Hand Pose Estimation for Hand-Object Interaction Cases using Augmented Autoencoder

Shile Li[1,*], Haojie Wang[1,*] and Dongheui Lee[1,2]

***Abstract*— Hand pose estimation with objects is challenging due to object occlusion and the lack of large annotated datasets. To tackle these issues, we propose an Augmented Autoencoder based deep learning method using augmented clean hand data. Our method takes 3D point cloud of a hand with an augmented object as input and encodes the input to latent representation of the hand. From the latent representation, our method decodes 3D hand pose and we propose to use an auxiliary point cloud decoder to assist the formation of the latent space. Through quantitative and qualitative evaluation on both synthetic dataset and real captured data containing objects, we demonstrate state-of-the-art performance for hand pose estimation with objects, even using only a small number of annotated hand-object samples.**

I. INTRODUCTION

Hand pose estimation plays an important role in many human-robot interaction tasks, such as teleoperation, virtual/augmented reality and robot imitation learning [1][2][3][4][5]. These applications require real-time and accurate hand pose estimation in 3D space. Recently, deep learning based methods have made significant progress in this area, which can be categorized to depth-based approaches [6][7][8][9][10][11][12][13] and RGB-based approaches[17][18][19]. Despite the success of these methods, they rarely concern the hand-object interaction cases. These methods typically fail in manipulation tasks because of the occlusions caused by the grasped object.

Recently, several works start to take object occlusion problems for hand pose estimation task into consideration. The majority are tracking based approaches [20][21][22]. The robust performance of these methods relies on tracking algorithms to exploit the temporal constraints between consecutive frames in input sequence. However, a good initialization is required for the first frame, and sometimes tracking drift happens. Other conventional methods [22][23][24] resort to multi-camera setups to reduce the influence of object occlusions from multiple viewpoints. However, it is expensive and complex to set up a synchronous and calibrated system with multiple sensors.

Currently, hand pose estimation for hand-object interaction cases is limited by existing available datasets. Public large-scale datasets with reliable 3D ground-truth annotations are lacking due to the complexity of annotating 3D hand pose. Although some large-scale datasets, like *Hands2017Challenge* [26], have accurate 3D pose annotations, they are entirely composed from clean hand samples. Therefore, it is worth considering how to utilize existing clean hand datasets for hand-object cases.

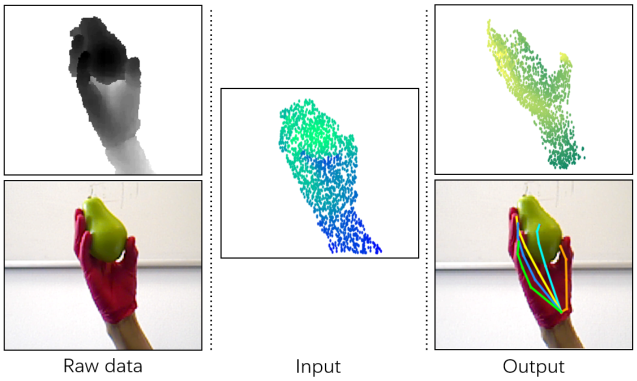

Fig. 1. The raw data are captured from a RGB-D camera. We use only the depth image to acquire the input cloud. The RGB image is used for visualization. For the output, besides the predicted pose, a clean hand is simultaneously reconstructed. (Brightness in point cloud indicates depth, i.e. darker denotes further.)

In this work, we propose a novel deep learning framework using Augmented Autoencoder to tackle hand-object interaction problem in hand pose estimation tasks. Our method takes 3D occluded hand point cloud as input, which is obtained by a random data augmentation process from clean hand samples. The encoder extracts point-wise features and fuses them to a latent vector. Addressing the problem of object occlusion in hand-object interaction cases, we use an auxiliary decoder to reconstruct the clean hand point cloud from the latent vector, and another decoder estimates simultaneously the 3D hand pose from the same latent vector. To the best of our knowledge, this is the first work that uses 3D point cloud data to tackle object occlusion problem in hand-object interaction tasks (Fig. 1).

Our contribution can be summarized as follows:

- We present an augmentation strategy to simulate hand-object interaction cases utilizing existing large clean hand datasets. Since unlimited types of objects could be augmented, the trained model is more generalizable on unknown objects.
- We propose an auxiliary clean hand reconstruction decoder to improve the quality of the latent space, which in turn improves the hand pose accuracy.
- We demonstrate the advantages of the proposed augmentation and reconstruction approaches both qualitatively and quantitatively through multiple experiments.

[1] Human-centered Assistive Robotic, Technical University of Munich.
[2] Institute of Robotics and Mechatronics, German Aerospace Center.
[*] The first two authors contributed equally to this work.

II. RELATED WORK

In the following, we first review some hand pose estimation works on both clean hand and hand-object interaction cases. Then we briefly introduce the backbone of our framework, Augmented Autoencoder and the utilized point cloud reconstruction method, FoldingNet.

A. Clean Hand Pose Estimation

In the past few years, a lot of 2D deep learning based methods for clean hand pose estimation has been proposed [12][13][11][14][15][16]. In particular, 2D depth image based methods demonstrate robust performance. Oberweger et al. [13] use 2D CNN to estimate the hand pose from the image features, where they introduce a bottleneck layer to force the predicted pose obey certain prior distribution. Wan et al. [12] estimate hand pose with a proposed pose parameterization strategy, which decomposes the pose parameters into a set of per-pixel estimations, i.e. 2D/3D heat maps and unit 3D directional vector fields, to leverage the 2D and 3D properties of the input depth map.

Recently, 3D deep learning methods gain more attention due to the abundant information in input data [7][8][29][9][10]. Ge et al. [8] present a PointNet [29] based approach that directly takes point cloud as input to regress 3D hand joint locations. In order to handle variations of hand global orientations, they introduce the oriented bounding box (OBB) to normalize the hand point clouds. Li et al. [7] propose a point-to-pose voting based residual permutation equivariant network for hand pose estimation task. Without the need of complex preprocessing steps, their method takes unordered 3D point cloud as input to compute point-wise features and through weighted fusion to obtain final hand pose estimates. Despite their good performance on hand pose estimation, they commonly ignore the crucial hand-object interaction cases.

B. Hand Pose Estimation with Object Interaction

There are some previous works that have taken the problem of object occlusion in hand pose estimation task into account [30][31][32][33]. The work by Tekin et al. [33] has impressive success of 3D hand pose estimation jointly with other parallel tasks. Their method takes a sequence of frames as input and outputs per-frame 3D hand and object pose predictions along with the estimates of object and action categories for the entire sequence, whereas it relies too much on a frame sequence rather than a single image. Gao et al. [31] propose an object-aware method to estimate 3D hand pose from a single RGB image, where they rely on a deep structure to infer the category of the grasped object shape under the assumption that objects of a similar category are grasped in a similar way. Boukhayma et al. [17] propose to use extracted hand parameters to control a mesh deformation hand model MANO [34] and project it into image domain to train the network. A similar hand model based work by Hasson et al. [25] uses a contact loss to describe the spatial state of hand and object when a hand manipulates object, i.e. using a repulsion loss to penalize interpenetration and an attraction loss to encourage the hand to be in contact with the object. These methods require complex annotation process and could not fully utilize existing annotated clean hand datasets for hand-object interaction cases.

C. Augmented Autoencoder and 3D Shape Reconstruction

Augmented Autoencoder is the backbone of our method, which is firstly proposed by Sundermeyer et al. [28] in their real-time RGB-based pipeline for object detection and 6D pose estimation. In order to remove the effects of object occlusions and background clutters, they use an augmentation process to generate input data, which superimposes artificial occlusions and clutters to the clean data. Their work demonstrates that this training procedure is able to enforce the invariance of the encoded latent variable against a variety of different input augmentations. Encouraged by the idea of augmentation invariance, we apply a random augmentation process on clean hand samples of existing datasets to generate our input, and recover corresponding clean hand samples with an auxiliary 3D shape reconstruction decoder.

3D Shape Reconstruction using deep learning has made a lot of advancement in recent years [35][36][37][38]. Yang et al. [37] propose a folding-based network, FoldingNet, which deforms a canonical 2D grid onto the underlying 3D target surface of a point cloud with two consecutive folding operations. For network complexity, FoldingNet consumes only about 7% parameters of a fully-connected layer based neural network to reconstruct a 3D target. Their method achieves low reconstruction errors even for targets with delicate structures. Therefore, we use FoldingNet for the clean hand reconstruction decoder.

A critical challenge in 3D shape reconstruction is to evaluate the predicted point cloud. The loss function should be not only computationally efficient but also differentiable with respect to point coordinates. The Chamfer Distance (CD) and the Earth Mover's Distance (EMD) [39] are two outstanding candidates to compare the reconstructed clean hand point cloud with ground-truth in our work.

III. METHOD

The overview of our method is illustrated in Fig. 2 (left). The framework is based on the structure of Variational Autoencoder (VAE) [27]. Our method takes 3D occluded hand point cloud as input, which is obtained by an augmentation process from clean hand and random objects. The encoder extracts point-wise features and fuses them to a latent vector, which is the latent representation of the input hand. Then, the acquired latent vector is used to reconstruct clean hand point cloud by the auxiliary Decoder 1 and predict 3D hand pose by Decoder 2.

A. Data Augmentation

The motivation behind our Augmented Autoencoder based hand pose estimation framework is to control what the latent vector encodes and which properties are ignored. To take advantages of current large-scale clean hand dataset, we apply a random augmentation process by superimposing random

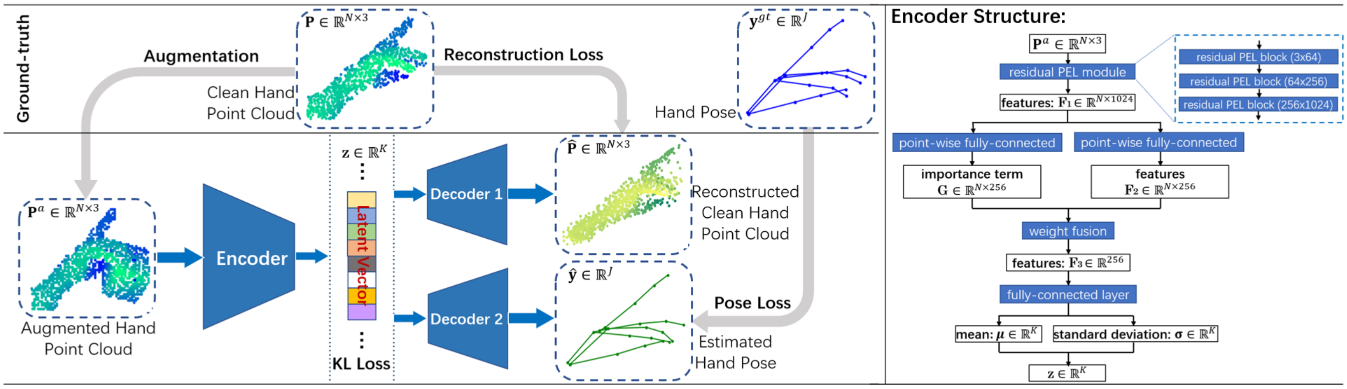
Ground-truth
Augmentation
$\mathbf{P} \in \mathbb{R}^{N\times 3}$
Clean Hand Point Cloud
Reconstruction Loss
$\mathbf{y}^{gt} \in \mathbb{R}^{J}$
Hand Pose
$\mathbf{P}^a \in \mathbb{R}^{N\times 3}$
Augmented Hand Point Cloud
Encoder
$\mathbf{z} \in \mathbb{R}^{K}$
Latent Vector
KL Loss
Decoder 1
Decoder 2
$\hat{\mathbf{P}} \in \mathbb{R}^{N\times 3}$
Reconstructed Clean Hand Point Cloud
$\hat{\mathbf{y}} \in \mathbb{R}^{J}$
Pose Loss
Estimated Hand Pose
Encoder Structure:
$\mathbf{P}^a \in \mathbb{R}^{N\times 3}$
residual PEL module
residual PEL block (3x64)
residual PEL block (64x256)
residual PEL block (256x1024)
features: $\mathbf{F_1} \in \mathbb{R}^{N\times 1024}$
point-wise fully-connected
point-wise fully-connected
importance term $\mathbf{G} \in \mathbb{R}^{N\times 256}$
features $\mathbf{F_2} \in \mathbb{R}^{N\times 256}$
weight fusion
features: $\mathbf{F_3} \in \mathbb{R}^{256}$
fully-connected layer
mean: $\boldsymbol{\mu} \in \mathbb{R}^{K}$
standard deviation: $\boldsymbol{\sigma} \in \mathbb{R}^{K}$
$\mathbf{z} \in \mathbb{R}^{K}$

Fig. 2. Overview of our method (left) and the structure of the encoder (right). The input of our network is occluded hand point cloud, which is obtained by a random augmentation process from clean hand point cloud. The encoder encodes the input hand to a latent vector. The obtained latent vector is then used to reconstruct clean hand point cloud by the auxiliary Decoder 1 and predict 3D hand pose by Decoder 2. There are three losses in our VAE based framework, which are the KL loss, reconstruction loss and pose loss. (Brightness in point cloud indicates depth, i.e. darker denotes further.)

objects from *ShapeNet* [40] on clean hands to simulate hand-object interaction scenarios in reality. Simultaneously, the clean hand point cloud also serves as the ground-truth for reconstructed points by the auxiliary Decoder 1. Through this approach, we make the latent representation invariant against object occlusions when a hand is in contact with an object.

The random augmentation process is shown in Fig. 3. In step 1, a randomly selected object from *ShapeNet* is superimposed on a clean hand point cloud sample after random rotation, scaling and translation. Step 2 renders the combined point cloud to depth image, where we only keep the point which is the closest to the camera among those projected to the same 2D image grid. Finally, step 3 converts the depth image to occluded hand point cloud.

B. Residual Permutation Equivariant Layer based Encoder

We use Residual Permutation Equivariant Layer (PEL) [7] as backbone to encode the input point cloud (Fig. 2 (right). The input occluded hand point cloud $\mathbf{P}^a \in \mathbb{R}^{N\times 3}$ represented by N unoredered 3D points passes firstly through a residual PEL module, which consists of 3 residual PEL blocks. Then point-wise feature $\mathbf{F_1} \in \mathbb{R}^{N\times 1024}$ is computed for each individual input point, where each row of $\mathbf{F_1}$ represents the local feature for one point. The obtained $\mathbf{F_1}$ is imported to two separate point-wise fully-connected modules respectively, resulting in two separate terms, an importance term $\mathbf{G} \in \mathbb{R}^{N\times 256}$ and a new feature term $\mathbf{F_2} \in \mathbb{R}^{N\times 256}$, where the local feature dimension for each point is shrunk to 256. Each element of $\mathbf{G}$ indicates the weight for corresponding feature value in $\mathbf{F_2}$ and provides vital information of the importance of current feature value. Then, by a weight fusion module, we merge the information of both terms to $\mathbf{F_3} \in \mathbb{R}^{256}$:

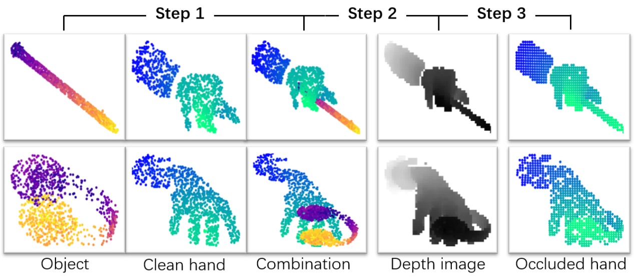

Fig. 3. Data augmentation process. Step 1: combine hand point cloud and object; Step 2: project combined point cloud to depth image; Step 3: convert depth image to occluded hand point cloud. (Brightness in point cloud indicates depth, i.e. darker denotes further.)

$$\mathbf{f}_i = \frac{\sum_{n=1}^{N} (\mathbf{G}_{ni} \mathbf{F_2}_{ni})}{\sum_{n=1}^{N} \mathbf{G}_{ni}}, \tag{1}$$

where $\mathbf{f}_i$ is the i-th feature value in $\mathbf{F_3}$.

In order to extract complex features, we use a 5-layer perceptron to encode $\mathbf{F_3}$ to the final K-dimensional latent vector, which consists of a latent mean vector $\boldsymbol{\mu} \in \mathbb{R}^K$, and a latent standard deviation vector $\boldsymbol{\sigma} \in \mathbb{R}^K$.

During training stage, a reparameterization process to sample from the distribution of the latent vector [27] is needed: $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \in \mathbb{R}^K$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\odot$ denotes element-wise multiplication. The final latent vector $\mathbf{z} \in \mathbb{R}^K$ is Gaussian distributed and $\mathbf{z} \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\sigma}^2\right)$.

C. Decoder and Training Loss

The obtained latent vector $\mathbf{z}$ from encoder is fed into decoders. The clean hand reconstruction Decoder 1 is based on FoldingNet [37]. The pose prediction Decoder 2 consists of multiple fully-connected layers.

Decoder 1 is a FoldingNet [37] that transforms (”folds”) 2d grid points of a square into 3D point cloud with two folding operations. In the folding operation, each grid point's coordinate is concatenated with the latent vector $\mathbf{z}$ and fed into a 4-layer perceptron to construct a more complex shape compared to the input. The final reconstructed points $\hat{\mathbf{P}}$ are evaluated by Chamfer Distance (CD) and Earth Mover's Distance (EMD) [39] with respect to the ground-truth clean hand point cloud $\mathbf{P} \in \mathbb{R}^{N\times 3}$. Note that the number of points in $\hat{\mathbf{P}}$ is required to be the same as $\mathbf{P}$.

The Chamfer Distance is defined as:

$$\mathcal{L}_{CD}\left(\hat{\mathbf{P}}, \mathbf{P}\right) = \frac{1}{|\hat{\mathbf{P}}|} \sum_{\hat{p} \in \hat{\mathbf{P}}} \min_{p \in \mathbf{P}} \|\hat{p} - p\| + \frac{1}{|\mathbf{P}|} \sum_{p \in \mathbf{P}} \min_{\hat{p} \in \hat{\mathbf{P}}} \|\hat{p} - p\|, \tag{2}$$

where the CD algorithm finds for each point the nearest neighbor in the other point cloud and sums up the Euclidean distances.

The Earth Mover's Distance requires that $\hat{\mathbf{P}}$ and $\mathbf{P}$ have the same size, i.e. $|\hat{\mathbf{P}}| = |\mathbf{P}|$, and it is defined as:

$$\mathcal{L}_{EMD}\left(\hat{\mathbf{P}}, \mathbf{P}\right) = \frac{1}{|\mathbf{P}|} \min_{\phi: \mathbf{P} \to \hat{\mathbf{P}}} \sum_{p \in \mathbf{P}} \|p - \phi(p)\|, \tag{3}$$

where ϕ denotes one-to-one bijective correspondences from the ground-truth $\mathbf{P}$ to the predicted point set $\hat{\mathbf{P}}$. The Euclidean distances of all matched point pairs are then summed.

Both loss functions have their own intrinsic characteristics. For example, while EMD roughly captures the shape corresponding to the mean value of the hidden variable of the hand point cloud, CD tends to give a splashy shape that blurs the shape's geometric structure [38]. To make the reconstruction by Decoder 1 more expressive, we combine both loss functions during training time. Therefore, implicitly, our method requires the reconstructed clean hand points have the same size N as the ground-truth.

For 3D hand pose prediction, Decoder 2, which consists of 5 fully-connected layers, takes the reparameterized latent vector as input and outputs the vectorized 3D hand pose $\hat{\mathbf{y}} \in \mathbb{R}^J$, where $J = 3 \times \#joints$. The training loss between predicted hand pose $\hat{\mathbf{y}}$ and ground-truth pose $\mathbf{y}^{gt} \in \mathbb{R}^J$ is the $L2$ loss:

$$\mathcal{L}_{pose} = \frac{1}{2} \sum_{j=1}^{J} \left(\hat{\mathbf{y}}_j - \mathbf{y}_j^{gt}\right)^2. \tag{4}$$

As the proposed framework is based on VAE, a KL (Kullback–Leibler divergence) loss is essential to force the computed latent vector $\mathbf{z}$ given observed occluded data to be close to the centered isotropic multivariate Gaussian $\mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$ (Fig. 2 left) . The KL loss is defined as:

$$\mathcal{L}_{KL} = \frac{1}{2} \sum_{k=1}^{K} \left(\mu_k^2 + \sigma_k^2 - \log\left(\sigma_k^2\right) - 1\right), \tag{5}$$

where K denotes the number of dimensions of the latent vector $\mathbf{z}$, μ_k is the k-th dimension of the latent mean $\boldsymbol{\mu}$ and σ_k denotes the k-th dimension of the latent standard deviation $\boldsymbol{\sigma}$.

The resulting total loss for our method is the summation of $\mathcal{L}_{CD}$, $\mathcal{L}_{EMD}$, $\mathcal{L}_{pose}$ and weighted $\mathcal{L}_{KL}$ terms:

$$\mathcal{L}_{total} = \mathcal{L}_{CD} + \mathcal{L}_{EMD} + \mathcal{L}_{pose} + \alpha \mathcal{L}_{KL}, \tag{6}$$

where α is the weight factor.

IV. EXPERIMENT AND RESULT

Our method is implemented using the TensorFlow framework with the ADAM optimizer. The learning rate is tapered down from 0.01 to 0.00001 during the course of training. For all experiments, we use an input and reconstruction point size of $N = 625$ for training, and $N = 900$ for testing. For the latent vector $\mathbf{z} \in \mathbb{R}^K$, we set the number of dimension $K = 64$ and the KL Loss is weighted using a factor of $\alpha = 0.001$. Before our object augmentation process, we perform for each hand sample random translation in all three dimensions within $[-15, 15]$ mm, random scaling within $[0.75, 1.25]$ and random rotation around z-axis within $[-\pi, \pi]$ $radian$. The trained model can be employed for real-time applications, since the network backbones, the ResPEL [7] and FoldingNet [37], are both real-time capable.

A. Datasets

For training and evaluating the proposed network, we use the *Hands2017Challenge* dataset [26], the *SynthHands* dataset [41] and also the *EgoDexter* dataset [41]. The *Hands2017Challenge* is collected from parts of the *BigHand2.2M* [42] and the *First-Person Hand Action (FHAD)* [43]. The training set contains 957032 depth images, and the test set contains 295510 depth images. All samples in *Hands2017Challenge* are clean hands, where the hands are not in contact with objects. The egocentric dataset *SynthHands* is a synthetic dataset created by posing a photorealistic hand model with real hand motion data. It captures multiple variations in natural hand motion, such as pose, skin color, shape, texture, background clutter as well as camera viewpoint. This dataset contains accurate annotated 92536 RGB-D images of clean hands and 91600 RGB-D images of hands interacting with objects, of which we use 69402 clean samples and 68700 interacting hand samples for training. Except the training samples, the rest 23134 clean samples serve as our *clean test set* and 22900 interacting samples as our *interacting test set*. The benchmark dataset *EgoDexter* consists of four real sequences with hand-object interactions (Rotunda, Desk, Kitchen, Fruits), which contain in total 1485 frames with 3D finger tip annotations. We compare the accuracy to the state-of-the-art method in [41] using this dataset. We exclude the Kitchen sequence due to its many annotation errors, and use the other three sequences for evaluation.

For the random augmentation process for clean hand samples, we use objects from *ShapeNetCore*, which is a subset of the object repository *ShapeNet* [40] and covers 55 object categories with about 51300 unique 3D models. As preprocessing, we sample these 3D models to point clouds.

B. Evaluation Metrics

We evaluate the performance of our method only qualitatively on real data for the trained model on *Hands2017Challenge*, because it contains no annotated samples for hand-object interaction cases. For the *SynthHands* dataset, two standard metrics are used for evaluation. The first one is the mean joint error (mm), which measures the average Euclidean distance error for all joints across the whole test set. The second metric is correct frame proportion, which indicates the percentage of frames that have all joint

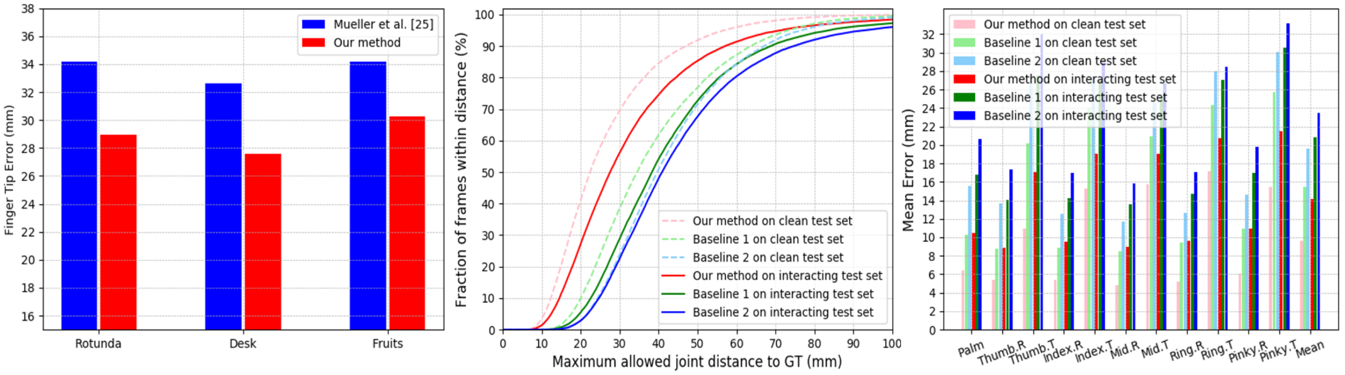

Fig. 4. Left: Comparison to state-of-the-art method on *EgoDexter* benchmark. Middle: Comparison to baselines on *SynthhandsTest* - Proportion of correct frames with respect to different error thresholds. Right: Comparison to baseline on *SynthhandsTest* - Mean errors of different joints.

errors within a certain threshold compared to the ground-truth. The correct frame proportion metric is challenging, since a single joint violation will cause an incorrect frame. For the *EgoDexter* dataset with only finger tip annotations, we use finger tip error for evaluation, which is the mean joint error for 3D finger tip positions.

C. Comparison to state-of-the-art Method

Since the *EgoDexter* dataset is only annotated on 3D finger tip positions, we use the finger tip error to compare the performance of our method with the kinematic pose tracking method proposed by Mueller et al. [41]. We follow the same training dataset in their work, where all samples in *SynthHands* are used. As shown in Fig. 4 (left), our method outperforms the state-of-the-art method on the test sequences, achieving the average error of 28.70 mm. Note that the objects in *EgoDexter* are different from the objects in *SynthHands* training data. It shows the generalization ability of our method to unknown objects.

D. Ablation Study

In the **first** ablation experiment, we mix different proportions of interacting hand samples to training set to compare the performance of different trained models. Then we use the optimal mixing proportion for the next experiments. c

Using the training samples from *SynthHands*, we set 4 different training datasets with varying percentages of hand-object interaction samples:

- Dataset A: 100% clean hand samples.
- Dataset B: 75% clean + 25% interacting hand samples.
- Dataset C: 50% clean + 50% interacting hand samples.
- Dataset D: 25% clean + 75% interacting hand samples.

Note that the interacting hand samples are not augmented during training time. Also, note that the performance of interacting hand is usually much worse than the clean hand samples due to occlusion.

The detailed comparison of mean joint errors on our both test sets can be found in Table I. We can already obtain a reasonably good result on 100% clean hand Dataset A. Even if using only augmented hand samples from clean hand without any interacting hand samples, the error on interacting test set is 19.13 mm, which indicates the effectiveness of the augmentation strategy.

TABLE I
Comparison of different training methods on SynthHands.

Training Dataset	Error on Test Dataset (mm)	
	clean hand	interacting hand
A	9.67	19.13
B	**9.63**	**14.16**
C	10.69	14.35
D	12.52	15.99

Furthermore, the best performance is achieved with training Dataset B, which contains 25% interacting hand samples. Compared to Dataset A, the mean joint error is decreased for 5 mm on interacting hand test set by mixing only a small proportion of real interacting hand samples in the training dataset. However, with the increasing proportion of interacting hand for training, the results become slightly worse, even on the interacting test set. The possible reason for this is that the decrease of clean hand proportion leads to less data augmentation, which means less random objects are seen for the training process, resulting in less generalizability on the unseen objects in the test set. Moreover, for the interacting training samples, hand reconstruction part were not trained since there is no available clean hand ground-truth to guide reconstruction, this leads to insufficient training of the reconstruction decoder and in turn influences the quality of the latent space. This experiment shows that, in practice, we can utilize large clean hand dataset and mix a small proportion of interacting hand samples, which are expensive to annotate, to achieve robust performance.

In the **second** experiment, for ablation study, we set the following baselines to show the effects of the data augmentation and points reconstruction approaches:

- Baseline 1. Ours without object augmentation.
- Baseline 2. Ours without clean hand reconstruction.

Both baselines are trained using Dataset B. As seen in Fig. 4 (middle and right), our method outperforms the two baselines on both clean hand test set and interacting hand test set. Table II shows that the results of baselines are worse even on clean hand test set. The possible reason for this is that the latent representation in baselines is implicitly correlated

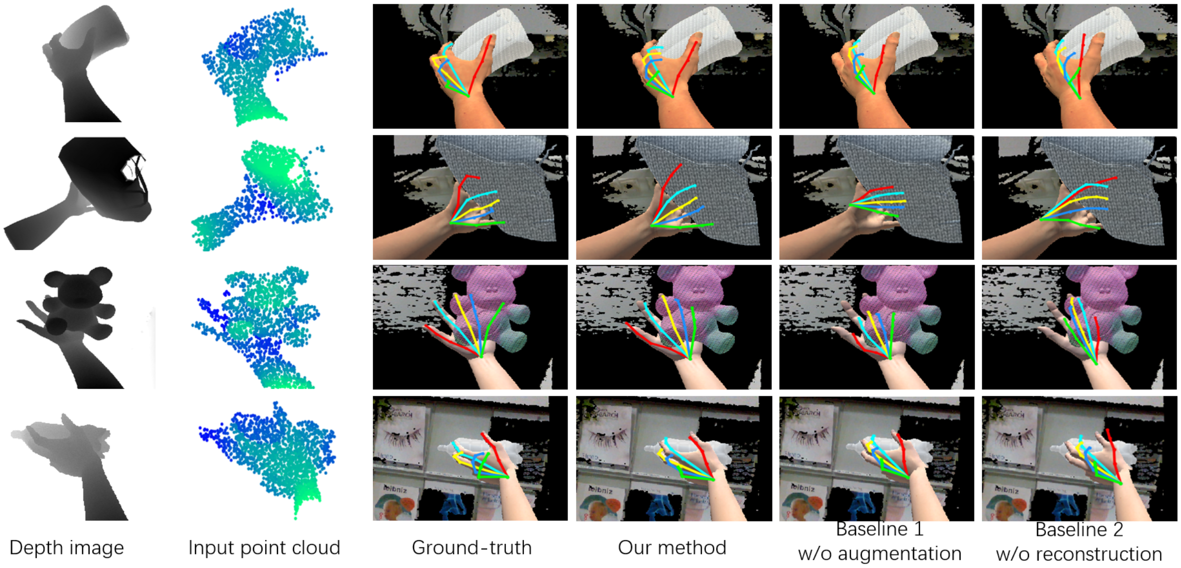

Fig. 5. Qualitative results compared with baselines on *SynthHands*. (Brightness in point cloud indicates depth, i.e. darker denotes further.)

TABLE II
COMPARISON WITH BASELINES ON SYNTHHANDS.

Model	**Error on Test Dataset (mm)**	
	clean hand	interacting hand
Our method	**9.63**	**14.16**
Baseline 1	15.44	20.78
Baseline 2	19.60	23.46

to the mixture of clean hands and interacting hands rather than clean hands alone in our Augmented Autoencoder based framework. By this result, we demonstrate the significant effects of the augmentation component and the reconstruction component in our method.

E. Qualitative Results

For the *SynthHands* dataset, the qualitative comparison of our method with two baselines is shown in Fig. 5 on the interacting test set.

For the *Hands2017Challenge* dataset, as the training set and test set contain only clean hands, we train our model without mixing any interacting hands. Furthermore, we just give a qualitative result on the trained model with this dataset for evaluation. Fig. 6 shows qualitative results on real data, where the hand interacts with different objects, such as ball, bucket, phone, paper box, which are not seen during training. Although the model is trained only with clean hand data on the *Hands2017Challenge* dataset, the results shows good performance.

Note that high quality point cloud reconstruction is not strictly required in our method. Fig. 6 shows that occluded objects are roughly removed after reconstruction, indicating the importance of Decoder 1 for the formation of the latent space of the clean hand.

V. CONCLUSION

In this paper, we propose a novel deep learning framework using Augmented Autoencoder to handle hand pose estimation tasks for hand-object interaction cases. Our method consumes 3D hand point cloud and predicts accurate 3D hand pose. The proposed augmentation process and auxiliary clean hand reconstruction decoder implicitly force the latent representation of the pose only to be correlated to clean hand and the reconstructed clean hand despite the object occlusion in hand-object interaction cases. Furthermore, the proposed hand pose estimation training strategy is able to utilize existing clean hand datasets to tackle hand-object interaction cases. Quantitative and qualitative evaluation results show that our framework is capable of achieving low joint errors on both clean hand input ($\sim 9\ mm$) and interacting hand input ($\sim 14\ mm$). In the future work, more aspects of joint hand-object case will be investigated such as object pose estimation [44] and physical constraints. Another interesting aspect will be evaluating the grasp quality of reconstrcuted hand pose.

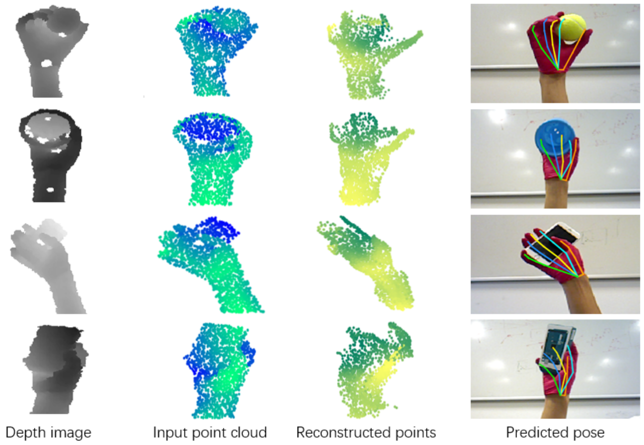

Fig. 6. Qualitative results on real data. (Brightness in point cloud indicates depth, i.e. darker denotes further.)

Acknowledgement: This work has been partially supported by Helmholtz Association.

References

[1] L. Zollo, S. Roccella, E. Guglielmelli, M. Carrozza, and P. Dario. Biomechatronic design and control of an anthropomorphic artificial hand for prosthetic and robotic applications. *IEEE/ASME Transactions On Mechatronics*, 12(4):418–429, 2007.

[2] Y. Jang, S. Noh, H. Chang, T. Kim, and W. Woo. 3d finger cape: Clicking action and position estimation under self-occlusions in egocentric viewpoint. *IEEE Transactions on Visualization and Computer Graphics*, 21(4):501–510, 2015.

[3] T. Piumsomboon, A. Clark, M. Billinghurst, and A. Cockburn. User-defined gestures for augmented reality. In *IFIP Conference on Human-Computer Interaction*, pages 282–299. Springer, 2013.

[4] B. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[5] S. Calinon and D. Lee Learning Control, Humanoid Robotics: a Reference *Springer, 2017*, 57(5):469–483, 2017

[6] S. Yuan, G. Garcia-Hernando, B. Stenger, G. Moon, J. Chang, K. Lee, P. Molchanov, J. Kautz, S. Honari, L. Ge, et al. Depth-based 3d hand pose estimation: From current achievements to future goals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2636–2645, 2018.

[7] S. Li and D. Lee. Point-to-pose voting based hand pose estimation using residual permutation equivariant layer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11927–11936, 2019.

[8] L. Ge, Y. Cai, J. Weng, and J. Yuan. Hand pointnet: 3d hand pose estimation using point sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8417–8426, 2018.

[9] L. Ge, H. Liang, J. Yuan, and D. Thalmann. Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3593–3601, 2016.

[10] L. Ge, H. Liang, J. Yuan, and D. Thalmann. 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1991–2000, 2017.

[11] C. Wan, T. Probst, L. Van Gool, and A. Yao. Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 680–689, 2017.

[12] C. Wan, T. Probst, L. Van Gool, and A. Yao. Dense 3d regression for hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5147–5156, 2018.

[13] M. Oberweger, P. Wohlhart, and V. Lepetit. Hands deep in deep learning for hand pose estimation. *arXiv:1502.06807*, 2015.

[14] C. Zimmermann and T. Brox. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4903–4911, 2017.

[15] A. Spurr, J. Song, S. Park, and O. Hilliges. Cross-modal deep variational hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 89–98, 2018.

[16] L. Yang and A. Yao. Disentangling latent hands for image synthesis and pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9877–9886, 2019.

[17] A. Boukhayma, R. Bem, and P. Torr. 3d hand shape and pose from images in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10843–10852, 2019.

[18] P. Panteleris, I. Oikonomidis, and A. Argyros. Using a single rgb frame for real time 3d hand pose estimation in the wild. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 436–445. IEEE, 2018.

[19] L. Yang, S. Li, D. Lee, and A. Yao. Aligning latent spaces for 3d hand pose estimation. In *Proceedings of the International Conference on Computer Vision*, 2019.

[20] N. Kyriazis and A. Argyros. Physically plausible 3d scene tracking: The single actor hypothesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9–16, 2013.

[21] H. Hamer, K. Schindler, E. Koller-Meier, and L. Van Gool. Tracking a hand manipulating an object. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1475–1482. IEEE, 2009.

[22] I. Oikonomidis, N. Kyriazis, and A. Argyros. Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *2011 International Conference on Computer Vision*, pages 2088–2095. IEEE, 2011.

[23] L. Ballan, A. Taneja, J. Gall, L. Van Gool, and M. Pollefeys. Motion capture of hands in action using discriminative salient points. In *European Conference on Computer Vision*, pages 640–653, 2012.

[24] D. Tzionas, L. Ballan, A. Srikantha, P. Aponte, M. Pollefeys, and J. Gall. Capturing hands in action using discriminative salient points and physics simulation. *International Journal of Computer Vision*, 118(2):172–193, 2016.

[25] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. Black, I. Laptev, and C. Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11807–11816, 2019.

[26] S. Yuan, Q. Ye, G. Garcia-Hernando, and T. Kim. The 2017 hands in the million challenge on 3d hand pose estimation. *arXiv:1707.02237*, 2017.

[27] D. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv:1312.6114*, 2013.

[28] M. Sundermeyer, Z. Marton, M. Durner, M. Brucker, and R. Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 699–715, 2018.

[29] C. Qi, L. Yi, H. Su, and L. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.

[30] Q. Ye and T. Kim. Occlusion-aware hand pose estimation using hierarchical mixture density network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–817, 2018.

[31] Y. Gao, Y. Wang, P. Falco, N. Navab, and F. Tombari. Variational object-aware 3d hand pose from a single rgb image. *IEEE Robotics and Automation Letters*, 2019.

[32] M. Madadi, S. Escalera, A. Carruesco, C. Andujar, X. Baró, and J. Gonzàlez. Occlusion aware hand pose recovery from sequences of depth images. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, p. 230–237, 2017.

[33] B. Tekin, F. Bogo, and M. Pollefeys. H+ o: Unified egocentric recognition of 3d hand-object poses and interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4511–4520, 2019.

[34] J. Romero, D. Tzionas, and M. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (TOG)*, 36(6):245, 2017.

[35] C. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, p. 628–644., 2016.

[36] T. Groueix, M. Fisher, V. Kim, B. Russell, and M. Aubry. Atlasnet: A papier-m\ˆ ach\'e approach to learning 3d surface generation. *arXiv:1802.05384*, 2018.

[37] Y. Yang, C. Feng, Y. Shen, and D. Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *IEEE Conference on Computer Vision and Pattern Recognition*, p. 206–215, 2018.

[38] H. Fan, H. Su, and L. Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.

[39] Y. Rubner, C. Tomasi, and L. Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.

[40] A. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv:1512.03012*, 2015.

[41] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1284–1293, 2017.

[42] S. Yuan, Q. Ye, B. Stenger, S. Jain, and T. Kim. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4866–4874, 2017.

[43] G. Garcia-Hernando, S. Yuan, S. Baek, and T. Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 409–419, 2018.

[44] S. Li, S. Koo and D. Lee Real-time and Model-free Object Tracking using Particle Filter with Joint Color-Spatial Descriptor. In International Conference on Intelligent Robots and Systems (IROS 2015) .