Image-free Domain Generalization via CLIP for 3D Hand Pose Estimation

Seongyeong Lee[1,2†] Hansoo Park[1] Dong Uk Kim[1] Jihyeon Kim[1]
Muhammadjon Boboev[1] Seungryul Baek[1]

[1]UNIST, South Korea [2]NC Soft, South Korea

Abstract

RGB-based 3D hand pose estimation has been successful for decades thanks to large-scale databases and deep learning. However, the hand pose estimation network does not operate well for hand pose images whose characteristics are far different from the training data. This is caused by various factors such as illuminations, camera angles, diverse backgrounds in the input images, etc. Many existing methods tried to solve it by supplying additional large-scale unconstrained/target domain images to augment data space; however collecting such large-scale images takes a lot of labors. In this paper, we present a simple image-free domain generalization approach for the hand pose estimation framework that uses only source domain data. We try to manipulate the image features of the hand pose estimation network by adding the features from text descriptions using the CLIP (Contrastive Language-Image Pre-training) model. The manipulated image features are then exploited to train the hand pose estimation network via the contrastive learning framework. In experiments with STB and RHD datasets, our algorithm shows improved performance over the state-of-the-art domain generalization approaches.

1. Introduction

3D hand pose estimation has been essential for human-machine interaction applications such as augmented and virtual reality and robotics. Depth-based 3D hand pose estimation [2, 44, 14, 23, 30, 39, 53, 55, 15, 24, 45, 16, 17, 52, 20] has been popular using the Kinect sensor. Recently, RGB-based 3D hand pose estimation has been widely used and developed thanks to its simple and practical setup compared to depth-based approaches. Many researchers have tried to improve the performance of RGB-based hand pose estimation either by developing the novel deep learning architectures [65, 63, 25] or by increasing data via additional data collection or real and synthetic data generation. For example, Lin et al. [36] recently proposed a Transformer-based model and showed improvement in hand pose estimation performance. Although the model shows the excellent performance, it is still challenging to secure the generalization ability if there is a huge gap in perspective or severe occlusions [1]. Using additional data apparently helps improve the generalization ability [11, 7]. However, it is challenging to collect RGB pose data in diverse environments with varied poses securing the accurate annotations, and thus the performance is limited for in-the-wild environments.

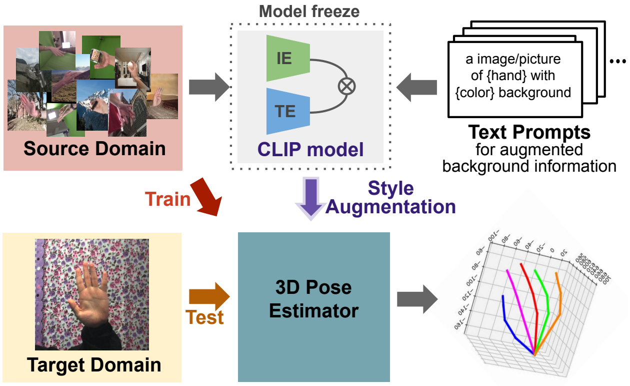

Figure 1. Overview of our domain generalization method in the hand pose estimation task exploiting the CLIP model. Given a source domain image, we train the 3D hand pose estimator by exploiting the CLIP text manipulated features as the new features to the pose estimation network.

Due to the challenges in scaling up the dataset size, there have appeared research trends that propose the algorithm for efficient data usage. As an example, various domain adaptation and generalization techniques [21, 29, 37, 47, 46, 31, 33, 57, 56, 59, 66, 19, 49, 58] have been proposed. Also,

This research was conducted when Seongyeong Lee was a graduate student (Master candidate) at UNIST†.

studies that implements the self-supervision via the contrastive learning [12, 50] is also shown in the literature. For generalization beyond the training domain, Zhang et al. [60] proposed a method for causal representation learning which explicitly exploits the causal structure of the task. By applying domain adaptation and generalization techniques, they achieved better generalization ability on hand pose estimation domain. In addition, Spurr et al. [50] presented a methodology that enables self-supervised contrastive learning for hand pose estimation with many unlabeled data. They encouraged equal variances for geometric transformations during representation learning. As a result, they improved generalization between datasets using the above methodology.

In this paper, we aim to improve domain generalization performance by efficiently using the features of a given data rather than using additional datasets. Especially, we proposed to use the CLIP [43] model to extract generalized feature representation from the source domain images. The CLIP model used a large number of image-text pairs and contrastive learning for training. So, it can extract a more generalized feature representation than models using only images. Using this advantage, we improve the generalization ability across unseen domains by providing the text encoder of CLIP with various texts for augmentation. To the best of our knowledge, this is the first work to adapt domain generalization in hand pose estimation domain using only the source dataset. An overview of our pose estimator with CLIP is shown in Figure 1.

2. Related work

In this section, we will review recent literature on hand pose estimation, contrastive learning and domain adaptation/generalization.

2.1. 3D hand pose estimation

Hand pose estimation is the task of estimating the x, y and z coordinates of 21 hand joints from either depth maps [22, 38] or RGB images [4, 26]. The depth map is invariant to the lighting conditions and shadows. It also has the advantage of being strong against clutter [2, 38]; while it has the disadvantage of not being able to capture various features such as textures and colors of the scene. The RGB image has the advantage of being able to capture detailed hand attributes such as unique colors, textures, and outlines. Besides, RGB cameras are more ubiquitous than depth sensors in our daily life, as we are holding RGB cameras in our smart phones; while we are not holding depth sensors in our pocket. However, unlike the depth map, the RGB image completely loses the 3D depth information. Therefore, compared to the depth map, there is much difficulty in achieving 3D hand pose estimation from RGB images [5, 26]. The useful yet challenging setting of RGB-based 3D hand pose estimation recently accelerates the development of many algorithms [40, 35, 64, 32]. More recently, RGB-based 3D hand mesh estimation has been established as well [26, 4]. The 3D hand model (ie. MANO) [4, 5, 34] has been exploited when constructing these pipelines. The graph convolutional network (GCN) is further exploited to better capture the relationship between vertices in the mesh topology [61, 18].

In the aspect of data, hands require data collection with much diverse camera perspectives, poses and shapes compared to other domains such as body pose estimation [9, 10, 6, 3, 48]. This is due to the fact that hands exhibit severe self-occlusions and diverse camera perspectives.

2.2. Contrastive learning

Contrastive learning is a way to achieving the self-supervised learning. It has been utilized in various works [28, 42, 12, 27, 51] for extracting the view-invariant representation from multiple views by exploiting the different views of the same content as positive samples and other content as negative samples. Chen et al. [12] proposed the contrastive learning for the data augmentation to learn better representation. They used data-augmented samples as positive; while other data samples as negative. The performance of these approaches could be improved by increasing the number of negative samples. However, the number of negative samples is limited by the GPU memory size. So, [13, 27] proposed a method to increase the number of negative samples with the momentum encoder. Zhu et al. [62] proposed a method for controlling the margin term according to the number of negative samples. Caron et al. [8] addressed the limitation by clustering the augmented data instead of comparing features. Contrastive learning has been also used in various tasks such as image and video classification [12, 27, 51] and object detection [28].

Recently, Spurr et al. [50] proposed the contrastive learning framework for the 3D hand pose estimation task. They extended the SimCLR [12] framework to be applicable to structured regression tasks such as the hand pose estimation. They used the geometric and appearance transformed hand images as positive and other images as negative samples for achieving the contrastive learning.

Most contrastive learning methods have been dependent on the data augmentation techniques such as scale manipulation, cut out, noise and rotation transformation. In our paper, we use the CLIP model, which is robust to the zero-shot learning and often used for the domain generalization tasks. We augment the data space in terms of style by manipulating the text prompts in CLIP and perform the contrastive learning between features of original and augmented data to make the feature of original data robust to domain generalization.

2.3. Domain adaptation/generalization

One of the important problems in computer vision is the dataset bias and domain shift between different datasets. To resolve this problem, domain adaptation (DA) /domain generalization (DG) approaches have recently attracted a lot of attention.

Domain adaptation (DA) is a methodology used in situations where the label of the target domain does not exist or is insufficient. DA utilizes source domain data, sparsely labeled, unlabeled target domain data as the training dataset to effectively learn the target domain distribution. DA is largely divided into three types: supervised domain adaptation (SDA) [21, 29, 37, 47], semi-supervised domain adaptation (SSDA) [46, 31, 33] and unsupervised domain adaptation (UDA) [57, 56, 59, 66]. SDA is a method that uses both labeled source and target domain data, SSDA is a method that exploits source domain data whose label is completely annotated and target domain data whose label is partially annotated. Finally, UDA is a method that uses source domain and target domain data without any labels.

Above mentioned methods (ie. SDA, SSDA and UDA) all require the target domain data to train the model. However, it is non-trivial to obtain the target data and also collecting the target data with labels is even harder. Domain generalization (DG) [19, 49, 58, 60] has emerged to relieve the challenge from collecting target data. DG differs from DA in that it makes features generalized in the target domain by using the source and additional unseen domain data which might be similar to the target domain data. Recently, Zhang et al. [60] proposed the DG methodology of the pose estimation using unconstrained dataset and could improve the generalization ability. However, it is limited by the fact that it requires to collect additional unseen domain data which might be similar to the target domain data. In our paper, we propose the domain generalization method which does not require additional dataset collection.

3. Our hand domain generalization framework

In this section, we introduce our hand pose estimation method utilizing CLIP model and contrastive learning mechanism for domain generalization. Overall, Our domain generalization framework receives an $256 \times 256 \times 3$-sized single RGB image $\mathbf{x} \in X$ as input and outputs 21×3-dimensional 3D joint coordinates $\mathbf{c} \in C$. In the remainder of this section, we will first explain our baseline hand pose estimation network f^{HPE}, then explain about the CLIP network f^{CLIP} and finally describe how we combine CLIP network f^{CLIP} into our hand pose estimation network f^{HPE} to construct the overall domain generalization framework. The overall schematic diagrams of our framework is shown in the Figure 2 and a list of used notations is provided in Table 1.

Table 1. Summary of notations.

$X \subset \mathbb{R}^{256\times256\times3}$	RGB image space.
$T \subset \mathbb{R}^{3,920\times1}$	Text prompt space.
$E \subset \mathbb{R}^{128\times1}$	Encoding vector space.
$H \subset \mathbb{R}^{21\times32\times32}$	2D heatmap space.
$C \subset \mathbb{R}^{21\times3}$	3D joint coordinate space.
$f^{\text{CLIP}} : [X,T] \to \mathbb{R}^{512\times1}$	CLIP model.
$f^{\text{CI}} : X \to \mathbb{R}^{512\times1}$	CLIP image encoder.
$f^{\text{CT}} : T \to \mathbb{R}^{512\times1}$	CLIP text encoder.
$f^{\text{HPE}} : X \to C$	Baseline 3D hand pose estimator.
$f^{\text{H2D}} : X \to [H,E]$	2D heatmap net.
$f^{\text{PP}} : H \to C$	Poseprior net.

3.1. Baseline 3D hand pose estimator f^{HPE}

We constitute our baseline hand pose estimator f^{HPE} samely as that of [63] which consists of 1) a 2D heatmap net f^{H2D} that estimates the 2D heatmap $\mathbf{h} \in H$ from input RGB image $\mathbf{x} \in X$, and (2) a Poseprior net f^{PP} that uses the estimated heatmap $\mathbf{h} \in H$ to predict 3D joint coordinates $\mathbf{c} \in C$. In the remainder of this subsection, we will explain details for two sub-networks (ie. f^{H2D}, f^{PP}).

2D heatmap net f^{H2D}. The 2D heatmap network f^{H2D} receives a $256 \times 256 \times 3$-sized single RGB image $\mathbf{x}$. Then, it generates 21 32×32-dimensional 2D heatmap $\mathbf{h}$. The input image $\mathbf{x}$ is sequentially applied to multiple ConvBlocks through the 'Branch1' described in the Fig. 2 and it is mapped to the 2D heatmap $\mathbf{h} \in H$. The 'Branch2' operations in Fig. 2 is related to involving CLIP features and this will be described in Sec. 3.3. We use the convolutional pose machine (CPM) architecture [54] for constructing our 2D heatmap network f^{H2D} following [63] and the detailed operations composing the architecture of the 2D heatmap network f^{H2D} is described in the supplemental.

Poseprior net f^{PP}. After estimating the heatmap $\mathbf{h} \in H$ from the 2D heatmap net f^{H2D}, the Poseprior network f^{PP} receives the estimated heatmap $\mathbf{h}$ and estimates corresponding 3D joint coordinates $\mathbf{c} \in C$. We train our model to predict the relative 3D coordinates within a given image frame and then converts them to absolute 3D coordinates as in [63]. In details, the network consists of two parallel processing streams. One stream predicts 3D hand pose in the canonical space. Another stream predicts a rotation matrix so that the 3D hand pose can be aligned in the camera space.

3.2. CLIP (Contrastive Language-Image Pre-Training) network f^{CLIP}

The CLIP network f^{CLIP} receives a $256\times256\times3$-sized single RGB image $\mathbf{x} \in X$ and the text prompt $\mathbf{t} \in T$. The text encoder f^{CT} and image encoder f^{CI} of the CLIP model f^{CLIP} generates a 512-dimensional encoder features $f^{\text{CI}}(\mathbf{x}) \subset \mathbb{R}^{512\times1}$ and $f^{\text{CT}}(\mathbf{t}) \subset \mathbb{R}^{512\times1}$, respectively. In the original CLIP, ResNet-50 or ViT (VisionTransformer)

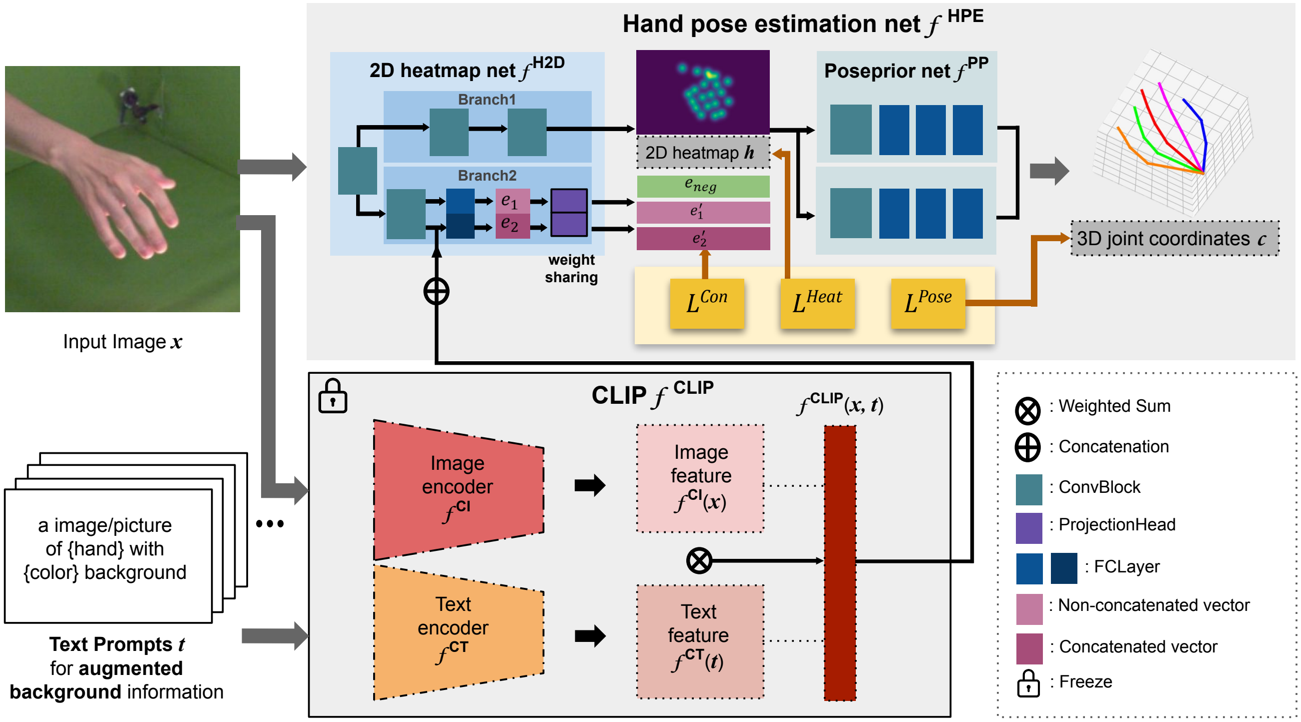

Figure 2. A schematic diagram of of our domain generalization framework for 3D hand pose estimation. 1) The **CLIP network** f^{CLIP} receives the RGB image $\mathbf{x}$ and the text prompt $\mathbf{t}$ as inputs and generates a weighted sum of features $f^{\text{CLIP}}(\mathbf{x},\mathbf{t})$ from the CLIP image/text encoders, 2) **2D heatmap net** f^{H2D} receives the same RGB image $\mathbf{x}$ as input and generates the 2D heatmap $\mathbf{h}$ and encoding vectors $\mathbf{e}'_1$ and $\mathbf{e}'_2$. 3) **Poseprior net** f^{PP} receives the 2D heatmap $\mathbf{h}$ and generates 3D joint coordinates $\mathbf{c}$. In the 2D heatmap net f^{H2D}, the 'Branch2' is taken to generate the encoding vector $\mathbf{e}'_1$ from the original sample and encoding vector $\mathbf{e}'_2$ combining the CLIP-augmented feature. Then, we applied the contrastive learning by regarding $\mathbf{e}'_1$ as the anchor, $\mathbf{e}'_2$ as the positive samples, and picking the negative sample $\mathbf{e}_{\text{neg}}$ based on the heatmap distance among samples in the same batch (See texts in Sec. 3.4 for details). Via this, we are able to make our feature space of the 2D heatmap net f^{H2D} robust to various domains. Three losses (ie. L^{Con}, L^{Heat} and L^{Pose}) are used to train our hand pose estimation net f^{HPE}.

is used as the baseline structure for the image encoder; also Transformer based on the byte pair encoding is used for the text encoder. In this paper, we used the ViT-B/32 as the image encoder and the Transformer network as the text encoder, which are the pre-trained models provided by OpenAI [1] (We freeze the weights of CLIP model during the training).

3.3. CLIP-augmented feature encoding

Our CLIP-augmentation method is described in this section. We actually have two stages for this: 1) CLIP feature generation and 2) encoding vector generation stages.

CLIP feature generation. The image $\mathbf{x}$ is input to the CLIP image encoder f^{CI} to extract the feature vector $f^{\text{CI}}(\mathbf{x})$. This vector contains rich information from the task-agnostic CLIP. However, it does not have information about images in other domains; while only having the information about images $\mathbf{x}$ that come from the source domain. To augment extra information from style and context, we proposed Table 2 to generate text prompt $\mathbf{t}$ reflecting diverse aspects of hand pose images. We can create text prompts by simply combining words in Table 2. This method can generate $3,920$ text prompt configurations and some text prompts automatically generated are exampled in the supplemental. While more sophisticated text prompts could be made for each image and it would further improve the performance; we demonstrated that this simple method works well for our pipeline.

The generated text prompt $\mathbf{t}$ is used as the input to the CLIP text encoder f^{CT} to extract the feature vector $f^{\text{CT}}(\mathbf{t})$. To mix up the information, we defined $f^{\text{CLIP}}(\mathbf{x},\mathbf{t})$ as the weight-summed vector of $f^{\text{CI}}(\mathbf{x})$ and $f^{\text{CT}}(\mathbf{t})$. We chose the ratio between image encoder and text encoder as 6:4 and 9:1 for STB and RHD, respectively via the 10-fold cross-validation. Ablation experiments on the ratio for weight-summation are shown in Table 4(a). Via this process, we enforce the CLIP feature $f^{\text{CLIP}}(\mathbf{x},\mathbf{t})$ to contain the extra information in addition to the source domain information.

Encoding vector generation. Given the CLIP feature

[1] https://openai.com/

Table 2. Compsition of text prompts

head	hand color	hand	color		background
a cropped image of	white	hand with	mountain	lake	room
a image of	dark brown	right hand with	bright	dark	background
a cropped photo of	peach		green	purple	
a picture of	brown		white	yellow	
one	pale yellow		sky blue	black	
a photo of	light beige		orange	red	
a photo of right	black		blue	yellow	
			gray	beige	
			pink	brown	
			dotted	flower	

$f^{\text{CLIP}}(\mathbf{x}, \mathbf{t})$ extracted from the source image $\mathbf{x}$ and the text prompt $\mathbf{t}$, the 2D heatmap network f^{H2D} receives a $256 \times 256 \times 3$-sized single RGB image $\mathbf{x}$ and generates 128-dimensional encoding vectors $\mathbf{e}'_1 \in E$ and $\mathbf{e}'_2 \in E$ via the 'Branch2' operations of 2D heatmap net f^{H2D} (see Fig 2). In this branch, the feature vector of the 2D heatmap network f^{H2D} is sequentially applied to ConvBlock and FC layer to generate the 512-dimensional intermediate feature $\mathbf{e}_1$. Also, the feature vector of the 2D heatmap network f^{H2D} is applied to ConvBlock and is concatenated with the CLIP feature $f^{\text{CLIP}}(\mathbf{x}, \mathbf{t})$ and applied again to the FC layers to generate the 512-dimensional intermediate feature $\mathbf{e}_2$. Intermediate features $\mathbf{e}_1$ and $\mathbf{e}_2$ are applied to the same (weight-shared) ProjectionHead layer to generate the 128-dimensional encoding vectors $\mathbf{e}'_1$ and $\mathbf{e}'_2$, respectively. Here, the intermediate feature $\mathbf{e}_2$ is the enriched version of the intermediate feature $\mathbf{e}_1$ in the aspect of context and backgrounds by concatenating source domain information with the CLIP features $f^{\text{CLIP}}(\mathbf{x}, \mathbf{t})$. The ProjectionHead is composed of two MLP layers that first projects 512-dimensional vectors into 512-dimensional vectors and then projects them again into the 128-dimensional vectors.

Afterwards, the encoding vectors $\mathbf{e}'_1$ and $\mathbf{e}'_2$ are further exploited for the contrastive learning mechanism using the Eq. 4. The entire pipeline is trained in the end-to-end manner through 2D heatmap loss, 3D pose loss and contrastive loss. Via the contrastive loss, our pipeline becomes robust to images from various domains.

3.4. Training

Our domain generalization framework is composed of end-to-end trainable networks based on 1) input image $\mathbf{x}$ from the source domain and 2) custom-created text prompt $\mathbf{t}$. We used only the source dataset (ie. FreiHAND [65]) for training and did not involve additional images or labels from the target domain or other unconstrained datasets. We trained the overall framework using three losses (ie. L^{Heat}, L^{Pose} and L^{Con}) as follows:

$$L(f^{\text{HPE}}) \quad = \quad \lambda_1 L^{\text{Heat}} + \lambda_2 L^{\text{Pose}} + \lambda_3 L^{\text{Con}} \tag{1}$$

where λ_1, λ_2 and λ_3 are balance parameter controlling the weight of each loss function. From the 10-random fold cross validation, we set λ_1 and λ_2 as 1 and set λ_3 as 0.1. Also, we used the Adam optimizer with $\beta = (0.9, 0.999)$ and a learning rate of 10^{-4}. In the remainder of this section, we will explain about three losses.

2D heatmap loss L^{Heat}. The 2D heatmap loss L^{Heat} is defined as the standard mean square error (MSE) loss to close the predicted heatmaps $f^{\text{H2D}}(\mathbf{x}) = \mathbf{h}$ to their corresponding ground-truth 3D joint coordinates $\mathbf{h}^{\text{GT}}$ as follows:

$$L^{\text{Heat}}(f^{\text{H2D}}) = \|f^{\text{H2D}}(\mathbf{x}) - \mathbf{h}^{\text{GT}}\|_2^2. \tag{2}$$

3D pose loss L^{Pose}. The 3D pose loss L^{Pose} is also defined as the standard mean square error (MSE) loss as follows:

$$L^{\text{Pose}}(f^{\text{H2D}}, f^{\text{PP}}) = \|f^{\text{PP}}(f^{\text{H2D}}(\mathbf{x})) - \mathbf{c}^{\text{GT}}\|_2^2 \tag{3}$$

where $\mathbf{c}^{\text{GT}}$ denotes the ground-truth 3D joint coordinates.

Contrastive loss L^{Con}. The contrastive loss is employed to maximize the latent space agreement with the positive samples while minimizing the agreement with negative samples. The encoding vector $\mathbf{e}'_1$ becomes the anchor while the encoding vector $\mathbf{e}'_2$ becomes the positive sample. After this, among the samples in the same batch, the negative sample $\mathbf{e}_{\text{neg}}$ is selected as the sample whose ground-truth heatmap is farthest from the predicted heatmap of the anchor sample. The contrastive loss L^{Con} is then defined as follows:

$$L^{\text{Con}}(f^{\text{H2D}}) = \|\mathbf{e}'_1 - \mathbf{e}'_2\|_1 - \max(0.5, \|\mathbf{e}'_1 - \mathbf{e}_{\text{neg}}\|_1). \tag{4}$$

3.5. Testing

Only the hand pose estimation net f^{HPE} disregarding the 'Branch2' of 2D heatmap net f^{H2D} is exploited during the testing stage. The test RGB image $\mathbf{x}$ is input to the 2D heatmap net f^{H2D} and it takes the 'Branch1' route (see Fig. 2) to generate 21 32×32-dimensional heatmap $\mathbf{h}$. Then the poseprior net f^{PP} is further applied to map it towards the 21×3-dimensional 3D joint coordinates $\mathbf{c}$.

4. Experiment

In this section, we elaborate our experimental settings and analyze the results qualitatively and quantitatively. We demonstrate that the domain generalization method utilizing CLIP has better generalization capability than the previous state-of-the-art methods.

4.1. Dataset

We conduct our experiments using three types of RGB-based 3D hand pose benchmark datasets which have RGB images and corresponding ground-truth 3D pose annotations.

FreiHAND dataset. FreiHAND [65] is a large 3D hand dataset consisting of 130,240 training images and 3,960 testing images. This dataset also includes the mesh annotation as well as the hand pose annotation. Testing data are created both having indoor and outdoor environments;

while training data consist of data taken in an indoor environment with a green background. Afterwards, it provides training data which are artificially synthesized with the background. We use this dataset as the source domain data.

STB dataset. Stereo Hand Pose Tracking Benchmark(STB) [41] provides 2D and 3D annotations of 21 keypoints with a resolution of 640×480. It is a real dataset, composed of STB-BB, STB-SK subsets of images. Two different subsets are captured by the Point Grey Bumblebee2 stereo camera and the Intel F200 depth camera, respectively. We follow the training and testing splits of Zimmerman et al. [63] ($15,000$ images for training and $3,000$ images for testing) and use only the test splits as the target domain dataset in our experiments.

RHD dataset. Rendered hand pose dataset(RHD) [63] is a synthetic dataset captured by *Blender* software using 20 different characters from Mixamo performing 39 actions. It has overall $43,986$ images with a resolution of 320×320 pixels and accurate 21 keypoints annotations and segmentation masks. The background images having cities and landscapes are randomly sampled from *Flickr*. We follow the training and testing splits of Zimmerman et al. [63] ($41,258$ images for training, $2,728$ images for testing) and use only the test splits as target domain dataset in our experiments.

Evaluation method. We evaluated the proposed algorithm on two hand pose estimation datasets (ie. STB and RHD) based on the end-point-error (EPE) measure that calculates the distance between estimated 3D joint coordinates $\mathbf{c}$ and ground-truth 3D joint coordinates $\mathbf{c}^{\text{GT}}$ in the mm unit.

4.2. Results.

This paper deals with the domain generalization (DG) problem in the 3D hand pose estimation task. We mainly compared our method with the existing DG method: Zhang et al. [60] that solves the same problem with ours and achieves the best performance before us. We involved their results [60] for involving only 'source domain images'. In Table 3, we compared ours with existing methods and confirm that our method showed the superior performance based only on the source images, without involving target images (ie. STB, RHD datasets) or additional unconstrained images. Compared to Zhang et al. [60], via our method, error rates are decreased by 3.33% and 3.11% in STB and RHD datasets, repectively.

Figure 3 shows the visualization of T-SNE distribution for the CLIP image and text encoder features (ie. $0.5 \times (f^{\text{CI}}(\mathbf{x}) + f^{\text{CT}}(\mathbf{t}))$) obtained from samples in the source dataset (FreiHAND) and the CLIP image encoder feature (ie. $f^{\text{CI}}(\mathbf{x})$) obtained from samples in the existing datasets (ie. STB, RHD and FreiHAND). The 'aug' denotes the samples obtained from the CLIP image and text encoder features (ie. $0.5 \times (f^{\text{CI}}(\mathbf{x}) + f^{\text{CT}}(\mathbf{t}))$) for the source dataset (FreiHAND); while 'stb', 'rhd' and 'frei' denote samples obtained from the CLIP image encoder feature (ie. $f^{\text{CI}}(\mathbf{x})$) for STB, RHD and FreiHAND datasets, respectively. Figure 3 shows that the distribution of augmented features (ie. 'aug' samples) cover the distribution of STB and RHD datasets, which are target domain datasets. Through this experiment, we qualitatively visualize the effect of our proposed method.

The validity of the augmented distribution is proven in Figure 4 which shows that our method maintains the context information of source domain and style of text prompt. In Figure 4, the image of the source domain dataset (FreiHAND) and text associated with the image are given to the CLIP image encoder (IE) f^{CI} and CLIP text encoder (TE) f^{CT}, respectively and transformed to output features. And then, the two output features (ie. image feature and text feature) are merged by the weight summation (a). The rest of the images in the source domain dataset, except for the input image are given to the image encoder to extract image features (b). After extracting features, the cosine similarities between (a) and (b) are calculated. In the right side of the Fig. 4, we visualized text prompts with source domain images ranked by their similarity. The samples having the highest cosine similarity might be most relevant samples to the combined input image $\mathbf{x}$ and text prompt $\mathbf{t}$. From the visualization, we could observe that such an assumption is valid.

Methods	FreiHAND→STB	FreiHAND→RHD
	EPE↓	EPE↓
Zhang et al. [60]	36.1	48.3
Ours	**34.9**	**46.8**

Table 3. Comparison with the state-of-the-art methods on STB, RHD datasets. Units are in mm scale.

4.3. Ablation study.

We assess the contribution of the CLIP model applied to hand pose estimation in our method: We configure four variants in our framework: 1) We prove the effectiveness of the CLIP model on the domain generalization problem through comparisons with and without the CLIP network, and study the optimal ratio between $f^{\text{CI}}(\mathbf{x})$ and $f^{\text{CT}}(\mathbf{t})$ when generating the combined features $f^{\text{CLIP}}(\mathbf{x}, \mathbf{t})$ from them in Table 4(a). 2) For the optimal combination of hand pose estimator features and CLIP features, we study which operation is effective to combine CLIP features to the original features between 'concatenation' and 'summation'. We also obtained the results for the case when we only used the CLIP text encoder for applying the style augmentation in Table 4(b). 3) We experimented to find the optimal weight ratio (ie. λ_3) between contrastive loss and other loss functions in Table 5. 4) Finally, we compared the performance of our method and the method applied with normal data

Methods	Ratio	STB	RHD
		EPE↓	EPE↓
Baseline		36.11	48.36
IE		35.00	48.37
IE+TE	0.9	36.69	**46.82**
	0.8	35.75	49.43
	0.6	**34.97**	47.34
	0.4	47.25	63.01

(a)

Methods	Combination	STB	RHD
	-	EPE↓	EPE↓
Baseline	-	36.11	48.36
TE	Concatenation	38.12	49.46
	Summation	35.93	48.36
IE+TE	Concatenation	**34.97**	**47.34**
	Summation	42.98	48.24

(b)

Table 4. Ablation study for hyper-parameters. (a) We compared results for various ratios of CLIP encoders. Here, the ratio indicates the weight assigned to the CLIP image encoder f^{CI}, 1-ratio is the weight assigned to the text encoder f^{CT}. The weight summation using the ratio of 0.6 and 0.9 works best for STB and RHD datasets, respectively. (b) We compared the combination methods among the concatenation and summation operations. We could find that the concatenation method works better than summation method.

Methods	λ_3	STB	RHD
		EPE↓	EPE↓
IE+TE	1	37.67	53.14
	0.1	34.97	**47.34**
	0.01	35.05	50.92
	0.001	**30.65**	50.97

Table 5. (a) A comparison of our methods under various λ_3 values of the contrastive loss. The best λ_3 value is set as 0.001 and 0.1 for STB and RHD datasets, respectively. However from the cross-validation, we set our λ_3 as 0.1 and report it as the final accuracy.

Methods	STB	RHD
	EPE↓	EPE↓
Ours	**34.97**	**46.82**
Normal Augmentation	38.92	49.86

Table 6. (a) A comparison of our method versus the normal augmentation method. At the normal augmentation method, there are six types of augmentation applied (ie. color jitter, cut out, gaussian noise, sobel filter, color drop, and gaussian blur). Ours consistently and significantly works better than the normal augmentation method.

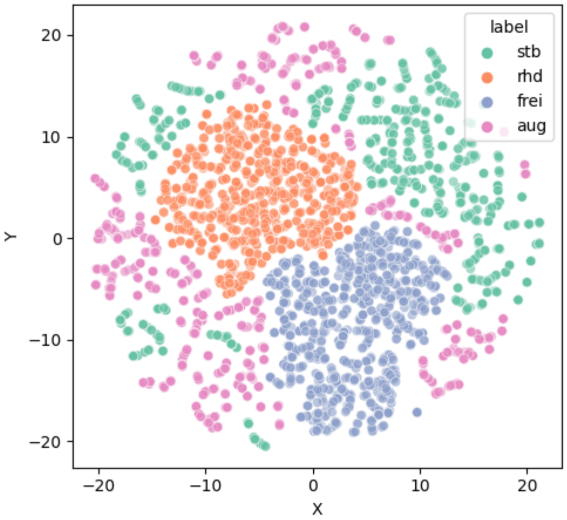

Figure 3. Visualization of T-SNE distribution obtained using CLIP image and text encoder features from source dataset (FreiHAND) (ie. 'aug') and CLIP image features from existing datasets (ie. 'stb' for STB, 'rhd' for RHD and 'frei' for FreiHAND datasets).

augmentation in Table 6. In Table 4(a), we compared the performance of several variants: baseline, our method using only CLIP image encoder (IE) and our method using CLIP image encoder + CLIP text encoder (IE+TE). The error rate is reduced by 3.08% in STB even when only CLIP image encoder is further involved. However, there is an significant improvement in both STB and RHD when both CLIP image encoder and CLIP text encoder are involved. In Table 4(a), the optimal weight ratio between $f^{\mathrm{CI}}(\mathbf{x})$ and $f^{\mathrm{CT}}(\mathbf{t})$ is seemingly 0.6 and 0.9 in STB and RHD, respectively. We obtained the values from the 10-fold cross-validation. In Table 4(b), we study the effectiveness of IE+TE method on the style augmentation. From our results, we can observe that the sole CLIP text encoder (TE) is not enough to achieve the domain generalization to other domains since it does not have the context information from the source domain. On the other hand, 'IE+TE' can maintain context information of source domain and augmented style information via both CLIP image encoder and CLIP text encoder, thereby achieving the complete domain generalization. We also study the effect of two operations when combining hand pose estimator features and CLIP features: concatenation and summation. Also, the results showed that the concatenation is more effective operation than the summation to incorporate CLIP features into the hand pose estimation features. In Table 5, we showed results for the optimal weight between the contrastive loss and other loss functions (ie. λ_3). The best performance is obtained when $\lambda_3 = 0.001$ and $\lambda_3 = 0.1$ for STB and RHD, respectively in the table; while we set

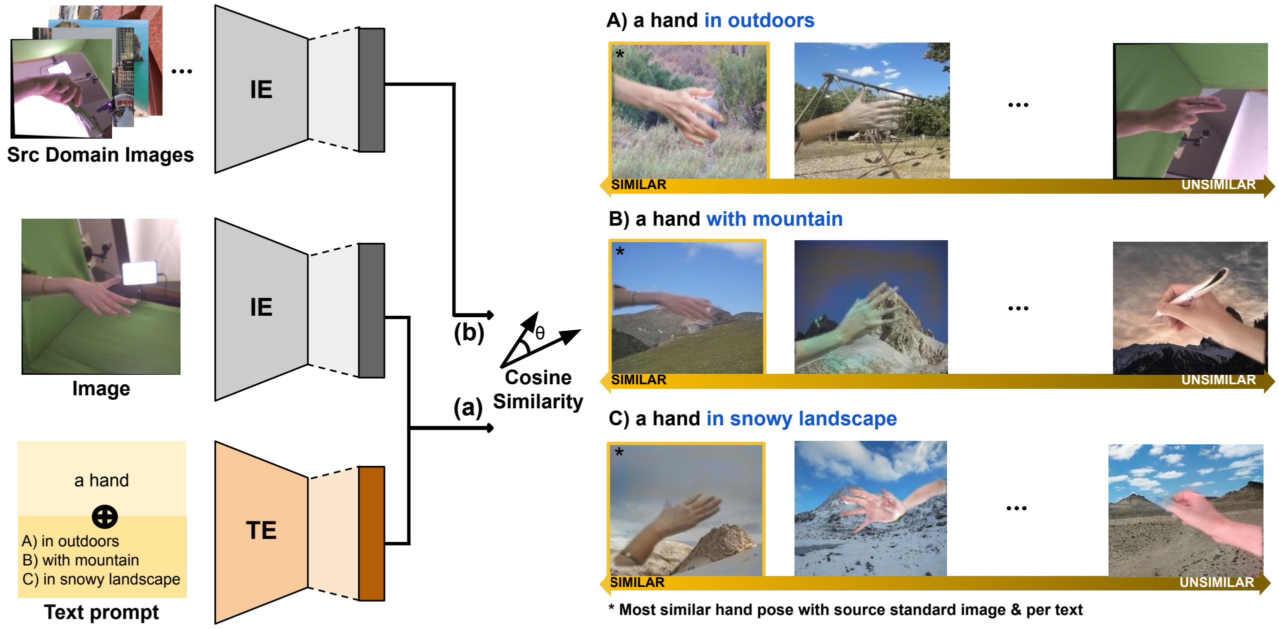

Figure 4. A visualization of FreiHAND images which has similar features to CLIP-augmented features. When the image of the source domain (FreiHAND) and the text prompt for the augmentation are given (left), in (a), we first weight-sum the features obtained from CLIP image encoder (IE) and CLIP text encoder (TE). After that, for (b), we apply the CLIP image encoder to all FreiHAND source data without the input image used in (a). After calculating the cosine similarity between features obtained in (a) and (b), images could be ranked according to their similarity and ranked images are visualized in the right side of this figure. The ranked images are seemingly well aligned with the text prompts..

$\lambda_3 = 0.1$ for both datasets as the value is obtained from 10-fold cross-validation. Thus, we report $\lambda_3 = 0.1$ results both for STB and RHD datasets. Finally, we compare the normal data augmentation method with ours in Table 6. We showed that our augmentation method consistently and significantly performs better than the normal data augmentation method. For the normal data augmentation, we involved six types of augmentation: color jitter, cut out, gaussian noise, sobel filter, color drop, and gaussian blur.

5. Conclusion

In this paper, we propose the 3D hand pose estimation framework that generalizes well to unseen domain datasets. Existing domain adaptation/generalization approaches tried to relieve the dataset-bias or domain-shift problem among different datasets by supplying additional unconstrained/-target domain dataset during the training stage. However, it takes a lot of time and effort to collect such datasets or sometimes impossible to get the exact target domain distribution. In this paper, we proposed image-free domain generalization framework that involves the CLIP model for generating style-augmented feature by the text prompt which is related to hand domains. The hand pose estimator becomes to have the style-augmented features through the contrastive learning mechanism, so the hand pose estimation network becomes robust to unseen domain data. In experiments, we trained our network on FreiHAND dataset and test on two popular hand pose estimation datasets (ie. RHD and STB). For two datasets, we have achieved the state-of-the-art performance in domain generalization setting by decreasing the error rates by 3.33% and 3.11% compared to previous state-of-the-art method in STB and RHD datasets, respectively. We demonstrated that the text prompt can augment the style information and makes our model to generalize to other domain datasets. We expect that our method can be extended to other tasks or in other pose estimation tasks (ie. human body pose and animal pose estimation).

Acknowledgements. This work was supported by IITP grants (No. 2021-0-01778 Development of human image synthesis and discrimination technology below the perceptual threshold 20%; No. 2020-0-01336 Artificial intelligence graduate school program (UNIST) 20%; No. 2021-0-02068 Artificial intelligence innovation hub 20%; No. 2022-0-00264 Comprehensive video understanding and generation with knowledge-based deep logic neural network 20%) and the NRF grant (No. 2022R1F1A1074828 20%), all funded by the Korean government (MSIT).

References

[1] Anil Armagan, Guillermo Garcia-Hernando, Seungryul Baek, Shreyas Hampali, Mahdi Rad, Zhaohui Zhang, Shipeng Xie, Neo Chen, Boshen Zhang, Fu Xiong, Yang Xiao, Zhiguo Cao, Junsong Yuan, Pengfei Ren, Weiting Huang, haifeng sun, Marek Hrúz, Jakub Kanis, Zdeněk Krňoul, Qingfu Wan, Shile Li, Dongheui Lee, Linlin Yang, Angela Yao, Yun-Hui Liu, Adrian Spurr, Pavlo Molchanov, Umar Iqbal, Philippe Weinzaepfel, Romain Brégier, Grégory Rogez, Vincent Lepetit, and Tae-Kyun Kim. Measuring generalisation to unseen viewpoints, articulations, shapes and objects for 3D hand pose estimation under hand-object interaction. In *ECCV*, 2020.

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

[3] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Real-time online action detection forests using spatio-temporal contexts. In *WACV*, 2017.

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

[5] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Weakly-Supervised Domain Adaptation via GAN and Mesh Model for Estimating 3D Hand Poses Interacting Objects. In *CVPR*, 2020.

[6] Seungryul Baek, Zhiyuan Shi, Masato Kawade, and Tae-Kyun Kim. Kinematic-layout-aware random forests for depth-based action recognition. In *BMVC*, 2017.

[7] Binod Bhattarai, Seungryul Baek, Rumeysa Bodur, and Tae-Kyun Kim. Sampling Strategies for GAN Synthetic Data. In *ICASSP*, 2020.

[8] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *NIPS*, 2020.

[9] Junuk Cha, Muhammad Saqlain, Donguk Kim, Seungeun Lee, Seongyeong Lee, and Seungryul Baek. Learning 3D skeletal representation from Transformer for action recognition. *IEEE Access*, 2022.

[10] Junuk Cha, Muhammad Saqlain, GeonU Kim, Mingyu Shin, and Seungryul Baek. Multi-Person 3D Pose and Shape Estimation via Inverse Kinematics and Refinement. In *ECCV*, 2022.

[11] Junuk Cha, Muhammad Saqlain, Changhwa Lee, Seongyeong Lee, Seungeun Lee, Donguk Kim, Won-Hee Park, and Seungryul Baek. Towards single 2D image-level self-supervision for 3D human pose and shape estimation. *Applied Sciences*, 2021.

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[13] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *Arxiv*, 2020.

[14] Xinghao Chen, Guijin Wang, Hengkai Guo, and Cairong Zhang. Pose guided structured region ensemble network for cascaded hand pose estimation. *Neurocomputing*, 2020.

[15] Yujin Chen, Zhigang Tu, Liuhao Ge, Dejun Zhang, Ruizhi Chen, and Junsong Yuan. So-handnet: Self-organizing network for 3d hand pose estimation with semi-supervised learning. In *ICCV*, 2019.

[16] Jian Cheng, Yanguang Wan, Dexin Zuo, Cuixia Ma, Jian Gu, Ping Tan, Hongan Wang, Xiaoming Deng, and Yinda Zhang. Efficient virtual view selection for 3d hand pose estimation. *ArXiv*, 2022.

[17] Wencan Cheng, Jae Hyun Park, and Jong Hwan Ko. Handfoldingnet: A 3d hand pose estimation network using multiscale-feature guided folding of a 2d hand skeleton. In *ICCV*, 2021.

[18] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David J Crandall. Hope-net: A graph-based model for hand-object pose estimation. In *CVPR*, 2020.

[19] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Advances in neural information processing systems. In *NIPS*, 2019.

[20] Linpu Fang, Xingyan Liu, Li Liu, Hang Xu, and Wenxiong Kang. Jgr-p2o: Joint graph reasoning based pixel-to-offset prediction network for 3d hand pose estimation from a single depth image. In *ECCV*, 2020.

[21] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016.

[22] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *CVPR*, 2018.

[23] Liuhao Ge, Yujun Cai, Junwu Weng, and Junsong Yuan. Hand pointnet: 3d hand pose estimation using point sets. In *CVPR*, 2018.

[24] Liuhao Ge, Zhou Ren, and Junsong Yuan. Point-to-point regression pointnet for 3d hand pose estimation. In *ECCV*, 2018.

[25] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *CVPR*, 2020.

[26] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *CVPR*, 2019.

[27] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[28] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *ICML*, 2020.

[29] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018.

[30] Weiting Huang, Pengfei Ren, Jingyu Wang, Qi Qi, and Haifeng Sun. Awr: Adaptive weighting regression for 3d hand pose estimation. In *AAAI*, 2020.

[31] Pin Jiang, Aming Wu, Yahong Han, Yunfeng Shao, Meiyu Qi, and Bingshuai Li. Bidirectional adversarial training for semi-supervised domain adaptation. In *IJCAI*, 2020.

[32] Dong Uk Kim, Kwang In Kim, and Seungryul Baek. End-to-End detection and pose estimation of two interacting hands. In *ICCV*, 2021.

[33] Taekyung Kim and Changick Kim. Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation. In *ECCV*, 2020.

[34] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *CVPR*, 2020.

[35] Fanqing Lin, Connor Wilhelm, and Tony Martinez. Two-hand global 3d pose estimation using monocular rgb. In *WACV*, 2021.

[36] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *ICCV*, 2021.

[37] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *NIPS*, 2018.

[38] Jameel Malik, Ibrahim Abdelaziz, Ahmed Elhayek, Soshi Shimada, Sk Aziz Ali, Vladislav Golyanik, Christian Theobalt, and Didier Stricker. Handvoxnet: Deep voxel-based network for 3d hand shape and pose estimation from a single depth map. In *CVPR*, 2020.

[39] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *CVPR*, 2018.

[40] 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 *ECCV*, 2020.

[41] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *CVPR*, 2018.

[42] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *Arxiv*, 2018.

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[44] Pengfei Ren, Haifeng Sun, Jiachang Hao, Jingyu Wang, Qi Qi, and Jianxin Liao. Mining multi-view information: A strong self-supervised framework for depth-based 3d hand pose and mesh estimation. In *CVPR*, 2022.

[45] Pengfei Ren, Haifeng Sun, Weiting Huang, Jiachang Hao, Daixuan Cheng, Qi Qi, Jingyu Wang, and Jianxin Liao. Spatial-aware stacked regression network for real-time 3d hand pose estimation. *Neurocomputing*, 2021.

[46] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *ICCV*, 2019.

[47] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Adversarial dropout regularization. *ArXiv*, 2017.

[48] Muhammad Saqlain, Donguk Kim, Junuk Cha, Changhwa Lee, Seongyeong Lee, and Seungryul Baek. 3DMesh-GAR: 3D human body mesh-based method for group activity recognition. *Sensors*, 2022.

[49] Seonguk Seo, Yumin Suh, Dongwan Kim, Geeho Kim, Jongwoo Han, and Bohyung Han. Learning to optimize domain specific normalization for domain generalization. In *ECCV*, 2020.

[50] Adrian Spurr, Aneesh Dahiya, Xi Wang, Xucong Zhang, and Otmar Hilliges. Peclr: Self-supervised 3d hand pose estimation from monocular rgb via contrastive learning. In *ICCV*, 2021.

[51] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, 2020.

[52] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Dual grid net: Hand mesh vertex regression from single depth maps. In *ECCV*, 2020.

[53] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Dense 3d regression for hand pose estimation. In *CVPR*, 2018.

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

[55] Fu Xiong, Boshen Zhang, Yang Xiao, Zhiguo Cao, Taidong Yu, Joey Tianyi Zhou, and Junsong Yuan. A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In *ICCV*, 2019.

[56] Jiaolong Xu, Liang Xiao, and Antonio M López. Self-supervised domain adaptation for computer vision tasks. *IEEE Access*, 2019.

[57] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *CVPR*, 2020.

[58] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *ICCV*, 2019.

[59] Qiming Zhang, Jing Zhang, Wei Liu, and Dacheng Tao. Category anchor-guided unsupervised domain adaptation for semantic segmentation. *NIPS*, 2019.

[60] Xiheng Zhang, Yongkang Wong, Xiaofei Wu, Juwei Lu, Mohan Kankanhalli, Xiangdong Li, and Weidong Geng. Learning causal representation for training cross-domain pose estimator via generative interventions. In *ICCV*, 2021.

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

[62] Benjin Zhu, Junqiang Huang, Zeming Li, Xiangyu Zhang, and Jian Sun. Eqco: Equivalent rules for self-supervised contrastive learning. *Arxiv*, 2020.

[63] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *ICCV*, 2017.

[64] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *ICCV*, 2017.

[65] 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 *CVPR*, 2019.

[66] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018.