Cross-Constrained Progressive Inference for 3D Hand Pose Estimation with Dynamic Observer-Decision-Adjuster Networks

Zhehan Kan[1], Xueting Hu[1], Zihan Liao[1], Ke Yu[1], Zhihai He[1,2]*

[1]Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen, China
[2]Pengcheng Laboratory, Shenzhen, China
{kanzh2021,huxt2022,liaozh2020,yuk2020}@mail.sustech.edu.cn, hezh@sustech.edu.cn

Abstract

Generalization is very important for pose estimation, especially for 3D pose estimation, where small changes in the 2D images could trigger large structural changes in the 3D space. To achieve generalization, the system needs to have the capability of detecting estimation errors by cross-validating the projection consistency between the 3D and 2D spaces and adapting its network inference process based on this feedback. Current pose estimation is one-time feed-forward and lacks the capability to gather feedback and adapt the inference outcome. To address this problem, we propose to explore the concept of dynamic progressive inference based on an observer-decision-adjuster network design, where we train an observer to continuously detect the prediction error based on constraints matching as well as an adjuster to refine its inference outcome based on these constraints errors. Within the context of 3D hand pose estimation, we find that this observer-adjuster design is relatively unstable since the observer is operating in the 2D image domain while the adjuster is operating in the 3D domain. To address this issue, we propose to construct two sets of observers-adjusters with complementary constraints from different perspectives. They operate in a dynamic, sequential manner controlled by a decision network to progressively improve the 3D pose estimation. We refer to this method as Cross-Constrained Progressive Inference (CCPI). Our extensive experimental results on FreiHAND and HO-3D benchmark datasets demonstrate that the proposed CCPI method is able to significantly improve the generalization capability and performance of 3D hand pose estimation.

Introduction

Hand pose estimation aims to correctly detect and localize the hand joint points from 2D images and use positional relationships to infer corresponding hand poses. It is an important task in computer vision and plays a crucial role in various downstream applications, such as human-computer interactions (Cheng, Yang, and Liu 2016), 3D human modeling (Zuffi et al. 2017), and contactless operations (Song et al. 2019). Recently, with the development of deep neural network methods, remarkable progress has been made in hand pose estimation. For simple hand gestures, existing deep learning methods have achieved very accurate estimations of hand key points (Choi, Moon, and Lee 2020). However, 3D hand pose estimation remains very challenging, especially for complex gestures with overlapping hands, crossed fingers, or occluded objects.

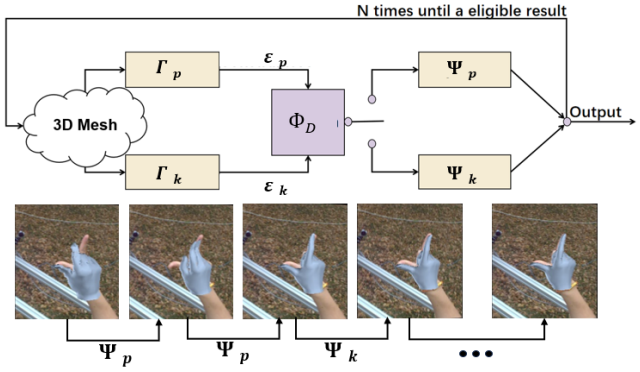


Figure 1: The top row is the diagram of CCPI, Γ_p and Γ_k are observer networks, Ψ_p and Ψ_k are adjuster networks, Φ_D is the decision network. The bottom row is one example of the dynamic sequential process.

We observe that generalization is one of the most important challenges in 3D hand pose estimation. Network models well-learned on the training set often suffer from significant performance degradation on test samples collected from different scenarios. For example, very accurate hand pose estimation is obtained on the training set; however, on the testing set, the hand pose edges are blurred and the hand morphology is missing. This is due to the complexity of the estimation process and the uncertainty of hand gestures in 3D hand pose estimation, where 3D pose structures are inferred from 2D images. Small changes in the 2D images could trigger large structural changes estimated in 3D space. Two different 3D hand poses with a large structural difference could yield very similar 2D images during the 3D-to-2D projection process. In this case, overlapping, intersection, and occlusion of hand key points are often difficult to fully recover.

Notably, to achieve successful generalization for 3D hand pose estimation, the system needs to have the capability of detecting estimation errors by cross-validating the projec-

*Corresponding author.

tion consistency between the 3D and 2D spaces. For example, in this work, once we have obtained the estimated 3D pose, we can synthesize the 3D hand graphics model based on these 3D keypoints, and then project this 3D graphics model into the 2D image and check the consistency between the projected 3D graphics model and the original image. Inconsistency between them indicates incorrect pose estimation. This inconsistency error can then be used to adjust the network inference process to refine the 3D pose estimation. Current pose estimation (Liu et al. 2021) is one-time feed-forward, where the learned network is executed once to regress pose estimation results. Clearly, it lacks the capability to gather feedback and adapt the inference outcome.

To address this problem, as illustrated in Figure 1, we propose to explore the concept of dynamic progressive inference, where we train an observer to continuously sense the prediction errors based on constraints matching, and then an adjuster network to refine the inference outcome based on these constraints errors. Within the context of 3D hand pose estimation, we find that this observer-adjuster design is relatively unstable since the observer is operating in the 2D image domain while the adjuster is operating in the 3D domain. To address this issue, we propose to construct two sets of observers-adjusters with complementary constraints from different perspectives. These two sets of constraints allow us to cross-check the prediction performance of these two sets of observer-adjuster networks. They operate in a dynamic sequential manner controlled by a decision network to progressively improve the 3D pose estimation. Our extensive experimental results on FreiHAND and HO-3D benchmark datasets demonstrate that the proposed CCPI method is able to significantly improve the generalization capability and performance of 3D hand pose estimation by up to 9.7% over the current state-of-the-art methods.

The rest of the paper is organized as follows. Section 2 reviews existing methods closely related to our work. The proposed CCPI method is presented in Section 3. Section 4 provides the experimental results, performance comparisons, and ablation studies. Section 5 concludes the paper.

Related Work and Summary of Contributions

In this section, we review related works on 3D hand pose estimation, hand pose refinement, iterative decision for visual tasks and progressive inference.

(1) 3D hand pose estimation. After the publication of hand object interaction benchmark datasets such as HO-3D (Hampali et al. 2020) and Freihand (Zimmermann et al. 2019), a number of researches have been conducted on these datasets to recover 3D hand poses from 2D images. This problem is inherently an ill-posed estimation problem since the depth and structural information are lost during the 3D-to-2D projection process. Recently, deep learning-based approaches have been developed for 3D pose estimation. Mueller *et al.*(Mueller et al. 2018) addressed the issue of insufficient annotated data by utilizing generative methods to convert synthetic training data into realistic training images Yang *et al.*(Yang and Yao 2019) introduced disentangled variational autoencoders to handle different pose variations in images. Recent methods (Park et al. 2022; Ye, Gupta, and Tulsiani 2022; Hampali et al. 2022; Christen et al. 2022) have been focusing on hand-object interaction. Mueller *et al.*(Park et al. 2022) addresses the limitations of existing spatial attention-based methods. Yang *et al.*(Ye, Gupta, and Tulsiani 2022) explored the relationship between hand articulation and the object it interacts with to predict object shape based on hand pose and the input image.

(2) Hand pose refinement. Our work is related to hand pose refinement. Several approaches have been developed for refining hand pose estimation (Dibra et al. 2017; Ge et al. 2018; Tang, Yu, and Kim 2013; Deng et al. 2022; Yang et al. 2022; Chen et al. 2020). Tang *et al.*(Tang, Yu, and Kim 2013) introduced a pseudo-kinematic joint refinement algorithm to handle occlusions and noisy articulations. Dibra *et al.*(Dibra et al. 2017) developed a 3D hand pose estimation method incorporating a depth loss component and physical and collision regularizers, which accurately estimates 3D hand pose without annotated data. Ge *et al.*(Ge et al. 2018) aimed to enhance the accuracy of fingertip localization by using a PointNet network to refine initial estimates with neighboring points in the point cloud. Chen *et al.*(Chen et al. 2020) proposed a pose-guided structured region ensemble network (Pose-REN) with novel feature extraction and hierarchical fusion methods to iteratively refine hand pose estimation.

The iterative method of refinement has been tested on hand pose estimation. Yang *et al.*(Yang et al. 2022) proposed DIR-Net to refine the computational cost by dynamic iterations. Laskaridis *et al.*(Laskaridis et al. 2020) proposed SPINN, which uses collaborative device-cloud computing and progressive inference methods to provide fast and robust CNN inference in different environments. Debasis (Kundu 2008) discussed Bayesian inference for the unknown parameters of the progressively censored Weibull distribution.

In existing work, progressive inference is used to perform the prediction in a step-by-step manner, gradually refining the prediction as more data becomes available. In this work, our proposed progressive inference is significantly different since it is a dynamic observation-decision-adjustment process, aiming to achieve improved generalization capability for pose estimation.

(3) Major contributions. The major contribution of this work can be summarized as follows: (a) We have developed a new dynamic progressive inference approach based on dynamic observer-decision-adjuster network design to address the important generalization problem in 3D hand pose estimation. (b) We establish two different constraints, projection and Kinematic Chain Space (KCS) (Wandt and Rosenhahn 2019) constraints, based on which two sets of observers-adjusters are trained to sense prediction errors and adaptively adjust the results. (c) Our extensive experimental results demonstrates that our proposed cross-constrained progressive inference method is able to significantly improve the performance of 3D pose estimation. The proposed method can also be applied to other machine learning tasks.

Method

In this section, we present our Cross-Constrained Progressive Inference (CCPI) for 3D hand pose estimation.

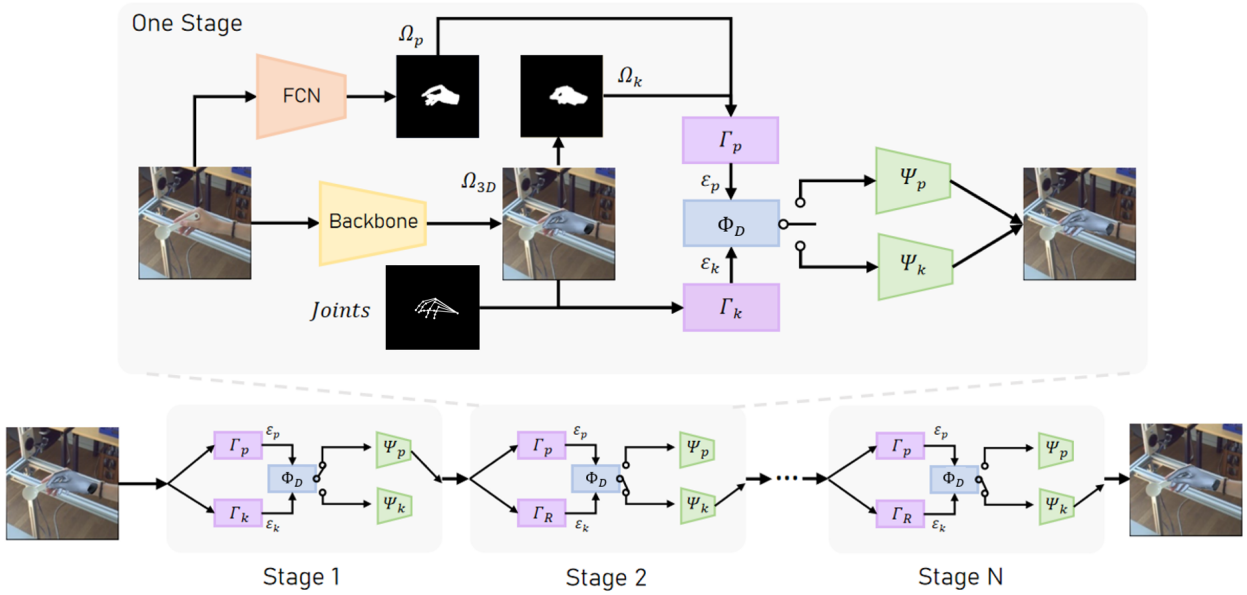


Figure 2: Pipeline of Cross-Constrained Progressive Inference (CCPI). Stage 1, Stage 2, and Stage N show the progressive process. The details of one stage are presented. Observers Γ_p and Γ_k use a reprojected mask and segmented mask for the KCS condition to sense the prediction errors in the 3D hand pose estimation results. Decision networks decide which adjuster to choose according to observation results. The difference in each stage reflects on which adjuster to choose.

Overview

The overall framework of the proposed CCPI method is illustrated in Figure 2. The baseline 3D hand pose estimation network Φ_0 predicts the 3D hand mesh Ω_{3D} from the input image I. We introduce two sets of constraints: the projection constraint and the relational KCS constraint. Based on these two constraints, we construct two observers Γ_p and Γ_k to sense the prediction errors in the 3D hand pose estimation results by verifying if these two constraints are satisfied. The corresponding errors are referred to constraint errors $\mathcal{E}_p$ and $\mathcal{E}_k$. For each observer, we construct an adjuster network, Ψ_p or Ψ_k, which learns to adaptively adjust the 3D hand pose estimation result with guidance by the constraint error. In this way, we construct two sets of observer-adjuster networks. They observe the 3D pose estimation inference process from two different perspectives so as to provide complementary guidance. We recognize that this cross-constrained observer-adjuster design is necessary since the observer operates in the 2D image domain while the adjuster operates in the 3D keypoints domain. This cross-constraint design can help us address this ill-posed problem. As illustrated in Figure 2 (bottom), this observer-adjuster operation can be performed with multiple stages to progressively improve the accuracy of the 3D pose estimation. At each stage, a decision network is trained to make the following decision: to use the observer-adjuster pair $[\Gamma_p, \Psi_p]$ or the pair $[\Gamma_k, \Psi_k]$. The decision depends on the current status of the 3D pose estimation results, specifically which observed constraint error is more significant and which adjuster is more effective for the current status. In the following sections, we will explain the proposed CCPI algorithm, elucidating its core principles and framework with comprehensive detail.

Cross-Constrained Observer Networks

In this section, we explain how to construct the cross-constrained observer networks based on two constraint errors: (1) projection constraint error and (2) Kinematic Chain Space (KCS) constraint error, to sense the prediction error during 3D hand pose estimation.

(1) Projection constraint error. We use an image segmentation network Φ_g (Long, Shelhamer, and Darrell 2015) to segment the hand region from the 2D image and the corresponding image region masked from the background, and the segmented region is denoted by Ω_s. When doing this segmentation, we can use the 3D pose estimation as a reference, for example, to locate the hand. In the meantime, we render the 3D mesh estimation Ω_{3D} into a 3D hand graphics model and project this model into the 2D space. The corresponding masked image region is denoted by Ω_p. If the 3D hand pose estimation is accurate, the hand mask Ω_s obtained from image segmentation should match well with Ω_p obtained from the 2D projection of the 3D hand model. This leads to the first 3D model projection constraint. We train a network to characterize the difference between these two mask images and generate the so-called projection constraint error

$$\mathcal{E}_p = \Gamma_p[\Omega_p, \Omega_s]. \tag{1}$$

Figure 3 shows five examples of incorrect estimation results detected by the projection constraint.

(2) Kinematic Chain Space constraint error. The KCS matrix is a well-established representation of human pose that characterizes the skeletal structure of the hand pose. It contains information about joint angles and bone lengths and can be derived by means of two matrix multiplications. Within this framework, each bone b_k is defined as the vector

Figure 3: Examples of error estimation results adjusted by the projection constraint.

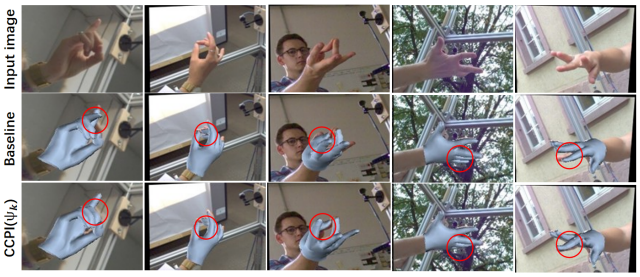


Figure 4: Examples of incorrect estimation results adjusted by the KCS constraint.

spanning the r-th and t-th joints, thereby characterizing the skeletal structure of the pose,

$$b_k = p_r - p_t. \tag{2}$$

The bones b are concatenating to a matrix B, which can be defined as:

$$B = (b_1, b_2, ..., b_b). \tag{3}$$

Multiplying B with its transpose, we compute the so-called KCS matrix:

$$\phi = B^T B = \begin{bmatrix} {l_1}^2 & \cdot & \cdot & \cdot \\ \cdot & {l_2}^2 & \cdot & \cdot \\ \cdot & \cdot & {l_3}^2 & \cdot \\ \cdot & \cdot & \cdot & {l_b}^2 \end{bmatrix} \tag{4}$$

The KCS matrix ϕ in skeletal animation has bone lengths l on the diagonal and scaled angular representations on the off-diagonal, which are obtained from inner products of bone vectors. It is crucial for inverse kinematics algorithms to determine joint angles for desired poses.

In this work, we use this KCS condition of hand poses to define the KCS constraint error. Let J be the set of joints obtained from the pose estimation. V represents the mesh vertices of the reconstructed 3D hand model. Based on this KCS condition, we train a network to perform joint analysis of hand pose estimation results to derive the KCS constraint error $\mathcal{E}_k$: (J, V) as follows:

$$\begin{aligned} \mathcal{E}_k &= \Gamma_k(KCS(J), V) \\ &= \mathbf{MLP_1}(KCS(J), \mathbf{MLP_2}(V)). \end{aligned} \tag{5}$$

Here, MLP_1 and MLP_2 are two multi-layer perceptron (MLP) networks, and Γ_k is the observer network. Figure 4 shows five examples of incorrect estimation results detected by the KCS constraint.

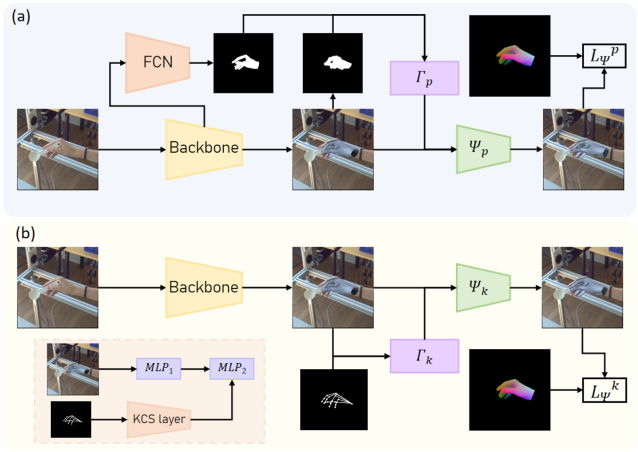


Figure 5: Joint training for observer and adjuster. (a) for Ψ_p and Γ_p, (b) for Ψ_k and Γ_k.

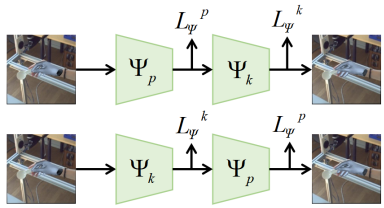


Figure 6: Coupled training of two sets of observer-adjuster networks in two different orders.

Cross-Constrained Adjuster Network Design

In the above section, we have derived the projection constraint error $\mathcal{E}_p$ and the KCS constraint error $\mathcal{E}_k$ to characterize the current 3D pose estimation result. In this section, we explain how to construct and train the corresponding adjuster networks Ψ_p and Ψ_k. In the following, we take Ψ_p as an example. The input to the adjuster network Ψ_p includes the current estimation results (J, V), the projection constraint error, and the image feature $\mathbf{f}$. It aims to adjust the pose estimation to $(J', V') = \Psi_p[(J, V), \mathcal{E}_p, \mathbf{f}]$. Let $(\bar{J}, \bar{V})$ be the ground-truth joints and mesh vertices of the hand pose. Thus, the loss function for training the adjuster network Ψ_p for the projection constraint error is given by

$$\mathcal{L}_\Psi^p = ||\Psi_p[(J, V), \mathcal{E}_p, \mathbf{f}] - (\bar{J}, \bar{V})||_2. \tag{6}$$

Similarly, for the KCS adjuster network Ψ_k, its training loss is given by

$$\mathcal{L}_\Psi^k = ||\Psi_k[(J, V), \mathcal{E}_k, \mathbf{f}] - (\bar{J}, \bar{V})||_2. \tag{7}$$

Notice that, the projection and KCS constraint errors $\mathcal{E}_p$ and $\mathcal{E}_k$ are obtained by the observer networks Γ_p and Γ_k. So, the observer networks and the adjuster networks are jointly trained, as illustrated in Figure 5.

These two sets of observer-adjuster networks analyze the 3D pose estimation results from two different perspectives. In our experiments, we find that it is more efficient to perform coupled training between these two sets of observer-adjuster networks. As illustrated in the figure. 6, during joint

	PA-MPVPE↓	PA-MPJPE↓	F@5 mm↑	F@15 mm↑
w/o	5.8	5.9	0.771	0.985
w/	**5.6**	**5.7**	**0.783**	**0.987**

Table 1: Performance of w/o couple train and w/ couple train on FreiHAND.

training, we concatenate these two sets of observer-adjuster networks in two different orders. Table 1 compares the performance with and without coupled training on the FreiHAND dataset. We can see that the proposed coupled training is able to improve the performance significantly.

Dynamic Observer-Decision-Adjuster Design

In the above section, we have developed two sets of observer-adjuster networks, $\{\Gamma_p, \Psi_p\}$ and $\{\Gamma_k, \Psi_k\}$, based on the projection and KCS constraint, respectively. Once successfully trained, during the actual 3D hand pose estimation process, they will be able to adjust the estimation results in different manners and achieve different outcomes. During experiments, we find that (1) it is necessary to perform multiple stages of observer-adjuster of the pose estimation results. (2) At different stages, it is also necessary to select different sets of observer-adjuster networks, depending on the error pattern. This leads to a dynamic multi-stage error observation and pose adjustment process. To decide which set of observer-adjuster networks needs to be used at each stage, we train a decision network Φ_D to connect all observers and adjusters together and build an observer-decision-adjuster structure.

Our purpose is to train the decision network Φ_D that models the input to the decision network including the projection and KCS constraint errors $[\mathcal{E}_p, \mathcal{E}_k]$. The output is the probability of selecting one of these two sets of observer-adjuster networks $\mathbf{p}(\{\Gamma_p, \Psi_p\})$ and $(\mathbf{p}\{\Gamma_k, \Psi_k\})$. When training the decision network, we use the following loss function, which aims to minimize the overall pose estimation error during the past N stages of adjustments.

$$
\begin{aligned}
\mathbf{E}(N) &= \sum_{i=0}^{N} \gamma^i \mathcal{R}^i \\
\mathcal{R}^i &= {\Phi_D}^i(\mathcal{E}_p, \mathcal{E}_k|\Psi_p) \\
&\quad \times ||\Psi_p(\mathcal{E}_p, (J^i, V^i); \mathbf{f}) - (\bar{J}, \bar{V})||_2 \\
&\quad + {\Phi_D}^i(\mathcal{E}_p, \mathcal{E}_k|\Psi_k) \\
&\quad \times ||\Psi_k(\mathcal{E}_k, (J^i, V^i); \mathbf{f}) - (\bar{J}, \bar{V})||_2)
\end{aligned} \tag{8}
$$

Here, the γ represents the attenuation factor. Φ_D^i represents the decision network output stage i, which are the probabilities in selecting one of these two sets of observer-adjuster networks. $\mathcal{R}^i$ represents the average error after adjustment by selecting the corresponding observer-adjuster network.

In our CCPI method, we use the decision network to select the set of observer-adjuster networks at each stage. This dynamic, sequential selection of these two sets of observer-adjuster networks is more efficient than combining them together. Figure 7 shows the training losses for these two different cases. We can see that the training loss of our CCPI method with dynamic sequential selection is much smaller than that of these two being combined.

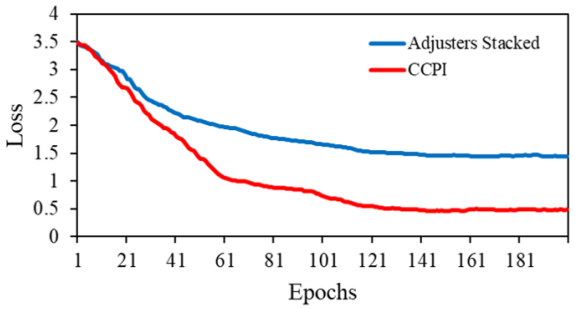


Figure 7: Loss of adjusters combined together and CCPI.

Experimental Results

In this section, we present experimental results, performance comparisons, and ablation studies to demonstrate the performance of our CCPI method.

Datasets

We conduct comparison and ablation experiments on the FreiHAND (Zimmermann et al. 2019) dataset and HO-3D (Hampali et al. 2020) dataset, both of which have been extensively used and contain challenging scenes.

FreiHAND: Freihand is an RGB dataset with hand pose and shape labels, which contains 32 different subjects, 130K training samples, and 4K evaluation samples. We train our model with all 130k training samples and test it with all the remaining evaluation samples. In performance evaluations, we use the metric of reconstruction error (PA-MPJPE/MPVPE), with F@5mm and F@15mm metrics utilized to report the accuracy both at the fine and coarse scale.

HO-3D: The HO-3D dataset contains over 66k training images and 11k test images from a total of 68 sequences with automatically generated annotations by algorithm. In evaluation, we introduce joint metric to evaluate our model on predicting key points of hand pose.

Implementation Details

For fair comparisons, we use Mesh Graphormer, METRO, I2LMEshNet, and Pose2Mesh as our backbone and follow the same training configuration as (Lin, Wang, and Liu 2021b), (Lin, Wang, and Liu 2021a), (Moon and Lee 2020) and (Choi, Moon, and Lee 2020). The model is trained with the Adam optimizer. We choose a batch size of 32 and an initial learning rate of 0.0001. More details are provided in the Supplemental Material.

Evaluation Metrics and Methods

MPJPE/MPVPE measures the mean per joint/vertex position error by Euclidean distance (mm) between the estimated and ground-truth coordinates. **F-Score** is the harmonic mean

Method	PA-MPVPE ↓	PA-MPJPE ↓	F@5 mm ↑	F@15 mm ↑
Hasson *et al.*(Hasson et al. 2020)	13.2	-	0.436	0.908
Boukhayma *et al.*(Boukhayma, Bem, and Torr 2019)	13.0	-	0.435	0.898
FreiHAND (Zimmermann et al. 2019)	10.7	-	0.529	0.935
Pose2Mesh (Choi, Moon, and Lee 2020)	7.8	7.7	0.674	0.969
I2LMEshNet (Moon and Lee 2020)	7.6	7.4	0.681	0.973
METRO (Lin, Wang, and Liu 2021a)	6.7	6.8	0.717	0.981
Mesh Graphormer (Lin, Wang, and Liu 2021b)	6.2	6.3	0.751	0.983
CCPI(Ours)	**5.6**	**5.7**	**0.783**	**0.987**
Performance Gain	**-0.6**	**-0.6**	**+0.032**	**+0.004**

Table 2: Comparison with the state-of-the-art methods on FreiHAND test-dev.

Method	Joint ↓	Mesh ↓	F@5 ↑	F@15 ↑
Pose2Mesh	12.5	12.7	44.1	90.9
Hasson *et al.*	11.4	11.4	42.8	93.2
I2LMeshNet	11.2	13.9	40.9	93.2
Hasson *et al.*	11.1	11.0	46.0	93.0
Hampali *et al.*	10.7	10.6	50.6	94.2
METRO	10.4	11.1	48.4	94.6
Liu *et al.*	10.2	9.8	52.9	95.0
HandOccNet	9.1	8.8	56.4	96.3
CCPI(Ours)	**8.7**	**8.5**	**58.4**	**96.8**
Performance Gain	**-0.4**	**-0.3**	**+2.0**	**+0.5**

Table 3: Comparison with the state-of-the-art methods on HO-3D test-dev.

	PA-MPVPE ↓	PA-MPJPE ↓	F@5 ↑	F@15 ↑
Baseline	6.2	6.3	0.751	0.983
+ $\mathbf{\Psi_k}$	6.0	6.1	0.760	0.984
+ $\mathbf{\Psi_p}$	5.9	6.0	0.765	0.985
CCPI	**5.6**	**5.7**	**0.783**	**0.987**

Table 4: Ablation study on FreiHAND.

between recall and precision between two meshes with respect to a specific distance threshold. F@5/F@15 corresponds to a threshold of 5mm/15mm.

Performance Results

In Table 2, we compare the performance of our CCPI method with the following state-of-the-art methods on the FreiHAND test set: the Hasson *et al.*(Hasson et al. 2020), Boukhayma *et al.*(Boukhayma, Bem, and Torr 2019), FreiHAND (Zimmermann et al. 2019), Pose2Mesh (Choi, Moon, and Lee 2020), I2LMEshNet (Moon and Lee 2020), METRO (Lin, Wang, and Liu 2021a), and Mesh Graphormer (Lin, Wang, and Liu 2021b) methods. We can see that our CCPI method outperforms the state-of-the-art method by 9.7%.

Table 3 shows the results of the challenging experiments on the HO-3D dataset. We compare the performance of our CCPI method with the following state-of-the-art methods: the Pose2Mesh (Choi, Moon, and Lee 2020), Hasson *et al.*(Hasson et al. 2020), I2LMeshNet (Moon and Lee 2020), Hasson *et al.*(Hasson et al. 2019), Hampali *et al.*(Hampali et al. 2020), METRO (Lin, Wang, and Liu 2021a), Liu *et al.*(Liu et al. 2021) and HandOccNet (Park et al. 2022) methods. Compared to the current best method, HandOccNet (Park et al. 2022), our CCPI method can still improve the average precision by 0.4, which shows that our method has a powerful ability to perform precise pose estimation on multi-person scenes with challenging scenarios of occlusion.

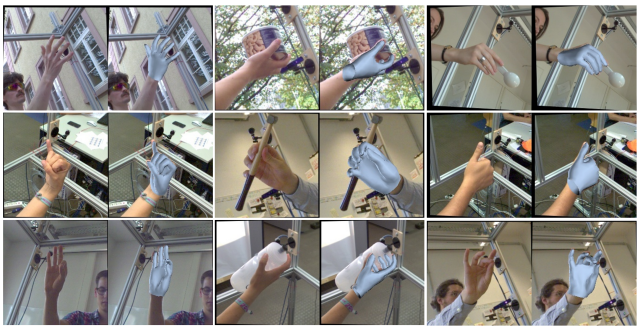


Figure 8: Nine results of our method on the FreiHAND test set. We selected the pictures of fingers bending, overlapping and being blocked. The left side of each pair of images is the original image, and the right side is the predicted mesh result.

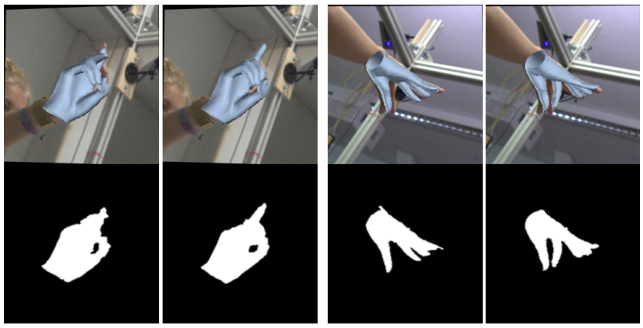
Figure 8 shows nine results of our CCPI method on the FreiHAND test set. We can see that our CCPI method can precisely complete the reconstruction of the hand, even when the fingers of the hand are crossed, overlapped, and partially occluded.

Ablation Studies

In order to systematically evaluate our method and investigate the contribution of each algorithm component, we use the Mesh Graphormer backbone to perform a number of ablation experiments on the FreiHAND dataset. Our algorithm has two major new components, the observer-adjuster

Method	PA-MPVPE ↓	PA-MPJPE ↓	F@5 mm ↑	F@15 mm ↑
Pose2Mesh	7.8	7.7	0.674	0.969
+ cross validation	9.1	9.3	0.564	0.961
I2LMeshNet	7.6	7.4	0.681	0.973
+ cross validation	8.6	8.5	0.594	0.967
METRO	6.7	6.8	0.717	0.981
+ cross validation	7.6	7.7	0.686	0.972
Mesh Graphormer	6.2	6.3	0.751	0.983
+ cross validation	7.1	7.2	0.694	0.978
CCPI	**5.6**	**5.7**	**0.783**	**0.987**
	6.3	**6.4**	**0.746**	**0.981**

Table 5: Ablation study of cross validation.



(a) (b)

Figure 9: Two examples with different adjusters. Subfigure (a) refers to smaller projection constraint errors. (b) for the example that the KCS constraint error is smaller.

network based on the projection constraint $[\Gamma_p, \Psi_p]$ and the observer-adjuster network based on the KCS constraint $[\Gamma_k, \Psi_k]$. In the first row of Table 4, we report the baseline (Mesh Graphormer) results. The second row shows the results with the Ψ_k, while the third row shows the results with the Ψ_p. The last row shows the final results with both Ψ_k and Ψ_p. We can clearly see that each algorithm component significantly contributes to the overall performance.

In Table 5, in order to demonstrate the robust generalizability of our method, we conduct cross-validation experiments. Specifically, we train our model on the HO3D dataset and test it on the FreiHAND dataset. The results demonstrate that the outcomes achieved by the CCPI method exhibit the highest accuracy while also minimizing the performance degradation attributed to dataset transfer. This substantiates the robust generalization capability of our proposed method.

Figure 9 shows two examples with different adjustment methods. (a) shows one example where the projection constraint error is small while the KCS constraint error is large. So the KCS-based adjuster Ψ_k is applied. (b) shows an opposite example where the projection constraint error is large while the KCS constraint error is small. So, in this case, the projection-based adjuster Ψ_p is applied. These two different adjustment methods are dynamically selected by the decision network at each stage of the estimation process.

Complexity Analysis

Table 6 shows the comparison between Mesh Graphormer and our CCPI method. The results show the parameters of CCPI just increased by 55% and the inference speed only increased by 62% with the accuracy significantly improved.

	Parameters (M)	Time(ms)
Mesh Graphormer	**128.05**	**91.94**
CCPI	198.74	149.05

Table 6: Complexity analysis ablation study of Mesh Graphormer on FreiHAND.

Conclusions

This work aims to address the generalization problem in 3D hand pose estimation by developing a dynamic observer-decision-adjuster network scheme for progressive pose estimation. It trains an observer to continuously detect the prediction error based on constraints matching and an adjuster to refine its inference outcome based on these constraints errors. We construct two sets of observer-adjuster networks with complementary constraints from different perspectives. They operate in a dynamic, sequential manner controlled by a decision network to progressively improve the 3D pose estimation. Our extensive experimental results on FreiHAND and HO-3D benchmark datasets demonstrate that our CCPI method is able to significantly improve the generalization capability and performance of 3D hand pose estimation.

References

Boukhayma, A.; Bem, R. d.; and Torr, P. H. 2019. 3D Hand Shape and Pose From Images in the Wild. In *CVPR*.

Chen, X.; Wang, G.; Guo, H.; and Zhang, C. 2020. Pose guided structured region ensemble network for cascaded hand pose estimation. *Neurocomputing*, 395: 138–149.

Cheng, H.; Yang, L.; and Liu, Z. 2016. Survey on 3D Hand Gesture Recognition. *IEEE TCSVT*, 26(9): 1659–1673.

Choi, H.; Moon, G.; and Lee, K. M. 2020. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *ECCV*, 769–787. Springer.

Christen, S.; Kocabas, M.; Aksan, E.; Hwangbo, J.; Song, J.; and Hilliges, O. 2022. D-grasp: Physically plausible dynamic grasp synthesis for hand-object interactions. In *CVPR*, 20577–20586.

Deng, X.; Zuo, D.; Zhang, Y.; Cui, Z.; Cheng, J.; Tan, P.; Chang, L.; Pollefeys, M.; Fanello, S.; and Wang, H. 2022. Recurrent 3D Hand Pose Estimation Using Cascaded Pose-guided 3D Alignments. *IEEE TPAMI*, 45(1): 932–945.

Dibra, E.; Wolf, T.; Oztireli, C.; and Gross, M. 2017. How to refine 3D hand pose estimation from unlabelled depth data? In *3DV*, 135–144. IEEE.

Ge, L.; Cai, Y.; Weng, J.; and Yuan, J. 2018. Hand pointnet: 3d hand pose estimation using point sets. In *CVPR*, 8417–8426.

Hampali, S.; Rad, M.; Oberweger, M.; and Lepetit, V. 2020. Honnotate: A method for 3d annotation of hand and object poses. In *CVPR*, 3196–3206.

Hampali, S.; Sarkar, S. D.; Rad, M.; and Lepetit, V. 2022. Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation. In *CVPR*, 11090–11100.

Hasson, Y.; Tekin, B.; Bogo, F.; Laptev, I.; Pollefeys, M.; and Schmid, C. 2020. Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In *CVPR*, 571–580.

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*, 11807–11816.

Kundu, D. 2008. Bayesian Inference and Life Testing Plan for the Weibull Distribution in Presence of Progressive Censoring. *Technometrics*, 50(2): 144–154.

Laskaridis, S.; Venieris, S. I.; Almeida, M.; Leontiadis, I.; and Lane, N. D. 2020. SPINN: Synergistic Progressive Inference of Neural Networks over Device and Cloud. In *MobiCom*. New York, NY, USA: Association for Computing Machinery. ISBN 9781450370851.

Lin, K.; Wang, L.; and Liu, Z. 2021a. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, 1954–1963.

Lin, K.; Wang, L.; and Liu, Z. 2021b. Mesh Graphormer. In *ICCV*, 12939–12948.

Liu, S.; Jiang, H.; Xu, J.; Liu, S.; and Wang, X. 2021. Semi-supervised 3d hand-object poses estimation with interactions in time. In *CVPR*, 14687–14697.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully Convolutional Networks for Semantic Segmentation. In *CVPR*.

Moon, G.; and Lee, K. M. 2020. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *ECCV*, 752–768. Springer.

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 *CVPR*, 49–59.

Park, J.; Oh, Y.; Moon, G.; Choi, H.; and Lee, K. M. 2022. Handoccnet: Occlusion-robust 3d hand mesh estimation network. In *CVPR*, 1496–1505.

Song, X.; Wang, P.; Zhou, D.; Zhu, R.; Guan, C.; Dai, Y.; Su, H.; Li, H.; and Yang, R. 2019. ApolloCar3D: A Large 3D Car Instance Understanding Benchmark for Autonomous Driving. In *CVPR*.

Tang, D.; Yu, T.-H.; and Kim, T.-K. 2013. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *ICCV*, 3224–3231.

Wandt, B.; and Rosenhahn, B. 2019. Repnet: Weakly supervised training of an adversarial reprojection network for 3d human pose estimation. In *CVPR*, 7782–7791.

Yang, J.; Bhalgat, Y.; Chang, S.; Porikli, F.; and Kwak, N. 2022. Dynamic iterative refinement for efficient 3d hand pose estimation. In *WACV*, 1869–1879.

Yang, L.; and Yao, A. 2019. Disentangling latent hands for image synthesis and pose estimation. In *CVPR*, 9877–9886.

Ye, Y.; Gupta, A.; and Tulsiani, S. 2022. What's in your hands? 3D Reconstruction of Generic Objects in Hands. In *CVPR*, 3895–3905.

Zimmermann, C.; Ceylan, D.; Yang, J.; Russell, B.; Argus, M.; and Brox, T. 2019. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *ICCV*, 813–822.

Zuffi, S.; Kanazawa, A.; Jacobs, D. W.; and Black, M. J. 2017. 3D Menagerie: Modeling the 3D Shape and Pose of Animals. In *CVPR*.