2021 IEEE/CVF International Conference on Computer Vision (ICCV) | 978-1-6654-2812-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCV48922.2021.01100

End-to-End Detection and Pose Estimation of Two Interacting Hands

Dong Uk Kim Kwang In Kim Seungryul Baek
UNIST

Abstract

Three dimensional hand pose estimation has reached a level of maturity, enabling real-world applications for single-hand cases. However, accurate estimation of the pose of two closely interacting hands still remains a challenge as in this case, one hand often occludes the other. We present a new algorithm that accurately estimates hand poses in such a challenging scenario. The crux of our algorithm lies in a framework that jointly *trains the estimators of interacting hands, leveraging their interdependence. Further, we employ a GAN-type discriminator of interacting hand pose that helps avoid physically implausible configurations, e.g. intersecting fingers, and exploit the visibility of joints to improve intermediate 2D pose estimation. We incorporate them into a single model that learns to detect hands and estimate their pose based on a unified criterion of pose estimation accuracy. To our knowledge, this is the first attempt to build an end-to-end network that detects and estimates the pose of two closely interacting hands (as well as single hands). In the experiments with three datasets representing challenging real-world scenarios, our algorithm demonstrated significant and consistent performance improvements over state-of-the-arts.*

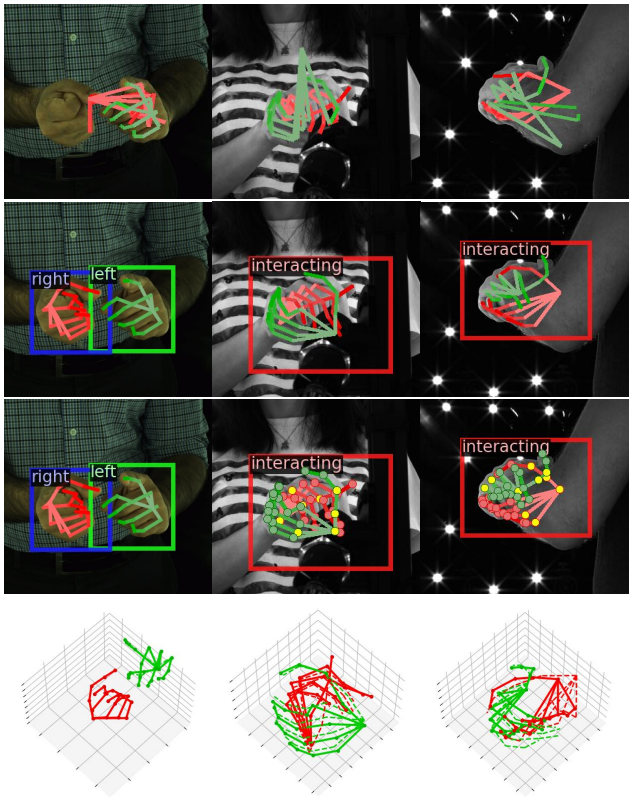


Figure 1: Example hand detection and 3D pose estimation results on the InterHand2.6M dataset [25]: (Rows 1–2) skeletal joints (overlaid on the inputs) estimated by Moon et al.'s state-of-the-art system [25] and ours, respectively. Ours provides more accurate pose estimation than [25] when two hands closely interact with each other (last two columns). For interacting hands, our joint estimation process is guided by the predicted joint visibility as shown in the third row (yellow: invisible, green and red: visible). For single-hands, visibility is not estimated (see supplemental for cases where single-hand joint visibility is also used). The last row visualizes our results in different views.

1. Introduction

Estimating hand pose finds numerous applications including augmented and virtual reality, sign language recognition, and gesture-based interfaces. The past decade has observed significant progress in this field thanks to advances in deep learning techniques. In particular, for isolated hands, skeletal pose estimation techniques are mature enough for use in practical applications. As such, recent effort has focused on challenging cases where one estimates e.g. the pose of hands captured in egocentric camera views [28, 54, 13] or interacting with objects [47, 7, 8, 14, 4], or restores hand shape as well as their skeletal pose [24, 3, 5, 57]. However, only recently, has attention been paid to estimating the pose of two interacting hands. This problem is challenging as interacting hands often cause severe self-occlusions (see Fig. 1 for examples).

Most existing work in this scenario takes generative, model fitting-based approaches, e.g. on depth maps [30, 49, 27, 42] or RGB images [52] while very recently, convolutional neural network (CNN)-based discriminative learning approaches have been investigated [25, 21]. CNN-based approaches have been particularly successful in addressing occlusions occurring in the context of egocentric views or objects under interac-

978-1-6654-2812-5/21/$31.00 ©2021 IEEE
DOI 10.1109/ICCV48922.2021.01100

Authorized licensed use limited to: ShanghaiTech University. Downloaded on October 19,2026 at 14:23:58 UTC from IEEE Xplore. Restrictions apply.

tion [28, 54, 13, 47, 7, 8, 14, 4]. However, applying these techniques to cases of interacting hands has been limited due to the lack of training data: The *Ego3D* dataset [21] provides synthetic hands of simulated characters from *Mixamo* while *Tzionas* dataset [49] provides few two-hand examples provided with 2D skeleton annotations. These datasets are limited in their scale, especially in their coverage of closely interacting hands cases. Recent *InterHand2.6M* dataset provides million-scale real images captured by multi-view cameras [25]. Still, closely interacting cases (with the bounding box intersection over union, IOU score larger than 0.5) therein are limited to around 18,000 instances.

In this paper, we propose a new CNN-based hand pose estimation framework. Our system is trained on existing datasets (*Ego3D* and *InterHand2.6M*) containing limited instances of interacting hands. However, when tested on interacting hands, it provides an accuracy level comparable to that of state-of-the-art systems on single-hand pose estimation cases.

Our approach builds upon a hypothesis that the visible hands contain useful information for inferring the pose of occluded hands. We experimentally validate this via a statistical test of independence across the joint positions of closely interacting hands, and instantiate this into a new framework that leverages such dependence by *jointly* estimating their pose. Further, we exploit the structural dependence of two hands by training a GAN-type discriminator helping avoid physically implausible joint hand configurations, e.g. two intersecting fingers. We also explicitly estimate visibility of the each joint and incorporate this information to improve 2D pose estimation.

To facilitate the training of the hand pose estimator in this scenario, we embed a hand detection network into our framework and classify the detected hands into (closely) *interacting* and *non-interacting (or single-hand)* categories, which are subsequently fed to the respective pose estimators. This enables us 1) to tailor our hand pose estimation system to challenging cases of interacting hands (via the pose estimator of interacting cases) while still retaining state-of-the-art performance on single-hand cases (via the single-hand pose estimator) and 2) to train the entire system in an end-to-end manner.

To the best of our knowledge, our system is the first end-to-end trainable pipeline that performs both detection and pose estimation of a single- or (two) interacting hands. In the experiments with *Ego3D* [21], *InterHands2.6M* [25], and *Tzionas* [49], we demonstrate that our joint estimation approach significantly improves upon 1) the baseline system that independently estimates hands and 2) state-of-the-art pose estimation systems.

2. Related work

Pose estimation of single (isolated) hands. Three-dimensional pose estimation of single hands has made significant progress in the past few years, either based on depth maps [56, 29, 43, 50, 29, 55, 23, 34, 55, 2, 51, 1] or RGB images [16, 6, 58, 17]. Depth-based 3D hand pose estimation has been especially successful thanks to the rich 3D information captured by depth maps [55]. Automatic data collection and synthesis pipelines [56, 8] further help achieve high-levels of accuracy. In the RGB domain, automatic data generation has been more challenging: Synthetic datasets often exhibit visible gaps to real-world data [34, 26, 38]. Recent attempts to acquire quality 3D annotations exploited multi-view and/or temporal information [12, 59, 39, 20]: Simon et al. [39] pioneered collecting hand pose annotations enforcing label consistency in a multi-camera setup [17] while Zimmermann et al. [59] collected 3D annotations using eight multi-view RGB cameras. In [12], Hampali et al. proposed a fully-automatic data collection pipeline involving 5 RGBD cameras and sptaio-temporal consistency. Further, differentiable renderers and perspective models [5, 9, 14, 22] have enabled training CNNs for 3D mesh reconstruction from single RGB images without requiring explicit 3D mesh supervision. They typically use 2D/3D skeletons and 2D segmentation masks as weak-supervision signals.

Existing methods can also be categorized into generative and discriminative approaches: Generative approaches optimize the parameters of 3D models (e.g. MANO [37]) to explain the input point clouds and depth values [48, 46, 37, 44, 41, 36, 31]. In the RGB domain, the 3D model is fit to intermediate representations such as 2D skeletons [32]. Most of generative approaches, however suffer from local optima or slow convergence speed. With the advent of CNNs and large-scale datasets [39, 55, 45, 55], discriminative methods have shown promising results proving powerful alternatives to generative approaches. More classical approaches including the iterated closest point and random forests-based methods can be found in [33, 18].

HPE for interacting hands. Only few existing work has considered the pose estimation of interacting hands [30, 49, 27, 52, 25, 21, 42, 40]. Oikonomidis et al. [30] pioneered this domain by fitting 3D models to interacting hands captured in RGBD sequences. Sridhar et al. [40] aligned 3D articulated Gaussian mixtures to hands interacting with objects. Tzionas et al. [49] constructed a database provided with 2D annotations, and developed a generative model that uses discriminatively detected salient points. Their approach requires either single RGBD images or multi-view RGB images.

Taylor et al. [42] proposed to combine CNNs and random forests for estimating palm orientations and hand segmentation masks to fit 3D models. Mueller et al. [27] constructed a new depth map dataset and presented a two-hand pose estimation pipeline that fits the MANO hand model [37].

Recently, Wang et al. [52] proposed a singe RGB image-based approach that fits the MANO model to estimated 2D segmentation masks and 2D skeletons. Moon et al. [25] presented the *InterHand2.6M* dataset containing millions of frames including closely interacting hands. Based on this dataset, they trained a new CNN pose estimator *InterNet* tailored for two interacting hands. This provides state-of-the-art performances on *Ego3D* and *InterHand2.6M*. As their architecture does not incorporate the hand detection capability, it requires an external

Authorized licensed use limited to: ShanghaiTech University. Downloaded on October 19,2026 at 14:23:58 UTC from IEEE Xplore. Restrictions apply.

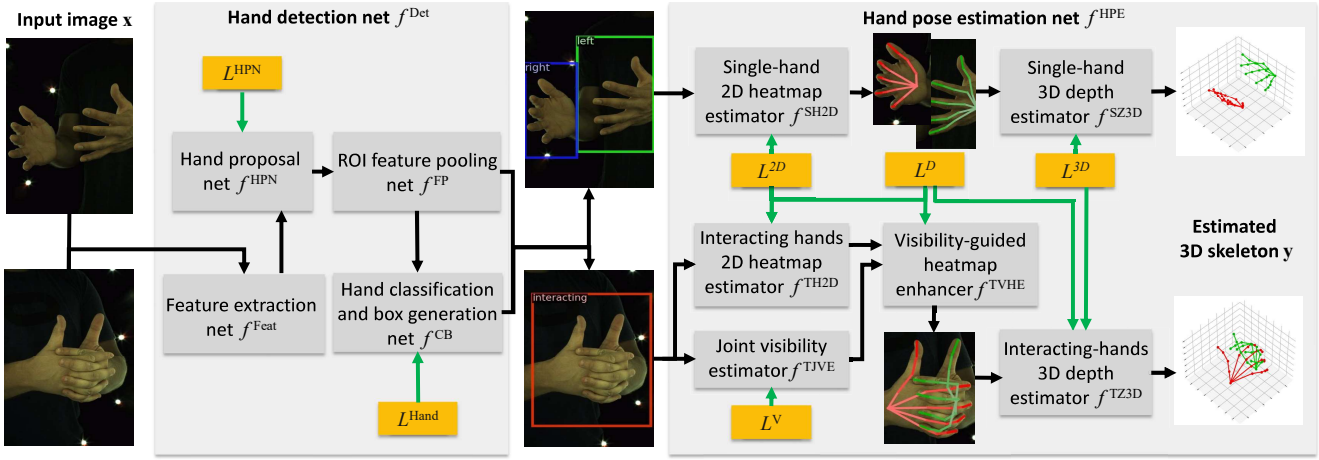


Figure 2: A schematic diagram of the training process of our end-to-end hand detection and 3D pose estimation framework. Our system first detects hands, extracts features therein, and classify them into three *handedness* classes: 'left' and 'right', and 'interacting' hands (f^{Det}). For single-hand cases ('left' and 'right'), the corresponding 3D pose is independently estimated by applying a 2D joint heatmap estimator f^{SH2D} followed by a 3D depth estimator f^{SZ3D}, each supervised by the respective losses L^{2D} and L^{3D}. For interacting hands, f^{Det} generates a single feature map that encodes both hands. These features are fed to the 2D joint heatmap estimator f^{TH2D}, joint visibility estimator f^{TJVE}, visibility-guided heatmap enhancer f^{TVHE}, and 3D depth estimator f^{TZ3D} guided by the respective skeletal joint position losses L^{2D} and L^{3D}, visibility loss L^{V}, plus the supervision L^{D} coming from a GAN-type joint hand pose discriminator. This structure helps the entire pipeline exploit 1) the statistical dependence (via joint training) and 2) structural dependence (via GAN discriminator) of two interacting hands. The black and green arrows denote the forward pass and supervision signals, respectively.

hand detector. In [21], Lin et al. proposed a pipeline combining the detection and pose estimation of two hands. However, their detection step is based on segmentation masks and therefore, the entire pose estimation pipeline is not end-to-end trainable. In the experiments, we demonstrate that our approach significantly outperforms these state-of-the-art approaches.

3. End-to-end detection and pose estimation of interacting hands

Problem definition and motivation. Our system receives an input RGB image of arbitrary size and generates the 3D joint positions of hands appearing therein. While the system can detect and estimate the pose of arbitrary number of hands (e.g. when multiple people appear in the input image), our experiments will focus on cases where only one or both hands of a person appear in each image. In this case, our system generates one (for single-hands) or two (for interacting hands) vectors of size $J \times 3$ with J being the number of joints encoding the pose of a hand. We fix J at 21 throughout the entire experiments.

Estimating hand pose becomes especially challenging when they closely interact with each other: In such cases, one hand often occludes the other as exemplified in Fig. 1. Our approach to face this challenge is to exploit the information of visible hands to improve the estimation of their occluded counterparts.

Dependence of interacting hands. We performed preliminary experiments to verify the hypothesis that the 3D pose of visible hands communicate relevant information about the occluded hand pose: A statistical test of independence of two interacting hand joints was conducted based on the Hilbert-Schmidt independence criterion (HSIC) [11] of two random vectors each encoding the pose of a hand. The bounding box of each hand is centered to remove the effect of spurious dependence caused by similar absolute joint positions. With 95% quantile of the null distribution (independence hypothesis), our test provided a strongly positive answer (33 times higher test statistics value than the pass threshold, indicating high certainty; see [11] for details). The same independence test applied to cases where two hands do not interact closely (i.e. when their IOU score is 0) also turned out to be positive but with much lower certainty: The test statistics value was only 1.56 times larger than the pass threshold.

Overview of detection and pose estimation networks. We exploit the underlying statistical dependence of two closely interacting hands by *jointly training* the corresponding pose estimators: By sharing the early layers, our pose estimation networks (for left and right hands) implicitly capture and take advantage of such dependence. This makes our framework an instance of multi-task learning. However, the results of our test above also indicate that statistical dependence is weaker when visible hands do not closely interact. We observed that incorporating such cases into joint training can degrade the performance (see the accompanying supplemental document for the corresponding experiments). Therefore, we classify each input instance into two categories 1) (closely) *interacting* and 2) *non-interacting* hands based on the IOU of the hand bounding

Authorized licensed use limited to: ShanghaiTech University. Downloaded on October 19,2026 at 14:23:58 UTC from IEEE Xplore. Restrictions apply.

boxes. For interacting cases, the pose estimators are jointly trained and tested while for non-interacting hands, single-hand pose estimators are individually applied similarly to existing hand pose estimation approaches. To facilitate this process, our system incorporates a hand detection network.

For interacting hands, we also take account of their *structural dependence* by simultaneously training a GAN-type skeletal pose discriminator. This helps prevent generating physically implausible joint configurations (e.g. two intersecting fingers).

The entire system is trained in an end-to-end manner streamlining the training of all network components in a single unified manner. Figure 2 shows an overview of our framework.

3.1. Network architectures

Our system f consists of a hand detector f^{Det} and pose estimator f^{HPE}: $f = f^{\text{HPE}} \circ f^{\text{Det}}$. The hand detector $f^{\text{Det}}: X \to F \times H$[1] receives an input image $\mathbf{x} \in X$ and generates $28 \times 28 \times 256$-dimensional feature maps of localized hands $\{\mathbf{f}_i\} \subset F$ and their *handedness* $\{h_i\} \subset H$ ('left hand', 'right hand' and 'two (interacting) hands'). If the detected hands are interacting, f^{HPE} *jointly* estimates two $J \times 3$-sized pose vectors ($\mathbf{y}$'s). For other cases, f^{HPE} independently generates a single pose vector $\mathbf{y}$ per hand.

3.1.1 Hand detection network f^{Det}

Our hand detector combines a feature extraction network f^{Feat}, hand proposal network f^{HPN}, hand classification and box generation network f^{CB}, and region of interest (ROI) feature pooling network f^{FP} [35]: $f^{\text{Det}} = [f^{\text{FP}}, f^{\text{CB}} \circ f^{\text{FP}}] \circ f^{\text{HPN}} \circ f^{\text{Feat}}$. We employ the ImageNet pre-trained ResNet-50 [15] for feature extraction: f^{Feat} receives an image $\mathbf{x}$ of size $H \times W$ and generate $\lfloor H/32 \rfloor \times \lfloor W/32 \rfloor \times 2{,}048$-sized global feature map $\mathbf{g}$. Taking $\mathbf{g}$ as input, $f^{\text{CB}} \circ f^{\text{FP}} \circ f^{\text{HPN}}$ estimates the handedness h. In parallel, f^{FP} extracts a $28 \times 28 \times 256$-sized local feature representation $\mathbf{f}$ for each bounding box proposed by f^{HPN} [35].

Hand proposal network f^{HPN}. We adopt the Faster R-CNN approach representing object bounding box proposals based on *anchor boxes* of different aspect ratios as references [35]. It is jointly trained with the feature extractor f^{Feat} by minimizing the loss below defined per image:

$$L^{\text{HPN}}(f^{\text{HPN}}, f^{\text{Feat}}) = \frac{1}{N_{\text{Cls}}} \sum_i L^{\text{Cls}}(p_i, p_i^*) + \frac{1}{N_{\text{REG}}} \sum_i p_i^* L^{\text{Reg}}(\mathbf{t}_i, \mathbf{t}_i^*), \quad (1)$$

where $\mathbf{t}_i$ is a vector of four variables representing the i-th box proposal. To facilitate training, f^{HPN} also estimates auxiliary variables $\{p_i\}$ representing the probability that the corresponding box proposals contain hands: For this, the ground-truth labels $\{p_i^*\}$ are determined as $p_i^* = 1$ when $\mathbf{t}_i$ overlaps significantly with a ground-truth hand box (i.e. the IOU score is higher than 0.7) and $p_i^* = 0$, otherwise. L^{Cls} is the standard cross-entropy loss and L^{Reg} is the smooth approximation of L_1 loss proposed in [10]. p_i^* is multiplied to the second term as L^{Reg} is used only when $\mathbf{t}_i$ overlaps with a ground-truth. The balancing parameters N_{Cls} and N_{REG} are fixed at 256 and 240, respectively, following [35].

Hand classification and box generation network f^{CB}. This 1) generates hand bounding boxes by selecting from and refining the box proposals of f^{HPN} and 2) determines the class of each output box as 'left hand', 'right hand' or '(two) interacting hands'. As many bounding boxes proposed by f^{HPN} do not actually contain any hand, we introduce an additional 'background' class to facilitate the training of f^{CB}. During training, it minimizes

$$L^{\text{Hand}}(f^{\text{CB}}, f^{\text{Feat}}) = \mathbb{1}_{\text{Hand}} L^{\text{Reg}}(\mathbf{t}, \mathbf{t}^*) + L^{\text{Cls}}(p_c, p_c^*), \quad (2)$$

where p_c and p_c^* are the predicted and ground-truth class probabilities (of four class including 'background'), respectively, and $\mathbb{1}_{\text{Hand}}$ is the indicator variable of three hand classes, i.e. L^{Reg} is not applied for 'background' boxes. The ground truth p_c^* is determined based on the hand IOU: If the IOU of the two hand boxes is larger than a threshold τ, it is classified as 'interacting' hands.

Discussion. both f^{HPN} and f^{CB} predict bounding box locations and their classes. They differ in that f^{HPN} classifies boxes into 'background' or 'hand' class, while f^{CB} classifies bounding boxes into single hands or interacting hands. Also, f^{CB} is designed to improve the initial bounding boxes obtained from the f^{HPN} (similarly to Faster-RCNN).

Determining the τ value is crucial: In training, the value of τ determines the size of the training set for the *joint* pose estimators: Large τ-values lead to small training sets focusing on challenging *closely interacting* cases. On the other hand, small τ values will offer large training sets, but they might include *loosely interacting* (easy) cases. We observed that $\tau = 0.3$ provides a good trade-off between rich and focused (on challenging cases) training sets. For testing, τ value was more conservatively determined at 0.5. The effect of varying τ values are provided in the supplemental material.

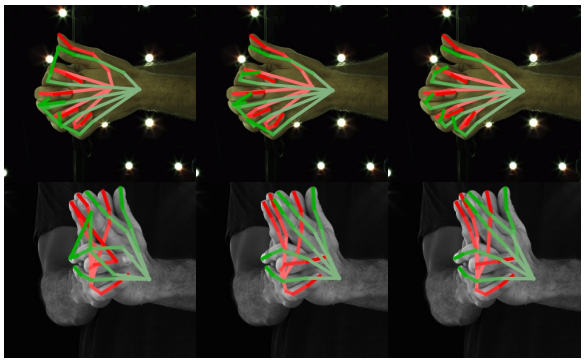
3.1.2 Hand pose estimation network f^{HPE}

This consists of two sub-networks f^{SHPE} and f^{THPE} tailored for single-hand ('left' or 'right') and interacting hands, respectively.

Single-hand 3D pose estimation network f^{SHPE}. This network combines a 2D heatmap estimator $f^{\text{SH2D}}: F \to M$ and a 3D depth value estimator $f^{\text{SZ3D}}: [F, M] \to Z$. The heatmap estimator f^{SH2D} converts the input feature map $\mathbf{f}$ to J 2D heatmaps of size 28×28 each specialized on a skeletal joint. The resulting combined heatmap $\mathbf{m} \in \mathbb{R}^{J \times 28 \times 28}$ and $\mathbf{f}$ are fed to f^{SZ3D} to estimate a depth map vector $\mathbf{z}$: $[\mathbf{z}]_i$ corresponds to the i-th joint.

[1] This is a slight abuse of notation: The range of f^{Det} is actually the power set $2^{F \times H}$ of $F \times H$ as it can generate more than one detections.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on October 19,2026 at 14:23:58 UTC from IEEE Xplore. Restrictions apply.



(a) Without L^{D} (b) With L^{D} (c) Ground-truth

Figure 3: Example hand pose estimation results of our system trained (a) without and (b) with the hand pose discriminator loss L^{D} (Eq. 6). Hand regions are cropped for improved visualization. Without L^{D}, our system can sometimes generate physically implausible configurations as shown in (a): (top) thumb and index finger intersect (bottom) third and middle finger intersect. Such configurations are detected and penalized by the discriminator d^{THPE} providing the corresponding supervision signal via L^{D}, helping correct such cases (b).

The final J skeletal joints $\{(x^i,y^i,z^i)\}_{i=1}^{J}$ are obtained by combining the peak location (x,y) of each 2D heatmap in $\mathbf{m}$ with $\mathbf{z}$.

Interacting hand pose estimation network f^{THPE}. This consists of a 2D heatmap estimator f^{TH2D} and a 3D depth estimator f^{TZ3D}. The size of f^{THPE}'s output is twice that of f^{SHPE} as the former generates the skeletal joints of two hands.

2D heatmap estimation networks f^{SH2D} and f^{TH2D}. These consist of nine 2D convolutional layers each accompanying a ReLU activation. For training, we use the standard L_2 losses:

$$\begin{aligned} L^{\mathrm{2D}}(f^{\mathrm{SH2D}}) &= \|f^{\mathrm{SH2D}}(\mathbf{f})-\mathbf{m}^{\mathrm{SGT}}\|_2^2, \\ L^{\mathrm{2D}}(f^{\mathrm{TH2D}}) &= \|f^{\mathrm{TH2D}}(\mathbf{f})-\mathbf{m}^{\mathrm{TGT}}\|_2^2, \end{aligned} \tag{3}$$

where $\mathbf{m}^{\mathrm{SGT}}$ and $\mathbf{m}^{\mathrm{TGT}}$ denote ground-truth heatmaps of single-hand (of size $J\times 28\times 28$) and interacting hands (of size $2\times J\times 28\times 28$).

Joint visibility estimation network f^{TJVE} and visibility-guided heatmap enhancement network f^{TVHE}. Our joint visibility estimate $\mathbf{v}\in V$ is a 42-dimensional vector each taking values in $[0,1]$ representing the visibility of the corresponding joint. The joint visibility estimation network $f^{\mathrm{TJVE}}: F\rightarrow V$ estimates $\mathbf{v}$ and the visibility-guided heatmap enhancement network $f^{\mathrm{TVHE}}: F\times M\rightarrow M$ receives $\mathbf{v}$ and improves the initial heatmaps $\mathbf{m}$ by weighting them with the predicted visibility $\mathbf{v}$. These networks are trained based on the L_2 losses:

$$\begin{aligned} L^{\mathrm{V}}(f^{\mathrm{TJVE}}) &= \|f^{\mathrm{TJVE}}(\mathbf{f})-\mathbf{v}^{\mathrm{TGT}}\|_2^2, \\ L^{\mathrm{2D}}(f^{\mathrm{TVHE}}) &= \|f^{\mathrm{TVHE}}(\mathbf{f},\mathbf{v}\odot\mathbf{m})-\mathbf{m}^{\mathrm{TGT}}\|_2^2, \end{aligned} \tag{4}$$

where $\mathbf{v}\odot\mathbf{m}$ denotes the heatmaps each weighted by the corresponding visibility value. Weighting the visibility in this way helps f^{TJVE} focus on more reliable estimates, as joints with high visibility $\mathbf{v}$ get more accurate heatmaps $\mathbf{m}$. The details of constructing the (pseudo) visibility ground-truths $\mathbf{v}^{\mathrm{TGT}}$ are presented in the supplemental.

3D depth estimation networks f^{SZ3D} and f^{TZ3D}. Both networks consist of two convolutional layers with ReLU activations followed by two fully connected layers with sigmoidal activations. Using sigmoid ensures that the output depth values lie in the normalized interval of $[0,1]$. Both networks receive the concatenation of holistic image features $\mathbf{f}$ and the corresponding 2D heatmaps $\mathbf{m}$. Before training, we normalize the 3D joint values to $[0,1]$ and locate the Metacarpophalangeal (MCP) joint of the middle finger at $(0.5,0.5,0.5)$. The networks are trained based on the L_2 losses:

$$\begin{aligned} L^{\mathrm{3D}}(f^{\mathrm{SZ3D}}) &= \|f^{\mathrm{SZ3D}}([\mathbf{f},\mathbf{m}])-\mathbf{z}^{\mathrm{SGT}}\|_2^2, \\ L^{\mathrm{3D}}(f^{\mathrm{TZ3D}}) &= \|f^{\mathrm{TZ3D}}([\mathbf{f},\mathbf{m}])-\mathbf{z}^{\mathrm{TGT}}\|_2^2 \end{aligned} \tag{5}$$

with $\mathbf{z}^{\mathrm{SGT}}$ and $\mathbf{z}^{\mathrm{TGT}}$ being the corresponding singe-hand and two-hand ground-truths. The detailed architectures of f^{SH2D} and f^{SZ3D} are provided in the supplemental.

Interacting hand pose discriminator d^{THPE}. While training the hand pose estimator f^{THPE} jointly on interacting hands helps exploit their underlying statistical dependence, it sometimes generates physically implausible configurations (Fig. 3a). We account for this deficiency by capturing the *structural* dependence via a GAN-type joint hand pose discriminator d^{THPE}. As our interacting hand pose estimation network f^{THPE} consists of two sub-networks f^{TH2D} and f^{TZ3D}, we decompose d^{THPE} into the corresponding discriminators: The heatmap discriminator $d^{\mathrm{TH2D}}: M\rightarrow[0,1]$ differentiates real heatmaps from those synthesized by f^{TH2D} while the 3D pose discriminator $d^{\mathrm{TZ3D}}: Y\rightarrow[0,1]$ distinguishes real 3D skeletons from the estimated ones constructed by combining $\mathbf{m}$ and $\mathbf{z}$. Our discriminator d^{TZ3D} sees the entire 3D joints $\mathbf{y}$ instead of only their depth parts $\mathbf{z}$. This provides a better context to d^{TZ3D}: The depth values by themselves do not provide enough information to check the realism. It should be noted that d^{TZ3D} cannot provide supervision to the heatmap estimator f^{TH2D} as the $x-y$ values detected from the 2D heatmaps are non-differentiable, hence we use d^{TH2D}.

3.2. Training

Our network f is trained by minimizing a combination of the losses for sub-networks (Eqs. 1–5):

$$\begin{aligned} L = {} & L^{\mathrm{HPN}}(f^{\mathrm{HPN}},f^{\mathrm{Feat}})+L^{\mathrm{Hand}}(f^{\mathrm{CB}},f^{\mathrm{Feat}}) \\ & +L^{\mathrm{2D}}(f^{\mathrm{H2D}},f^{\mathrm{TVHE}})+\lambda_1 L^{\mathrm{3D}}(f^{\mathrm{Z3D}}) \\ & +\lambda_2 L^{\mathrm{D}}(f^{\mathrm{TH2D}},f^{\mathrm{TZ3D}},f^{\mathrm{TVHE}})+\lambda_3 L^{\mathrm{V}}(f^{\mathrm{TJVE}}), \end{aligned} \tag{6}$$

where the loss L^{D} represents supervision provided by the interacting hand pose discriminator d^{THPE}. The weighting

Authorized licensed use limited to: ShanghaiTech University. Downloaded on October 19,2026 at 14:23:58 UTC from IEEE Xplore. Restrictions apply.

parameters λ_1, λ_2, and λ_3 are determined at 30, 0.01 and 1, respectively based on cross-validation on the *InterHand2.6M* dataset. Similarly to the Faster-RCNN training scheme [35], we alternate between the updates of 1) the feature extraction network f^{Feat} and hand proposal network f^{HPN} (via L^{HPN}) and 2) the feature extraction network f^{Feat} and hand classification and box generation network f^{CB} (via L^{Hand}), both followed by the update of f^{HPE} using L^{2D}, L^{3D}, and L^{D}.

We observed that the hand detection network f^{Det} converges faster than the hand pose estimation network f^{HPE} and thus, we freeze the weight of f^{Det} after 5 out of 10 total epochs, to speed up the training process.

Data augmentation. We enlarged the original training dataset by applying the standard data augmentation steps including 1) translation up to 10 pixels (within 10% of the ground-truth box sizes), 2) rotation within $[-45,45]$ degrees, 3) horizontal flipping, 4) brightness and saturation changes in $[-20\%,60\%]$ and $[-10\%, 20\%]$ of the original values, respectively. The resulting dataset is around four times larger than the original set.

Implementation details. Python and PyTorch library are used for our implementation.[2] All network weights were initialized by sampling from i.i.d Gaussian distribution with mean zero and standard deviation 0.001. For optimization, we use the Adam optimizer [19] with the initial learning rate of 0.001 and its default parameter $\beta=(0.9,0.999)$. The size of mini-batch and the number of total epochs were fixed at 3 and 10, respectively.

4. Experiments

Datasets. We evaluated our method on three challenging datasets containing two-hands interaction cases: *Ego3D* [21], *InterHand2.6M* [25] and *Tzionas* [49] datasets.

For *Ego3D*, we adopted their 'Static' evaluation protocol [21] having 50,000 training and 5,000 testing images. It provides 3D ground-truth annotations of 21 skeletal joints (1 for wrist and 4 for each finger). For each data instance, the $x-y$-coordinate values (the height and width of the image pane) were normalized to $[0,1]$ while its z-value was scaled such that the bone length between the wrist and middle MCP becomes 10cm.

InterHand2.6M (v0.0) is the first realistic dataset having RGB and 3D pose annotations for two-hand interactions. It contains 2.6 million 512×334-sized images of 26 subjects (7 females, 19 males). We use the 'Train (H)' protocol suggested by the authors of this dataset [25]: Total 284,716 images (76,445 isolated hands and 208,271 interacting hands) were used for training while 66,722 image were used for testing (18,399 single-hands and 48,323 interacting hands).

For testing on *Tzionas* [49], we used their 7 two-hand sequences containing 1,307 frames total. Since this dataset does not have a separate training set, we applied our system trained on *InterHand2.6M*.

[2]Our code builds upon the Faster R-CNN implementation provided by detectron2: https://github.com/facebookresearch/detectron2

Baselines and evaluation metric. We compare with 3 state-of-the-art hand pose estimation approaches that are explicitly designed for interacting hands: Moon et al.'s *InterHand2.6M*-based system [25], Lin et al.'s global two-hand pose estimation approach based on *Ego3D* [21], and Wang et al.'s model-based approach [52]. To assess the performance of our algorithm when applied to single-hand cases, we also compare with Wei et al.'s convolutional pose machine [53] (for 2D hand pose only) and Boukhayma et al.'s joint hand pose and mesh estimation approach [5].

For all baselines that we compare with, we show the results reported in the respective publications. Our results are obtained based on the same training and testing set splits making direct comparisons possible (per dataset; shown shortly). However, only [21, 52] and ours provide the explicit hand detection capability while the remaining algorithms assume that each input image focuses on individual hands. For the latter approaches ([25, 5, 53]), the reported results were obtained based on hand-focused images cropped using the ground-truth bounding box annotations.

For evaluation, three error measures were used: 3D end point error (EPE) and mean per joint position error (MPJPE) both in mm unit, and 2D end point error (EPE) in pixel unit. For *Ego3D* [21], 2D and 3D EPEs were used as in [53, 25, 21]. For *InterHand2.6M*, MPJPE was used following [25]. *Tzionas* offers only 2D annotations for every 5 frames, therefore we used 2D EPE to facilitate direct comparisons with [25, 5, 52].

Results. Table 1 summarizes the results. The previous state-of-the-art results on *Ego3D* were reported by Moon et al. [25] (in 3D EPE) and Wei et al. [53] (in 2D EPE). Our algorithm achieved significant performance improvements from these results (by 3.93% and 32.49% reduction of error rates, respectively). It should be noted that Moon et al.'s algorithm used the ground-truth hand bounding box annotations at testing while ours achieved lower error rates even without relying on such annotations. Wei et al.'s [53] approach is designed for 2D hand pose estimation: For comparison in 2D, we projected the initial 3D pose estimation results onto the image pane.

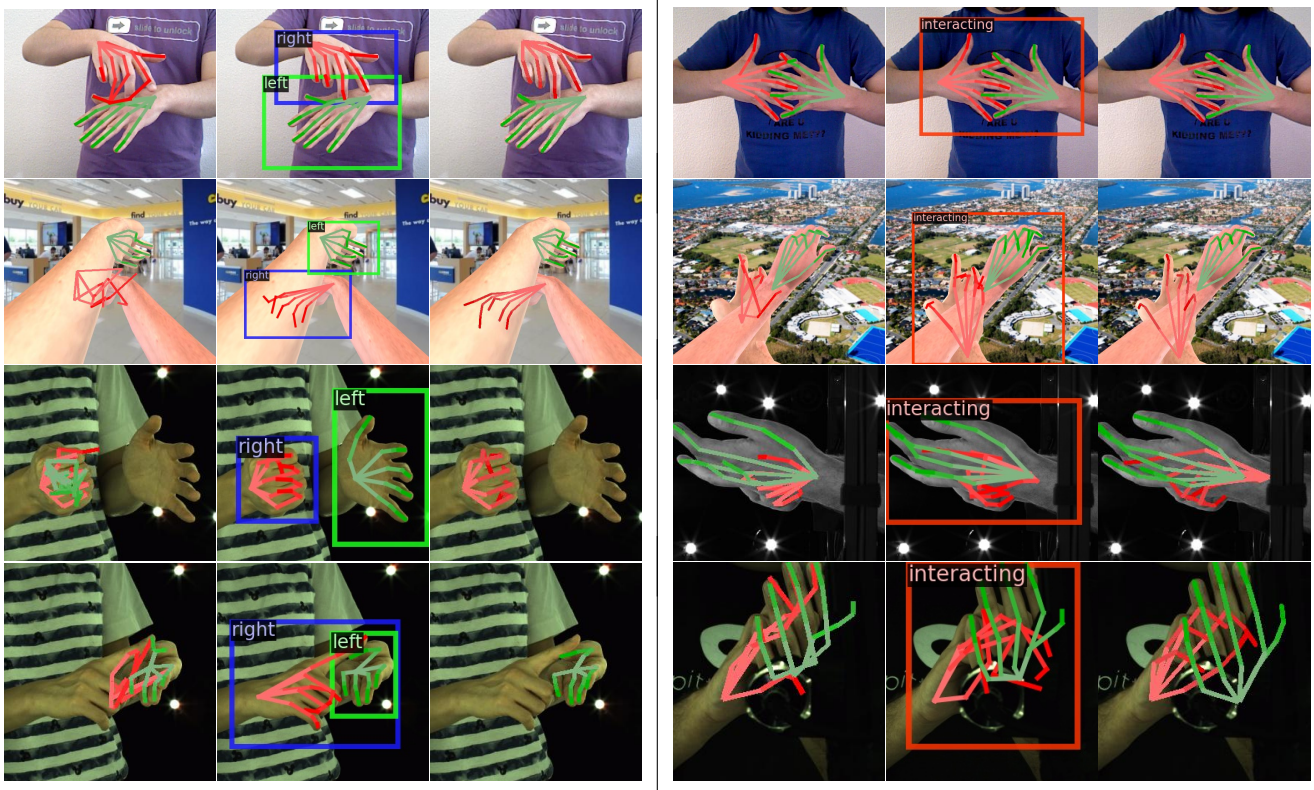
Our algorithm also improved Moon et al.'s state-of-the-art results on *InterHand2.6M* by 4.05%. For *Tzionas*, the previous best results were achieved by Boukhayma et al.'s approach [5]. Ours outperformed theirs by 3.80%. Overall, ours consistently ranked best across all datasets.

Apart from our algorithm, Moon et al.'s approach ranked the best on both *Ego3D* and *InterHand2.6M* [25]. We also made an attempt to compare with this approach on *Tzionas* using their publicly available code and network weights:[3] The corresponding results (Table 1) indicate that our approach provide much more stable performance across different datasets.

Figure 4 shows example images and the corresponding hand pose estimation results of Moon et al.'s algorithm [25]

[3]https://github.com/facebookresearch/InterHand2.6M

Authorized licensed use limited to: ShanghaiTech University. Downloaded on October 19,2026 at 14:23:58 UTC from IEEE Xplore. Restrictions apply.



(a) Moon et al. [25] (b) Ours (c) Ground-truths (d) Moon et al. [25] (e) Ours (f) Ground-truths

Single-hand | Interacting hands

Figure 4: Example hand pose estimation results on *Tzionas* (top), *Ego3D* (second row), and *InterHand2.6M* (last two rows): Hand regions are further cropped for improved visualization. Our system automatically detects hands in each input image. The accompanying supplemental provides additional examples.

(using their code) and ours on single- and interacting-hands cases. Both Moon et al.'s approach and our algorithm generated highly accurate pose estimates for single-hand cases. However, for interacting hands, severe occlusions can pose significant challenges even for state-of-the-art Moon et al.'s approach (fourth-column in Fig. 4). By exploiting the dependence lying in interacting hands and thereby jointly training the corresponding estimators, our approach can provide higher quality estimates.

The hand detection accuracy was measured at the mean AP of 98.62 on InterHand2.6M. For handedness classification and joint visibility estimation, the average classification accuracies were 97.64% and 78.38%, respectively.

We used the joint visibility only at interacting hands as constructing the ground-truth visibility can be challenging for single-hands. Our supplemental document shows that when available, such joint visibility can lead to a slight performance gain for single-hands.

Ablation study. We assessed the contributions of the *interacting hands*-specific designs in our system: We constructed four variations of our original system by 1) removing the contribution of the GAN discriminator L^{D} in Eq. 6, denoted as 'Ours ($-L^{\mathrm{D}}$)', 2) further removing the joint visibility estimation and visibility-guided heatmap enhancement networks ('Ours ($-L^{\mathrm{V}}$,L^{D})'), and 3) completely removing the joint training of instances that belong to *interacting hands* class denoted as 'Ours ($-$Interaction, L^{D})'. These variations remove the capabilities of capturing *structural* and *statistical* dependence of interacting hands from our final system. We also assessed the effectiveness of our end-to-end hand detection and pose estimation design via 4) a system that constructs and freezes the hand detector before the training of the pose estimator ('Ours (Separate detection)').

We assessed their performances in two test cases of *InterHand2.6M*: The first case focuses on a subset containing only closely interacting hands (with the corresponding IOU score less than τ) while the second evaluates on the entire dataset. Table. 2 shows the results. The results for closely interacting cases demonstrate that our GAN discriminator, visibility-guided heatmap enhancer, and joint training (of interacting hands) strategy collectively and individually contribute to improving

Authorized licensed use limited to: ShanghaiTech University. Downloaded on October 19,2026 at 14:23:58 UTC from IEEE Xplore. Restrictions apply.

Table 1: Error rates of different hand pose estimation approaches. The best and second best results are highlighted in **blue** and *green*, respectively. The algorithms marked with '✓' on Box inf. column provide the hand detection capability. For the other algorithms (with ✗), the bounding boxes of hands in each image were generated based on the ground-truth box labels. The results of Moon et al.'s algorithm on *Tzionas* was obtained using code publicly available by Moon (see text for details).

Method	Box inf.	*Ego3D* [21]		*InterHand2.6M* [25]	*Tzionas* [49]
		2D EPE (px)	3D EPE (mm)	MPJPE (mm)	2D EPE (px)
Wei et al. [53]	✗	*7.11*	N/A	N/A	N/A
Lin et al. [21]	✓	8.11	17.42	N/A	N/A
Boukhayma et al. [5]	✗	N/A	N/A	N/A	*12.91*
Wang et al. [52]	✓	N/A	N/A	N/A	13.31
Moon et al. [25]	✗	N/A	*12.20*	*12.58*	17.61
Ours	✓	**4.53**	**11.63**	**12.08**	**12.42**

Table 2: Performances (MPJPE in mm) of alternative design choices of our algorithm on *InterHand2.6M*. For interacting hands, the error rates measured at only visible joints are also shown in parentheses (calculated using the joint visibility ground-truths which are available only for interacting hands).

Method	MPJPE
Entire dataset	
Ours ($-$Interaction, L^{D})	12.39
Ours ($-L^{\mathrm{V}}$,L^{D})	12.23
Ours ($-L^{\mathrm{D}}$)	12.17
Ours (Separate detection)	13.69
Ours	12.08
Only 'interacting hands' cases	
Ours ($-$Interacting class, L^{D})	14.36 (14.16)
Ours ($-L^{\mathrm{V}}$,L^{D})	12.95 (12.47)
Ours ($-L^{\mathrm{D}}$)	12.39 (11.93)
Ours (Separate detection)	12.48 (12.04)
Ours	11.52 (11.11)

the performance. The corresponding accuracy improvements on the entire dataset are less pronounced since all four algorithms generate the same outputs for single-hand cases. It should be noted that the average error rate of our final algorithm on interacting hands is only 0.8% higher than that of the entire dataset indicating that our algorithm achieves similar levels of accuracy for single-hand and more challenging interacting hands cases.

5. Conclusion and discussion

Hand interactions pose a major challenge to pose estimation due to severe occlusion of one hand by the other. We empirically verified our conjecture that the information of visible parts of interacting hands can help infer the pose of occluded hands: Our pose estimation network is trained to jointly estimate the pose of two interacting hands exploiting the underlying statistical dependence as well as the visibility of individual joints. We further enhanced the structural consistency of the estimated hand joints using a GAN-type discriminator. Our algorithm is instantiated as a new end-to-end network that automatically detects and estimates the pose of hands on arbitrary RGB images. Evaluated on three challenging datasets representing real-world scenarios, our algorithm demonstrated significant performance improvements over state-of-the-art approaches (either specialized on interacting hands or single hands).

Our GAN discriminator sees only skeletal joints. While this helps avoid generating physically implausible skeletal configurations, it cannot directly capture the mechanics of interacting hand surface geometry, e.g. skin deformations, which can provide additional information for the pose of occluded hands. Future work should explore the possibilities of estimating and leveraging hand shape (meshes) e.g. by fitting MANO model [37] as well as their dynamics e.g. by employing recurrent neural networks.

Our statistical dependence test across the joint positions of interacting hands further supports the hypothesis that visible hands contain useful information about the pose of occluded hands. However, it is possible that our test reflects spurious correlations which might exist even between non-interacting hands. Such a possibility could be ruled out based on experiments on datasets that record hands of multiple persons. Future work should also investigate this, as well as the possibility of applying our approach to estimating the poses and shapes of hands and objects under interaction.

Acknowledgments

This work was supported by the NRF grants (No. 2021R1F1A1047920 and No. 2021R1A2C2012195) and IITP grants (No. 2020–0–01336, AIGS of UNIST, No. 2021–0–01778, Development of human image synthesis and discrimination technology below the perceptual threshold, No. 2020–0–00537, Development of 5G based low latency device - edge cloud interaction technology, and No. 2021–0–00537, Visual Common Sense Through Self-supervised Learning for Restoration of Invisible Parts in Images), all funded by the Korea government (MSIT).

Authorized licensed use limited to: ShanghaiTech University. Downloaded on October 19,2026 at 14:23:58 UTC from IEEE Xplore. Restrictions apply.

References

[1] Anil Armagan, Guillermo Garcia-Hernando, Seungryul Baek, Shreyas Hampali, Mahdi Rad, Zhaohui Zhang, Shipeng Xie, MingXiu Chen, Boshen Zhang, Fu Xiong, et al. Measuring generalisation to unseen viewpoints, articulations, shapes and objects for 3d hand pose estimation under hand-object interaction. In *ECCV*, 2020. 2

[2] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Augmented skeleton space transfer for depth-based hand pose estimation. In *CVPR*, 2018. 2

[3] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Pushing the envelop for RGB-based dense 3D hand pose estimation via neural rendering. In *CVPR*, 2019. 1

[4] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Weakly-supervised domain adaptation via GAN and mesh model for estimating 3D hand poses interacting objects. In *CVPR*, 2020. 1, 2

[5] Adnane Boukhayma, Rodrigo de Bem, and Philip H. S. Torr. 3D hand shape and pose from images in the wild. In *CVPR*, 2019. 1, 2, 6, 8

[6] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *ECCV*, 2018. 2

[7] Enric Corona, Albert Pumarola, Guillem Alenya, Francesc Moreno-Noguer, and Gregory Rogez. GanHand: Predicting human grasp affordances in multi-object scenes. In *CVPR*, 2020. 1, 2

[8] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with RGB-D videos and 3D hand pose annotations. In *CVPR*, 2018. 1, 2

[9] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3D hand shape and pose estimation from a single RGB image. In *CVPR*, 2019. 2

[10] Ross Girshick. Fast R-CNN. In *ICCV*, 2015. 4

[11] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In *ALT*, 2005. 3

[12] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. HOnnotate: A method for 3D annotation of hand and objects poses. In *CVPR*, 2020. 2

[13] Shangchen Han, Beibei Liu, Randi Cabezas, Christopher D. Twigg, Peizhao Zhang, Jeff Petkau, Tsz-Ho Yu, Chun-Jung Tai, Muzaffer Akbay, Zheng Wang, Asaf Nitzan, Gang Dong, Yuting Ye, Lingling Tao, Chengde Wan, and Robert Wang. MEgATrack: monochrome egocentric articulated hand-tracking for virtual reality. In *SIGGRAPH*, 2020. 1, 2

[14] Yana Hasson, Gül Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J. Black, Ivan Laptev, and Cordelia Schmid. 3D hand shape and pose estimation from a single RGB image. In *CVPR*, 2019. 1, 2

[15] Kaming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[16] Umar Iqbal, Pavlo Molchanov, Thomas Breuel, Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5D heatmap regression. In *ECCV*, 2018. 2

[17] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: a massively multiview system for social motion capture. In *ICCV*, 2015. 2

[18] Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. Real time hand pose estimation using depth sensors. 2013. 2

[19] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. 6

[20] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Self-supervised learning of 3D human pose using multi-view geometry. In *CVPR*, 2019. 2

[21] Fanqing Lin, Connor Wilhelm, and Tony Martinez. Two-hand global 3D pose estimation using monocular RGB. In *ArXiv*, 2020. 1, 2, 3, 6, 8

[22] Jameel Malik, Ahmed Elhayek, Fabrizio Nunnari, Kiran Varanasi, Kiarash Tamaddon, Alexis Heloir, and Didier Stricker. DeepHPS: End-to-end estimation of 3D hand pose and shape by learning from synthetic depth. In *3DV*, 2018. 2

[23] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2V-PoseNet: Voxel-to-voxel prediction network for accurate 3D hand and human pose estimation from a single depth map. In *CVPR*, 2018. 2

[24] Gyeongsik Moon, Takaaki Shiratori, and Kyoung Mu Lee. DeepHandMesh: Weakly-supervised deep encoder-decoder framework for high-fidelity hand mesh modeling from a single RGB image. In *ECCV*, 2020. 1

[25] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. InterHand2.6M: A new large-scale dataset and baseline for 3D single and interacting hand pose estimation from a single RGB image. In *ECCV*, 2020. 1, 2, 6, 7, 8

[26] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. GANerated hands for real-time 3D hand tracking from monocular RGB. In *CVPR*, 2018. 2

[27] Franziska Mueller, Micah Davis, Florian Bernard, Oleksandr Sotnychenko, Mickeal Verschoor, Miguel A. Otaduy, Dan Casas, and Christian Theobalt. Real-time pose and shape reconstruction of two interacting hands with a single depth camera. In *SIGGRAPH*, 2019. 1, 2

[28] Franziska Mueller, Dushyant Mehta, Oleksandr Sotnychenko, Srinath Sridhar, Dan Casas, and Christian Theobalt. Real-time hand tracking under occlusion from an egocentric RGB-D sensor. In *ICCV*, 2017. 1, 2

[29] Markus Oberweger and Vincent Lepetit. DeepPrior++: Improving fast and accurate 3D hand pose estimation. In *ICCV HANDS Workshop*, 2017. 2

[30] Iasonas Oikonomidis, Nikolaos Kyriazis, and Antonis A. Arguros. Tracking the articulated motion of two strongly interacting hands. In *CVPR*, 2012. 1, 2

[31] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A. Argyros. Efficient model-based 3D tracking of hand articulations using kinect. In *BMVC*, 2011. 2

[32] Paschalis Panteleris, Iason Oikonomidis, and Antonis Argyros. Using a single RGB frame for real time 3D hand pose estimation in the wild. In *WACV*, 2018. 2

[33] Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. Realtime and robust hand tracking from depth. In *CVPR*, 2014. 2

[34] Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Feature mapping for learning fast and accurate 3D pose inference from synthetic images. In *CVPR*, 2018. 2

Authorized licensed use limited to: ShanghaiTech University. Downloaded on October 19,2026 at 14:23:58 UTC from IEEE Xplore. Restrictions apply.

[35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks . In *CVPR*, 2015. 4, 6

[36] Konstantinos Roditakis, Alexandros Makris, and Antonis A. Argyros. Generative 3D hand tracking with spatially constrained pose sampling. In *BMVC*, 2017. 2

[37] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. In *SIGGRAPH Asia*, 2017. 2, 8

[38] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. Learning from simulated and unsupervised images through adversarial training. In *CVPR*, 2017. 2

[39] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017. 2

[40] Srinath Sridhar, Franziska Mueller, Franziska, Michael Zollhöfer, Dan Casas, Antti Oulasvirta, and Christian Theobalt. Real-time joint tracking of a hand manipulating an object from rgb-d input. In *ECCV*, 2016. 2

[41] Jonathan Talyor, Richard Stebbing, Varun Ramakrishna, Cem Keskin, Jamie Shotton, Shahram Izadi, Aaron Hertzmann, and Andrew Fitzgibbon. User-specific hand modeling from monocular depth sequences. In *CVPR*, 2014. 2

[42] Jonathan Talyor, Vladimir Tankovich, Danhang Tang, Cem Keskin, David Kim, Philip Davidson, Adarsh Kowdle, and Shahram Izadi. Articulated distance fields for ultra-fast tracking of hands interacting. In *SIGGRAPH*, 2017. 1, 2

[43] Danhang Tang, Hyung Jin Chang, Alykhan Tejani, and Tae-Kyun Kim. Latent regression forest: structural estimation of 3D articulated hand posture. *TPAMI*, 2016. 2

[44] Danhang Tang, Jonathan Taylor, Pushmeet Kohli, Cem Keskin, Tae-Kyun Kim, and Jamie Shotton. Opening the black box: hierarchical sampling optimization for estimating human hand pose. In *ICCV*, 2015. 2

[45] Danhang Tang, Tsz-Ho Yu, and Tae-Kyun Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *ICCV*, 2013. 2

[46] Jonathan Taylor, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Toby Sharp, Eduardo Soto, David Sweeney, Julien Valentin, Benjamin Luff, Arran Topalian, Erroll Wood, Sameh Khamis, Pushmeet Kohli, Shahram Izadi, Richard Banks, Andrew Fitzgibbon, and Jamie Shotton. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. In *SIGGRAPH*, 2016. 2

[47] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+O: Unified egocentric recognition of 3D hand-object poses and interactions. In *CVPR*, 2019. 1, 2

[48] Anastasia Tkach, Mark Pauly, and Andrea Tagliasacchi. Sphere-meshes for real-time hand modeling and tracking. In *SIGGRAPH Asia*, 2016. 2

[49] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics simulation. *IJCV*, 2016. 1, 2, 6, 8

[50] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Crossing nets: dual generative models with a shared latent space for hand pose estimation. In *CVPR*, 2017. 2

[51] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Self-supervised 3D hand pose estimation through training by fitting. In *CVPR*, 2019. 2

[52] Jiayi Wang, Franziska Mueller, Florian Bernard, Suzanne Sorli, Neng Qian, Oleksandr Sotnychenko, Miguel A. Otaduy, Dan Casas, and Christian Theobalt. RGB2Hands: Real-time tracking of 3D hand interactions from monocular RGB video. In *SIGGRAPH Asia*, 2020. 1, 2, 6, 8

[53] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016. 6, 8

[54] Qi Ye and Tae-Kyun Kim. Occlusion-aware hand pose estimation using hierarchical mixture density network. In *ECCV*, 2018. 1, 2

[55] Shanxin Yuan, Guillermo Garcia-Hernando, Björn Stenger, Gyeongsik Moon, Ju Yong Chang, Kyoung Mu Lee, Pavlo Molchanov, Jan Kautz, Sina Honari, Liuhao Ge, Junsong Yuan, Xinghao Chen, Guijin Wang, Fan Yang, Kai Akiyama, Yang Wu, Qingfu Wan, Meysam Madadi, Sergio Escalera, Shile Li, Dongheui Lee, Iason Oikonomidis, Antonis Argyros, and Tae-Kyun Kim. Depth-based 3D hand pose estimation: From current achievements to future goals. In *CVPR*, 2018. 2

[56] Shanxin Yuan, Qi Ye, Björn Stenger, Siddhant Jain, and Tae-Kyun Kim. Big hand 2.2M benchmark: hand pose data set and state of the art analysis. In *CVPR*, 2017. 2

[57] Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. Monocular real-time hand shape and motion capture using multi-modal data. In *CVPR*, 2020. 1

[58] Christian Zimmermann and Thomas Brox. Learning to estimate 3D hand pose from single RGB images. In *ICCV*, 2017. 2

[59] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. FreiHand: A dataset for markerless capture of hand pose and shape from single RGB image. In *ICCV*, 2019. 2

Authorized licensed use limited to: ShanghaiTech University. Downloaded on October 19,2026 at 14:23:58 UTC from IEEE Xplore. Restrictions apply.