Hand Mesh and Object Pose Reconstruction Using Cross Model Autoencoder

Chaitanya Bandi[a] and Ulrike Thomas
Robotics and Human Machine Interaction Lab, Technical University of Chemnitz, Chemnitz, Germany

Keywords: Hand, Object, Pose, Reconstruction, Autoencoder.

Abstract: Hands and objects severely occlude each other, making it extremely challenging to estimate the hand-object pose during human-robot interactions. In this work, we propose a framework that jointly estimates 3D hand mesh and 6D object pose in real-time. The framework shares the features of a single network with both the hand pose estimation network and the object pose estimation network. Hand pose estimation is a parametric model that regresses the shape and pose parameters of the hand. The object pose estimation network is a cross-model variational autoencoder network for the direct reconstruction of an object's 6D pose. Our method shows substantial improvement in object pose estimation on two large-scale open-source datasets.

1 INTRODUCTION

Hands are the primary tools that interpret the actions of humans and interact with the real environment. To understand human action and behavior in human-robot interaction environments, the poses of the hand and the poses of the interacting objects are necessary. With advancements in computer vision and deep learning, both hand pose estimation (Zimmermann and Brox, 2017; Spurr et al., 2018; Mueller et al., 2018; Ge et al., 2019; Hasson et al., 2019; Park et al., 2022; Mueller et al., 2017; Garcia-Hernando et al., 2018; Yuan et al., 2018; Moon et al., 2018; Zhou et al., 2020) and object pose estimation (Kehl et al., 2017; Xiang et al., 2018; Rad and Lepetit, 2017; Tekin et al., 2018; Peng et al., 2019; Hu et al., 2019) have made significant progress independently. 3D hand and object pose estimation is a central part of applications like human-robot interaction (Yang et al., 2021; Ortenzi et al., 2021), virtual reality (Höll et al., 2018), and augmented reality (Piumsomboon et al., 2013). To avoid implausible mesh representations, the strict relationship between the hand and the object must be understood. Although joint hand-object pose estimation has gained interest in recent studies, it requires further attention.

Combined hand-object pose estimation is quite challenging because of the self and mutual occlusions of hands and objects. 3D hand-object pose estimation research falls into two categories: optimization- based and learning-based. The optimization requires refinement where the process is repeated multiple times to achieve convergence, unlike the end-to-end learnable models. Because convergence takes a considerable amount of time for optimization methods, real-time applicability is out of the question. Similarly, in this study, we focus on a learning-based approach with

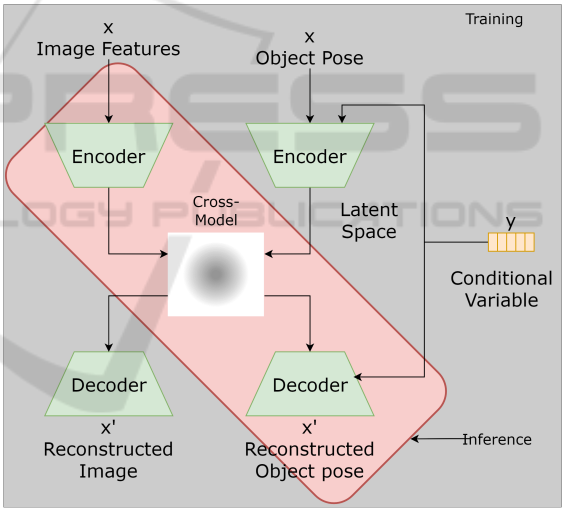

Figure 1: The basic structural process of the cross-model conditional variational autoencoder. Each autoencoder consists of an encoder, a low dimensional latent distribution, and a decoder. The latent space is shared between the models and paths are switched depending on the decoder.

[a] https://orcid.org/0000-0001-7339-8425

Bandi, C. and Thomas, U.
Hand Mesh and Object Pose Reconstruction Using Cross Model Autoencoder.
DOI: 10.5220/0012370700003660

In *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2024) - Volume 4: VISAPP*, pages 183-193
ISBN: 978-989-758-679-8; ISSN: 2184-4321

a user-friendly scenario in which hand pose and object pose reconstruction are performed using a single RGB image. Learning-based approaches can be broadly categorized as those that rely on implicit representations and parametric mesh models. The well-known parametric hand model is MANO (Romero et al., 2017). With prior shape knowledge and actual 3D human hand scans, MANO produces anthropomorphically valid hand meshes. Although parametric meshes have limited resolution, it is difficult to recover intricate interactions from them. In addition reconstructing 3D objects in hands is quite challenging. The complexity further increases during interaction with the objects. Recent research (Tekin et al., 2019; Doosti et al., 2020; Hasson et al., 2019; Tse et al., 2022; Liu et al., 2021) has been successful in addressing the challenges of estimating or reconstructing hand-object pose estimation from a single RGB image. Recently, there have been encouraging findings regarding object reconstruction using neural implicit representations (Karunratanakul et al., 2020). In their work, the authors demonstrate how to model hand-object interactions using the joint representation of unified signed distance fields (SDFs). The network does not consider any explicit prior details about hands and objects that cause unrealistic meshes.

In this work, we propose reconstructing an object's 6D pose from a deep generative model known as a conditional variational autoencoder (CVAE) (Sohn et al., 2015) with cross-modality. Autoencoders usually generate n number of output samples for a given input. In our case, we need one input and one output from the generative model; therefore, we consider cross-model CVAE. The basic cross-model CVAE architecture is illustrated in Figure 1. We also exploit the idea of using an attention module from the transformer (Vaswani et al., 2017) to enhance the encoder and decoder features of CVAE.

To summarize, the core contributions of these work areas are as follows:

[1] We propose a joint hand-object reconstruction model from a single RGB image.

[2] We designed a novel framework for object pose reconstruction using autoencoder models.

[3] We evaluate the framework on ObMan (Hasson et al., 2019) and DexYCB (Chao et al., 2021) large-scale open-source datasets and show that our framework outperforms the state-of-the-art methods on object mesh reconstruction.

2 RELATED WORK

Our research is related to hand pose estimation, object pose estimation, joint hand-object pose estimation, and variational autoencoders. Different input information, such as RGB, depth, and point cloud information, is used to estimate the hand-object pose. Recent research work on hand pose estimation completely focused on regressing 2D and 3D hand poses from a single RGB image.

Hand Pose Estimation. Zimmerman et al. (Zimmermann and Brox, 2017) present a cascaded architecture with segmentation, pose, and pose-prior networks. Initially, the region of the hand is segmented and forwarded to a pose network for 2D heatmap regression of hand joints. Later, the pose prior network elevates the 2D keypoints to 3D keypoints. Adrian et al. (Spurr et al., 2018) propose a cross-model latent space reconstruction of the hand pose using a variational autoencoder. A simple regression of 2D or 3D hand pose does not convey the shape of the hand, Ge et al. (Ge et al., 2019) present a graph convolutional network-based architecture to recover 3D hand mesh. Model-based approaches rely on a differentiable MANO model (Romero et al., 2017) to obtain a 3D hand pose and shape with a mesh. Later, the research works were extended to model-based methods that regress the pose and shape parameters of a hand (Boukhayma et al., 2019; Park et al., 2022; Hasson et al., 2019; Kulon et al., 2020; Zhang et al., 2019) instead of 3D keypoints.

Object Pose Estimation. The research work suggests that there are two different methods of 6D object pose estimation: direct regression and regression of 3D object points for recovery of 6D object pose using the perspective-n-point (PnP) algorithm. Yu et al. (Xiang et al., 2018) propose a convolutional neural network-based model for 6D object pose estimation using regression translation and rotation as a quaternion. The authors also introduce a large-scale 6D object pose dataset known as the YCB dataset, which is widely used. Due to the limitation of direct regression, the works (Peng et al., 2019; Hu et al., 2019) rely on a two-stage process of detecting 2D keypoints in RGB images using convolutional neural networks and then using the known 3D correspondences to obtain 6D pose using the PnP algorithm. To further improve the accuracy of 6D object pose, (Labb'e et al., 2020) introduce multi-view multi-object pose estimation.

Unified Hand-Object Pose Estimation. The earliest unified hand-object pose estimation (Tekin et al., 2019) solves four tasks simultaneously, i.e., object pose estimation, 3D hand pose estimation, object recognition, and action classification, using a single-

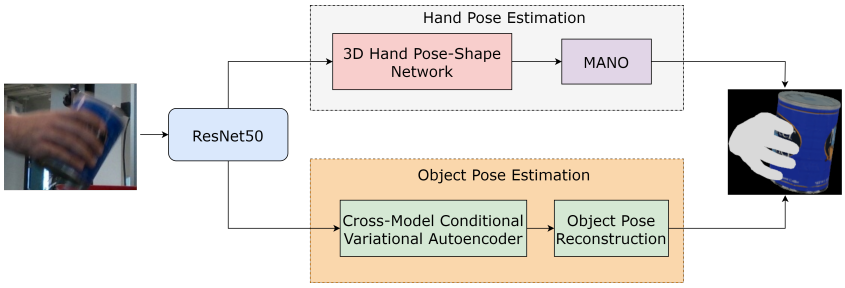

Figure 2: The basic overview of the proposed architecture. The input image is a closely cropped region of hand manipulating object. The input image is passed through the ResNet50 architecture to obtain the shared features infromation for the hand pose estimation and the object pose estimation. The output from the architectuer is a hand mesh from MANO model and 6D object pose from autoencoder.

shot neural network. To compute the object pose, the authors abandon the notion of 2D–3D correspondences and instead regress the direct 3D bounding box coordinates of the object. Doosti et al. (Doosti et al., 2020) propose Graph UNet architecture to further enhance the accuracy of combined 3D hand-object pose estimation. For hand-object manipulation applications, Hasson et al. (Hasson et al., 2019) propose an end-to-end model for the regression of plausible hand-object poses using a shared feature backbone. Sharing the feature network for the hand-object pose implicitly encodes contextual information. Leveraging the context information of hand and object, (Liu et al., 2021) introduces semi-supervised learning for hand-object interactions. The work generates pseudolabels by considering spatial-temporal consistencies. The architecture consists of two different streams that share a similar FPN architecture with the ResNet50 backbone. The features of the hand and object were extracted from the FPN architecture, and contextual reasoning was performed for object pose estimation. The features are then forwarded to independent decoders to regress the hand mesh and 6D object pose. The work (Tse et al., 2022), presents collaborative learning for hand-object pose reconstruction using unsupervised associative loss. The hand-object features are encoded independently at the input without a shared backbone, and the information from attention-guided graph convolution is shared with the object mesh network and the hand mesh network. The work (Wang et al., 2022) propose a dense mutual attention module to refine the hand and object meshes estimated in the first stage. The network models the fine-grained dependencies between hands and objects using a graph convolution network and attention. AlignSDF (Chen et al., 2022) is one of the early works to propose a hybrid model that combines a parametric model with an implicit representation model known as SDFs. The authors consider pose priors to the SDFs, unlike the work in (Karunratanakul et al., 2020) to further enhance the hand-object reconstruction.

Variational and Conditional Variational Autoencoder. Conditional variational autoencoders (CVAE) (Sohn et al., 2015) are an extension of the variational autoencoders (VAE) (Kingma and Welling, 2014; Rezende et al., 2014). VAEs are deep learning generative models for learning the latent distribution of input samples instead of fixed vector learning. VAE comprises an encoder, latent distribution, and a decoder. To convert the input data samples to a compressed low-dimensional latent representation, a probabilistic encoder with a mean and standard deviation is used. In CVAE, an additional condition variable is added to the encoder input and respectively to a decoder. Li et al. (Li et al., 2020) propose an augmented autoencoder for hand pose estimation during object occlusions. The authors used a variational autoencoder to estimate the 3D hand pose during object occlusion from a point cloud input. The work in (Chen et al., 2020) presents an idea of learning shape using VAE for size estimation, in addition to pose and shape. The work (Spurr et al., 2018) introduces cross-model variational autoencoders that result in a single latent space for input with multiple modalities, and we draw inspiration from this work for the cross-model CVAE that we propose.

3 METHODOLOGY

In Figure 2, we introduce our hand-object joint reconstruction network. The network consists of a feature extraction network in the first stage that takes an RGB image $\mathbb{R}^{3\times256\times256}$ of a combined hand-object region. The feature extraction network is a well-known ResNet-50 (He et al., 2016) architecture. The features from the ResNet50 backbone are passed through a deep neural network layer with an attention module to obtain the pose and shape parameters

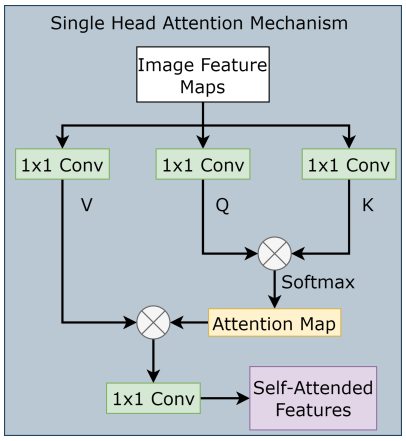

Figure 3: The single head attention mechanism used in the transformer.

for the parametric MANO (Romero et al., 2017) hand model. The object pose estimation network comprises a cross-model variational autoencoder to reconstruct the object pose.

3.1 Attention Mechanism

The attention mechanism introduced in (Vaswani et al., 2017) works well for many applications such as natural language processing and computer vision (Dosovitskiy et al., 2021). The attention mechanism takes n features as input and returns n output features. The basic operation of attention is that it learns to pay more attention to the necessary features. The attention mechanism is also known as scaled dot-product attention and consists of queries (Q), keys (K), and values (V) as inputs. The same input features are copied to queries, keys, and values, and the attention is computed as follows:

$$\text{Attention}\,(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (1)$$

where $\sqrt{d_k}$ is a scaling factor. The attention mechanism can be used for n dimensional (D) space. The attention mechanism is illustrated in Figure 3 single head attention block. The single-head attention mechanism is further extended to multi-head attention by combining multiple heads in parallel. The attention mechanism works for an n-dimensional input.

3.2 Hand Pose Shape Regression Network

The hand pose estimation network is a direct shape and pose parameter regression model with self-attention combined with residuals as represented in Figure 4, and a MANO (Romero et al., 2017) model to obtain hand vertices and 3D hand joints. The process of hand mesh extraction is similar to the work in (Liu et al., 2021). The only difference is the pose and shape regression network, in which the attention modules are included to further enhance the features. For learning supervision, we consider L_2 loss between the groundtruth joints J_{gt} and the predicted joints J_p from the MANO model.

$$L_{J^{3D}} = \sum_{j=i}^{21} \|J_{gt} - J_p\|_2^2 \qquad (2)$$

The hand pose and shape regression network consists of three residual attention modules with pooling layers. Each residual layer consists of two convolutional attention layers. As the attention layer has the same input and output sizes, the pooling layers are introduced after the residual attention as represented in the Figure 4. After the residual attention layers, two fully connected layers are connected with a batch normalization and ReLU activation unit. The final layers output a total of $\mathbb{R}^{58}$. The output from the regression network consists of pose $\theta \in \mathbb{R}^{48}$ and shape $\beta \in \mathbb{R}^{10}$ parameters for the MANO model. The MANO model reconstructs the hand vertices and 3D hand joints from the pose and shape parameters of the regression network. The L_2 loss is computed on the pose parameters θ, and shape parameters β for further regularization. The overall loss of the hand branch is the sum of the 2D heatmap loss, pose-shaped regression loss, and MANO (Romero et al., 2017) loss. Due to the limitations of the parametric MANO model, we include an additional loss known as biomechanical constraint loss L_{BMC} using hand kinematics as in (Spurr et al., 2020). This loss is also considered during training to avoid undesirable hand joint kinematics.

$$L_{handloss} = \lambda_J L_{J^{3D}} + \lambda_\theta L_\theta + \lambda_\beta L_\beta + L_{BMC} \qquad (3)$$

where $\lambda_J = 0.5$, $\lambda_\theta = 5 \times 10^{-7}$, $\lambda_\beta = 5 \times 10^{-5}$ to balance the joint loss and pose-shape parameters.

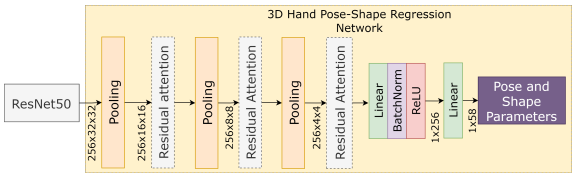

Figure 4: The hand pose and shape regression network. The network consists of three residual attention layers with pooling in between to reduce the feature size. The output from the attention layers is further passed to the linear layers to regress hand pose and shape parameters.

3.3 Object Pose Estimation Network

The object pose estimation network comprises a cross-model conditional variational autoencoder network for 6D pose reconstruction. The cross-model CVAE network consists of two VAEs or one CVAE and one VAE that share the latent space and decoders, as shown in Figure 5 and Figure 1. There are two different branches in cross-model CVAE: the object reconstruction branch and the image reconstruction branch.

3.4 Cross-Model Conditional Variational Autoencoder

A CVAE (Sohn et al., 2015) is a deep generative model with a conditional argument that is widely used in applications such as robotics (Ivanovic et al., 2021), image classification (Bao et al., 2017), and object detection. Simple CVAE takes object pose x as input, a conditional variable y, and learns to reconstruct x', which is similar to the input object pose. CVAE comprises an encoder network that resembles the latent distribution z, or $q_\phi(z|x,y)$. The latent distribution is Gaussian with unit variance. The following section describes a decoder network that is an approximation of $p_\theta(x|z,y)$. The decoder network generates a grasp pose from the learned distribution $p_\theta(x|z,y)$ given the conditional variable y and a sample from the latent distribution. The loss function for weight learning is given by Eq. 4.

$$\mathcal{L}(\theta,\phi) = -\mathbb{E}_{z\sim q_\phi(z|x,y)}[\log p_\theta(x|z,y)] + \beta_{vae} D_{KL}(q_\phi(z|x,y)||p_\theta(z|y)) \quad (4)$$

Eq. 4 consists of a L_2 reconstruction loss in the first term, which represents the disparity between the real pose x and the reconstructed pose x' obtained from the decoder. The Kullback-Leibler divergence (KLD) between the learned latent distribution and the unit-variance Gaussian is managed in the second part of Eq. 4. KL divergence serves as a regularization to maintain the variational posterior near the prior distribution over latent variables. The parameter β_{vae} balances the capacity of the latent variables with the reconstruction error (Higgins et al., 2017).

$$KLD = -0.5(1+\log(\sigma^2)-\mu^2-\sigma^2) \quad (5)$$

For the first branch, the object pose vector is considered as the input. The input object pose x is forwarded to the CVAE encoder layers. The encoder consists of three fully connected layers with batch normalization and leaky ReLU, except for the last layer. The first, second, and third layer output features are 128, 256, and 512, respectively. We pass the fully connected features to a self-attention module as in Figure 3 and a linear layer to obtain the mean and variance features for latent distribution z, or $q_\phi(z|x,y)$. Similarly, in the decoder, the latent features are forwarded to two fully connected layers with output sizes of 512 and 256. Finally, we add another self-attention layer with a linear module to obtain enhanced features and obtain the reconstructed 9D object pose vector. The object pose vector is then transformed into a 6D object pose.

The second branch is utilized for the reconstruction of the input hand-object image. The second branch consists of six convolutional layers with batch normalization and leaky ReLU activation in the encoder. The features are then forwarded to two fully connected layers and a self-attention module to obtain the mean and variance of the latent distribution. The complete process is illustrated in Figure 5. The input to the second branch or VAE is of size $256\times32\times32$. Each convolution layer in the encoder consists of a convolution block, batch normalization, and activation layer. The first convolution layer outputs 64 features with kernel size 1 and zero padding, resulting in $64\times32\times32$ features. The kernel size and padding for the next layer are changed to 3 and 1, respectively, and the output feature size is $32\times32\times32$. The parameters of the first and second convolutions are repeated two more times, resulting in features of size $2\times32\times32$. Finally, the features are linearized and downsampled to 1×1024. The features were further reduced to 256 by applying two fully connected layers, and the features were enhanced via self-attention. The attended features are used for the latent distribution z computation. Finally, in the decoder, the process is repeated with 2D convolution transpose to obtain an output image of size $256\times256\times3$. The cross-model autoencoders are trained in a manner similar to the work in (Spurr et al., 2018). Each sample is trained in a data-pairs manner so that the model outputs a single latent space with multiple modalities. Finally, we obtain the modality based on the selection of the decoder. In this work, during inference, we select the encoder from branch 2 and the decoder from branch 1 to estimate the single object pose based on the input image and neglect the other decoder from branch 2 because it is not necessary for this application. The loss function of the overall pose estimation network is computed individually for both branches, summed, and updated during training. During experimentation, we added a conditional variable to branch 1 as in Figure 1. The conditional variable is a one-hot coded vector of the object ID number of known objects.

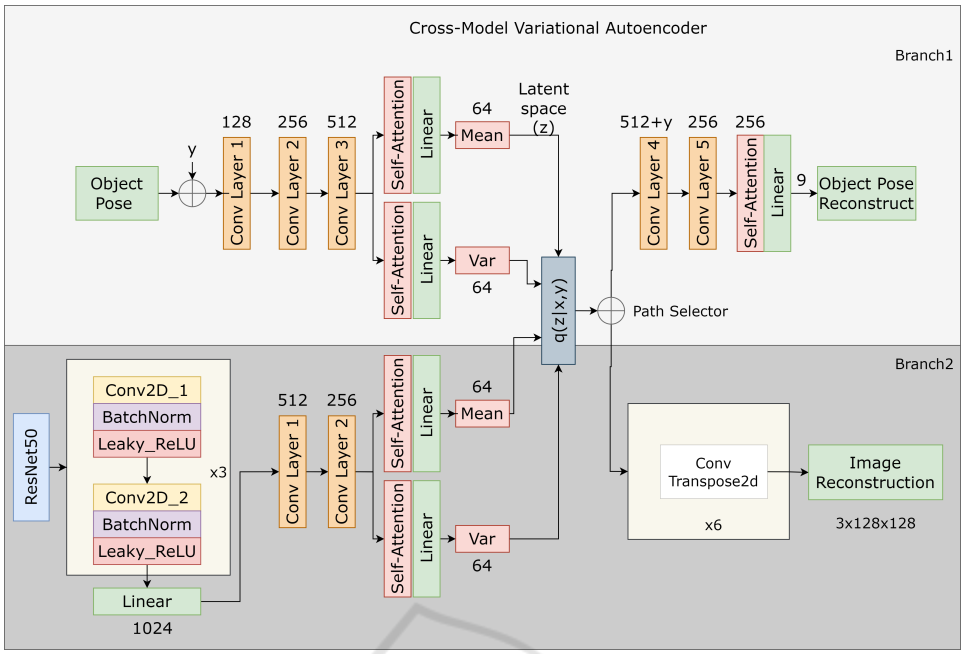

Figure 5: The basic structural process of the cross-model conditional variational autoencoder. Each autoencoder consists of an encoder, a low dimensional latent distribution, and a decoder. The latent space is shared between the models and paths are switched depending on the decoder.

3.5 Object Pose Representation

The input is an object pose with a translation T and rotation R component. Different representations of rotation components exist, such as rotation matrices, quaternions, and Euler angles. Singularities and the antipodal issue for regression are constraints of Euler angles and quaternions. Additionally, (Zhou et al., 2019) has shown that any rotation representation in 3D with fewer than five dimensions is discontinuous and more difficult to learn. As a result, we use the Gram-Schmidt process to take advantage of the orthogonal features of a rotation matrix and create an orthonormal basis from two vectors, as shown in Eq.6. The third column of the rotation matrix component is unnecessary. The rotation matrix R is reconstructed by multiplying the first and second column vectors, e_1 and e_2, respectively. The object pose input for CVAE is the translation vector T and the first two columns from the rotation matrix R.

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} = \begin{pmatrix} \vec{e_1} & \vec{e_2} & \vec{e_1} \times \vec{e_2} \end{pmatrix} \quad (6)$$

$$x = \begin{pmatrix} t_1 & t_2 & t_3 & r_{11} & r_{21} & r_{31} & r_{12} & r_{22} & r_{32} \end{pmatrix} \quad (7)$$

Once the object pose x' is reconstructed from CVAE, the 6D rotation representation is converted to a rotation matrix by applying the abovementioned process.

The total loss function is a combination of the hand pose estimation loss and the object pose estimation loss. The hand pose estimation loss is the sum of the heatmap loss, pose-shaped regression loss, BMC loss, and MANO loss, as shown in Eq. 3. The object pose estimation loss is the sum of the two VAE losses, as shown in Eq. 4. The β_{cvae} in the CVAE loss can either be fixed or varied.

$$L_{Total} = L_{handloss} + \lambda_{branch_1} L(\theta,\phi)_{branch_1} + \lambda_{branch_2} L(\theta,\phi)_{branch_2} \quad (8)$$

where the β_{cvae} parameter is set to 0.01 in the initial stages and is modified during the experimentation. The λ_{branch_1} and λ_{branch_2} is the loss scaling factors for training the proposed end-to-end model, which is empirically computed. The β_{cvae} parameter is modified with respect to the epoch number as suggested in (Higgins et al., 2017). The value of β_{cvae} is set to 0.5 until epoch 10, which changes to 0.1 from epoch 10 to epoch 20. From epochs 20 to 40, the value is set to 0.01, and the reduction process is repeated every 20 epochs until the β_{cvae} reaches 0.0001.

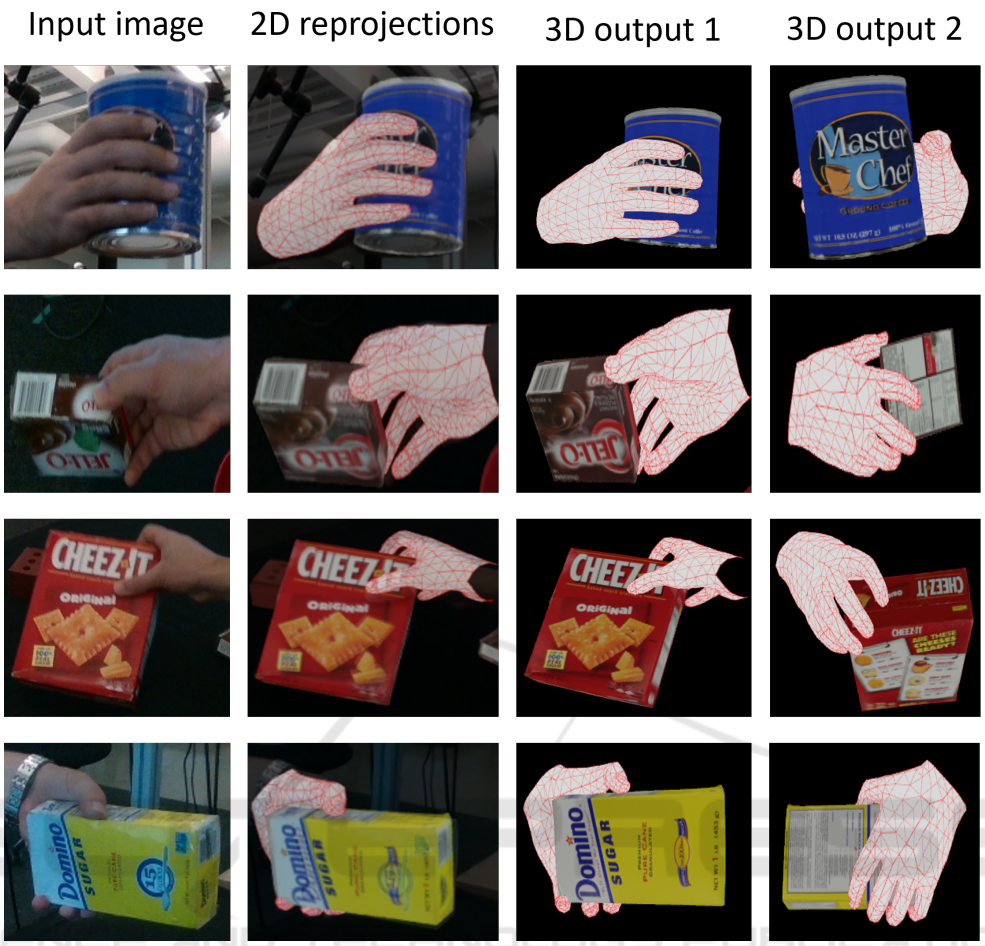

Figure 6: The visualization of the results from the proposed architecture. The first image in each row is the input image, the second image is the output reprojections on 2D images, and the last two images are 3D meshs of hand and object0 from two different views.

4 EXPERIMENTS

4.1 Implementation Details

The introduced framework is entirely implemented in PyTorch (Paszke et al., 2019). The complete model is trained in an end-to-end manner with Adam optimizer (Kingma and Ba, 2015). We intiliatize the ResNet50 (He et al., 2016) architecture with pre-trained weights and shared input features for the hand pose estimation and object pose estimation networks. The model is trained with an initial learning rate of 1e-4 with a decay factor for every 20 epochs until we reach 100 epochs and a constant rate until 200 epochs with a batch size of 128. The input images are resized to $256 \times 256 \times 3$, and we perform simple data augmentation techniques such as scaling, color jitter, brightness, and contrast. The input image is a closely cropped region of hands and objects with bounding box information provided by the datasets.

4.2 Datasets and Evaluation Metrics

ObMan (Hasson et al., 2019). A large-scale synthetic dataset contains various hand-grasping poses with high-quality meshes on a variety of imported ShapeNet objects but for experimentation, we con-

Table 1: The architecture is trained in two different ways. The hand pose and shape regression is trained individually and end-to-end with object pose reconstruction on DexYCB dataset.

Architecture	Hand MJE ($mm \downarrow$)
Joint Training	13.1
Independent Hand	12.1

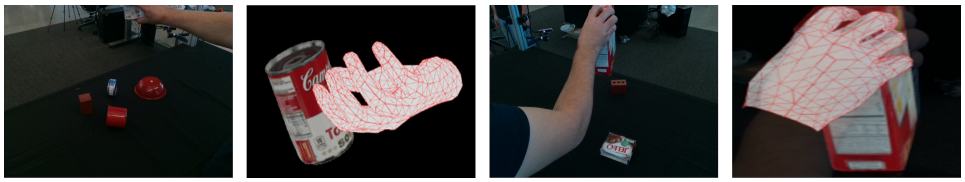

Figure 7: The qualitative reconstructed outputs of failure scenarios.

Table 2: Comparison of the proposed architecture to the state-of-the-art methods on DexYCB dataset.

Method	Hand MJE(cm↓)	Object MCE(cm↓)	Interaction PD(mm↓)
(Hasson et al., 2019)	1.76	-	-
(Hasson et al., 2021)	1.88	5.25	0.79
(Tse et al., 2022)	1.53	-	-
(Wang et al., 2022)	1.27	3.26	0.67
Ours	1.21	3.02	0.60

Table 3: Comparison to the state-of-the-art methods on ObMan dataset.

Method	Hand MJE(mm↓)	Object CD(mm↓)	Interaction PD(mm↓)
(Hasson et al., 2019)	11.6	637.8	9.2
(Tse et al., 2022)	9.1	385.7	7.4
(Chen et al., 2022)	-	338	6.6
Ours	8.7	315.6	6.8

sider 8 different objects. To generate plausible grasps between synthetic hands and objects, GraspIt software was utilized. From this dataset, we consider 80K samples for training and over 6K samples for testing.

DexYCB (Chao et al., 2021). The DexYCB dataset is a large-scale real dataset with over 580K RGB-D images from 10 human subjects manipulating 20 YCB objects. The dataset is captured from 8 Realsense cameras simultaneously at a rate of 30 fps with a resolution of 640×480. The evaluation setup consists of different scenarios with unseen subjects, unseen views, and grasping from which we select the default setup known as S0. The val/test split does not share anything except the sequences in the S0 setup. The dataset consists of videos of subjects grasping the ycb objects where the distance between hands and objects is quite large at the beginning and some times not visible in the scene. For a fair comparison, we follow a process similar to (Wang et al., 2022) to neglect images where the distance between hands and objects is greater than 1*cm* to assume contact between them.

Evaluation Metrics. The standard evaluation metrics of hand pose estimation are mean hand joint error (MJE). For object pose estimation, we compute the mean corner error (MCE). To measure the reconstruction quality of joint meshes we also compute the penetration depth (PD) in *mm* to check for plausible collisions between hands and objects similar to (Wang et al., 2022) for a fair comparison of the DexYCB dataset. Similarly for the ObMan dataset, we use mean joint error (MJE) for hand pose evaluation, chamfer distance (CD) in *mm* for object pose estimation, and penetration depth (PD) for hand-object interactions as proposed in (Hasson et al., 2019) for fair comparison.

4.3 Results

We test the performance of the proposed hand-object framework on the ObMan and the DexYCB datasets. Few reconstructed qualitative samples can be observed in Figure 6. The first column consists of input images to the network which is a closely cropped hand object region. The second column represents the output reprojections on 2D images. The last two columns are the output 3D hand and object pose from two different views. Two scenarios where the proposed architecture achieved low or failed to reconstruct are shown in Figure 7. From this we can observe that, the proposed algorithm fails to reconstruct the right poses when partial hand is visible in the image or when the objects are highly occluded by the hands.

Hand Only Experiments. Although we propose the joint learning framework, we evaluate the performance of training the hand pose estimation network

Table 4: Ablation study and the effect of attention layers.

Methods	Hand MJE ($cm \downarrow$)	Object MCE ($cm \downarrow$)
w/out Attention-Hand	1.76	-
With Attention-Hand	1.27	-
With Attention-Hand and BMC Loss	1.21	-
w/out y and w/out Attention-Object	-	9.8
with y and w/out Attention-Object	-	4.8
with Attention-Object	-	3.02

individually. From the experiments, we noticed that there is a slight performance improvement in the hand pose estimation when trained independently. The mean hand joint errors (MJE) for both experiments are mentioned in Table 1. From the experiments, we can observe that the hand pose estimation achieves better outcomes when trained independently.

Comparison with the State-of-the-Art Methods. The model trained on the DexYCB (Chao et al., 2021) dataset and the comparison results are shown in Table 2. As shown, our method achieved a mean hand joint error of 12.1 mm. Recent works such as (Chen et al., 2022; Chen et al., 2023) also achieved better results on the DexYCB dataset but the results are represented in chamfer distance rather than the traditional MJE and MCE so, we could not compare their methods. The results on the ObMan (Hasson et al., 2019) dataset are represented in Table 3. We can observe that the hand pose estimation achieved a mean joint error of 8.7*mm* and object chamfer distance is 315.6*mm*. Although the chamfer distance achieved state-of-the-art performance, the interaction parameter is a bit higher than in previous works. To this end, adding a further refinement stage would improve the interaction parameter further.

4.4 Ablation Study

The ablation study gives a deeper understanding of the architecture and the importance of each block in the proposed network. To study the network, we consider the DexYCB (Chao et al., 2021) dataset. The significance of each block can be observed in Table 4. For hand mesh reconstruction, we first use just the ResNet block and regress the pose and shape parameters for the MANO (Romero et al., 2017) model. From that, we obtain the hand mean joint error of 1.76*cm* and after adding the hand pose shape regression network with attention blocks the error is reduced to 1.27*cm* and it is further improved to 1.21*cm* by using BMC loss during training. Similarly, conduct these experiments for object reconstruction block where the conditional variable y and attention mechanism are modified. At first, we consider training without the conditional variable y and attention mechanism and achieve object MCE of 9.8*cm* which is quite high. Further, we added variable y and noticed a huge improvement resulting in 4.8*cm* object MCE. Finally, by considering all the parameters and proper hyperparameter tuning we achieve object MCE of 3.02*cm*. In adddition to that, we test the real-time usability of the architecture by estimating the frames per second. From the experimentation, we achieve around 19 fps.

5 CONCLUSION

In this work, we propose a joint learning framework for hand-object pose estimation. The features from a single backbone are shared between the hand pose estimation network and the object pose estimation network. We propose a unique cross-model conditional variational autoencoder for 6D object pose estimation via multi-model pair learning. We trained the architecture on the DexYCB and the ObMan open-source large-scale dataset and achieved good performance on hand pose estimation and better than state-of-the-art performance on object pose estimation. The performance of interaction parameter needs further attention and we are currently aiming to improve it by adding a refinement stage.

REFERENCES

Bao, J., Chen, D., Wen, F., Li, H., and Hua, G. (2017). Cvae-gan: Fine-grained image generation through asymmetric training. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2764–2773.

Boukhayma, A., Bem, R. d., and Torr, P. H. (2019). 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.

Chao, Y.-W., Yang, W., Xiang, Y., Molchanov, P., Handa, A., Tremblay, J., Narang, Y. S., Van Wyk, K., Iqbal, U., Birchfield, S., Kautz, J., and Fox, D. (2021). DexYCB: A benchmark for capturing hand grasping of objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chen, D., Li, J., and Xu, K. (2020). Learning canonical shape space for category-level 6d object pose and size estimation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11970–11979.

Chen, Z., Chen, S., Schmid, C., and Laptev, I. (2023). gSDF: Geometry-Driven signed distance functions for 3D hand-object reconstruction. In *CVPR*.

Chen, Z., Hasson, Y., Schmid, C., and Laptev, I. (2022). Alignsdf: Pose-aligned signed distance fields for hand-object reconstruction. *ArXiv*, abs/2207.12909.

Doosti, B., Naha, S., Mirbagheri, M., and Crandall, D. J. (2020). Hope-net: A graph-based model for hand-object pose estimation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6607–6616.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929.

Garcia-Hernando, G., Yuan, S., Baek, S., and Kim, T.-K. (2018). First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 409–419.

Ge, L., Ren, Z., Li, Y., Xue, Z., Wang, Y., Cai, J., and Yuan, J. (2019). 3d hand shape and pose estimation from a single rgb image. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10825–10834.

Hasson, Y., Varol, G., Schmid, C., and Laptev, I. (2021). Towards unconstrained joint hand-object reconstruction from rgb videos. *2021 International Conference on 3D Vision (3DV)*.

Hasson, Y., Varol, G., Tzionas, D., Kalevatykh, I., Black, M. J., Laptev, I., and Schmid, C. (2019). Learning joint reconstruction of hands and manipulated objects. In *CVPR*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Higgins, I., Matthey, L., Pal, A., Burgess, C. P., Glorot, X., Botvinick, M. M., Mohamed, S., and Lerchner, A. (2017). beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*.

Höll, M., Oberweger, M., Arth, C., and Lepetit, V. (2018). Efficient physics-based implementation for realistic hand-object interaction in virtual reality. In *Proc. of Conference on Virtual Reality and 3D User Interfaces*.

Hu, Y., Hugonot, J., Fua, P. V., and Salzmann, M. (2019). Segmentation-driven 6d object pose estimation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3380–3389.

Ivanovic, B., Leung, K., Schmerling, E., and Pavone, M. (2021). Multimodal deep generative models for trajectory prediction: A conditional variational autoencoder approach. *IEEE Robotics and Automation Letters*, 6(2):295–302.

Karunratanakul, K., Yang, J., Zhang, Y., Black, M. J., Muandet, K., and Tang, S. (2020). Grasping field: Learning implicit representations for human grasps. *2020 International Conference on 3D Vision (3DV)*, pages 333–344.

Kehl, W., Manhardt, F., Tombari, F., Ilic, S., and Navab, N. (2017). Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1530–1538.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. *CoRR*, abs/1312.6114.

Kulon, D., Güler, R. A., Kokkinos, I., Bronstein, M. M., and Zafeiriou, S. (2020). Weakly-supervised mesh-convolutional hand reconstruction in the wild. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4989–4999.

Labb'e, Y., Carpentier, J., Aubry, M., and Sivic, J. (2020). Cosypose: Consistent multi-view multi-object 6d pose estimation. In *ECCV*.

Li, S., Wang, H., and Lee, D. (2020). Hand pose estimation for hand-object interaction cases using augmented autoencoder. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 993–999.

Liu, S., Jiang, H., Xu, J., Liu, S., and Wang, X. (2021). Semi-supervised 3d hand-object poses estimation with interactions in time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Moon, G., Chang, J., and Lee, K. M. (2018). V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mueller, F., Bernard, F., Sotnychenko, O., Mehta, D., Sridhar, S., Casas, D., and Theobalt, C. (2018). Ganerated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*.

Mueller, F., Mehta, D., Sotnychenko, O., Sridhar, S., Casas, D., and Theobalt, C. (2017). Real-time hand tracking under occlusion from an egocentric rgb-d sensor. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 1284–1293.

Ortenzi, V., Cosgun, A., Pardi, T., Chan, W. P., Croft, E. A., and Kulić, D. (2021). Object handovers: A review for robotics. *IEEE Transactions on Robotics*, 37:1855–1873.

Park, J., Oh, Y., Moon, G., Choi, H., and Lee, K. M. (2022). Handoccnet: Occlusion-robust 3d hand mesh estimation network. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. Curran Associates Inc., Red Hook, NY, USA.

Peng, S., Liu, Y., Huang, Q., Bao, H., and Zhou, X. (2019). Pvnet: Pixel-wise voting network for 6dof pose estimation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4556–4565.

Piumsomboon, T., Clark, A., Billinghurst, M., and Cockburn, A. (2013). User-defined gestures for augmented reality. In *CHI '13 Extended Abstracts on Human*

Factors in Computing Systems, CHI EA '13, page 955–960, New York, NY, USA. Association for Computing Machinery.

Rad, M. and Lepetit, V. (2017). Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3848–3856.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic back-propagation and variational inference in deep latent gaussian models. *ArXiv*, abs/1401.4082.

Romero, J., Tzionas, D., and Black, M. J. (2017). Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):245:1–245:17.

Sohn, K., Lee, H., and Yan, X. (2015). Learning structured output representation using deep conditional generative models. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Spurr, A., Iqbal, U., Molchanov, P., Hilliges, O., and Kautz, J. (2020). Weakly supervised 3d hand pose estimation via biomechanical constraints. *ArXiv*, abs/2003.09282.

Spurr, A., Song, J., Park, S., and Hilliges, O. (2018). Cross-modal deep variational hand pose estimation. In *CVPR*.

Tekin, B., Bogo, F., and Pollefeys, M. (2019). H+o: Unified egocentric recognition of 3d hand-object poses and interactions. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4506–4515.

Tekin, B., Sinha, S. N., and Fua, P. V. (2018). Real-time seamless single shot 6d object pose prediction. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 292–301.

Tse, T. H. E., Kim, K. I., Leonardis, A., and Chang, H. J. (2022). Collaborative learning for hand and object reconstruction with attention-guided graph convolution. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1654–1664.

Vaswani, A., Shazeer, N. M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *ArXiv*, abs/1706.03762.

Wang, R., Mao, W., and Li, H. (2022). Interacting hand-object pose estimation via dense mutual attention. *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5724–5734.

Xiang, Y., Schmidt, T., Narayanan, V., and Fox, D. (2018). Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *ArXiv*, abs/1711.00199.

Yang, W., Paxton, C., Mousavian, A., Chao, Y.-W., Cakmak, M., and Fox, D. (2021). Reactive human-to-robot handovers of arbitrary objects. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE.

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., and Kim, T.-K. (2018). Depth-based 3d hand pose estimation: From current achievements to future goals. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2636–2645.

Zhang, X., Li, Q., Zhang, W., and Zheng, W. (2019). End-to-end hand mesh recovery from a monocular rgb image. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2354–2364.

Zhou, Y., Barnes, C., Lu, J., Yang, J., and Li, H. (2019). On the continuity of rotation representations in neural networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5738–5746.

Zhou, Y., Habermann, M., Xu, W., Habibie, I., Theobalt, C., and Xu, F. (2020). Monocular real-time hand shape and motion capture using multi-modal data. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5345–5354.

Zimmermann, C. and Brox, T. (2017). Learning to estimate 3d hand pose from single rgb images. In *IEEE International Conference on Computer Vision (ICCV)*. https://arxiv.org/abs/1705.01389.