Single-to-Dual-View Adaptation for Egocentric 3D Hand Pose Estimation

Ruicong Liu Takehiko Ohkawa Mingfang Zhang Yoichi Sato
The University of Tokyo, Japan
{lruicong, ohkawa-t, mfzhang, ysato}@iis.u-tokyo.ac.jp

Abstract

The pursuit of accurate 3D hand pose estimation stands as a keystone for understanding human activity in the realm of egocentric vision. The majority of existing estimation methods still rely on single-view images as input, leading to potential limitations, e.g., limited field-of-view and ambiguity in depth. To address these problems, adding another camera to better capture the shape of hands is a practical direction. However, existing multi-view hand pose estimation methods suffer from two main drawbacks: 1) Requiring multi-view annotations for training, which are expensive. 2) During testing, the model becomes inapplicable if camera parameters/layout are not the same as those used in training. In this paper, we propose a novel Single-to-Dual-view adaptation (S2DHand) solution that adapts a pre-trained single-view estimator to dual views. Compared with existing multi-view training methods, 1) our adaptation process is unsupervised, eliminating the need for multi-view annotation. 2) Moreover, our method can handle arbitrary dual-view pairs with unknown camera parameters, making the model applicable to diverse camera settings. Specifically, S2DHand is built on certain stereo constraints, including pair-wise cross-view consensus and invariance of transformation between both views. These two stereo constraints are used in a complementary manner to generate pseudo-labels, allowing reliable adaptation. Evaluation results reveal that S2DHand achieves significant improvements on arbitrary camera pairs under both in-dataset and cross-dataset settings, and outperforms existing adaptation methods with leading performance. Project page: `https://github.com/ut-vision/S2DHand`.

1. Introduction

Delving into the realm of egocentric vision (first-person view), the pursuit of refining 3D hand pose estimation stands as a keystone for understanding human activity. This quest not only forges new paths in human-computer interaction [31, 34, 38], but also empowers imitation learning [8, 13, 37]. Moreover, it enhances the immersive experience in augmented/virtual reality (AR/VR) to new heights [15, 33]. Recently, with the advancements of AR/VR headsets, egocentric data has become increasingly prevalent [5, 10], leading to an increasing demand for estimating 3D hand poses from egocentric viewpoints.

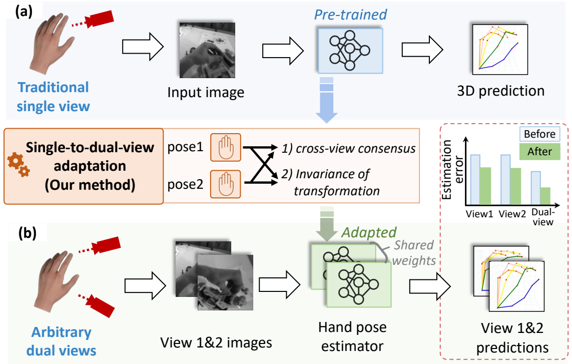

Figure 1. From (a) to (b), our single-to-dual-view adaptation method adapts a traditional single-view hand pose estimator to arbitrary dual views and achieves better accuracy. (a) Traditional single-view hand pose estimation. (b) Inference process of the adapted model under a dual-view setting.

To achieve better 3D hand pose estimation performance, recent years have witnessed many networks with various structures [7, 39, 46]. However, the majority of existing hand pose estimation methods are still under a single-view setting, which is convenient but leads to potential limitations, *e.g.*, limited field-of-view and ambiguity in depth. To address these problems, a potential solution is to add another camera to expand the field-of-view and reduce depth ambiguity by capturing the hand shape from an additional view angle. Furthermore, the use of multiple cameras also aligns with industry trends, as demonstrated by the latest AR/VR headsets such as the Apple Vision Pro and Meta Quest, which feature multiple egocentric cameras. Overall, an unavoidable trend towards multi-view settings in hand pose estimation is emerging, driven by its technological advantages and the direction of industrial development.

Currently, several existing studies [4, 12, 19] have paid attention to hand pose estimation under multi-view settings.

This work was supported by JSPS KAKENHI Grant Number JP20H04205 and JST ACT-X Grant Number JPMJAX2007.

Table 1. Differences among traditional single-view methods, multi-view training methods, and our proposed single-to-dual-view adaptation method. Green indicates lower training requirements, enhanced testing results, and reduced camera parameter requirements, respectively.

Methods	Pre-training dataset required	Test	Camera parameters
Traditional single-view methods [7, 39, 46]	Single-view (common single-camera setting)	Single-view	Not required
Multi-view training methods [4, 12, 19]	Dual-view (need multi-camera setting)	Dual-view & same camera poses with training	Required & same with training
Single-to-dual-view adaptation (Ours)	Single-view (common single-camera setting)	Dual-view & arbitrary camera poses	Not required

These methods typically process input images from multiple views simultaneously, utilizing a feature fusion module to arrive at a final prediction [22, 41]. However, all these methods have two significant drawbacks that limit their applicability. 1) The training, especially for the feature fusion module, necessitates multi-view labels, which are costly to annotate. 2) During testing, the same camera parameters as in training must be used. An estimator trained under a specific multi-camera setup becomes inapplicable if there are any changes to the camera layout or parameters.

Unlike existing multi-view training methods, we propose a new solution that adapts an estimator from single-view to dual-view without needing multi-view labels or camera parameters. As shown in Fig. 1, given a pre-trained estimator, our method adapts it to an arbitrary dual-view setting (from (a) to (b)), where two cameras are placed in any layout without knowing their parameters. Here, all we need is a pre-trained estimator and a sufficient number of unlabeled dual-view inputs from the two cameras. As compared in Tab. 1 (row 2-3), in contrast to multi-view training, our method only needs common and cheaper single-view data for training. During testing, unlike existing methods, our method is compatible with arbitrary dual-view pairs, making the model applicable to flexible and changeable camera settings. Specifically, when camera settings change, it is easy and swift to repeat our method's adaptation process to re-adapt the pre-trained estimator to work well with new camera parameters. For camera parameters, existing methods not only need them for training, but also require them the same with testing. Conversely, our method is clearly more practical since no camera parameters are required.

Building on these advancements, we present a novel unsupervised Single-to-Dual-view adaptation framework (S2DHand) for egocentric 3D hand pose estimation. It uses certain stereo constraints for adaptation, including cross-view consensus (pair-wise) and invariance of transformation between both camera coordinate systems (to all input pairs). These two stereo constraints are used in a complementary manner to refine the accuracy of pseudo-labels, allowing the model to better fit to the dual views. Specifically, the cross-view consensus is leveraged through an attention-based merging module, and the invariance of transformation is utilized via a rotation-guided refinement module.

We evaluate our method by adapting a pre-trained estimator to several dual-camera pairs placed in arbitrary poses [30]. Our evaluation encompasses both in-dataset and cross-dataset scenarios. Experimental results reveal that our technique not only realizes notable improvements across all pairs but also surpasses state-of-the-art adaptation methods. The primary contributions of this paper are summarized as:

- We propose a novel unsupervised single-to-dual-view adaptation (S2DHand) solution for egocentric 3D hand pose estimation. Our method can adapt a traditional single-view estimator for arbitrary dual views without requiring annotations or camera parameters.
- We build a pseudo-label-based strategy for adaptation. It leverages cross-view consensus and invariance of transformation between both camera coordinate systems for reliable pseudo-labeling. This leads to two key modules: attention-based merging and rotation-guided refinement.
- Evaluation results demonstrate the benefits of our approach for arbitrarily placed camera pairs. Our method achieves significant improvements for all pairs both under in-dataset and cross-dataset settings.

2. Related Work

2.1. Multi-view hand pose estimation

Multi-view hand pose estimation accepts multi-view images as input and outputs a final 3D hand pose, which remains a relatively unexplored research area. Previous works [4, 12, 19] design various network structures to predict 3D hand poses through a multi-view fusion module. Similarly, multiple studies have been done [44, 45] through feature fusion for body pose estimation under multi-view settings.

All these studies have two limitations: 1) they require costly multi-view images and annotations for training, 2) during testing, camera poses are assumed to be known and covered by training data, thereby limiting their applicability. In contrast, our method eliminates the need for multi-view annotations and is adaptable to arbitrary dual views.

2.2. Adaptation in hand pose estimation

Adaptation aims at tailoring a model for specific application scenarios [20, 24, 29]. Existing adaptation methods in hand pose estimation mainly focus on adaptation across different domains (datasets), *e.g.*, entropy minimization [17, 32], consistency regularization [3, 21, 29], and pseudo-labeling methods [16, 35, 42]. Prior works only use constraints in single-view settings for adaptation, *e.g.*, bio-mechanical constraint [21, 42], and are thus limited to single-view inference. Unlike these methods, we propose stereo constraints from dual views for adaptation and supports dual-view inference, which extends the application scenarios.

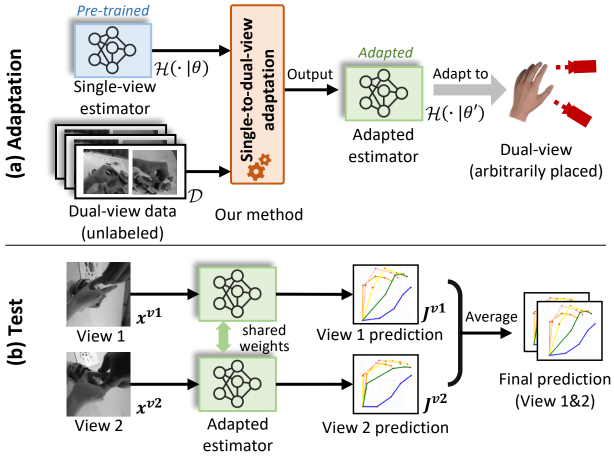

Figure 2. Problem setting of single-to-dual-view adaptation for hand pose estimation. (a) The input and output of adaptation. (b) The dual-view testing scheme after adaptation.

3. Problem Setting

Fig. 2 illustrates the task setting of single-to-dual-view adaptation for hand pose estimation. We denote unlabeled dual-view data as $\mathcal{D} = \{\mathbf{x}_i^{v1}, \mathbf{x}_i^{v2}|_{i=1}^N\}$, where $\mathbf{x}_i^{v1}$ and $\mathbf{x}_i^{v2}$ denote the i-th image from view1 and view2, respectively, N is the number of image pairs. The dual-view data $\mathcal{D}$ contains no ground-truth hand poses or camera parameters.

As shown in Fig. 2 (a), suppose we have a baseline hand pose estimator $\mathcal{H}(\cdot|\theta)$ with parameters θ pre-trained from common single-view data. Leveraging $\mathcal{D}$ can enhance its performance, as $\mathcal{D}$ provides additional information from a dual-view setup. Our objective is to adapt this pre-trained estimator, $\mathcal{H}(\cdot|\theta)$, to an arbitrary yet fixed dual-view setting (with unknown camera poses) without needing ground-truths or camera parameters. By inputting $\mathcal{H}(\cdot|\theta)$ and $\mathcal{D}$ into our method, it outputs an adapted estimator $\mathcal{H}(\cdot|\theta')$ with parameters θ' tailored for the dual-view scenario.

Upon adapting the estimator, its inference mechanism is correspondingly tailored for dual-view scenarios (Fig. 1 (b)). During testing, the adapted estimator $\mathcal{H}(\cdot|\theta')$ processes a dual-view input pair $(\mathbf{x}^{v1}, \mathbf{x}^{v2})$ and produces two predictions $(\mathbf{J}^{v1}, \mathbf{J}^{v2})$, where each $\mathbf{J}^v \in \mathbb{R}^{21\times3}$ represents the 21 3D joints of the hand. These predictions denote the 3D hand joints for each view and can be combined together to generate a final output, *e.g.*, through a simple average.

Camera layout for a multi-view headset. Fig. 3 illustrates an example of headset-mounted camera setups for multi-view egocentric data capture, with four cameras at each corner for different views. The top-right of Fig. 3 displays images from these cameras. Six distinct dual-view pairs can be created from these four views. As a supplement, the bottom of Fig. 3 shows the synthetic training data, highlighting variations in style and lighting. Such data helps to explore the performance of our method under cross-dataset or simulate-to-real settings. See Sec. 5.1 for dataset details.

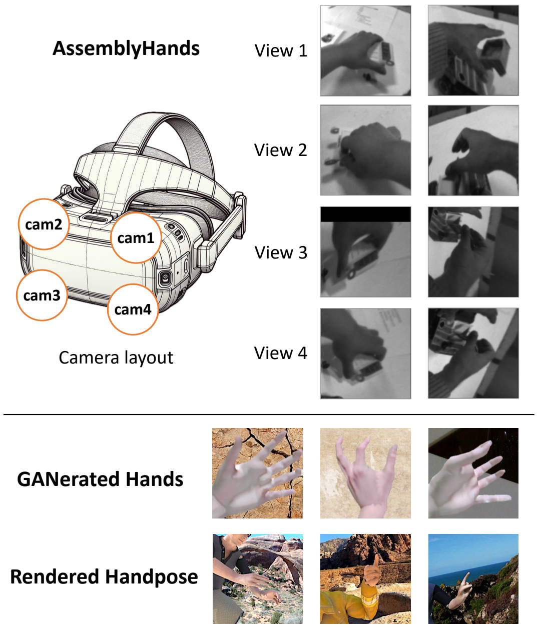

Figure 3. Top: Headset and its camera layout to collect multi-view data, and samples from the four views. Bottom: Samples of synthetic data. Image samples are from AssemblyHands [30], GANerated Hands [28], and Rendered Handpose [47], respectively.

4. Proposed Method

We propose a novel unsupervised single-to-dual-view adaptation framework (S2DHand). Before adaptation, an initialization step is performed to initialize the rotation matrix between both views (Sec. 4.1). The rotation matrix is essential to establish the transformation between two camera coordinate systems. The architecture overview, as illustrated in Fig. 4, comprises two branches, an estimator $\mathcal{H}$ and a its momentum version $\overline{\mathcal{H}}$. The adaptation process is designed from two stereo constraints, pair-wise cross-view consensus and invariant rotation transformation between both camera coordinate systems. This leads to two key pseudo-labeling modules: attention-based merging and rotation-guided refinement (Secs. 4.3 and 4.4). Notably, these two modules function in a complementary manner, depending on the prediction accuracy, ensuring reliable pseudo-labeling.

4.1. Initialization

The initialization step aims to estimate a relatively accurate rotation matrix R, since R is necessary to link the two camera coordinate systems [23, 43]. It should be noted that translation vector between the two cameras is not necessary, as the predicted hand poses are usually aligned by the wrist during testing [30, 46]. Assuming that the initial pre-trained estimator is sufficiently accurate to generate reasonable predictions, we estimate the R using the predictions of unlabeled dual-view data $\mathcal{D} = \{\mathbf{x}_i^{v1}, \mathbf{x}_i^{v2}|_{i=1}^N\}$. Given $\mathcal{D}$, the estimator $\mathcal{H}$ can output N pairs of predictions $\{\mathbf{J}_i^{v1}, \mathbf{J}_i^{v2}|_{i=1}^N\}$,

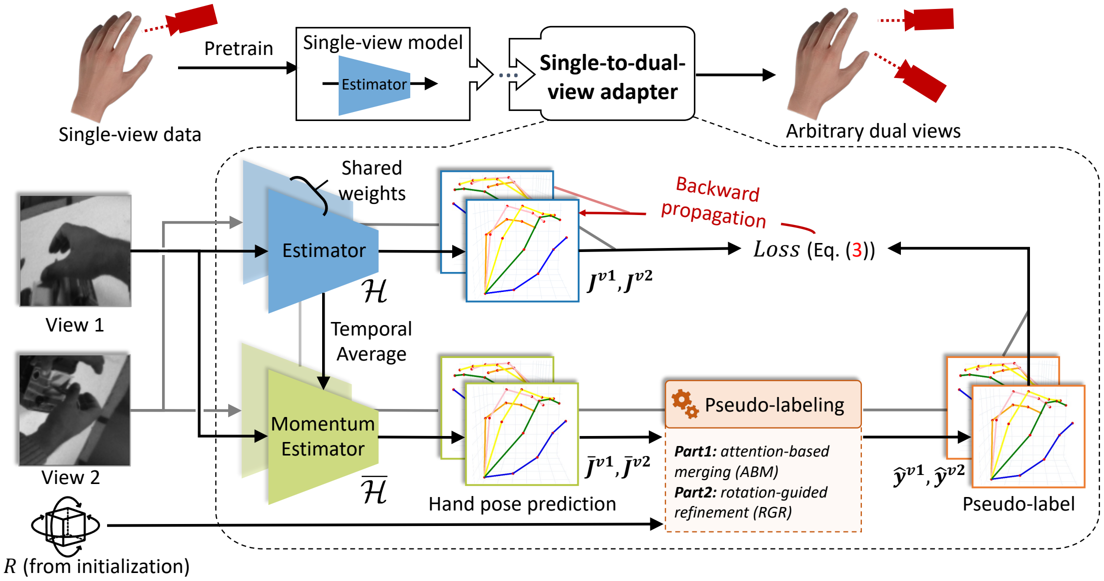

Figure 4. Overview of the proposed S2DHand, image pairs captured from arbitrarily placed dual cameras are input for adaptation. The architecture of S2DHand is illustrated in the dark dashed box, which contains a dynamically updated estimator and a momentum estimator. The momentum estimator's predictions are used to generate pseudo-labels, which are then processed by our pseudo-labeling module (Secs. 4.3 and 4.4). Using the pseudo-labels, a loss function is computed to update the estimator. The rotation matrix R from the initialization step (Sec. 4.1) is required for the pseudo-labeling.

where $\mathbf{J}_i^v \in \mathbb{R}^{21\times 3}$ (21 is the number of 3D joints). Then, the rotation matrix R is estimated by:

$$R^{(0)} = \frac{1}{N}\sum_{i=1}^{N} rot(\mathbf{J}_i^{v1}, \mathbf{J}_i^{v2}), \tag{1}$$

where the superscript of R denotes the iteration number, (0) indicates that it is before the first iteration. The rot function [18] generates a 3×3 rotation matrix from two 21×3 joint predictions. Note that the average in Eq. (1) is not element-wise, but an average of rotation matrices.

4.2. Single-to-dual-view adaptation

With the initialized R, the adaptation process begins. The S2DHand framework comprises two branches, an estimator $\mathcal{H}(\cdot|\theta)$ with dynamically updating parameters θ, and its momentum version $\overline{\mathcal{H}}(\cdot|\overline{\theta})$, which updates its parameters $\overline{\theta}$ using temporal moving average. Temporal moving average has been proved by many works [6, 14, 25, 27] that can help to stabilize the training process. The $\overline{\theta}$ is updated as:

$$\overline{\theta}^{(T)} = \eta_\theta \overline{\theta}^{(T-1)} + (1-\eta_\theta)\theta, \tag{2}$$

where $\overline{\theta}^{(T-1)}$ indicates the temporal averaged parameters in the previous iteration $T-1$, and η_θ represents the ensembling momentum, which is set as 0.99 [25, 27].

As shown in Fig. 4, during the single-to-dual-view adaptation, the role of the momentum model $\overline{\mathcal{H}}$ is to generate pseudo-labels, which are then utilized to supervise the model $\mathcal{H}$. The pseudo-labeling module (Secs. 4.3 and 4.4) outputs pseudo-labels $\hat{\mathbf{y}}^{v1}, \hat{\mathbf{y}}^{v2}$ based on the predictions $\overline{\mathbf{J}}^{v1}, \overline{\mathbf{J}}^{v2}$ from $\overline{\mathcal{H}}$. These pseudo-labels are then used to supervise the predictions $\mathbf{J}^{v1}, \mathbf{J}^{v2}$ of $\mathcal{H}$. The loss function is computed as:

$$\mathcal{L} = \|\mathbf{J}^{v1} - \hat{\mathbf{y}}^{v1}\|_2 + \|\mathbf{J}^{v2} - \hat{\mathbf{y}}^{v2}\|_2. \tag{3}$$

The estimator follows the implementation of DetNet [46], where $\mathcal{H}$ directly outputs heatmaps, and $\mathbf{J}$ is calculated from the heatmaps. Therefore, the loss function is actually computed from corresponding heatmaps, here we write these 3D-joint variables only for better understanding.

4.3. Pseudo-labeling: attention-based merging

The attention-based merging (ABM) module, which constitutes the first part of pseudo-labeling, is derived from cross-view consensus. Cross-view consensus refers to the concept of achieving agreement or consistency between different views of the same data [40]. Theoretically, when transformed into the same coordinate system, the two predictions $\overline{\mathbf{J}}^{v1}$ and $\overline{\mathbf{J}}^{v2}$ from different views should be identical, *i.e.*, $R\overline{\mathbf{J}}^{v1} = \overline{\mathbf{J}}^{v2}$, with $\overline{\mathbf{J}}^{v1}$ $\overline{\mathbf{J}}^{v2}$ being aligned with wrist joint.

This stereo constraint is the foundation for this module to generate accurate pseudo-labels. Prior works utilize a simple average [26, 29] (*e.g.*, $(R\overline{\mathbf{J}}^{v1} + \overline{\mathbf{J}}^{v2})/2$) or sample-wise confidence [2, 29] to improve the quality of pseudo-labels. However, these approaches overlook the varying confidence in joints that is caused by differences in image capture across views. For instance, a joint that is occluded

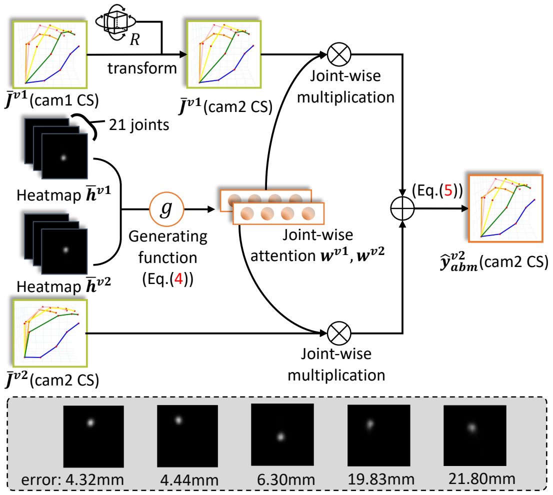

Figure 5. Top: illustration of the first part of pseudo-labeling: attention-based merging module. The generating process of $\hat{y}^{v2}_{abm}$ in view2 is shown as an example, the process of view1 is the same. Bottom: visualizations of heatmaps with different accuracy.

in one view but fully visible in another could lead to reliable predictions being hindered by unreliable ones.

To address this, we propose joint-wise attention $\mathbf{w} \in \mathbb{R}^{21\times 1}$ to represent each joint's prediction confidence. It is derived from the 2D heatmap $\overline{\mathbf{h}} \in \mathbb{R}^{21\times 32\times 32}$ output from $\overline{\mathcal{H}}$. The $\overline{\mathbf{h}}$ indicates the probability of each joint's presence at every pixel in the 2D image space. This approach is from an observation (bottom of Fig. 5): as the error of prediction increases, the intensity of the heatmap's hotspot decreases, *i.e.*, darker indicates low accuracy. Inspired by this, we propose an attention-generating function:

$$\mathbf{w}^v_j = \frac{\beta^{max(\overline{\mathbf{h}}^v_j)}}{\sum_{v\in\{v1,v2\}}\beta^{max(\overline{\mathbf{h}}^v_j)}}, \tag{4}$$

where the subscript j indicate the index of joint, *i.e.*, $j = 1, 2, ..., 21$. We introduce a hyper-parameter β here to adjust softness. Please refer to Sec. 5.8 for parameter choosing.

The workflow of this attention-based merging module is illustrated at the top of Fig. 5. First, we transform both predictions into the same camera coordinate system. Then, a joint-wise multiplication is performed for each of them using the attention $\mathbf{w}^{v1}, \mathbf{w}^{v2}$. Finally, the pseudo-label $\hat{\mathbf{y}}_{abm}$ is calculated through a summation operation. In summary:

$$\begin{aligned}\hat{\mathbf{y}}^{v1}_{abm} &= \mathbf{w}^{v1}\overline{\mathbf{J}}^{v1} + \mathbf{w}^{v2}R^T\overline{\mathbf{J}}^{v2},\\ \hat{\mathbf{y}}^{v2}_{abm} &= \mathbf{w}^{v1}R\overline{\mathbf{J}}^{v1} + \mathbf{w}^{v2}\overline{\mathbf{J}}^{v2}.\end{aligned} \tag{5}$$

4.4. Pseudo-labeling: rotation-guided refinement

The rotation-guided refinement (RGR) module is based on another stereo constraint: invariance of rotation transformation between both views. This implies that the estimated rotation matrix should remain unchanged since the cameras are fixed, *i.e.*, theoretically $rot(\overline{\mathbf{J}}^{v1}, \overline{\mathbf{J}}^{v2}) = C$. In light of this, this module aims to refine the predictions $\overline{\mathbf{J}}^{v1}, \overline{\mathbf{J}}^{v2}$ such that $rot(\overline{\mathbf{J}}^{v1}, \overline{\mathbf{J}}^{v2})$ becomes invariant across all input data. The refinement result becomes the pseudo-label $\hat{\mathbf{y}}_{rgr}$.

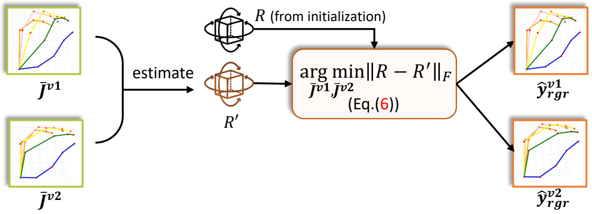

Figure 6. Illustration of the second part of pseudo-labeling: rotation-guided refinement module.

The workflow is shown in Fig. 6. Given a pair of predictions, our method estimates a new rotation matrix $R' = rot(\overline{\mathbf{J}}^{v1}, \overline{\mathbf{J}}^{v2})$. Subsequently, the more accurately estimated R from the initialization step is set as the target for refinement, aiming to minimize $\|R - R'\|_F$. The refinement process can be expressed as:

$$\hat{\mathbf{y}}^{v1}_{rgr}, \hat{\mathbf{y}}^{v2}_{rgr} = \underset{\overline{\mathbf{J}}^{v1}, \overline{\mathbf{J}}^{v2}}{\arg\min} \|R - rot(\overline{\mathbf{J}}^{v1}, \overline{\mathbf{J}}^{v2})\|_F. \tag{6}$$

In detail, we employ BFGS [1] algorithm for minimizing.

Final pseudo-label. Finally, the pseudo-label is calculated from a weighted average of $\hat{\mathbf{y}}_{abm}$ and $\hat{\mathbf{y}}_{rgr}$:

$$\hat{\mathbf{y}} = \alpha\hat{\mathbf{y}}_{abm} + (1-\alpha)\hat{\mathbf{y}}_{rgr}. \tag{7}$$

Here, we introduce another pre-fixed hyper-parameter α, which adjusts the the weight of the two parts of the pseudo-label. Empirically, we set $\alpha = 0.7$ (see Sec. 5.8).

Complement between two pseudo-labels. When the predictions $\overline{\mathbf{J}}^{v1}$ and $\overline{\mathbf{J}}^{v2}$ are accurate, R' closely approximates R, making $\hat{\mathbf{y}}_{rgr}$ redundant. In such cases, $\hat{\mathbf{y}}_{abm}$ is beneficial as it merges the two accurate predictions. Conversely, if $\overline{\mathbf{J}}^{v1}$ and $\overline{\mathbf{J}}^{v2}$ are unreliable, $\hat{\mathbf{y}}abm$ will consequently also lack accuracy. In such instances, $\hat{\mathbf{y}}rgr$ steps in as a complementary tool for refining pseudo-labels. By minimizing $\|R - R'\|_F$, it optimizes the predictions towards an alignment with real-world condition.

Update rotation matrix R. Clearly, the rotation matrix R from the initialization step plays an important role in bridging the two camera coordinate systems. As can be expected, its accuracy significantly affects the final performance. To enhance its accuracy with each iteration, we also employ a temporal moving average for its updates. Given an input batch containing B image pairs, the R is updated as:

$$R^{(T)} = \eta_R R^{(T-1)} + (1-\eta_R)\cdot\frac{1}{B}\sum_{i=1}^{B} rot(\overline{\mathbf{J}}^{v1}_i, \overline{\mathbf{J}}^{v2}_i), \tag{8}$$

where the ensembling momentum η_R is set as 0.999 [14]. This way, the rotation matrix R can be updated slowly and provide a more accurate rotation function with iteration.

Table 2. Adaptation results for all dual-camera pairs. The camera pairs are from $\mathcal{D}_{ah}$ dataset [30], which is divided into two parts according to the collecting headset. In "In-dataset" and "Cross-dataset" settings, the baseline model is pre-trained on $\mathcal{D}_{ah}$ and $\mathcal{D}_{syn}$ [28, 47], respectively. The adaptation yields a uniquely adapted model for each pair, and "Overall" averages the results across all 6 pairs.

Camera pair	Method	In-dataset ($\mathcal{D}_{ah} \rightarrow \mathcal{D}_{ah}$)				Cross-dataset ($\mathcal{D}_{syn} \rightarrow \mathcal{D}_{ah}$)			
		$\mathcal{D}_{ah} - Headset1$		$\mathcal{D}_{ah} - Headset2$		$\mathcal{D}_{ah} - Headset1$		$\mathcal{D}_{ah} - Headset2$	
		Mono-M	Dual-M	Mono-M	Dual-M	Mono-M	Dual-M	Mono-M	Dual-M
cam 1, 2	Baseline	43.00	39.20	54.71	52.38	67.93	60.48	70.32	62.26
	S2DHand	**31.01** ▼ 27.9%	**31.36** ▼ 20.0%	**45.52** ▼ 16.8%	**45.14** ▼ 13.8%	**63.46** ▼ 6.6%	**59.32** ▼ 1.9%	**70.09** ▼ 0.3%	**60.97** ▼ 2.1%
cam 1, 3	Baseline	25.00	23.29	22.59	21.08	57.79	51.42	64.00	60.25
	S2DHand	**19.73** ▼ 21.1%	**19.92** ▼ 14.5%	**17.90** ▼ 20.8%	**17.68** ▼ 16.1%	**50.84** ▼ 12.0%	**47.55** ▼ 7.5%	**61.81** ▼ 3.4%	**58.34** ▼ 3.2%
cam 1, 4	Baseline	24.90	22.70	16.73	14.91	52.71	46.55	54.32	50.57
	S2DHand	**20.88** ▼ 16.1%	**20.87** ▼ 8.1%	**14.64** ▼ 12.5%	**14.29** ▼ 4.2%	**46.05** ▼ 12.6%	**42.50** ▼ 8.7%	**46.59** ▼ 14.2%	**45.66** ▼ 9.7%
cam 2, 3	Baseline	17.96	15.23	17.10	15.08	53.36	48.42	52.84	48.84
	S2DHand	**14.97** ▼ 16.6%	**14.44** ▼ 5.2%	**14.42** ▼ 15.7%	**14.20** ▼ 5.8%	**40.26** ▼ 24.6%	**39.32** ▼ 18.8%	**43.61** ▼ 17.5%	**42.88** ▼ 12.2%
cam 2, 4	Baseline	22.09	19.84	23.24	20.96	59.44	54.32	61.13	57.41
	S2DHand	**17.98** ▼ 18.6%	**17.75** ▼ 10.5%	**18.31** ▼ 21.2%	**18.41** ▼ 12.2%	**50.59** ▼ 14.9%	**49.41** ▼ 9.0%	**52.45** ▼ 14.2%	**51.48** ▼ 10.3%
cam 3, 4	Baseline	16.83	15.77	19.93	18.08	45.82	42.34	49.84	48.99
	S2DHand	**16.36** ▼ 2.8%	**15.55** ▼ 1.4%	**19.25** ▼ 3.4%	**17.80** ▼ 1.5%	**39.46** ▼ 13.9%	**37.43** ▼ 11.6%	**44.04** ▼ 11.6%	**42.88** ▼ 12.5%
Overall	Baseline	24.96	22.67	25.72	23.75	56.18	50.59	58.74	54.72
	S2DHand	**20.16** ▼ 19.2%	**19.98** ▼ 11.9%	**21.67** ▼ 15.7%	**21.25** ▼ 10.5%	**48.44** ▼ 13.8%	**45.92** ▼ 9.2%	**53.11** ▼ 9.6%	**50.37** ▼ 7.9%

5. Experiment

5.1. Dataset

We employ **AssemblyHands** [30] ($\mathcal{D}_{ah}$) as the evaluation set, as it is the newest large-scale benchmark dataset with high-quality multi-view 3D hand pose annotations. As for the training set, we set two adaptation scenarios: 1) in-dataset setting where the training set is drawn from the same dataset $\mathcal{D}_{ah}$ and 2) cross-dataset setting where we use synthetic dataset ($\mathcal{D}_{syn}$) as the training set, consisting of **Rendered Handpose** [47] and **GANerated Hands** [28]. The details of the datasets are as below:

- *AssemblyHands* [30] is a large-scale benchmark dataset featuring accurate 3D hand pose annotations. Collected using two AR headsets, it comprises images captured from four synchronized egocentric cameras. The dataset includes 412K training samples and 62K testing samples.
- *GANerated Hands* [28] includes over 330K color images of hands. The images are synthetically generated and then fed to a GAN [9] to make the features closer to real hands.
- *Rendered Handpose* [47] contains about 44K samples. The images are rendered with freely available characters.

In detail, AssemblyHands is collected by two VR headsets, as shown in Fig. 3, each headset has four egocentric cameras at four corners. Following the collecting devices, we separate $\mathcal{D}_{ah}$ into two parts, each part is collected using one headset, namely $\mathcal{D}_{ah} - Headset1/2$.

5.2. Experimental setup

Evaluation metric. We compare the predictions from our model with the ground-truth labels in root-relative coordinates, and use the common mean per joint position error (MPJPE) in millimeters as the evaluation metric. However, since our focus is on the single-to-dual-view adaptation task, the adapted estimator is expected to perform in dual-view settings. This implies that traditional MPJPE computed from single-view (monocular MPJPE, Mono-M) cannot be sufficient. As a result, we propose a new dual-view MPJPE metric Dual-M in addition to monocular MPJPE. The metrics are defined as follows:

- Mono-M: the traditional monocular MPJPE, which collects all the single-view errors from both views and calculates their average.
- Dual-M: the proposed metric under dual views. To calculate it, first the predictions from both views are averaged using a rotation matrix R. Then, we calculate the MPJPE of the averaged predictions as the Dual-M. Usually, the R is from the initialization step of our method.

Implementation detail. We employ PyTorch for implementation. All experiments run on a single NVIDIA A100 GPU. DetNet from [46] is adopted as the backbone of our hand pose estimator. Adam optimizer is employed with a learning rate of 1×10^{-3} to pre-train the 3D hand pose estimation network. For the single-to-dual-view adaptation, we use the Adam optimizer with a learning rate of 5×10^{-4}.

5.3. Adaptation results for all camera pairs

In this section, we use our method to adapt the same pre-trained single-view hand pose estimator to all dual-view pairs from the evaluation set $\mathcal{D}_{ah}$ independently, yielding one adapted model for each pair. Experiments are conducted under both in-dataset and cross-dataset settings. Under the in-dataset setting (Tab. 2, $\mathcal{D}_{ah} \rightarrow \mathcal{D}_{ah}$), the baseline model is pre-trained on $\mathcal{D}_{ah}$, while under the cross-dataset setting ($\mathcal{D}_{syn} \rightarrow \mathcal{D}_{ah}$), the baseline model is pre-trained on $\mathcal{D}_{syn}$ before being adapted to the camera pairs from $\mathcal{D}_{ah}$.

As shown in Tab. 2, compared with the pre-trained model (Baseline), our S2DHand offers significant accuracy gains under both settings among all camera pairs. This indicates that our method can adapt well to arbitrary dual views regardless of the camera positions or pre-training datasets.

Quantitative results demonstrate that the S2DHand offers substantial improvements. On average, the improvement in both monocular (Mono-M) and dual-view (Dual-M) metrics exceeds 10%, with the maximum improvement exceeding 20%. Interestingly, we can see that the improvement for cam $1, 2$ under the cross-dataset setting is relatively small. This indicates that low initial accuracy limits the performance.

5.4. Comparison under cross-dataset settings

Our method is compared with state-of-the-art adaptation techniques in cross-dataset settings. Considering the prevalence and significance of cross-dataset scenarios in real-world applications, this experiment evaluates the capability of S2DHand in comparison to leading domain adaptation methods. Specifically, adaptation methods included in the comparison are: SFDAHPE [32], RegDA [17], DAGEN [11], and ADDA [36].

For fairness, we do not include existing multi-view methods [4, 12, 19] in this comparison. This is because these methods require 1) multi-view labels and 2) camera parameters, whereas our approach is unsupervised and does not require such parameters. In contrast, all the comparison methods are unsupervised, leading to a fair comparison.

In detail, SFDAHPE [32], RegDA [17] are developed for pose estimation. DAGEN [11] and ADDA [36] are originally proposed for gaze estimation and classification, respectively. We include these two methods here to show potential of these state-of-the-art methods in enhancing the cross-dataset performance of hand pose estimation. To make a fair comparison, their original networks are replaced with the same DetNet [46] as our baseline.

Quantitative results of different methods are shown in Tab. 3. Our method not only significantly outperforms the state-of-the-art methods, but also shows an advantage of being source-free. The superior performance verifies the effectiveness of the proposed S2DHand for single-to-dual-view adaptation under cross-dataset settings. For reference, we also provide the result of fine-tuning as the upper bound of this cross-dataset task.

5.5. Ablation study

We conducted ablation experiments to analyze the contribution of each component in our model. The following experiments are evaluated based on the $\mathcal{D}_{ah} \rightarrow \mathcal{D}_{ah}$ task for clearer observation. The components are shown below:

- ABM: Attention-based merging module, which generates pseudo-labels based on the cross-view consensus.
- RGR: Rotation-guided refinement module, which generates pseudo-labels based on the invariance of rotation transformation between both camera coordinate systems.

Table 3. Comparison with state-of-the-art adaptation methods under cross-dataset settings. "SF" indicates if the method is source-free (requiring no data from source dataset, $\mathcal{D}_{syn}$). * denotes that labels from the target dataset ($\mathcal{D}_{ah}$) are needed.

$\mathcal{D}_{syn} \rightarrow \mathcal{D}_{ah}$		$\mathcal{D}_{ah} - Headset1$		$\mathcal{D}_{ah} - Headset2$	
	SF	Mono-M	Dual-M	Mono-M	Dual-M
Source Only		56.18	50.59	58.74	54.72
Fine-tune*		45.03	38.11	47.75	42.19
ADDA [36]	✗	56.90	48.48	57.87	51.39
DAGEN [11]	✗	55.37	49.72	57.62	53.17
RegDA [17]	✗	51.41	47.85	54.75	51.50
SFDAHPE [32]	✓	54.06	49.11	57.22	53.39
S2DHand (Ours)	✓	**48.44**	**45.92**	**53.11**	**50.37**

Table 4. Ablation study of our method on $\mathcal{D}_{ah} \rightarrow \mathcal{D}_{ah}$ task. ABM and RGR stand for the two pseudo-labeling modules, respectively.

ABM	RGR	$\mathcal{D}_{ah} - Headset1$		$\mathcal{D}_{ah} - Headset2$	
		Mono-M	Dual-M	Mono-M	Dual-M
✗	✗	24.96	22.67	25.72	23.75
✓	✗	20.81	20.54	22.24	21.71
✗	✓	21.89	21.33	23.54	22.75
✓	✓	**20.16**	**19.98**	**21.67**	**21.25**

Tab. 4 shows the hand pose estimation errors under different combinations. We observe that both ABM and RGR can significantly improve the hand pose estimation performance over the pre-trained baseline (first row). Our final version achieves the best results for all metrics, confirming the optimality of our method.

5.6. Number of input image pairs

To find the optimal number of input image pairs (*i.e.*, N) for our method, we evaluate the S2DHand's performance under different numbers of input image pairs. Specifically, the experiments are conducted on the cam $2, 3$ pair in the $\mathcal{D}_{ah} \rightarrow \mathcal{D}_{ah}$ task. The results are illustrated in Fig. 7. It indicates that the performance of S2DHand constantly improves as the number of input image pairs increase. Our method's performance converges when $N \geq 1000$. Consequently, we choose $N = 1000$ for our S2DHand.

5.7. Complement between two pseudo-labels

Fig. 8 demonstrates the complementary nature, as stated in Sec. 4.4, of the pseudo-labeling. Using camera pair $cam1, 2 - Headset1$ under in-dataset setting, we analyze the error of pseudo-labels with and without the refinement term $\hat{\mathbf{y}}_{rgr}$, in relation to the prediction error of $\bar{\mathbf{J}}$. Note that $\bar{\mathbf{J}}$ is the prediction used to compute these pseudo-labels. For better observation, the predictions are first divided into seven equal intervals according to their errors, $[9.4, 34.0), [34.0, 58.6), ..., [157.1, 181.7]$. Then, the average pseudo-label error for each interval is calculated.

In Fig. 8, the bars are placed in the middle of each inter-

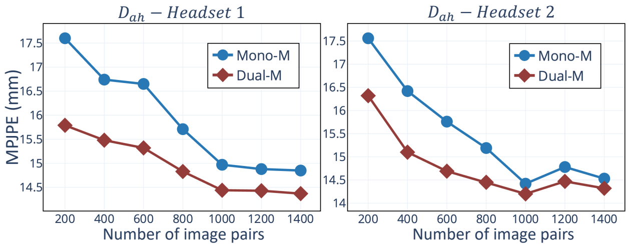

Figure 7. Evaluation of S2DHand's performance with gradually increasing the number of input image pairs.

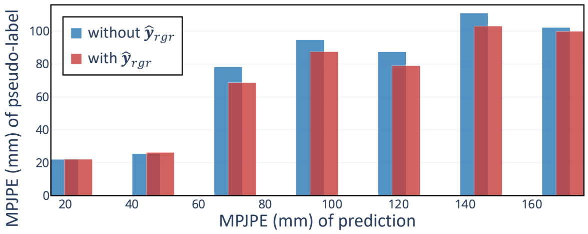

Figure 8. The error of pseudo-labels with and without $\hat{\mathbf{y}}_{rgr}$, in relation to the error of prediction $\overline{\mathbf{J}}$, where $\overline{\mathbf{J}}$ is what we use to compute the pseudo-labels. MPJPE in millimeter is the metric.

Table 5. Performance of our method with different hyper-parameters α (Eq. (7)) and β (Eq. (4)).

$\mathcal{D}_{ah} \to \mathcal{D}_{ah}$	$\mathcal{D}_{ah} - Headset1$		$\mathcal{D}_{ah} - Headset2$	
	Mono-M	Dual-M	Mono-M	Dual-M
$\alpha = 0.3$	20.83	20.54	22.39	21.82
$\alpha = 0.5$	20.34	20.17	21.90	21.49
$\alpha = 0.7$	**20.16**	**19.98**	**21.67**	**21.25**
$\alpha = 0.9$	20.46	20.18	22.17	21.53
$\beta = 1$	21.02	20.94	22.77	22.56
$\beta = e$	20.66	20.47	22.50	22.10
$\beta = \infty$	**20.16**	**19.98**	**21.67**	**21.25**

val, with y-axis representing the pseudo-label errors. The result indicates that the $\hat{\mathbf{y}}_{rgr}$ term is redundant for accurate predictions ($< 60mm$) but significantly reduces pseudo-label errors for larger prediction errors ($\geq 60mm$). This finding supports the statement in Sec. 4.4 about the importance of $\hat{\mathbf{y}}_{rgr}$ for complementing pseudo-labels in cases of inaccurate predictions.

5.8. Hyper-parameters

We evaluate how the S2DHand's performance varies with the change of weight parameter α (Eq. (7)). α controls the weights for averaging the two parts of pseudo-labeling. We test four values of α, 0.3, 0.5, 0.7, and 0.9. The results are shown in the row 1-4 of Tab. 5, where our method achieves the best performance when $\alpha = 0.7$.

We also test the performance with varying β (Eq. (4)). β is the parameter in the attention-generating function, which generates the joint-wise attention in attention-based merging module. In fact, the β acts the role of e in the softmax function. When $\beta = 1, e, \infty$, the merging becomes a simple average (where attention becomes invalidated), softmax function, and maximum function, respectively. We can see that the S2DHand achieves the best performance when $\beta = \infty$. This suggests that selecting the prediction with higher confidence as a pseudo-label in a joint-wise manner is the most effective strategy. Consequently, we set $\alpha = 0.7$ and $\beta = \infty$ for all the experiments.

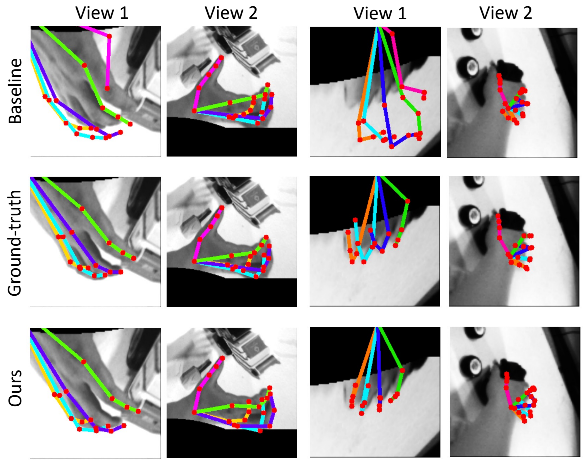

Figure 9. Visual examples of estimated 3D hand poses under both views. The joints are portrayed by projecting the final 3D predictions to the image plane.

5.9. Qualitative result

To understand how our method improves the performance of hand pose estimation under dual-view settings, we visually present typical cases by portraying 3D hand joints onto the input image pairs. In detail, the 3D joints are projected to the image plane, with the visualization depicted in Fig. 9.

Notably, when confronted with extreme view angles (see the left pair), the predictions of baseline model tend to be unreliable. Conversely, our technique gives a prediction much closer to the actual hand shape after adaptation. In the 3rd column, even when the hand is partially out of field-of-view, leading to a truncated hand, our S2DHand continues to deliver trustworthy predictions. These results indicate that S2DHand can utilize additional information from dual views to provide significant improvements even under extreme challenging cases.

6. Conclusion

In this paper, we present a novel single-to-dual-view adaptation framework (S2DHand), designed to adapt a single-view hand pose estimator to dual-view settings. The S2DHand is unsupervised, eliminating the need for multi-view labels. Our method also requires no camera parameters, enabling compatibility with arbitrary dual views. Two stereo constraints are employed as two pseudo-labeling modules in an complementary manner. Our method achieves significant performance gains across all dual-view pairs under both in-dataset and cross-dataset settings.

References

[1] Charles George Broyden. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, 6(1):76–90, 1970. 5

[2] Minjie Cai, Feng Lu, and Yoichi Sato. Generalizing hand segmentation in egocentric videos with uncertainty-guided model adaptation. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 14392–14401, 2020. 4

[3] Ching-Hang Chen, Ambrish Tyagi, Amit Agrawal, Dylan Drover, Rohith Mv, Stefan Stojanov, and James M Rehg. Unsupervised 3d pose estimation with geometric self-supervision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5714–5724, 2019. 2

[4] Liangjian Chen, Shih-Yao Lin, Yusheng Xie, Yen-Yu Lin, and Xiaohui Xie. Mvhm: A large-scale multi-view hand mesh benchmark for accurate 3d hand pose estimation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 836–845, 2021. 1, 2, 7

[5] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European conference on computer vision (ECCV)*, pages 720–736, 2018. 1

[6] Fabian Dubourvieux, Romaric Audigier, Angelique Loesch, Samia Ainouz, and Stephane Canu. Unsupervised domain adaptation for person re-identification through source-guided pseudo-labeling. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4957–4964. IEEE, 2021. 4

[7] Qichen Fu, Xingyu Liu, Ran Xu, Juan Carlos Niebles, and Kris M. Kitani. Deformer: Dynamic fusion transformer for robust hand pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 23600–23611, 2023. 1, 2

[8] Guillermo Garcia-Hernando, Edward Johns, and Tae-Kyun Kim. Physics-based dexterous manipulations with estimated hand poses and residual reinforcement learning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9561–9568. IEEE, 2020. 1

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 6

[10] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022. 1

[11] Zidong Guo, Zejian Yuan, Chong Zhang, Wanchao Chi, Yonggen Ling, and Shenghao Zhang. Domain adaptation gaze estimation by embedding with prediction consistency. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 7

[12] Shangchen Han, Po-chen Wu, Yubo Zhang, Beibei Liu, Linguang Zhang, Zheng Wang, Weiguang Si, Peizhao Zhang, Yujun Cai, Tomas Hodan, et al. Umetrack: Unified multi-view end-to-end hand tracking for vr. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022. 1, 2, 7

[13] Ankur Handa, Karl Van Wyk, Wei Yang, Jacky Liang, Yu-Wei Chao, Qian Wan, Stan Birchfield, Nathan Ratliff, and Dieter Fox. Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9164–9170. IEEE, 2020. 1

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 4, 5

[15] Markus Höll, Markus Oberweger, Clemens Arth, and Vincent Lepetit. Efficient physics-based implementation for realistic hand-object interaction in virtual reality. In *2018 IEEE conference on virtual reality and 3D user interfaces (VR)*, pages 175–182. IEEE, 2018. 1

[16] Linzhi Huang, Yulong Li, Hongbo Tian, Yue Yang, Xiangang Li, Weihong Deng, and Jieping Ye. Semi-supervised 2d human pose estimation driven by position inconsistency pseudo label correction module. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 693–703, 2023. 2

[17] Junguang Jiang, Yifei Ji, Ximei Wang, Yufeng Liu, Jianmin Wang, and Mingsheng Long. Regressive domain adaptation for unsupervised keypoint detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6780–6789, 2021. 2, 7

[18] Wolfgang Kabsch. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 34(5):827–828, 1978. 4

[19] Leyla Khaleghi, Alireza Sepas-Moghaddam, Joshua Marshall, and Ali Etemad. Multi-view video-based 3d hand pose estimation. *IEEE Transactions on Artificial Intelligence*, 2022. 1, 2, 7

[20] Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8602–8617, 2021. 2

[21] Qiuxia Lin, Linlin Yang, and Angela Yao. Cross-domain 3d hand pose estimation with dual modalities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17184–17193, 2023. 2

[22] Qunming Liu, Xiaodong Li, Xiaofei Zhang, Xiaojun Tan, and Bodong Shi. Multi-view joint learning and bev feature-fusion network for 3d object detection. *Applied Sciences*, 13 (9):5274, 2023. 2

[23] Ruicong Liu and Feng Lu. Uvagaze: Unsupervised 1-to-2 views adaptation for gaze estimation. *arXiv preprint arXiv:2312.15644*, 2023. 3

[24] Ruicong Liu, Yiwei Bao, Mingjie Xu, Haofei Wang, Yunfei Liu, and Feng Lu. Jitter does matter: Adapting gaze estimation to new domains. *arXiv preprint arXiv:2210.02082*, 2022. 2

[25] Ruicong Liu, Yunfei Liu, Haofei Wang, and Feng Lu. Pnp-ga+: Plug-and-play domain adaptation for gaze estimation using model variants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 4

[26] Shaowei Liu, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang. Semi-supervised 3d hand-object poses estimation with interactions in time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14687–14697, 2021. 4

[27] Yunfei Liu, Ruicong Liu, Haofei Wang, and Feng Lu. Generalizing gaze estimation with outlier-guided collaborative adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3835–3844, 2021. 4

[28] 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 Computer Vision and Pattern Recognition (CVPR)*, 2018. 3, 6

[29] Takehiko Ohkawa, Yu-Jhe Li, Qichen Fu, Ryosuke Furuta, Kris M Kitani, and Yoichi Sato. Domain adaptive hand keypoint and pixel localization in the wild. In *European Conference on Computer Vision*, pages 68–87. Springer, 2022. 2, 4

[30] Takehiko Ohkawa, Kun He, Fadime Sener, Tomas Hodan, Luan Tran, and Cem Keskin. Assemblyhands: Towards egocentric activity understanding via 3d hand pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12999–13008, 2023. 2, 3, 6

[31] Vladimir I Pavlovic, Rajeev Sharma, and Thomas S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):677–695, 1997. 1

[32] Qucheng Peng, Ce Zheng, and Chen Chen. Source-free domain adaptive human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4826–4836, 2023. 2, 7

[33] Thammathip Piumsomboon, Adrian Clark, Mark Billinghurst, and Andy Cockburn. User-defined gestures for augmented reality. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pages 955–960. 2013. 1

[34] Siddharth S Rautaray and Anupam Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial intelligence review*, 43:1–54, 2015. 1

[35] Dripta S Raychaudhuri, Calvin-Khang Ta, Arindam Dutta, Rohit Lal, and Amit K Roy-Chowdhury. Prior-guided source-free domain adaptation for human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14996–15006, 2023. 2

[36] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017. 7

[37] Stefan Vogt, Giovanni Buccino, Afra M Wohlschläger, Nicola Canessa, N Jon Shah, Karl Zilles, Simon B Eickhoff, Hans-Joachim Freund, Giacomo Rizzolatti, and Gereon R Fink. Prefrontal involvement in imitation learning of hand actions: effects of practice and expertise. *Neuroimage*, 37 (4):1371–1383, 2007. 1

[38] Christian Von Hardenberg and François Bérard. Bare-hand human-computer interaction. In *Proceedings of the 2001 workshop on Perceptive user interfaces*, pages 1–8, 2001. 1

[39] Yilin Wen, Hao Pan, Lei Yang, Jia Pan, Taku Komura, and Wenping Wang. Hierarchical temporal transformer for 3d hand pose estimation and action recognition from egocentric rgb videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21243–21253, 2023. 1, 2

[40] Wenhao Wu, Hau San Wong, and Si Wu. Semi-supervised stereo-based 3d object detection via cross-view consensus. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17471–17481, 2023. 4

[41] Yinsong Xu, Zhuqing Jiang, Aidong Men, Haiying Wang, and Haiyong Luo. Multi-view feature fusion for person re-identification. *Knowledge-Based Systems*, 229:107344, 2021. 2

[42] 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

[43] Mingfang Zhang, Jinglu Wang, Xiao Li, Yifei Huang, Yoichi Sato, and Yan Lu. Structural multiplane image: Bridging neural view synthesis and 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16707–16716, 2023. 3

[44] Yuxiang Zhang, Zhe Li, Liang An, Mengcheng Li, Tao Yu, and Yebin Liu. Lightweight multi-person total motion capture using sparse multi-view cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5560–5569, 2021. 2

[45] Yang Zheng, Ruizhi Shao, Yuxiang Zhang, Tao Yu, Zerong Zheng, Qionghai Dai, and Yebin Liu. Deepmulticap: Performance capture of multiple characters using sparse multiview cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6239–6249, 2021. 2

[46] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5346–5355, 2020. 1, 2, 3, 4, 6, 7

[47] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. Technical report, arXiv:1705.01389, 2017. https://arxiv.org/abs/1705.01389. 3, 6