HaMuCo: Hand Pose Estimation via Multiview Collaborative Self-Supervised Learning

Xiaozheng Zheng [1,2 †] Chao Wen [2 †] Zhou Xue [2] Pengfei Ren [1,2] Jingyu Wang [1 *]
[1] State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications
[2] PICO IDL, ByteDance, Beijing

Abstract

Recent advancements in 3D hand pose estimation have shown promising results, but its effectiveness has primarily relied on the availability of large-scale annotated datasets, the creation of which is a laborious and costly process. To alleviate the label-hungry limitation, we propose a self-supervised learning framework, HaMuCo, that learns a single-view hand pose estimator from multi-view pseudo 2D labels. However, one of the main challenges of self-supervised learning is the presence of noisy labels and the "groupthink" effect from multiple views. To overcome these issues, we introduce a cross-view interaction network that distills the single-view estimator by utilizing the cross-view correlated features and enforcing multi-view consistency to achieve collaborative learning. Both the single-view estimator and the cross-view interaction network are trained jointly in an end-to-end manner. Extensive experiments show that our method can achieve state-of-the-art performance on multi-view self-supervised hand pose estimation. Furthermore, the proposed cross-view interaction network can also be applied to hand pose estimation from multi-view input and outperforms previous methods under the same settings.

1. Introduction

3D hand pose estimation is essential in various application scenarios, from action recognition and sign language translation to AR/VR [19, 20]. Hand pose estimation has achieved a significant improvement in recent years. However, the progress heavily relies on the emergence of many hand pose datasets with accurate 3D annotations. Acquiring labeled datasets is quite time-consuming and laborious, exposing a realistic challenge for deep learning models to learn with limited and noisy data.

Self-supervised learning is an emerging solution to the challenge posed by manual annotation. Though worth exploring, self-supervised pose estimation with RGB hand images is a challenging and relatively unexplored area with only one pioneering method, S^{2}HAND [9]. S^{2}HAND aims to conduct 3D hand reconstruction from a single RGB image with the noisy off-the-shell 2D hand pose estimation results for supervision. Unfortunately, S^{2}HAND faces a predicament where its performance is significantly reliant on the quality of the pseudo label, and inferior labeling may result in reduced performance. Moreover, evaluating the quality of the pseudo label is an ill-posed problem that lacks clear criteria or input, further complicating the issue.

This observation motivates us to leverage multi-view information for enhancing self-supervised learning, as the complementary nature of multi-view observations can help mitigate the ambiguity inherent in pose estimation. Although the first 3D hand dataset with synchronized multi-view input (HanCo [62]) was proposed in 2021, to our knowledge, there is no previous work exploring the potential of multi-view for self-supervised hand pose estimation. Therefore, we turn to studies in the human body pose estimation, which share some similarities.

As mentioned in previous work [24], naively enforcing multi-view consistency is prone to generate degenerated solutions, thus they resorted to additional 2D labels of unrelated datasets and proposed a solution under the scope of weakly supervised learning. Other studies, such as EpipolarPose [27] and CanonPose [52], utilized multi-view data with special designs to enhance the supervision and achieved promising results under the scope of self-supervised learning.

In this paper, we push along this direction on hand pose estimation via multi-view collaborative learning. We take one step further by designing a learnable network, which utilizes multi-view information, to tackle 1) noisy pseudo labels and 2) unreliable multi-view "groupthink" issues causing training collapse in the early training stage. Formally, we name the pipeline HaMuCo, which stands for

Project page: https://zxz267.github.io/HaMuCo.

* Corresponding author. † Equal contribution.

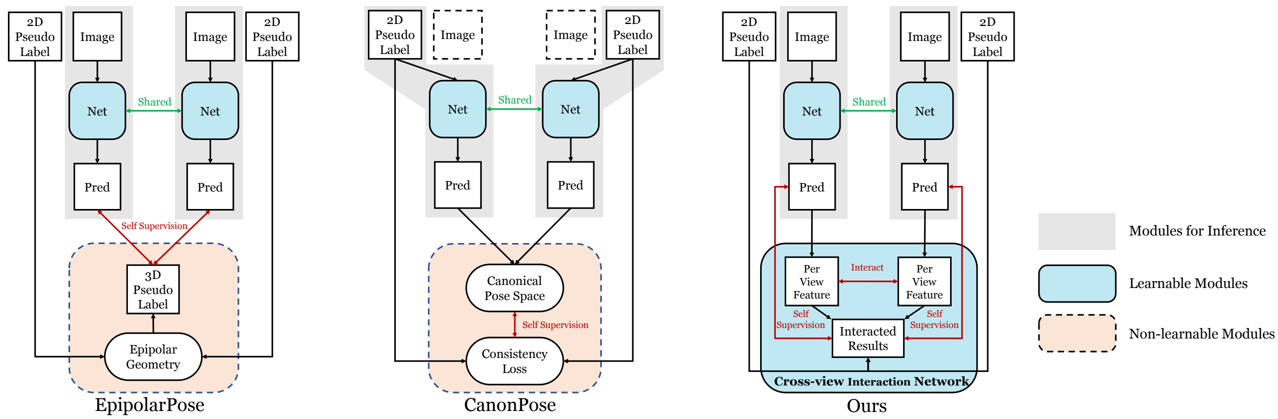

Figure 1. Overall pipeline comparison: HaMuCo learns a monocular 3D hand pose estimator from multi-view self-supervision via cross-view feature interaction. Our cross-view interaction network addresses the importance of introducing learnable feature interaction, which is absent from previous methods [27, 52]. At inference time only the gray part is applied.

Hand Multiview Collaborative learning.

The core idea of our approach is to enhance the single-view estimation by means of cross-view feature interaction and further integrate multi-view results to supervise the single-view output to achieve self-distillation in an end-to-end fashion. Thus, our framework is built with a single-view hand pose estimator and a cross-view interaction network for supervision. The single-view estimator uses the MANO [45] hand model as the decoder, which provides the hand prior to regularizing irrational anatomy when supervised by noisy pseudo labels. The cross-view interaction network captures cross-view features and utilizes several consistent losses among different views to guide collaborative learning.

We conduct comprehensive experiments on the HanCo [62] dataset and our approach outperforms previous methods by a considerable margin for self-supervised 3D hand pose estimation. Notably, our results demonstrate competitive performance compared to a state-of-the-art fully supervised approach proposed by Chen *et al*. [7]. Our proposed framework is highly versatile, as it can be trained with or without calibration, and is capable of incorporating the cross-view interaction network to achieve superior multi-view inference results when multi-view test data is available. Moreover, our model can generalize well to other datasets [29, 46, 64] and in-the-wild images.

In summary, our contributions are the following:

- We propose the first self-supervised learning framework for single-view hand pose estimation without any training data annotation and achieve state-of-the-art performance via multi-view collaborative learning.
- We propose a cross-view interaction network to supervise the single-view estimator by enforcing multi-view consistency and capturing cross-view features for collaborative learning among multiple views.
- The proposed framework is capable of multi-view inference by incorporating the cross-view interaction network and achieves state-of-the-art performance without bells and whistles.

2. Related Work

Hand Pose Estimation. Hand pose estimation can be categorized into RGB-based methods [23, 49, 63] and depth-based methods [14, 15, 36], depending on the input modality. In this paper, we focus our attention on RGB-based hand pose estimation. The RGB-based methods can be further divided into three categories, *skeleton-based methods* [5, 12, 23, 31, 38, 39, 48, 49, 55–57, 63], *model-based methods* [1, 2, 4, 9, 58, 59, 64], and *mesh-based* methods [7,8,10,16,28,30,32,33,37,50,61]. *Skeleton-based methods* regress the hand joints directly. Zimmermann *et al*. [63] introduces a multi-stage network that lifts the regressed 2D joints to 3D ones. Variational autoencoder [26] is employed to learn a cross-modal latent space to achieve better hand pose estimation and disentanglement [49, 56, 57]. Latent 2.5D representation regression is proved more effective than direct coordinates regression for hands by [23], which is also adopted by [13, 31, 48, 61]. There are also many works solving hand pose estimation with two hands interactions [13, 29, 38] and hand-object interactions [2, 12, 29]. Recent *model-based methods* make use of MANO [45], which can incorporate the hand prior and predict the hand mesh simultaneously. Those methods [1, 4, 9, 58, 59] rely on additional supervisions [1, 4, 9, 58, 59] or inputs [4]. In contrast, *mesh-based methods* regress each vertex directly, which is more accurate but requires large-scale datasets with hand mesh annotations [18, 29, 38, 64]. Most of these methods utilize graph convolutional network (GCN) [7, 8, 10, 16, 28, 50, 61] or transformers [32] or both [30, 33] for regression. I2L-MeshNet [37] regresses each vertex by predicting 1D heatmaps of three axes. Chen *et al*. [6] uses an image-to-image translation network to predict the UV map of the mesh. Similar to previous works [30, 33], we also use transformer and GCN. However, we employ them for cross-view interaction.

Multi-View Fully-Supervised Pose Estimation. Multi-view information is widely explored to improve 3D human pose estimation by tackling occlusions and depth-ambiguity in a fully-supervised manner [3, 22, 25, 41–43, 47, 60]. Volume-based methods [25, 41, 42, 51] unproject 2D features or heatmaps of joints to a 3D space for estimation. Another kind of method [22, 43, 60] utilizes the geometry information to fuse the features in 2D space directly and efficiently. Recently, some works [35,47] utilize transformers for implicit cross-view fusion without camera extrinsics.

Label-Efficient Learning. Label-efficient learning aims to reduce the 3D label requirements. Many works devote to solving hand pose estimation in a label-efficient manner [1, 4, 5, 7, 9, 39, 48, 55, 59, 63]. Synthetic data is used to avoid manual annotation [7,39,63], but may need domain transfer [39]. Or use weakly supervised learning [4,48] to obtain 3D results by manually annotating 2D labels to assist with hand priors. Multi-view label-efficient learning is also explored in 3D pose estimation [24, 27, 44, 52]. Rhodin *et al*. [44] trains a semi-supervised network with only a small amount of labeled 3D data and multi-view consistency constraints. Iqbal *et al*. [24] mixes single-view images with 2D labels and unlabelled multi-view images for training. Our goal is the same as that of previous methods, which is to train without any manual 3D labels.

Self-supervised 3D Pose Estimation. (1) Single-view training and inference. To the best of our knowledge, there is only one method for self-supervised 3D hand pose estimation, proposed by Chen *et al*. [9]. Their framework, S^2Hand, uses only single-view 2D noisy labels for training and achieves self-supervision through rendering. However, the performance is limited due to the use of single-view information and the quality of the noisy labels. (2) Multi-view training, single-view inference. Our approach belongs to this category but is fundamentally different from the existing methods. EpipolarPose [27] triangulates multi-view 2D pseudo labels according to epipolar geometry to 3D ones for training. CanonPose [52] learns to lift 2D pseudo labels to 3D canonical pose space with multi-view consistency constraints. All the aforementioned methods use non-learnable self-supervised modules like geometric modules or consistency loss functions, as shown in Fig. 1. However, they [27, 52] ignore the importance of introducing cross-view interaction and multi-view collaborative learning. Previous methods struggle to achieve good performance since the pose of a hand can change drastically over time and different joints may have similar appearances.

3. Method

As depicted in the left part of Fig. 2, our framework consists of a simple yet effective single-view estimator and cross-view interaction network. The core idea of our approach is that prediction from a monocular view can be enhanced via cross-view feature interaction and the interacted results can further supervise the single-view output to achieve self-distillation.

3.1. Single-View Estimator

Overview. Our framework takes multi-view synchronized hand images $\mathcal{I} = \{\boldsymbol{I}_i\}_{i=0}^{v}$ with v views as input, each view is an image of $\boldsymbol{I}_i \in \mathbb{R}^{3\times h\times w}$. The output is a 3D hand mesh $\boldsymbol{M}$ on each view. We designed a simple yet effective model-based network as a single-view estimator. Using the hand model will reduce the adverse effects of using poor pseudo labels as supervision by providing hand prior information for regularization. Please refer to supplementary materials for more details about the single-view estimator.

Hand model. We employ MANO [45] as the hand model. The hand mesh can be derived from the MANO layer using parameters $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$, *i.e*. $\boldsymbol{M}(\boldsymbol{\beta},\boldsymbol{\theta})$. $\boldsymbol{\beta} \in \mathbb{R}^{10}$ and $\boldsymbol{\theta} \in \mathbb{R}^{16\times 3}$ control the shape and pose of the hand respectively. We can use a predefined regressor to obtain the 3D joints from the 3D mesh vertices by $\boldsymbol{P} = \boldsymbol{J}\boldsymbol{M}$, where $\boldsymbol{J} \in \mathbb{R}^{k\times n}$, where $n = 778$ and $k = 21$ are the joints number and vertices number. For more details, we recommend referring to [45].

Camera model. Following Boukhayma *et al*. [4], we model the geometry correspondence by the weak-perspective camera model and obtain camera parameters from the single-view network predictions. Given the translation $\boldsymbol{t}$ and scale s, the 2D coordinates in image plane can be obtained by: $\Pi(\boldsymbol{P}) = s\Omega(\boldsymbol{P}) + \mathbf{t}$, where Ω is the orthographic projection and Π denotes the weak-perspective projection.

Network Structure. Since the single-view estimator is not the main component, for the sake of simplicity, we employ a CNN as the encoder F_e, and an MLP as the decoder F_d for regressing the MANO parameters. We have 3D hand mesh: $\boldsymbol{M}_i(\boldsymbol{\theta}_i, \boldsymbol{\beta}_i) = F_s(\boldsymbol{I}_i)$, where $F_s = F_d(F_e(\cdot))$ denotes the entire single-view network. The estimator also passes different levels of features $\boldsymbol{H}^j$ (where $\boldsymbol{H}^j$ is the intermediate feature of the encoder after j residual blocks, j=1, 2, 3, 4) to our cross-view interaction network.

3.2. Cross-view Interaction Network

In this section, we introduce the cross-view interaction network (CVI-Net), which is the core of our system to enable the network to exploit multi-view information. This stage conducts cross-view interaction and distillation. The critical components of this stage are a cross-view interaction network for capturing cross-view features and several consistent losses for guiding collaborative learning.

3.2.1 View-Shared Graph Feature Extraction

The first step for interaction is to extract the appropriate features. Different from [7, 8, 61], our module collects useful information into a graph through view-shared graph feature extraction module (VSGFE) as shown in Fig. 2. Specifically, it makes use of multi-level feature maps from dif-

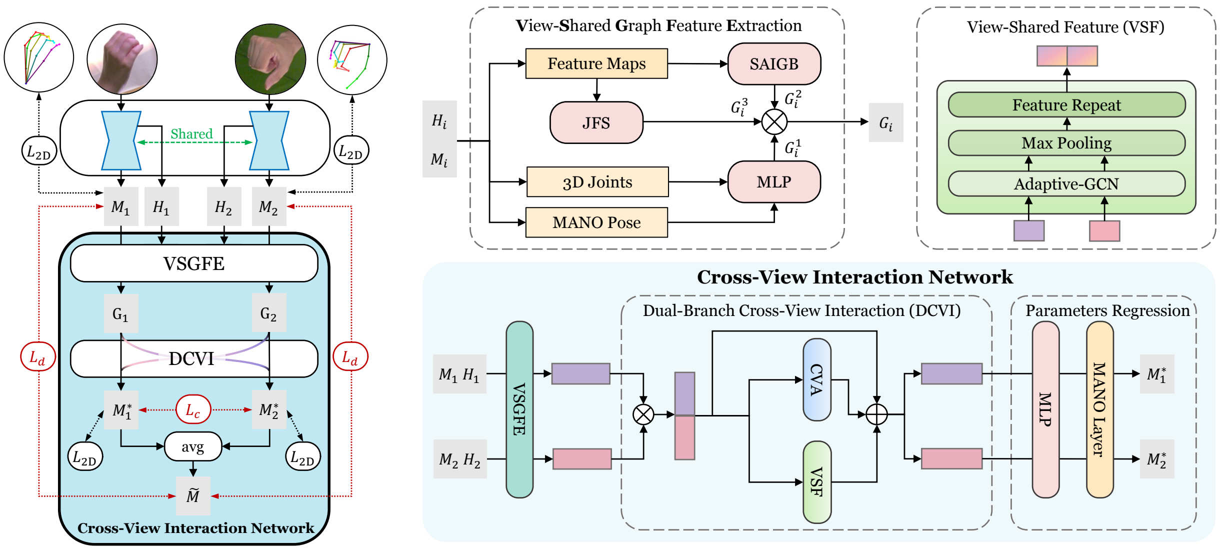

Figure 2. The left illustrates our whole pipeline (2 views for simplicity). During the training phase, the network takes multi-view hand images and pseudo-labels as inputs. The bottom right depicts our cross-view interaction networks. The top right shows the view-shared graph feature extraction (VSGFE) module and view-shared feature (VSF) module. $\oplus$ and $\otimes$ denotes add and concatenation respectively.

ferent views $\mathcal{H} = \{\boldsymbol{H}_i^j\}_{i=0}^v$, 3D joints $\mathcal{P} = \{\boldsymbol{P}_i\}_{i=0}^v$, and MANO pose parameters $\Theta = \{\boldsymbol{\theta}_i\}_{i=0}^v$ from the single-view estimator to extract a graph feature $\boldsymbol{G}$. The graph feature of each view $\boldsymbol{G}_i$ consists of three parts. $\boldsymbol{G}_i^1$, $\boldsymbol{G}_i^2$ and $\boldsymbol{G}_i^3$ aim to capture joint location features, global image features, and local image features, respectively. The first part is joint location embedding $\boldsymbol{G}_i^1 \in \mathbb{R}^{k \times c_1}$, providing the explicit geometric information. This embedding is obtained by using an MLP to map the single-view 3D joints locations $\boldsymbol{P}_i$ and pose parameters $\boldsymbol{\theta}_i$ to dimension c_1. The second part is joint-wise high-level image features $\boldsymbol{G}_i^2 \in \mathbb{R}^{k \times c_2}$ generated by spatial-aware initial graph building (SAIGB) [61] module using the last level feature maps $\boldsymbol{H}_i^4$. This part provides compact image clues of all views for interaction. The third part is joint-aligned features $\boldsymbol{G}_i^3 \in \mathbb{R}^{k \times c_3}$ gathered by joint feature sampler (JFS). JFS projects joints onto multi-level image feature maps $\{\boldsymbol{H}_i^j\}_{j=1}^3$ to gather fine-grained perceptual features like [53,54] for better local alignment. We then concatenate graph features to get $\boldsymbol{G}_i = [\boldsymbol{G}_i^1 \otimes \boldsymbol{G}_i^2 \otimes \boldsymbol{G}_i^3]$.

3.2.2 Dual-Branch Cross-View Interaction (DCVI)

We first stack $\{\boldsymbol{G}_i\}_{i=0}^v$ of all views to obtain multi-view graph feature $\boldsymbol{G} \in \mathbb{R}^{vk \times (c_1+c_2+c_3)}$. We design a component to effectively capture complementary information from other views on multi-view graph feature $\boldsymbol{G}$. The interaction module has two branches, (1) *cross-view attention branch* (CVA) and (2) *view-shared feature branch* (VSF). *Cross-view attention branch* utilizes a cross-view transformer F_t consisting of several multi-head attention layers with token size vk and MLPs, which allows each joint to aggregate features from other joints or views. This branch implicitly captures the multi-view information. An explicit multi-view prior information is that the observed poses from all the views should be consistent in 3D. Therefore, we add a branch to excavate the multi-view shared information to enhance the feature representation. Specifically, *view-shared feature branch* first employs adaptive-GCN [12] F_a to map the view-specific features $\boldsymbol{G}_i$ to a canonical feature space $\boldsymbol{C}_i = F_a(\boldsymbol{G}_i)$, the nodes in adaptive-GCN represents the hand joints and the edges represents joint feature correlation. Then, we stack $\mathcal{C} = \{\boldsymbol{C}_i\}_{i=0}^v$ together to get multi-view canonical features $\boldsymbol{C} \in \mathbb{R}^{v \times k \times (c_1+c_2+c_3)}$. After that, we use max-pooling on $\boldsymbol{C}$ to get the max activated features of every joint then repeat them in the view dimension as the view-shared features $\boldsymbol{C}' \in \mathbb{R}^{vk \times (c_1+c_2+c_3)}$. We denote the dual-branch cross-view interaction as: $\boldsymbol{G}^* = \boldsymbol{G} + F_t(\boldsymbol{G}) + \boldsymbol{C}'$, where $\boldsymbol{G}^*$ is the updated graph feature.

Parameters regression. The view specific feature $\boldsymbol{G}_i^*$ after the interaction can be obtained by reshaping $\boldsymbol{G}^*$. We then employ a shared MLP F_r as a decoder to regress the pose parameters $\boldsymbol{\theta}_i^* = F_r(\boldsymbol{G}_i^*)$ to derive the hand mesh of each view $\boldsymbol{M}_i^*(\boldsymbol{\theta}_i^*, \boldsymbol{\beta}_i)$ and corresponding joints $\boldsymbol{P}_i^* = \boldsymbol{J}\boldsymbol{M}_i^*$.

3.2.3 Multi-View Collaborative Learning

To allow all the views and the networks to learn collaboratively, we utilize consistency losses L_c upon interaction outputs and distillation loss L_d between multi-view fusion results and single-view outputs, as shown in Fig. 2. L_c introduces collaborative learning between multiple views, guiding the poses from different views to be as close as possible. While L_d makes the CVI-Net and single-view estimator work in a collaborative manner, achieving a self-distillation effect.

Results fusion. Since we need to supervise the single-view estimator with the results after the interaction, instead of simply using the refined results $\boldsymbol{M}_i^*$ of each view, we ensemble all the results into a unified and more reliable result $\tilde{\boldsymbol{M}}$. Considering the lack of explicit guidance, we empirically introduce a prior that all the views contribute equally. Thus, we simply average all aligned results to obtain $\tilde{\boldsymbol{M}}$. Specifically, we use A to denote the align procedure. When the extrinsics are known, we use the relative camera pose for alignment. When the camera extrinsics are unavailable, we use Procrustes analysis [62, 64] to compute relative rotation and align meshes to a canonical view. The final result is calculated as follows: $\tilde{\boldsymbol{M}} = \frac{1}{v}\sum_{i=1}^{v} A(\boldsymbol{M}_i^*)$.

Consistency losses. We design two types of consistency loss L_c: *2D consistency loss* $L_{c_{2D}}$ and *Fusion consistency loss* L_{c_f}. The motivation behind $L_{c_{2D}}$ is that the 2D predictions in the x-axis and y-axis are more accurate than the depth prediction in the z-axis. Therefore, $L_{c_{2D}}$ utilizes the 2D predictions in every single view as the pseudo label to supervise other views, which explores the view-specific reliable information to collaboratively improve the predictions of all the views. *2D consistency loss* is defined as: $L_{c_{2D}} = \frac{1}{v^2}\sum_{i=1}^{v}\sum_{j=1}^{v}\|\Pi(\boldsymbol{M}_i^*) - \Pi(A_i(\boldsymbol{M}_j^*))\|_1$, where $A_i(\cdot)$ denotes the alignment operation to align other view-j to view-i. *Fusion consistency loss* uses the fused results to supervise each view. The loss is defined as:$L_{c_f} = \frac{1}{v}\sum_{i=1}^{v}\|\boldsymbol{M}_i^* - A_i^{-1}(\tilde{\boldsymbol{M}})\|_1$, where $A_i^{-1}(\cdot)$ denotes the inverse transformation from canonical view to view-i. $L_{c_{2D}}$ and L_{c_f} are complementary to each other. Only using $L_{c_{2D}}$ tends to get performance saturation faster. In contrast, only adopting L_{c_f} can lead to unstable training since there may exist the situation that the fusion results are worse due to the majority of the predictions being wrong, especially at the early training stage. During training, we alternately update $L_{c_{2D}}$ and L_{c_f} to achieve more stable optimization.

Multi-view distillation loss. Since the multi-view fusion results are much better than the 2D pseudo label, we introduce multi-view distillation loss $L_d = \frac{1}{v}\sum_{i=1}^{v}\|\boldsymbol{M}_i - A_i^{-1}(\tilde{\boldsymbol{M}})\|_1$ that uses the fusion results to supervise the single-view outputs to achieve self-distillation.

Total loss. Except for the losses for multi-view collaborative learning, our framework also adopts two general constraints, 2D joints loss, and hand prior regularization. The prior regularization regularizes the pose and shape parameters: $L_p = \frac{1}{v}\sum_{i=1}^{v}\alpha(\|\boldsymbol{\theta}_i\|_1 + \|\boldsymbol{\theta}_i^*\|_1 + \gamma\|\boldsymbol{\beta}_i\|_1)$, where α and γ are used to balance the loss scale. The 2D joints $L1$ loss L_{2D} is used to supervise the results from the 2D pseudo labels. The final loss is defined as: $L = L_c + L_d + L_{2D} + L_p$.

4. Experiments

4.1. Datasets and Metrics

FreiHAND [64] is a dataset for single-view 3D hand pose estimation, which contains 130,240 training images and 3,960 testing images. All images are captured from the real world with 3D annotations. The training set consists of 32,560 composited images with four types of real-world backgrounds and hands captured against a green screen.

HanCo [62] extends FreiHAND, which consists of 1,517 videos with multiple views and camera calibration. It has 860,304 frames in total, *i.e.* 107,538 time-step per view. Since HanCo does not have an official train/test split, we use the first 1,200 sequences for training and the last 317 sequences for testing in all experiments for fair comparisons.

Other datasets. We also provide additional results on other datasets. Assembly101 [46] is an action recognition dataset that consists of 4,321 videos sequence. H2O [29] is a hand-object interaction dataset with 571,645 frames. Please refer to supplementary materials for details.

Metrics. We report standard metrics for hand pose estimation as follows. (1) **MPJPE/MPVPE** (mean per joint/vertex position error) measures the average Euclidean distance in mm between the predicted and ground-truth joints/vertices. JE/VE are the abbreviations for MPJPE/MPVPE. (2) **NMPJPE/NMPVPE** (normalized mean per joint/vertex position error, N-JE/VE) computes MPJPE/MPVPE after performing translation and scale alignment. (3) **PA-MPJPE/PA-MPVPE** (PA-JE/VE) is a modification of MPJPE/MPVPE with Procrustes analysis [17]. This metric normalizes the absolute scale, center, and rotation. (4) **F-Score** [9] is the harmonic mean of recall and precision between two meshes w.r.t. a specific distance threshold. F@5mm and F@15mm are reported. (5) **AUC** means the area under the curve of the PCK, where the PCK refers to the percentage of correct joints.

4.2. Implementation Details

We implement all the networks in PyTorch [40]. We first train our framework without L_c and L_d for 10 epochs. Then, we train the whole framework for another 30 epochs. Each batch contains images from 8 time-step of 8 cameras. We use AdamW [34] optimizer and set the initial learning rate to 3e-4. We use 256×256 hand images as input. Please refer to supplementary materials for more details.

4.3. Comparisons with state-of-the-arts

In Sec. 4.3.1, we evaluate the performance of our method under the single-view inference setting. As self-supervised hand pose estimation is a relatively new task, there is limited literature available for comparison. To address this, we adapt self-supervised body pose estimation methods [27,52] to hand and compare them with our method on HanCo [62]. We then compare with the only existing self-supervised hand pose estimation method, S^{2}Hand [9]. As S^{2}Hand can only be trained on single-view images, we use our single-view network only (denote as Ours-SV) for both training and inference as baselines. We further conduct extensive evaluations of our full model and baselines to demonstrate

Method	Input	N-JE↓	PA-JE↓
Fully-Supervised Method:			
MobRecon [7]	image	9.9	5.7
EpipolarPose [27]	image	10.5	6.1
Self-Supervised Method:			
EpipolarPose [27]	image, 📷	19.7	9.3
CanonPose [52]	2D pose, 📷	30.9	12.6
Ours	image, 📷	**11.1**	**7.0**
EpipolarPose [27]	image	42.3	23.5
CanonPose [52]	2D pose	31.8	12.8
Ours	image	**15.2**	**7.7**

Table 1. Single-view inference comparisons on the HanCo [62] dataset. 📷 denotes the method using camera extrinsics during training. Notably, in the self-supervised setting, our method exhibits a significant improvement over previous methods.

Method	Data	Backbone	PA-JE↓	PA-VE↓	F@5↑
Fully-Supervised Method:					
YoutubeHand [28]	Frei.	Res50	8.4	8.6	0.61
I2UV-HandNet [6]	Frei.	Res50	6.7	6.9	0.71
MobRecon [7]	Frei.	Res50†	6.1	6.2	0.76
Ours-SV	Frei.	Res50	7.5	7.5	0.68
Self-Supervised Method:					
S^{2}HAND [9]	Frei.	EffiNet-b0	11.8	11.9	0.48
Ours-SV	Frei.	EffiNet-b0	11.6	11.7	0.49
Ours-SV	Frei.	Res50	11.9	12.0	0.47
Ours-SV	HanCo	EffiNet-b0	11.3	11.4	0.51
Ours-SV	HanCo	Res50	11.6	11.8	0.48
Ours	HanCo	EffiNet-b0	6.3	6.8	0.71
Ours	HanCo	Res50	**6.2**	**6.7**	**0.72**

Table 2. Quantitative results on the FreiHAND evaluation set. The notation † denotes using a stacked backbone structure. "Our-SV" refers to training only with our single-view network.

the efficacy of multi-view collaborative learning.

In addition, thanks to our cross-view interaction network, our approach is capable of performing multi-view inference by simply averaging individual view results when multi-view test data is available. In Sec. 4.3.2, we compare our method with state-of-the-art approaches under the multi-view inference setting.

4.3.1 Single-View Inference

Hanco. We train EpipolarPose and CanonPose using their open-source code. We also train fully-supervised methods [7, 27] as a reference for performance. Tab. 1 outlines the performance of fully-/self-supervised methods in the literature along with ours. In the case where camera extrinsics are available for training, CanonPose performs the worst because it lifts noisy 2D pseudo labels from OpenPose to 3D ones. When camera extrinsics are not available, all competitors experience a performance decline. This is due to the lack of collaborative interaction across multi-view features in previous self-supervised methods. In contrast, our method outperforms both of them by a large margin. Our cross-view interaction networks can enhance single-view inference, whether camera extrinsics are available during training or not. More details about the usage of cameras can be found in Sec. 3.2.3. Compared to previous self-supervised methods, our approach significantly improves performance, highlighting the importance of cross-view interaction among different views. Moreover, our approach can get comparable results to fully-supervised methods.

FreiHAND. The comparisons on the evaluation set are shown in Tab. 2. The experiments conducted under self-supervised settings indicate that our baselines, Ours-SV, already achieve performance comparable to S^{2}Hand. Moreover, directly equipping baselines with other backbones or more training data does not improve too much. We argue that performance improvements in single-view self-supervised hand pose estimation cannot be achieved by changing the backbone architecture or increasing the amount of training data. In contrast, our full model, *i.e.* Ours, substantially further improves the results on the FreiHand dataset, which justify the effectiveness of multi-view collaborative learning. Moreover, our self-supervised approach achieves competitive performance with recent fully-supervised state-of-the-art methods

Method	MPJPE ↓	PA-MPJPE ↓
Ttraditional Triangulation Method (w/o training):		
DLT [21]	16.8	13.2
Pictorial [11]	13.5	10.2
RANSAC [25]	12.3	9.8
Fully-Supervised Method:		
EpipolarTrans [22]	6.2	4.2
LT-Algebraic [25]	5.5	3.6
LT-Volumetric [25]	5.8	3.6
LT-Volumetric$^+$ [25]	**4.9**	3.6
EpipolarPose$^+$ [27]	8.0	4.4
Ours (Opt-Center)	6.0	**3.2**
Ours (RANSAC)	5.8	3.4
Self-Supervised Method:		
EpipolarTrans [22]	11.2	9.0
LT-Algebraic [25]	10.3	7.8
LT-Volumetric [25]	10.6	8.0
LT-Volumetric$^+$ [25]	9.5	7.2
CanonPose$^+$ [52]	21.6	10.5
EpipolarPose$^+$ [27]	17.2	8.3
Ours (Opt-Center)	8.8	**5.3**
Ours (RANSAC)	**8.5**	5.6

Table 3. Multi-view inference results on the HanCo dataset. The notation $^+$ indicates that methods require the GT 3D center.

4.3.2 Multi-View Inference

We show the quantitative results of our multi-view inference performance with other competitors on HanCo in Tab. 3. A naive solution is to triangulate pseudo labels without training. We show the performance of traditional meth-

ID	Method	NMPJPE ↓			PA-MPJPE ↓		
		Single	Interact	Fusion	Single	Interact	Fusion
		ResNet-50 as the backbone:					
1	Full	**11.14**$_{\uparrow 0.03}$	8.31$_{\downarrow 0.03}$	**7.65**$_{\uparrow 0.10}$	**7.05**$_{\uparrow 0.17}$	**5.35**$_{\uparrow 0.07}$	**5.34**$_{\uparrow 0.06}$
		ResNet-18 as the backbone:					
2	Full	11.17	**8.28**	7.75	7.22	5.42	5.40
3	– VSF	11.21$_{\downarrow 0.04}$	8.49$_{\downarrow 0.21}$	7.81$_{\downarrow 0.06}$	7.25$_{\downarrow 0.03}$	5.52$_{\downarrow 0.10}$	5.50$_{\downarrow 0.10}$
4	– CVA	11.31$_{\downarrow 0.14}$	8.45$_{\downarrow 0.17}$	7.81$_{\downarrow 0.06}$	7.29$_{\downarrow 0.07}$	5.48$_{\downarrow 0.06}$	5.46$_{\downarrow 0.06}$
5	– $\boldsymbol{G}^1$	11.31$_{\downarrow 0.14}$	8.56$_{\downarrow 0.28}$	7.77$_{\downarrow 0.03}$	7.31$_{\downarrow 0.09}$	5.52$_{\downarrow 0.10}$	5.49$_{\downarrow 0.09}$
6	– $\boldsymbol{G}^2$	11.33$_{\downarrow 0.16}$	8.38$_{\downarrow 0.10}$	7.83$_{\downarrow 0.08}$	7.34$_{\downarrow 0.08}$	5.45$_{\downarrow 0.03}$	5.42$_{\downarrow 0.02}$
7	– $\boldsymbol{G}^3$	11.30$_{\downarrow 0.13}$	8.99$_{\downarrow 0.69}$	7.82$_{\downarrow 0.07}$	7.30$_{\downarrow 0.08}$	5.45$_{\downarrow 0.03}$	5.44$_{\downarrow 0.04}$
8	– $L_{c_{2D}}$	11.25$_{\downarrow 0.08}$	8.43$_{\downarrow 0.15}$	7.90$_{\downarrow 0.15}$	7.32$_{\downarrow 0.10}$	5.58$_{\downarrow 0.16}$	5.57$_{\downarrow 0.17}$
9	– L_{c_f}	11.74$_{\downarrow 0.57}$	8.98$_{\downarrow 0.70}$	8.38$_{\downarrow 0.63}$	7.55$_{\downarrow 0.33}$	5.84$_{\downarrow 0.42}$	5.80$_{\downarrow 0.40}$
10	– DCVI	13.52$_{\downarrow 2.35}$	/	11.99$_{\downarrow 4.24}$	9.59$_{\downarrow 2.37}$	/	9.42$_{\downarrow 4.02}$
11	– L_c	14.04$_{\downarrow 2.87}$	17.03$_{\downarrow 8.75}$	10.32$_{\downarrow 2.57}$	9.04$_{\downarrow 1.82}$	10.21$_{\downarrow 4.79}$	7.92$_{\downarrow 2.52}$
12	– L_d	17.05$_{\downarrow 5.88}$	8.56$_{\downarrow 0.28}$	8.01$_{\downarrow 0.26}$	10.13$_{\downarrow 2.91}$	5.67$_{\downarrow 0.25}$	5.65$_{\downarrow 0.25}$

Table 4. Quantitative ablation studies. We remove each of our components here to show their contribution to our framework. Full denotes our complete model. CVI represents our whole cross-view interaction network. Other notations are consistent with Sec. 3. We report the errors of single-view outputs (Single, $\boldsymbol{M}$), cross-view interaction outputs (Interact, $\boldsymbol{M}^*$), and multi-view fusion results (Fusion, $\tilde{\boldsymbol{M}}$).

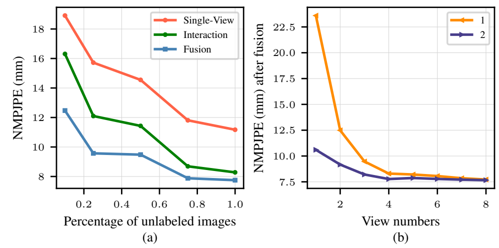

Figure 3. Error of using different (a) #training data, (b) (line-1)#view for training , and (line-2)#view for inference when trained with 8 views.

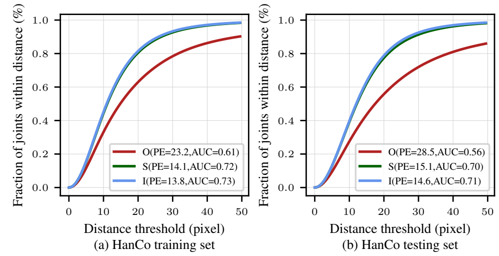

Figure 4. AUC of three 2D joint sets. O, S, I, PE denote OpenPose, single-view, interaction, and average pixel error in resolution 256×256.

ods. Such methods can serve as a reference for evaluating the effectiveness of self-supervised methods. We adapt fully-supervised multi-view 3D pose estimation methods LT [25] and EpipolarTrans [22] to a self-supervised manner. Under self-supervised settings, EpipolarTrans can only achieve limited performance improvements compared to traditional methods. LT-Algebraic [25], which incorporates learnable confidence into the triangulation. LT-Volumetric model [25], which unprojects 2D features into a 3D volume for inference, achieves better results, but the performance is dependent on the accuracy of the hand center. CanonPose [52] and EpipolarPose [27] obtain multi-view inference results through simple averaging like ours.

However, both of these methods are inferior to ours because they lack cross-view interaction. As our method predicts the root-relative 3D pose, we need to conduct post-processing to obtain the absolute coordinates. We introduce two different ways to achieve this: 1) using the 2D predictions of different views to triangulate and refine a center and 2) conducting RANSAC triangulation using our 2D predictions. Both methods have their merits. Opt-center can keep the root-relative results with hand prior, resulting in low PA-MPJPE. RANSAC gets better joint-wise accuracy, which is indicated by low MPJPE. We also provide qualitative results in the supplementary materials on the Assembly101 [46] dataset, which has a static camera setup. Even for challenging head-mounted moving cameras, we achieve convincing 3D pose estimates on the H2O [29] dataset. The experiments show that we have significantly pushed the performance of self-supervised methods to a comparable level with fully supervised methods.

4.4. Qualitative Result

Fig. 5 presents the visual comparisons of 2 views between 2D joints of OpenPose, ours, and ground-truth on the HanCo dataset. We can observe that our method is more robust for outliers and can generate predictions close to the labels. Fig. 6 shows the 3D predictions from two viewpoints of ours, EpipolarPose, and CanonPose on the HanCo dataset. The results indicate that our method can get more accurate results especially when the occlusions are severe. Please refer to supplementary materials for more results.

4.5. Ablation Study

As shown in Tab. 4, we conduct comprehensive ablation experiments on the HanCo [62] dataset to show the effectiveness of each component. Single, Interact and Fusion denotes the evaluation of $\boldsymbol{M}$, $\boldsymbol{M}^*$ and $\tilde{\boldsymbol{M}}$ respectively.

Different backbones. We first show our performance with different backbones. As shown in #1 and #2, using a large backbone like Res50, our performance can be further improved. For efficiency, we conduct ablation studies using Res18 as the backbone unless otherwise specified.

Two branches for cross-view interaction module. As presented in #3 and #4, both of the branches can reduce the error. VSF can explicitly model the view-shared information and add reliable information from every view. CVA can capture the self-/cross-view joint-level correlations.

Graph features. The results indicate that three kinds of features (#5, #6, #7) all lead to performance improvement. Especially, local feature (#7, $\boldsymbol{G}^3$) can notably reduce the error after the interaction by providing fine-grained details.

DCVI. We also conduct experiments to show the importance of DCVI by removing it and posing consistency constraints in single-view outputs like [24]. In this way, the performance drops dramatically (#10), proving the necessity of using DCVI to capture the features of all the views for self-supervised learning.

Two branches for multi-view consistency loss. Without enforcing cross-view interaction outputs to be consistent, the performance significantly drops (#9). If we do not explore relatively more reliable 2D predictions to enhance consistency, the performance can also get worse (#8).

Consistency losses. (L_c) The performance is unsatisfactory (#11) when employing the cross-view interaction network without any consistency constraints (*i.e.* discard #8 and #9). The interaction network should cooperate with consistency so that the constraints can guide the network to exploit multi-view information to function better.

Multi-view distillation loss. (L_d) Removing the multi-view distillation loss, all the metrics drop by a large margin (#12), especially in single-view estimation accuracy. This phenomenon proves the effectiveness of collaborative learning between single- and multi-view networks.

4.6. Model Analysis

Different percentage of unlabeled images. Fig. 3 (a) shows our method can get consistent performance improvement as the unlabeled training data increases.

Different view number for training. The line-1 in Fig. 3 (b) shows the performance of our method tested on a certain view when trained with different view numbers. The curve shows that our method can be consistently improved as the number of views increases. We also observe that using multiple views for training can significantly improve performance when the valid views are few.

Different view number for inference. Our model allows inferring with an arbitrary number of views. However, when the model is trained with a fixed view number, it could get the view number bias, resulting in better performance using the view number close to the training one. To avoid this, we add random masks in our interaction module and finetune the model for a few epochs. After that, results can get better by a small margin (the single-view error is 11.07mm and the fusion error 7.60mm, both in NMPJPE.). The line-2 in Fig. 3 (b) shows results on a certain view when trained on 8 views and tested on 1 to 8 views. We can observe consistent improvement with the inference view number increases.

Different 2D joint sets. Fig. 4 presents the accuracy of different 2D joint sets on the HanCo training and testing set. Our 2D predictions are extremely better than OpenPose 2D pseudo label used for training.

Iteratively training. Our approach can use the previous predictions as pseudo labels for iterative training. We find it helpful till iteration 3 and get saturated afterward. From 1 to 3 iterations, NMPJPE is 7.75, 7.68, and 7.64.

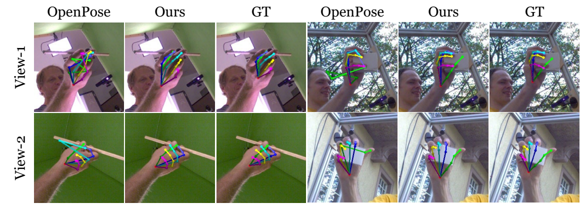

Figure 5. 2D prediction (overlayed in the images) comparisons between OpenPose, ours, and ground-truth on the HanCo dataset.

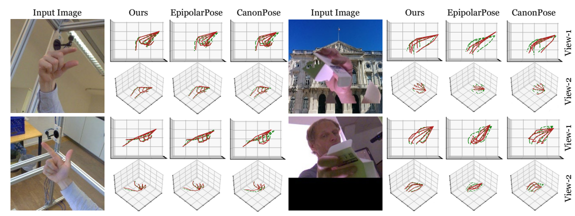

Figure 6. 3D prediction comparisons between our method, EpipolarPose, and CanonPose on the HanCo dataset. Our prediction and ground-truth are shown in solid red and dashed green respectively.

5. Conclusion and Future Work

To our best knowledge, we present the first self-supervised framework that aims to learn a single-view 3D hand estimator from unlabeled multi-view data. At the core of our approach, a cross-view interaction network is carefully designed to supervise the single-view output by leveraging the collaboration among multi-views. Specifically, the network captures the interdependencies of features among different views, resulting in improved accuracy of hand pose estimation after cross-view interaction. Additionally, the multi-view results are fused to supervise the single-view output for self-distillation. The effectiveness and versatility of the proposed framework are extensively evaluated through experiments, which demonstrate that our method not only establishes a new benchmark for self-supervised 3D hand pose estimation from single-view input but also offers flexible multi-view inference with state-of-the-art performance.

We focused on hand pose estimation without heavy occlusions in this work. Extending our work to more challenging scenarios, such as hand-object interaction or relaxing the synchronization constraints in multi-view inputs, would be interesting topics for further study.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grants (62071067, 62171057, 62201072), in part by the Ministry of Education and China Mobile Joint Fund (MCM20200202), Beijing University of Posts and Telecommunications-China Mobile Research Institute Joint Innovation Center, in part by the Project funded by China Postdoctoral Science Foundation (2023TQ0039).

References

[1] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1067–1076, 2019. 2, 3

[2] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6121–6131, 2020. 2

[3] Kristijan Bartol, David Bojanić, Tomislav Petković, and Tomislav Pribanić. Generalizable human pose triangulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11028–11037, 2022. 3

[4] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3d hand shape and pose from images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10843–10852, 2019. 2, 3

[5] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 666–682, 2018. 2, 3

[6] Ping Chen, Yujin Chen, Dong Yang, Fangyin Wu, Qin Li, Qingpei Xia, and Yong Tan. I2uv-handnet: Image-to-uv prediction network for accurate and high-fidelity 3d hand mesh modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12929–12938, 2021. 2, 6

[7] Xingyu Chen, Yufeng Liu, Yajiao Dong, Xiong Zhang, Chongyang Ma, Yanmin Xiong, Yuan Zhang, and Xiaoyan Guo. Mobrecon: Mobile-friendly hand mesh reconstruction from monocular image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20544–20554, 2022. 2, 3, 6

[8] Xingyu Chen, Yufeng Liu, Chongyang Ma, Jianlong Chang, Huayan Wang, Tian Chen, Xiaoyan Guo, Pengfei Wan, and Wen Zheng. Camera-space hand mesh recovery via semantic aggregation and adaptive 2d-1d registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13274–13283, 2021. 2, 3

[9] Yujin Chen, Zhigang Tu, Di Kang, Linchao Bao, Ying Zhang, Xuefei Zhe, Ruizhi Chen, and Junsong Yuan. Model-based 3d hand reconstruction via self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10451–10460, 2021. 1, 2, 3, 5, 6

[10] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *European Conference on Computer Vision*, pages 769–787. Springer, 2020. 2

[11] Junting Dong, Wen Jiang, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Fast and robust multi-person 3d pose estimation from multiple views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7792–7801, 2019. 6

[12] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David J Crandall. Hope-net: A graph-based model for hand-object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6608–6617, 2020. 2, 4

[13] Zicong Fan, Adrian Spurr, Muhammed Kocabas, Siyu Tang, Michael J Black, and Otmar Hilliges. Learning to disambiguate strongly interacting hands via probabilistic per-pixel part segmentation. In *2021 International Conference on 3D Vision (3DV)*, pages 1–10. IEEE, 2021. 2

[14] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3593–3601, 2016. 2

[15] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1991–2000, 2017. 2

[16] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10833–10842, 2019. 2

[17] John C Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975. 5

[18] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3196–3206, 2020. 2

[19] Shangchen Han, Beibei Liu, Randi Cabezas, Christopher D Twigg, Peizhao Zhang, Jeff Petkau, Tsz-Ho Yu, Chun-Jung Tai, Muzaffer Akbay, Zheng Wang, et al. Megatrack: monochrome egocentric articulated hand-tracking for virtual reality. *ACM Trans. Graph.*, 39(4):87, 2020. 1

[20] Shangchen Han, Po-chen Wu, Yubo Zhang, Beibei Liu, Linguang Zhang, Zheng Wang, Weiguang Si, Peizhao Zhang, Yujun Cai, Tomas Hodan, Cabezas Randi, Tran Luan, Akbay Muzaffer, Yu Tsz-Ho, Keskin Cem, and Wang Robert. Umetrack: Unified multi-view end-to-end hand tracking for vr. *ACM Transactions on Graphics*, 2022. 1

[21] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 6

[22] Yihui He, Rui Yan, Katerina Fragkiadaki, and Shoou-I Yu. Epipolar transformers. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 7779–7788, 2020. 3, 6, 7

[23] Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5 d heatmap regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 118–134, 2018. 2

[24] Umar Iqbal, Pavlo Molchanov, and Jan Kautz. Weakly-supervised 3d human pose learning via multi-view images

in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5243–5252, 2020. 1, 3, 8

[25] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7718–7727, 2019. 3, 6, 7

[26] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations*, 2014. 2

[27] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Self-supervised learning of 3d human pose using multi-view geometry. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1077–1086, 2019. 1, 2, 3, 5, 6, 7

[28] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4990–5000, 2020. 2, 6

[29] Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2o: Two hands manipulating objects for first person interaction recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10138–10148, 2021. 2, 5, 7

[30] Mengcheng Li, Liang An, Hongwen Zhang, Lianpeng Wu, Feng Chen, Tao Yu, and Yebin Liu. Interacting attention graph for single image two-hand reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2761–2770, 2022. 2

[31] Moran Li, Yuan Gao, and Nong Sang. Exploiting learnable joint groups for hand pose estimation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 1921–1929, 2021. 2

[32] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1954–1963, 2021. 2

[33] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12939–12948, 2021. 2

[34] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations*, 2019. 5

[35] Haoyu Ma, Zhe Wang, Yifei Chen, Deying Kong, Liangjian Chen, Xingwei Liu, Xiangyi Yan, Hao Tang, and Xiaohui Xie. Ppt: token-pruned pose transformer for monocular and multi-view human pose estimation. In *European Conference on Computer Vision*, pages 424–442. Springer, 2022. 3

[36] 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 *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pages 5079–5088, 2018. 2

[37] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *European Conference on Computer Vision*, pages 752–768. Springer, 2020. 2

[38] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2. 6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *European Conference on Computer Vision*, pages 548–564. Springer, 2020. 2

[39] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–59, 2018. 2, 3

[40] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS 2017 Workshop on Autodiff*, 2017. 5

[41] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Harvesting multiple views for marker-less 3d human pose annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6988–6997, 2017. 3

[42] Haibo Qiu, Chunyu Wang, Jingdong Wang, Naiyan Wang, and Wenjun Zeng. Cross view fusion for 3d human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4342–4351, 2019. 3

[43] Edoardo Remelli, Shangchen Han, Sina Honari, Pascal Fua, and Robert Wang. Lightweight multi-view 3d pose estimation through camera-disentangled representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6040–6049, 2020. 3

[44] Helge Rhodin, Jörg Spörri, Isinsu Katircioglu, Victor Constantin, Frédéric Meyer, Erich Müller, Mathieu Salzmann, and Pascal Fua. Learning monocular 3d human pose estimation from multi-view images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8437–8446, 2018. 3

[45] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6), 2017. 2, 3

[46] Fadime Sener, Dibyadip Chatterjee, Daniel Shelepov, Kun He, Dipika Singhania, Robert Wang, and Angela Yao. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21096–21106, 2022. 2, 5, 7

[47] Hui Shuai, Lele Wu, and Qingshan Liu. Adaptive multi-view and temporal fusing transformer for 3d human pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 3

[48] Adrian Spurr, Umar Iqbal, Pavlo Molchanov, Otmar Hilliges, and Jan Kautz. Weakly supervised 3d hand pose estimation via biomechanical constraints. In *European Conference on Computer Vision*, pages 211–228. Springer, 2020. 2, 3

[49] Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 89–98, 2018. 2

[50] Xiao Tang, Tianyu Wang, and Chi-Wing Fu. Towards accurate alignment in real-time 3d hand-mesh reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11698–11707, 2021. 2

[51] Hanyue Tu, Chunyu Wang, and Wenjun Zeng. Voxelpose: Towards multi-camera 3d human pose estimation in wild environment. In *European Conference on Computer Vision*, pages 197–212. Springer, 2020. 3

[52] Bastian Wandt, Marco Rudolph, Petrissa Zell, Helge Rhodin, and Bodo Rosenhahn. Canonpose: Self-supervised monocular 3d human pose estimation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13294–13304, 2021. 1, 2, 3, 5, 6, 7

[53] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 52–67, 2018. 4

[54] Chao Wen, Yinda Zhang, Zhuwen Li, and Yanwei Fu. Pixel2mesh++: Multi-view 3d mesh generation via deformation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1042–1051, 2019. 4

[55] Linlin Yang, Shicheng Chen, and Angela Yao. Semihand: Semi-supervised hand pose estimation with consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11364–11373, 2021. 2, 3

[56] Linlin Yang, Shile Li, Dongheui Lee, and Angela Yao. Aligning latent spaces for 3d hand pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2335–2343, 2019. 2

[57] Linlin Yang and Angela Yao. Disentangling latent hands for image synthesis and pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9877–9886, 2019. 2

[58] Xiong Zhang, Hongsheng Huang, Jianchao Tan, Hongmin Xu, Cheng Yang, Guozhu Peng, Lei Wang, and Ji Liu. Hand image understanding via deep multi-task learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11281–11292, 2021. 2

[59] Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular rgb image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2354–2364, 2019. 2, 3

[60] Zhe Zhang, Chunyu Wang, Weichao Qiu, Wenhu Qin, and Wenjun Zeng. Adafuse: Adaptive multiview fusion for accurate human pose estimation in the wild. *International Journal of Computer Vision*, 129(3):703–718, 2021. 3

[61] Xiaozheng Zheng, Pengfei Ren, Haifeng Sun, Jingyu Wang, Qi Qi, and Jianxin Liao. Sar: Spatial-aware regression for 3d hand pose and mesh reconstruction from a monocular rgb image. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 99–108. IEEE, 2021. 2, 3, 4

[62] Christian Zimmermann, Max Argus, and Thomas Brox. Contrastive representation learning for hand shape estimation. In *DAGM German Conference on Pattern Recognition*, pages 250–264. Springer, 2021. 1, 2, 5, 6, 7

[63] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE international conference on computer vision*, pages 4903–4911, 2017. 2, 3

[64] 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 images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 813–822, 2019. 2, 5