The sizes of the test sets for each evaluation criterion are shown in Table 1.

Fig. 4 shows the distributions of the training and test data for each task. The viewpoints are defined as elevation and azimuth angles of the hand *w.r.t.* the camera using the ground-truth joint annotations. The articulation of the

Table 1. Detailed analytics on the number of frames provided on the training and test sets for the different tasks.

		#Frames										
Dataset	Task id	Total	Ext.	Int.	Art.	View.	Sha.	Obj.	#Subjects	#Objects	#Actions	#Seq.
Test	1	125K	20%	16%	16%	32%	16%	✗	10	✗	✗	✗
	2	25K	14%	32%	✗	✗	37%	17%	4	37	71	292
	3	6.6K	24%	35%	✗	✗	14	27%	5	5	1	5
Training	1	175951							5	✗	✗	✗
	2	45713							4	26	45	539
	3	10505							3	4	1	12

hand is defined and obtained by clustering on the ground-truth joint angles in a fashion similar to [20], by using binary representations (open/closed) of each finger *e.g.* '00010' represents a hand articulation cluster with frames with the index finger closed and the rest of the fingers open, which ends up with $2^5 = 32$ clusters. Examples from the articulation clusters are provided in Appendix A.3. Note that the use of a low-dimensional embedding such as PCA or t-SNE is not adequate here to compare the two data distributions, because the dimensionality of the distributions is very high and a low-dimensional embedding would not be very representative. Fig. 4 further shows the splits in terms of subjects/shapes, where five seen subjects and five unseen subjects are present. Similarly, the data partition was done on objects. This way we can control the data to define the evaluation metrics.

Use of 3D Hand Models for HPEs. A series of methods [1,3,8,10,49] have been proposed in the literature to make use of 3D hand models for supervision of HPEs. Ge et al. [8] proposed to use Graph CNNs for mapping RGB images to infer the vertices of 3D meshes. Hasson et al. [10] jointly infers both hands and object meshes and investigated the effect of the 3D contact loss penalizing the penetration of object and hand surfaces. Others [1,3,49] attempted to make use of MANO [30], a parametric 3D hand model by learning to estimate low-dimensional PCA parameters of the model and using it together with differentiable model renderers for 3D supervision. All the previous works on the use of 3D models in learning frameworks have shown to help improving performance on the given task. Recently, [18] showed that fitting a 3D body model during the estimation process can be accelerated by using better initialization of the model parameters however, our goal is slightly different since we aim to explore the use of 3D models for better generalisation from the methods. Since the hand pose space is huge, we make use of a 3D hand model to fill the gaps in the training data distribution to help approaches to improve their extrapolation capabilities. In this study, we make use of the MANO [30] hand model by providing the model's parameters for each training image. We fit the 3D model for each image in an optimization-based framework which is described in more details below.

Gradient-based Optimization for Model Fitting. We fit the MANO [30] models' shape $\mathbf{s} = \{s_j\}_{j=1}^{10}$, camera pose $\mathbf{c} = \{c_j\}_{j=1}^{8}$, and articulation $\mathbf{a} =$

$\{a_j\}_{j=1}^{45}$ parameters to the i-th raw skeletons of selected articulations $\mathbf{z} = \{z_i\}_{i=1}^{K}$, by solving the following equation:

$$(\mathbf{s}^{i*}, \mathbf{c}^{i*}, \mathbf{a}^{i*}) = \arg\min_{(\mathbf{s},\mathbf{c},\mathbf{a})} O(\mathbf{s}, \mathbf{c}, \mathbf{a}, \mathbf{z}^i)), \forall i \in [1, K]\,, \tag{1}$$

where our proposed objective function $O(\mathbf{s}, \mathbf{c}, \mathbf{a}, \mathbf{z}^i)$ for the sample i is defined as follows:

$$O(\mathbf{s}, \mathbf{c}, \mathbf{a}, \mathbf{z}^i) = ||f^{reg}(V(\mathbf{s}, \mathbf{c}, \mathbf{a})) - \mathbf{z}^i||_2^2 + \sum_{j=1}^{10} \|\mathbf{s}_j\|_2^2 + R_{Lap}(V(\mathbf{s}, \mathbf{c}, \mathbf{a}))\,. \tag{2}$$

$V(\mathbf{s}, \mathbf{c}, \mathbf{a})$ denotes the 3D mesh as a function of the three parameters $\mathbf{s}, \mathbf{c}, \mathbf{a}$. Eq. (2) is composed of the following terms: *i*) the Euclidean distance between 3D skeleton ground-truths $\mathbf{z}^i$ and the current MANO mesh model's 3D skeleton values $f^{reg}(V(\mathbf{s}, \mathbf{c}, \mathbf{a}))$[1]; *ii*) A shape regularizer enforcing the shape parameters $\mathbf{s}$ to be close to their MANO model's mean values, normalized to 0 as in [30], to maximize the shape likelihood; and *iii*) A Laplacian regularizer $R_{Lap}(V(\mathbf{s}, \mathbf{c}, \mathbf{a}))$ to obtain the smooth mesh surfaces as in [16]. Eq. (1) is solved iteratively by using the gradients from Eq. (2) as follows:

$$(\mathbf{s}_{t+1}, \mathbf{c}_{t+1}, \mathbf{a}_{t+1}) = (\mathbf{s}_t, \mathbf{c}_t, \mathbf{a}_t) - \gamma \cdot \nabla O(\mathbf{s}_t, \mathbf{c}_t, \mathbf{a}_t, \mathbf{z}^i), \forall t \in [1, T]\,, \tag{3}$$

where $\gamma = 10^{-3}$ and $T = 3000$ are empirically set. This process is similar to the refinement step of [39,1], which refines estimated meshes by using the gradients from the loss. In Fig. 5, both the target and the fitted depth images during the process described by Eq. (3) are depicted. Minor errors of the fitting are not a problem for our purpose given that we will generate input and output pairs of the fitted model by exploiting fitted meshes' self-data generation capability while ignoring original depth and skeletons. Here the aim of fitting the hand model is to obtain a plausible and a complete articulation space. The model is fitted without optimizing over depth information from the model and the input depth image since we did not observe an improvement on the parameter estimation. Moreover, the optimization needs to be constrained to produce plausible hand shapes and noise and other inconsistencies may appear in the depth image.

5 Evaluated Methods

In this section, we present the gist of selected 14 methods among 36 participants (17 for Task 1, 10 for Task 2, 9 for Task 3) to further analyze their results in Section 6. Methods are categorized based on their main components and properties. See Tables 2,3 and 4 for a glance of the properties of the methods in HANDS'19.

2D and 3D supervision for HPEs. Approaches that embed and process 3D data obtain high accuracies but less efficient [47] in terms of their complexity compared to 2D-based approaches. 3D-based methods use 3D convolutional layers for point-clouds input similar to *NTIS* which uses an efficient voxel-based

[1] f^{reg} geometrically regresses the skeleton from the mesh vertex coordinates. It is provided with the MANO model and the weights are fixed during the process.

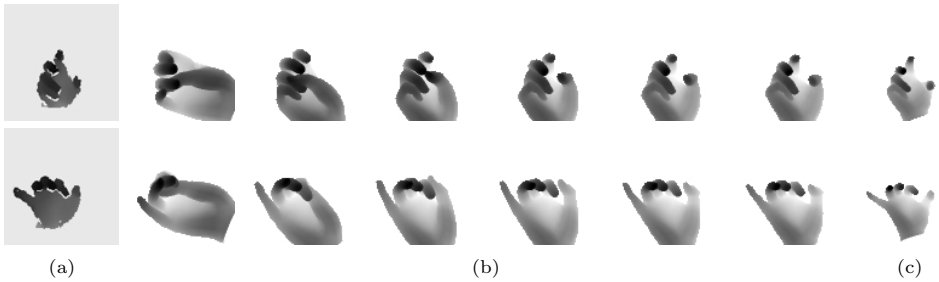


Fig. 5. Depth renderings of the hand model for different iterations in gradient-based optimization fitting. Target image (joints) (a), optimization iterations 0, 100, 300, 400, 600, 700 (b), final fitted hand pose at iteration 3000 (c).

representation V2V-PoseNet [22] with a deeper architecture and weighted sub-voxel predictions on quarter of each voxel representations for robustness. Some other approaches adopts 3D as a way of supervision similar to *Strawberryfg* [40] which employs a render-and-compare stage to enforce voxel-wise supervision for model training and adopts a 3D skeleton volume renderer to re-parameterize an initial pose estimate obtained similar to [36]. *BT* uses a permutation invariant feature extraction layer [19] to extract point-cloud features and uses a two branch framework for point-to-pose voting and point-to-latent voting. 3D supervision is employed by point-cloud reconstruction from a latent embedding in Task 1 whereas 3D hand model parameters are estimated and used in a differentiable model renderer for 3D supervision for the other tasks.

2D CNN-based approaches has been a standard way for learning regression models as used by *Rokid* [50] where they adopt a two stage regression models. The first regression model is used to predict an initial pose and the second model built on top of the first model. *A2J* [43] uses a 2D supervised method based on 2D offset and depth estimations with anchor points. Anchor points are densely set on the input image to behave as local regressors for the joints and able to capture global-local spatial context information. *AWR* [13] adopts a learnable and adaptive weighting operation that is used to aggregate spatial information of different regions in dense representations with 2D convolutional CNNs. The weighting operation adds direct supervision on joint coordinates and draw consensus between the training and inference as well as enhancing the model's accuracy and generalisation ability by adaptively aggregating spatial information from related regions. *CrazyHand* uses a hierarchically structured regression network by following the joints' distribution on the hand morphology. *ETH_NVIDIA* [34] adopts the latent 2.5D heatmap regression [14]; additionally an MLP is adopted for denoising the absolute root depth. Absolute 3D pose in scale-normalized space is obtained with the pinhole camera equations. *NLE* [29] first performs a classification of the hand into a set of canonical hand poses (obtained by clustering on the poses in the training set), followed by a fine class-specific regression of the hand joints in 2D and 3D. *NLE* adopts the only approach proposing multiple hand poses in a single stage with a Region Proposal Network (RPN) [28] integration.

Table 2. Task 1 - Methods' Overview

Username	Description	Input	Pre-processing	Post-processing	Synthetic Data	Backbone	Loss	Optimizer
Rokid [50]	2D CNN joint regression	Depth 224 × 224	Initial pose est. to crop	✗	570K Synthetic + Mixed Synthetic	EfficientNet-b0 [37]	Wing [5]	Adamax
A2J [43]	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 [13]	2D CNN, dense direction & offset rep. Learnable adaptive weighting	Depth 256 × 256 segm. 128 × 128 pose est.	Bbox crop ESPNet-v2 [21] for binary segm. iter. refinement of CoM	Ensemble from 5 models	✗	ResNet-50&101 SRN [27] HRNet [35]	Smooth L1	Adam
NTIS [22]	3D CNN Deeper V2V-PoseNet [22] 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 [40]	Integral Pose Regression [36] 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 [19]	3D supervision with cloud reconst. Permutation invariant [19] Point-to-pose + point-to-latent voting.	Point cloud 512 3D vectors	View correction [19]	✗	✗	ResPel [19] for feat. extract FoldingNet [45] for reconstruction	L2 Chamfer and EMD KL constraint	Adam

Table 3. Task 2 - Methods' Overview

Username	Description	Input	Pre-processing	Post-processing	Synthetic Data	Backbone	Loss	Optimizer
NTIS [22]	3D CNN Deeper V2V-PoseNet [22] 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 [43]	2D CNN offset and depth regression with anchor points and weighting	Depth 256 × 256	Bbox crop	Ensemble predictions from 3 training epochs	✗	SEResNet-101 [12]	Smooth L1	Adam
CrazyHand	2D CNN tree-like branch structure regression with hand morphology	Depth 128 × 128	Iterative CoM	✗	✗	ResNet-50	L2	-
BT [19]	Differentiable Mano [30] layer Permutation invariant [19] Point-to-pose+ point-to-latent voting	Point cloud 512 3D points	View correction [19]	✗	32K synthetic + random objects from HO-3D [9]	ResPel [19]	L2 pose L2 Mano vertex KL constraint	Adam

Table 4. Task 3 - Methods' Overview

Username	Description	Input	Pre-processing	Post-processing	Synthetic Data	Backbone	Loss	Optimizer
ETH_NVIDIA [34]	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 [29]	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 [28] for localization loss	SGD
BT [44]	Multi-modal input with latent space alignment [44] Differentiable Mano [30] layer	RGB 256 × 256 Point cloud - 356	Bbox cropping	✗	100K synthetic + random objects from HO-3D [9]	EncoderCloud: ResPEL [19] EncoderRGB: ResNet-18 DecoderMano: 6 fully-connected	L2 pose, L2 Mano vert. Chamfer, Normal and Edge length for mesh KL constraint	Adam

Detection, regression and combined HPEs. Detection methods are based on hand key-points and producing a probability density maps for each joint. *NTIS* uses a 3D CNN [22] to estimate per-voxel likelihood of each joint. Regression-based methods estimate the joint locations by learning a direct mapping from the input image to hand joint locations or the joint angles of a hand model [33,51]. *Rokid* uses joint regression models within two stages to estimate an initial hand pose for hand cropping and estimates the final pose from the cleaned hand image. *A2J* adopts regression framework by regressing offsets from anchors to final joint location. *BT*'s point-wise features are used in a voting scheme which behaves as a regressor to estimate the pose.

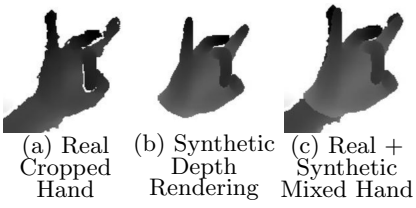
Some approaches take advantage of both detection-based and regression-based methods. Similarly, *AWR*, *Strawberryfg* estimates probability maps to estimate joint locations with a differentiable *soft-argmax* operation [36]. A hierarchical approach proposed by *CrazyHand* regresses the joint locations from joint probability maps. *ETH_NVIDIA* estimates 2D joint locations from estimated probability maps and regresses relative depth distance of the hand joints *w.r.t.* a root joint. *NLE* first localizes the hands and classifies them to anchor poses and the final pose is regressed from the anchors.

Method-wise ensembles. *A2J* uses densely set anchor points in a voting stage which helps to predict location of the joints in an ensemble way for better generalisation leveraging the uncertainty in reference point detection. In a similar essence, *AWR* adaptively aggregates the predictions from different regions and *Strawberryfg* adopts local patch refinement [41] where refinement models are adopted to refine bone orientations. *BT* uses the permutation equivariant features extracted from the point-cloud in a point-to-pose voting scheme where the votes are ensembled to estimate the pose. *NLE* ensembles anchor poses to estimate the final pose.

Ensembles in post-processing. Rather than a single pose estimator, an ensemble approach was adopted by multiple entries by randomly replicating the methods and fusing the predictions in the post-prediction stage, *e.g.* *A2J*, *AWR*, *NTIS*, *NLE* and *Strawberryfg*.

A2J ensembles predictions from ten different backbone architectures in Task 1 like *AWR* (5 backbones) and augments test images to ensemble the predictions with different scales and rotations as similar to rotation augmentation adopted by *NLE*. *NTIS* uses predictions obtained from the same model at 6 different training epochs. A similar ensembling is also adopted by *A2J* in Task 2. *NTIS* adopts a different strategy where N most confident sub-voxel predictions are ensembled to further use them in a refinement stage with Truncated SVDs together with temporal smoothing (Task 2). *NLE* takes advantage of ensembles from multiple pose proposals [29]. *Strawberryfg* employs a different strategy and ensembles the predictions from models that are trained with various input modalities.

Real + synthetic data usage. The methods *Rokid* in Task 1 and *BT* in Tasks 2 and 3 make use of the provided MANO [30] model parameters to synthesize more training samples. *Rokid* leverages the synthesized images and combines them the real images—see Fig. 6—to train their initial pose regression network which effectively boosts accuracies—see Table 8. However, the amount of synthetic data created is limited to 570K for *Rokid* and 32K in Task 2, 100K in Task 3 for *BT*. Considering the continuous high-dimensional hand pose space with or without objects, if we



(a) Real Cropped Hand (b) Synthetic Depth Rendering (c) Real + Synthetic Mixed Hand

Fig. 6. Visualization of synthetic depth images by *Rokid* [50]: (a) input depth image, (b) rendered depth image using 3D hand model, (c) the mixed by using the pixels with the closest depth values from real and synthetic images.

sub-sample uniformly and at minimum, for instance, 10^2(azimuth/elevation angles)$\times 2^5$(articulation)$\times 10^1$(shape)$\times 10^1$(object) = 320K, the number is already very large, causing a huge compromise issue for memory and training GPU hours. Random sampling was applied without a prior on the data distribution or smart sampling techniques [4,2]. *BT* generates synthetic images with objects and hands similar to [23] by randomly placing the objects from [9] to nearby hand locations without taking into account the hand and object interaction. The rest of the methods use the provided real training data only.

Multi-modal inputs for HPEs. *BT* adopts [44] in Task 3 to align latent spaces from depth and RGB input modalities and to embed the inherit depth information in depth images during learning. *Strawberryfg* makes use of multi-inputs where each is obtained from different representations of the depth image, *e.g.* point-cloud, 3D point projection [7], multi-layer depth map [31], depth voxel [22].

Dominating HPE backbones. ResNet [11] architectures with residual connections have been a popular backbone choice among many HPEs *e.g. A2J*, *AWR*, *Strawberryfg*, *CrazyHand*, *ETH_NVIDIA*, *NLE* or implicitly by *BT* within the ResPEL [19] architecture. *Rokid* adopts EfficientNet-b0 [37] as a backbone which uniformly scales the architecture's depth, width, and resolution.

6 Results and Discussion

We share our insights and analysis of the results obtained by the participants' approaches: 6 in Task 1, 4 in Task 2, and 3 in Task 3. Our analyses highlight the impacts of data pre-processing, the use of an ensemble approach, the use of MANO model, different HPE methods, and backbones and post-processing strategies for the pose refinement.

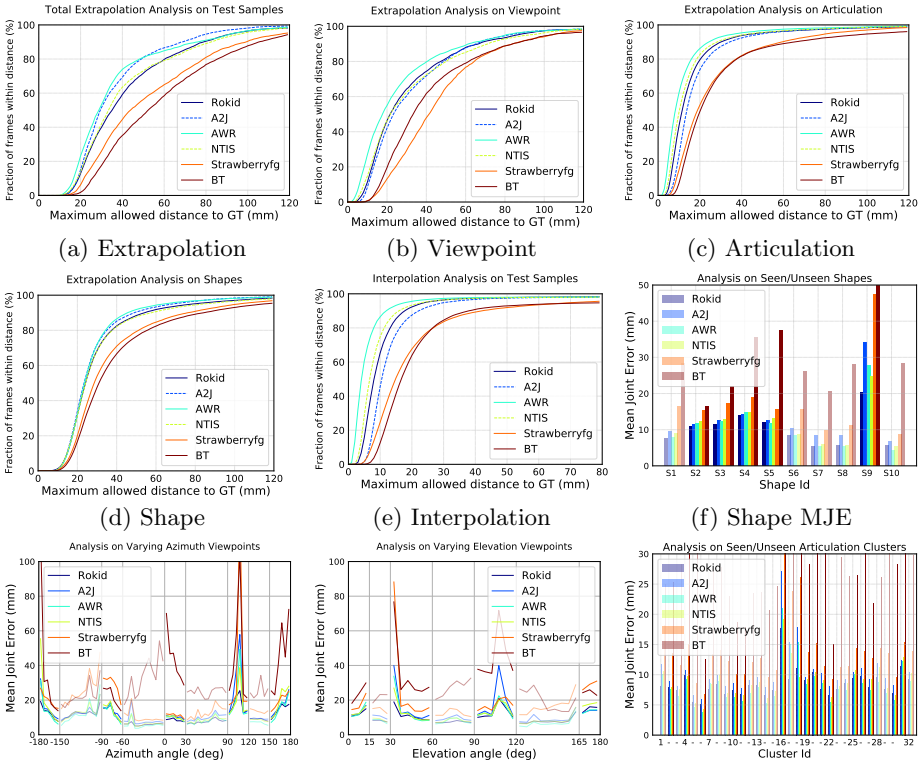
Analysis of Submitted Methods for Task 1. We consider the main properties of the selected methods and the evaluation criteria for comparisons. Table 5 provides the errors for the MJE metric and Fig. 7 show that high success rates are easier to achieve in absence of an object for low distance d thresholds. 2D-based approaches such as *Rokid*, with the advantage of additional data synthesizing, or *A2J*, with cleverly designed local regressors, can be considered to be best when the MJE score is evaluated for the Extrapolation criterion. *AWR* performs comparable to the other 2D-based approaches by obtaining the lowest MJE errors on the

Table 5. Task 1 - MJE (mm) and ranking of the methods on five evaluation criteria. Best results on each evaluation criteria are highlighted.

Username	Extrapolation	Interpolation	Shape	Articulation	Viewpoint
Rokid	**13.66** (1)	4.10 (2)	**10.27** (1)	4.74 (3)	**7.44** (1)
A2J	13.74 (2)	6.33 (6)	11.23 (4)	6.05 (6)	8.78 (6)
AWR	13.76 (4)	**3.93** (1)	11.75 (5)	**3.65** (1)	7.50 (2)
NTIS	15.57 (7)	4.54 (3)	12.05 (6)	4.21 (2)	8.47 (4)
Strawberryfg	19.63 (12)	8.42 (10)	14.21 (10)	7.50 (9)	14.16 (12)
BT	23.62 (14)	18.78 (16)	21.84 (16)	16.73 (16)	19.48 (14)

Table 6. Task 2 - MJE (mm) and ranking of the methods on four evaluation criteria.

Username	Extrapolation	Interpolation	Object	Shape
NTIS	**33.48 (1)**	**17.42 (1)**	29.07 (2)	23.62 (2)
A2J	33.66 (2)	17.45 (2)	**27.76 (1)**	**23.39** (1)
CrazyHand	38.33 (4)	19.71 (4)	32.60 (4)	26.26 (4)
BT	47.18 (5)	24.95 (6)	38.76 (5)	32.36 (5)



(a) Extrapolation (b) Viewpoint (c) Articulation

(d) Shape (e) Interpolation (f) Shape MJE

(g) Azimuth Viewpoint (h) Elevation Viewpoint (i) Articulation MJE

Fig. 7. Task 1 - Success rate analysis (a-e) and MJE analysis on extrapolation and interpolation using shapes (f), viewpoints (g, h) and articulations (i). Solid colors depict samples of extrapolation and transparent colors depict interpolation samples.

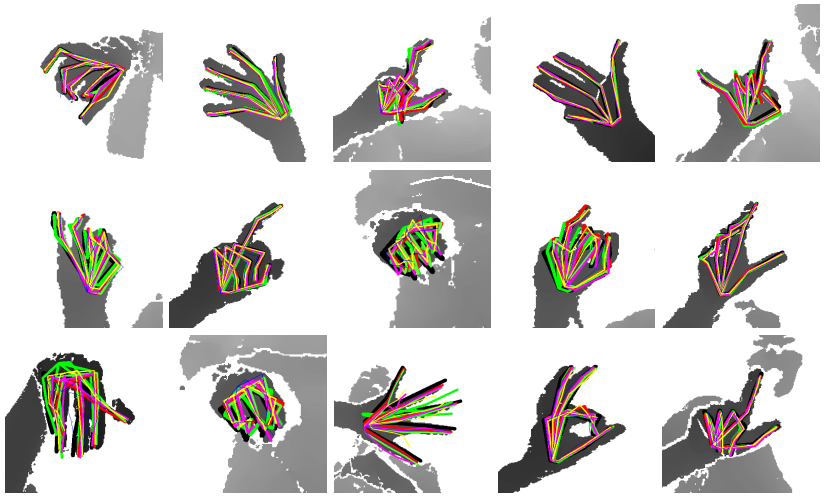


Fig. 8. Task 1 - Visualization of the **ground-truth** annotations and estimations of ***Rokid***, ***A2J***, ***AWR***, ***NTIS***, ***Strawberryfg***, ***BT***.

Interpolation and Articulation criteria. *AWR* performs best for the distances less than $50mm$ on Extrapolation as well as showing better generalisation to unseen Viewpoints and Articulations, while excelling to interpolate well. A similar trend is observed with the 3D-voxel-based approach *NTIS*. However, the other 3D supervised methods, *Strawberryfg* and *BT* show lower generalisation capability compared to other approaches while performing reasonably well on the Articulation, Shape, and Interpolation criteria but not being able to show a similar performance for the Extrapolation and Viewpoint criteria.

Analysis of Submitted Methods for Task 2. We selected four submitted methods to compare on Task 2, where a hand interacts with an object in an egocentric viewpoint. Success rates illustrated in Fig. 9 highlight the difficulty of extrapolation. All methods struggle to show good performance on estimating frames with joint errors less than $15mm$. On the other hand, all methods can estimate 20% to 30% of the joints correctly with less than $15mm$ error for the other criteria in this task.

NTIS (a voxel-based) and *A2J* (weighted local regressors with anchor points) perform similarly when MJEs for all joints are considered. However, *NTIS* obtains higher success rates on the frame-based evaluation for all evaluation criteria with low distance error thresholds (d), see Fig. 9. Its performance is relatively much higher when Extrapolation is considered, especially for the frames with unseen objects, see Fig. 9. This can be explained by having a better embedding of the occluded hand structure with the voxels in the existence of seen/unseen objects. *NTIS* interpolates well under low distance thresholds.

Note that the first three methods, *NTIS*, *A2J*, and *CrazyHand* perform very similar for high error thresholds *e.g.* $d > 30mm$. *CrazyHand* uses a structured detection-regression-based HPE where a heatmap regression is employed for the joints from palm to tips in a sequential manner which is highly valuable for egocentric viewpoints, helps to obtain comparable results with *A2J* where the structure is implicitly refined by the local anchor regressors.

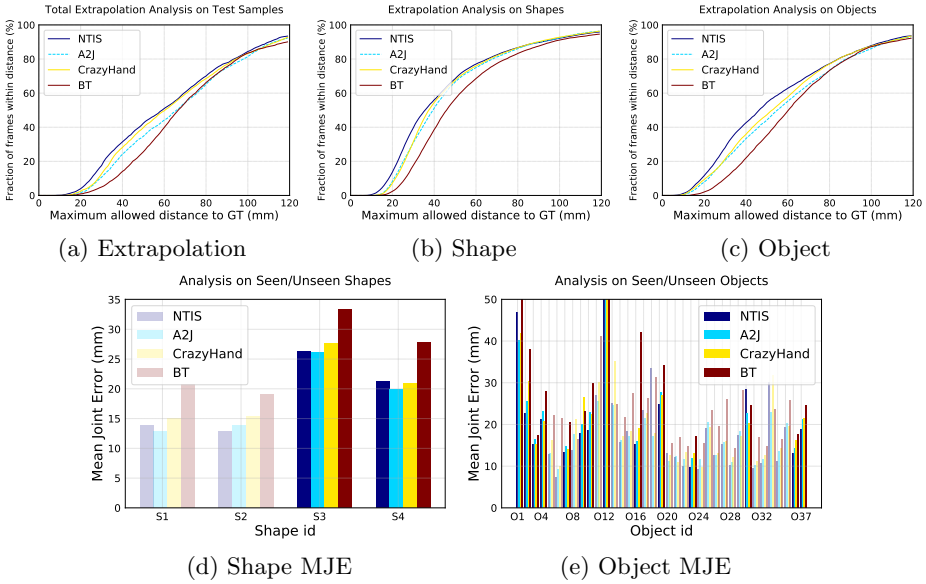
Analysis of Submitted Methods for Task 3. We selected 3 entries, with different key properties for this analysis. It is definitively harder for the participants to provide accurate poses compared to the previous tasks. None of the methods can estimate frames that have all joints estimated with less than $25mm$ error, see Fig. 11. The $25mm$ distance threshold shows the difficulty of estimating a hand pose accurately from RGB input modality even though the participants of this task were provided with the ground-truth wrist joint location.

Table 7. Task 3 - MJE (mm) and ranking of the methods on four evaluation criteria.

Username	Extrapolation	Interpolation	Object	Shape
ETH_NVIDIA	**24.74** (1)	6.70 (3)	27.36 (2)	**13.21** (1)
NLE	29.19 (2)	**4.06** (1)	**18.39** (1)	15.79 (3)
BT	31.51 (3)	19.15 (5)	30.59 (3)	23.47 (4)

Table 8. Impact of synthetic data reported by *Rokid* [50] with learning from different ratios of synthetic data and the Task 1 training set. $100\% = 570K$.

Synthetic Data %	-	10%	30%	70%	100%
Extrapolation MJE (mm)	30.11	16.70	16.11	15.81	15.73



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

(d) Shape MJE (e) Object MJE

Fig. 9. Task 2 - Success rate analysis (a,b,c) and interpolation (seen, transparent) and extrapolation (unseen, solid) errors for subject (d) and object (e).

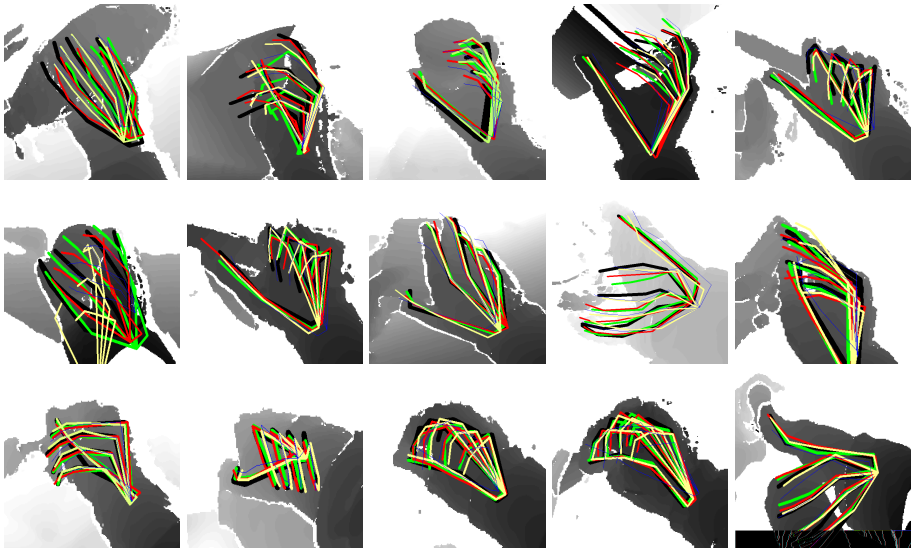
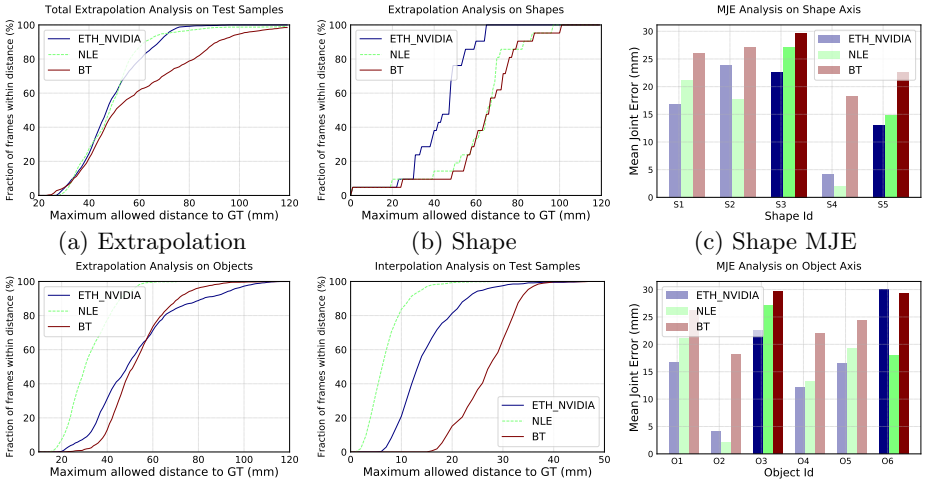


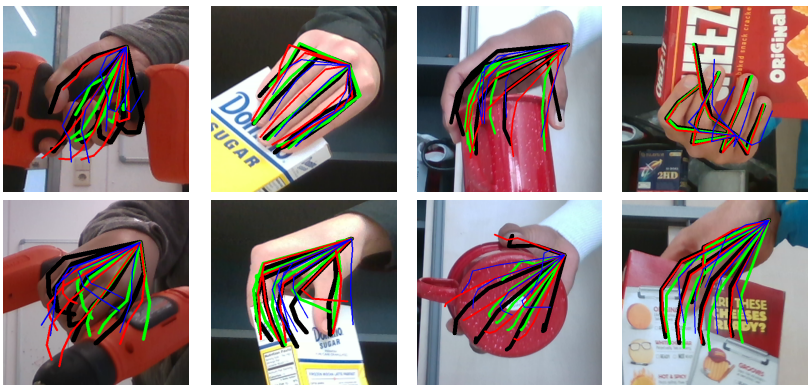
Fig. 10. Task 2 - Visualization of the **ground-truth** annotations and estimations of ***NTIS***, ***A2J***, ***CrazyHand***, ***BT***.



(a) Extrapolation (b) Shape (c) Shape MJE

(d) Object (e) Interpolation (f) Object MJE

Fig. 11. Task 3 - Success rate analysis on the evaluation criteria (a,b,d,e) and MJE error analysis on the seen/unseen subjects (c) and objects (f). For (c) and (f), solid and transparent colors are used to depict extrapolation and interpolation.



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

Fig. 12. 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.

The task is based on hand-object interaction in RGB modality. Therefore, the problem raises the importance of multi-modal data and learn from different modalities. Only *BT* uses the MANO parameters provided by the organizers to synthesize 100K images and adds random objects near the hand. This approach supports the claim on the importance of multi-modality and filling the real data gaps with synthetic data with its close performance to the two higher ranked methods in MJE.

The generalisation performance of *BT* in Task 3 compared to the team's approaches with similar gist in Tasks 1 and 2 supports the importance of multi-model learning and synthetic data augmentation. The close performance of the method to generalise to unseen objects compared to *ETH_NVIDIA* and to generalise to unseen shapes compared to *NLE* also supports the argument with the data augmentation. The approach is still outperformed in MJE for this task although it performs close to the other methods.

NLE's approach shows the impact of learning to estimate 2D joints+3D joints ($28.45mm$) compared to learning 3D joints alone ($37.31mm$) on the Object as well as improvements for the Interpolation. Object performance is further improved to $23.22mm$ with PPI integration. Further insights put by *NLE*'s own experiments on the number rotation augmentations (n) in post-processing helps to better extrapolate for unseen shapes ($17.35mm$, $16.77mm$, $15.79mm$ where $n = 1, 4, 12$, respectively).

Analysis on the Usage of Synthetic Images. The best performing method of Task 1 (*Rokid*) in MJE uses the 3D hand model parameters to create 570K synthetic images by either perturbing (first stage) the model parameters or not (second stage). Synthetic data usage significantly helps in training the initial model (see Fig. 6). Table 8 shows the impact of different proportions of the 570K synthetic data usage to train the model together with the real training images. Using synthetic data can boost such a simple 3D joint regressor's performance from MJE of $30.11mm$ to $15.73mm$, a $\sim 50\%$ improvement. Moreover, *Rokid*'s experiments with a regression model trained for 10 epochs shows the impact of the mixed depth inputs, Fig. 6, to lower the total extrapolation error ($26.16mm$) compared to the use of raw depth renderings ($30.13mm$) or the renderings averaged ($31.92mm$) with the real input images. *BT* uses synthetic images in a very small amount of $32K$ and $100K$ in Tasks 2 and 3 since 3D reconstruction is difficult to train at a larger scale. However, favorable impact the data can be observed by comparing performances in Tasks 1 and 2.

Analysis on Evaluation Criteria We discuss the generalisation power of the methods based on our evaluation criteria below. Fig. 7 (f-i) 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. HPEs 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.

Both Fig. 7 (g) for azimuth angles and (h) for elevation angles show the analysis for the viewpoints. Most of the extrapolation intervals (except the edges 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. 7 (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. 20 shows the example frames for the 32 clusters.
Shape extrapolation. Fig. 7 (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. 11 (c, f) show 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 Table 21. 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. 4 (bottom right) for the distributions of the training and test set. A similar aspect has been observed in [46] where different hand shape analysis has been provided, see Fig. 19. 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. 18.

7 Ablation Studies by the Participants

Here we present the experiments and their results conducted by the participants for the challenge. Section 7.1 presents experimental results conducted by the participated approaches based on different backbone architectures and similarly, Section 7.2 shows experimental evaluation on the ensembling techniques in pre-processing, post-processing and methodological level.

7.1 Experiments with Different Backbone Architectures

While Residual Network (ResNet) [11] 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 Tables 9 and 10.

Table 9. 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 9 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 9) including ResNet-101 (center2+segmented), ResNet-101 (center2+original), ResNet-50 (center2+original), SRN_multi_size_ensemble and HRNet_Resnet50-_shape_ensemble. Since ESPNetv2 [21] 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 approaches using ensembled networks, SRN [27] 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 10. 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 10 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 [22] but with a wider architecture. *NTIS*' experiment, see Table 11 shows that quadrupling the number of kernels in individual blocks provides the best results.

Table 11. Impact of widening the architecture used in V2V-PoseNet [22] in *NTIS* experiments. The number of kernels in each block in V2V-PoseNet architecture is quadrupled (wider).

Architecture V2V-PoseNet [22]	Extrapolation MJE (mm)
Original	38.33
Wider	**36.36**

7.2 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++[29] 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 12 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 12. 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**

Table 13. Importance of pose proposal integration [29] (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**

NLE's experiments later show the impact of learning and inferencing both 2D and 3D pose, and the impact of pose proposal integration [29] (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 poses. Later the pose proposal with 5 anchor poses brings a significant boost for extrapolation capabilities of the method.

Table 14. 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 14 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 15. The impact of each mentioned model is reported in Table 16. 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 16 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 17 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 15. 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	✓	✓	✓	✓

A2J uses 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,

Table 16. MJE (mm) obtained in *Strawberryfg* experiments by using different models trained with different input types, see Table 15. '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 17. 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 15.

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**

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 18 shows the impact of ensembling confident joint predictions and refinement stage with Truncated SVDs.

Table 18. 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 19 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.

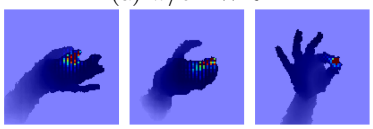
AWR methodological ensembling with AWR operation. Fig. 13 shows the impact of learnable adaptive weighting regression (AWR) approach on the probability

Table 19. Impact of temporal smoothing and the context size (k) for smoothing in *NTIS* experiments on Task 2 using exact same V2V-PoseNet [22] architecture.

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

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 20 shows the impact of the AWR operation on two other datasets, NYU [38] and HANDS'17 [46].

(a) w/o AWR



(b) w/ AWR

Fig. 13. Impact of AWR operation on the target joints' probability maps.

Table 20. *AWR* experiments for w/o adaptive weighting on NYU [38] and HANDS'17 [46] datasets. Results reported in MJE (mm) metric.

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

8 Conclusion

We carefully designed structured training and test sets for 3D HPEs and organized a challenge for the hand pose community to show state-of-the-art methods still tend to fail to extrapolate on large pose spaces. Our analyses highlight the impacts of using ensembles, the use of synthetic images, different type of HPEs *e.g.* 2D, 3D or local-estimators and post-processing. Ensemble techniques, both methodologically in 2D and 3D HPEs and in post-processing, help many approaches to boost their performance on extrapolation. The submitted HPEs were proven to be successful while interpolating in all the tasks, but their extrapolation capabilities vary significantly. Scenarios such as hands interacting with objects present the biggest challenges to extrapolate by most of the evaluated methods both in depth and RGB modalities.

Given the limited extrapolation capabilities of the methods, usage of synthetic data is appealing. Only a few methods actually were making use of synthetic data to improve extrapolation. $570K$ synthetic images used by the winner of Task 1 is still a very small number compared to how large, potentially infinite, it could be. We believe that investigating these possibilities, jointly with data sub-sampling strategies and real-synthetic domain adaptation is a promising and interesting line of work. The question of what would be the outcome if we sample 'dense enough' in the continuous and infinite pose space and how 'dense enough' is defined when we are limited by hardware and time is significant to answer.

Acknowledgements. This work is partially supported by Huawei Technologies Co. Ltd. and Samsung Electronics. S. Baek was supported by IITP funds from MSIT of Korea (No. 2020-0-01336 AIGS of UNIST, No. 2020-0-00537 Development of 5G based low latency device - edge cloud interaction technology).

References

1. Baek, S., Kim, K.I., Kim, T.K.: Pushing the envelope for RGB-based dense 3D hand pose estimation via neural rendering. In: CVPR (2019) 6, 7
2. Bhattarai, B., Baek, S., Bodur, R., Kim, T.K.: Sampling strategies for GAN synthetic data. In: ICASSP (2020) 11
3. Boukhayma, A., de Bem, R., Torr, P.H.: 3D hand shape and pose from images in the wild. In: CVPR (2019) 6
4. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: AutoAugment: Learning augmentation strategies from data. In: CVPR (2019) 11
5. Feng, Z.H., Kittler, J., Awais, M., Huber, P., Wu, X.J.: Wing loss for robust facial landmark localisation with convolutional neural networks. In: CVPR (2018) 9
6. Garcia-Hernando, G., Yuan, S., Baek, S., Kim, T.K.: First-person hand action benchmark with RGB-D videos and 3D hand pose annotations. In: CVPR (2018) 2, 3, 4
7. Ge, L., Liang, H., Yuan, J., Thalmann, D.: Robust 3D hand pose estimation in single depth images: from single-view CNN to multi-view CNNs. In: CVPR (2016) 11
8. Ge, L., Ren, Z., Li, Y., Xue, Z., Wang, Y., Cai, J., Yuan, J.: 3D hand shape and pose estimation from a single RGB image. In: CVPR (2019) 6
9. Hampali, S., Oberweger, M., Rad, M., Lepetit, V.: HO-3D: A multi-user, multi-object dataset for joint 3D hand-object pose estimation. In: arXiv preprint arXiv:1907.01481v1 (2019) 4, 9, 11, 31
10. Hasson, Y., Varol, G., Tzionas, D., Kalevatykh, I., Black, M.J., Laptev, I., Schmid, C.: 3D hand shape and pose estimation from a single RGB image. In: CVPR (2019) 6
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) 2, 3, 11, 17
12. Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E.: Squeeze-and-excitation networks. In: CVPR (2018) 9
13. Huang, W., Ren, P., Wang, J., Qi, Q., Sun, H.: AWR: Adaptive weighting regression for 3D hand pose estimation. In: AAAI (2020) 8, 9
14. Iqbal, U., Molchanov, P., Breuel, T., Gall, J., Kautz, J.: Hand pose estimation via latent 2.5D heatmap regression. In: ECCV (2018) 8
15. Joo, H., Liu, H., Tan, L., Gui, L., Nabbe, B., Matthews, I., Kanade, T., Nobuhara, S., Sheikh, Y.: Panoptic studio: a massively multiview system for social motion capture. In: ICCV (2015) 3
16. Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. In: ECCV (2018) 7
17. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: ICNN (1995) 2
18. Kolotouros, N., Pavlakos, G., Black, M.J., Daniilidis, K.: Learning to reconstruct 3D human pose and shape via model-fitting in the loop. In: ICCV (2019) 6
19. Li, S., Lee, D.: Point-to-pose voting based hand pose estimation using residual permutation equivariant layer. In: CVPR (2019) 8, 9, 11
20. Lin, J., Wu, Y., Huang, T.S.: Modeling the constraints of human hand motion. In: HUMO (2000) 6
21. Mehta, S., Rastegari, M., Caspi, A., Shapiro, L., Hajishirzi, H.: ESPNet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In: ECCV (2018) 9, 18

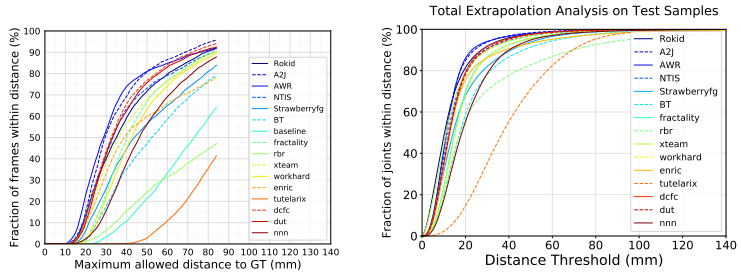
22. Moon, G., Chang, J.Y., Lee, K.M.: V2V-PoseNet: Voxel-to-voxel prediction network for accurate 3D hand and human pose estimation from a single depth map. In: CVPR (2018) 8, 9, 11, 19, 23
23. Mueller, F., Bernard, F., Sotnychenko, O., Mehta, D., Sridhar, S., Casas, D., Theobalt, C.: GANerated hands for real-time 3D hand tracking from monocular RGB. In: CVPR (2018) 2, 11
24. Mueller, F., Mehta, D., Sotnychenko, O., Sridhar, S., Casas, D., Theobalt, C.: Real-time hand tracking under occlusion from an egocentric RGB-D sensor. In: ICCV (2017) 2
25. Oberweger, M., Lepetit, V.: DeepPrior++: Improving fast and accurate 3D hand pose estimation. In: ICCV Workshop on HANDS (2017) 2
26. Oikonomidis, I., Kyriazis, N., Argyros., A.A.: Efficient model-based 3D tracking of hand articulations using kinect. In: BMVC (2011) 2, 5
27. Ren, P., Sun, H., Qi, Q., Wang, J., Huang, W.: SRN: Stacked regression network for real-time 3D hand pose estimation. In: BMVC (2019) 9, 18
28. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS (2015) 8, 9
29. Rogez, G., Weinzaepfel, P., Schmid, C.: 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) 8, 9, 10, 20
30. Romero, J., Tzionas, D., Black, M.J.: Embodied Hands: Modeling and capturing hands and bodies together. ACM Transactions on Graphics (Proc. SIGGRAPH Asia) **36**(6), 245:1–245:17 (2017) 3, 6, 7, 9, 10
31. Shin, D., Ren, Z., Sudderth, E.B., Fowlkes, C.C.: Multi-layer depth and epipolar feature transformers for 3D scene reconstruction. In: CVPR Workshops (2019) 11
32. Simon, T., Joo, H., Matthews, I., Sheikh, Y.: Hand keypoint detection in single images using multiview bootstrapping. In: CVPR (2017) 3
33. Sinha, A., Choi, C., Ramani, K.: DeepHand: Robust hand pose estimation by completing a matrix imputed with deep features. In: CVPR (2016) 9
34. Spurr, A., Iqbal, U., Molchanov, P., Hilliges, O., Kautz, J.: Weakly supervised 3D hand pose estimation via biomechanical constraints. arXiv preprint arXiv:2003.09282 (2020) 8, 9
35. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: CVPR (2019) 9
36. Sun, X., Xiao, B., Wei, F., Liang, S., Wei, Y.: Integral human pose regression. In: ECCV (2018) 8, 9, 10
37. Tan, M., Le, Q.V.: EfficientNet: Rethinking model scaling for convolutional neural networks. In: ICML (2019) 9, 11
38. Tompson, J., Stein, M., Lecun, Y., Perlin, K.: Real-time continuous pose recovery of human hands using convolutional networks. TOG (2014) 23
39. Tung, H.Y.F., Tung, H.W., Yumer, E., Fragkiadaki, K.: Self-supervised learning of motion capture. In: NIPS (2017) 7
40. Wan, Q.: SenoritaHand: Analytical 3D skeleton renderer and patch-based refinement for HANDS19 challenge Task 1 - Depth-based 3D hand pose estimation (Dec 2019), https://github.com/strawberryfg/Senorita-HANDS19-Pose 8, 9
41. Wan, Q., Qiu, W., Yuille, A.L.: Patch-based 3D human pose refinement. In: CVPRW (2019) 10
42. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: A convolutional neural network for 6D object pose estimation in cluttered scenes. In: RSS (2018) 4

43. Xiong, F., Zhang, B., Xiao, Y., Cao, Z., Yu, T., Zhou, J.T., Yuan, J.: A2J: Anchor-to-joint regression network for 3D articulated pose estimation from a single depth image. In: ICCV (2019) 8, 9
44. Yang, L., Li, S., Lee, D., Yao, A.: Aligning latent spaces for 3D hand pose estimation. In: ICCV (2019) 9, 11
45. Yang, Y., Feng, C., Shen, Y., Tian, D.: FoldingNet: Point cloud auto-encoder via deep grid deformation. In: CVPR (2018) 9
46. Yuan, S., Ye, Q., Garcia-Hernando, G., Kim, T.K.: The 2017 hands in the million challenge on 3d hand pose estimation. arXiv preprint arXiv:1707.02237 (2017) 3, 17, 23, 31
47. Yuan, S., Garcia-Hernando, G., Stenger, B., Moon, G., Chang, J.Y., Lee, K.M., Molchanov, P., Kautz, J., Honari, S., Ge, L., Yuan, J., Chen, X., Wang, G., Yang, F., Akiyama, K., Wu, Y., Wan, Q., Madadi, M., Escalera, S., Li, S., Lee, D., Oikonomidis, I., Argyros, A., Kim, T.K.: Depth-based 3D hand pose estimation: From current achievements to future goals. In: CVPR (2018) 2, 7
48. Yuan, S., Ye, Q., Stenger, B., Jain, S., Kim, T.K.: BigHand 2.2M Benchmark: hand pose data set and state of the art analysis. In: CVPR (2017) 2, 3, 4
49. Zhang, X., Li, Q., Mo, H., Zhang, W., Zheng, W.: End-to-end hand mesh recovery from a monocular RGB image. In: ICCV (2019) 6
50. Zhang, Z., Xie, S., Chen, M., Zhu, H.: HandAugment: A simple data augmentation method for depth-based 3D hand pose estimation. arXiv preprint arXiv:2001.00702 (2020) 8, 9, 10, 13
51. Zhou, X., Wan, Q., Zhang, W., Xue, X., Wei, Y.: Model-based deep hand pose estimation. In: IJCAI (2016) 9
52. Zimmermann, C., Brox, T.: Learning to estimate 3D hand pose from single RGB images. In: ICCV (2017) 3

A Appendix

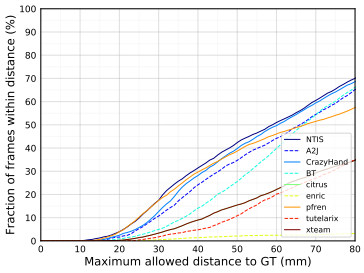
A.1 Frame Success Rates for All Participated Users in the Challenge

Figure 14 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. The challenge have received 16 submissions for Task 1, 9 submissions for Task 2 and 7 for Task 3 to be evaluated from different users.

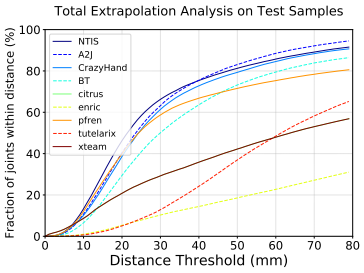


(a) Task 1 - Extrapolation frame

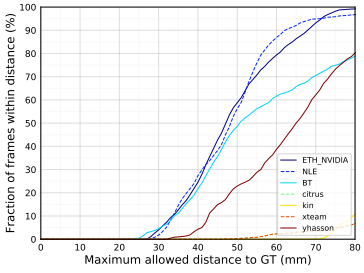
(b) Task 1 - Extrapolation joint



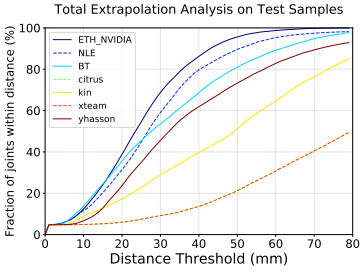
(c) Task 2 - Extrapolation frame



(d) Task 2 - Extrapolation joint



(e) Task3 - Extrapolation frame

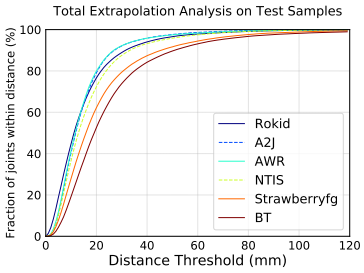


(f) Task3 - Extrapolation joint

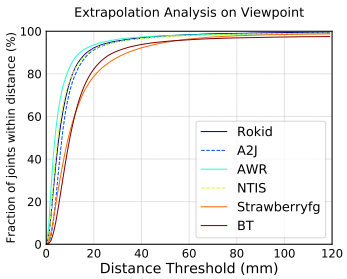
Fig. 14. All participated methods' total extrapolation accuracy analysis for each task. (a,c,e) represents the frame success rates where each frames' error is estimated by considering the maximum error of all joints in that frame. (b,d,f) shows the joint success rates.

A.2 Joint Success Rates of the Analysed Approaches

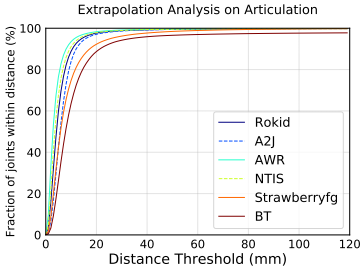
Success rate analyses for each of three tasks based on all joints in the test set are provided below. Please note the difference of the figures below compared to the success rate analysis based on frames as showed in Fig. 7, 9 and 11. Comparing the joint based analysis and the frame based analysis, we 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.



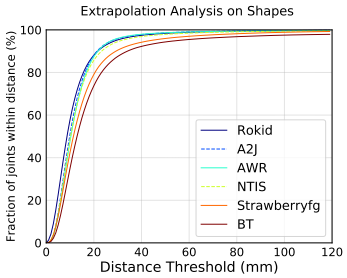
(a) Extrapolation



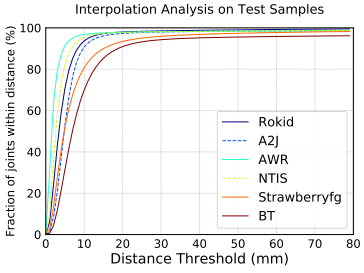
(b) Viewpoint



(c) Articulation

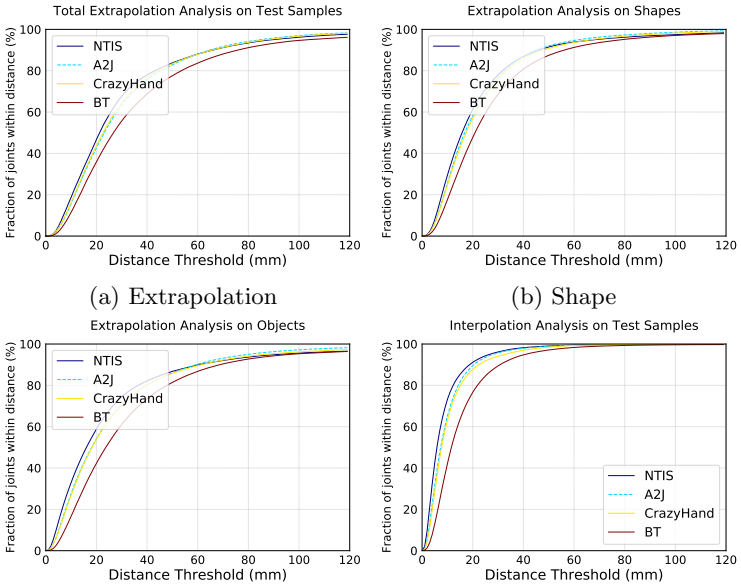


(d) Shape



(e) Interpolation

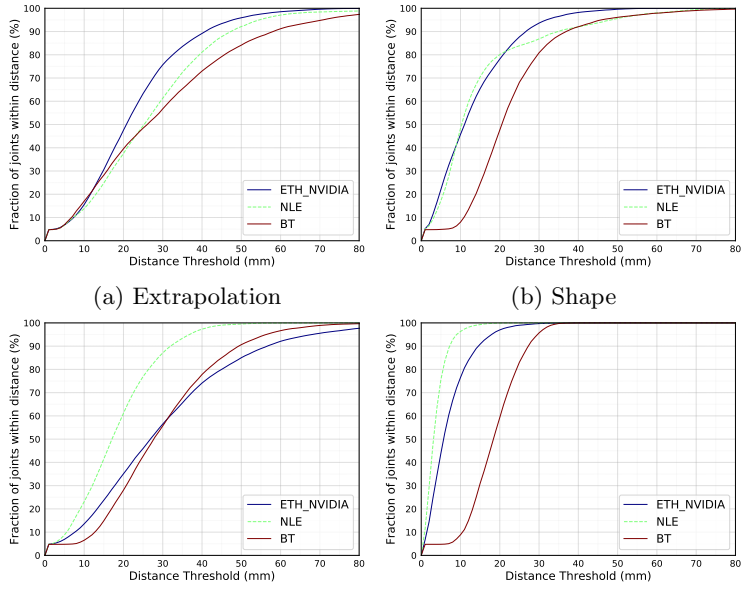
Fig. 15. Task 1 - Joint success rate analysis on different evaluation axis where each joints' error in the set is evaluated for measuring the accuracy.



(a) Extrapolation (b) Shape

(c) Object (d) Interpolation

Fig. 16. Task 2 - Joint success rate analysis on different evaluation axis where each joints' error in the set is evaluated for measuring the accuracy.

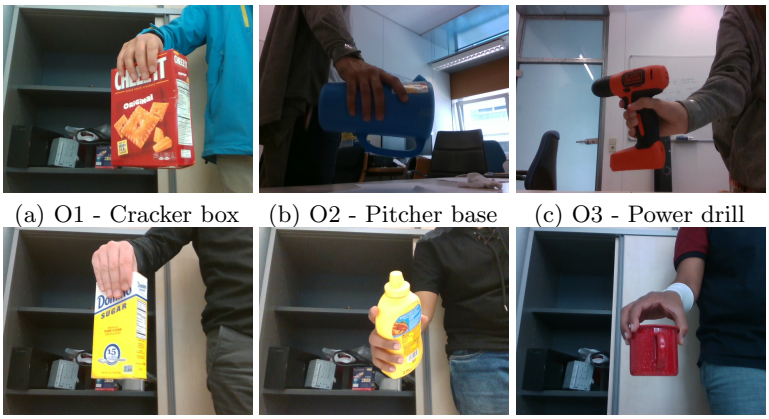


(a) Extrapolation (b) Shape

(c) Object (d) Interpolation

Fig. 17. Task 3 - Joint success rate analysis on different evaluation axis where each joints' error in the set is evaluated for measuring the accuracy.

A.3 Visualizations for Articulation Clusters, Hand Shapes and Object Types



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

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

Fig. 18. Example frames for the objects appear in Task 3, HO-3D [9] 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

Table 21. List of seen and unseen objects in the training dataset of Task 3.

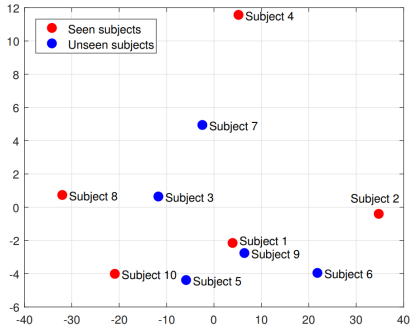


Fig. 19. Visualization of different hand shape distributions, appear in [46], by using the first two principal components of the hand shape parameters. Figure is taken from [46].

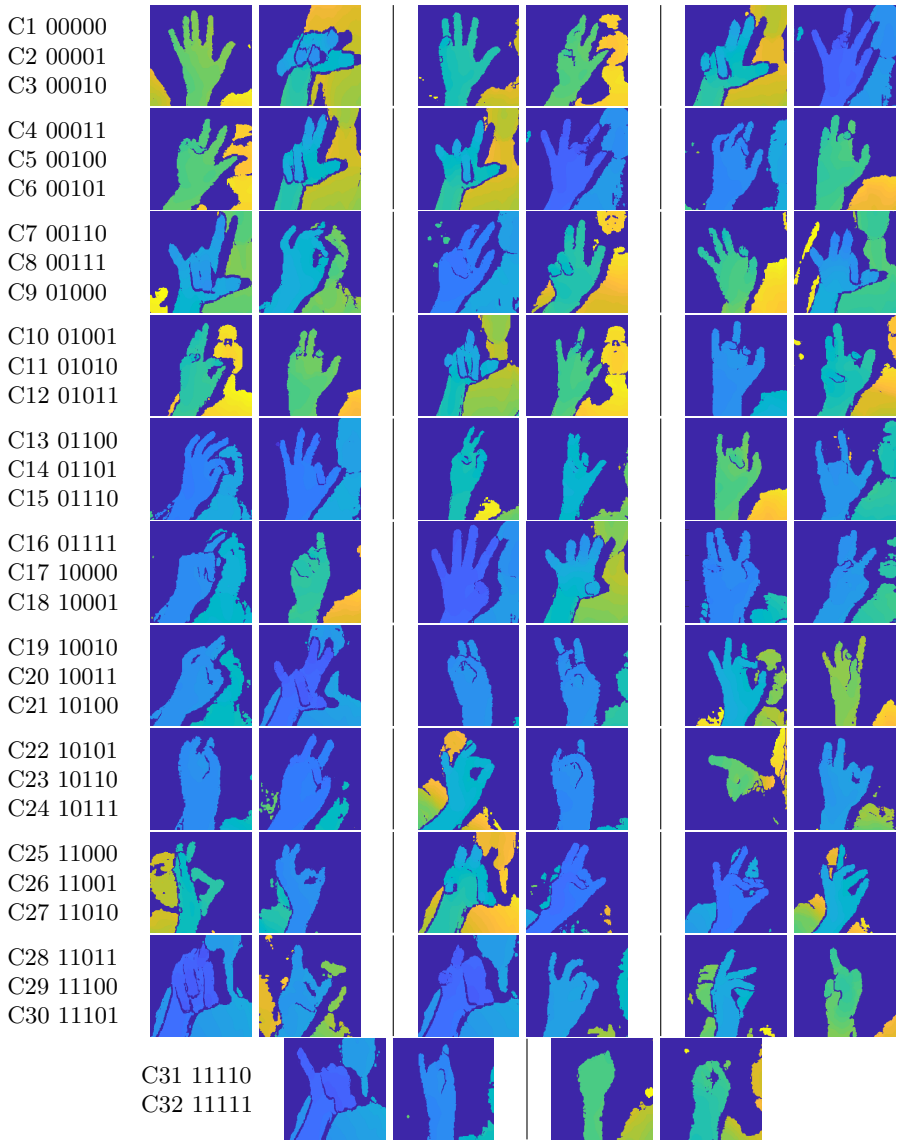


Fig. 20. Examples frames for 32 articulation clusters used in the evaluations. Each row shows cluster ids and their respective binary representations for two example images of three clusters. Each binary representation is constructed from thumb to pinky fingers with 0 representing closed and 1 representing open fingers.