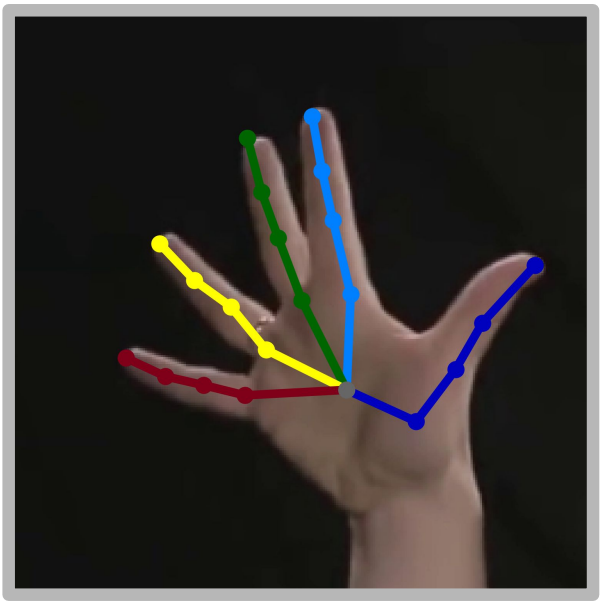
Exploring self-supervised learning techniques for hand pose estimation

Exploring self-supervised learning techniques for hand pose estimation



Aneesh Dahiya

Master Thesis
March 2021

Supervisors:
Adrian Spurr
Prof. Dr. Otmar Hilliges

ETH zürich

Advanced Interactive Technologies

Abstract

Estimating 3D hand pose from a monocular RGB image is a challenging task. This is largely due to the limited amount of available labeled data, as annotating images for 3D hand pose requires a complex multi-camera setup and a controlled lab-like setting. This in turn introduces a domain gap between the different hand pose datasets and the unconstrained settings of the real world. In this thesis, we develop a self-supervised method to use unlabeled data from different hand pose datasets to improve the accuracy of 3D hand pose estimation, and to bridge the domain gap. We propose a novel contrastive learning framework for pose estimation, inspired by the recent success of contrastive learning on image classification tasks. In a standard contrastive learning framework, a model tries to learn a feature representation that is invariant under any image augmentation. This can be beneficial, as the pose is invariant to appearance based image augmentations. However, geometric augmentations (like rotation) change the pose equivariantly. However using geometric augmentations with contrastive self-supervision leads to invariance. This can be detrimental to the pose estimation. We empirically show that the features learned with our equivariant contrastive framework lead to more improvement when compared to standard contrastive frameworks. Furthermore, we attain an improvement of 7.6% in PA MKP-3D on FreiHAND with a standard ResNet-152, trained with additional unlabeled data when compared to a fully supervised baseline. This enables us to achieve state-of-the-art performance in a purely data driven way, without any task-specific specialized architecture.

Acknowledgement

I would like to thank my supervisor Adrian Spurr whose insightful comments and ideas made this work possible. I would also like to extend my gratitude towards Prof. Hiliges and everyone at the AIT lab who gave their valuable feedback to improve the quality of this work.

Lastly, I would like to thank my mother, without her support, encouragement and sacrifices I would not be where I am today.

Contents

List of Figures

List of Tables

Introduction

The advancements in computing power, image capturing and machine learning have made a plethora of tasks possible. One of these tasks is reliably estimating the 3D hand pose from images. This has several applications in the field of robotics, virtual reality, augmented reality and mixed reality. Early approaches make use of RGB images combined with depth map to estimate the pose. In recent years the research community shifted its focus to the estimation of hand pose from solely monocular RGB images, as cameras without depth sensors/stereo are far more ubiquitous, cheaper and have lower power requirements. In this work we focus on estimating the hand pose from monocular RGB images.

The estimation of 3D pose from RGB image is a challenging task. Amongst others, conditions that significantly contribute to its difficulty are a large diversity in backgrounds, lighting conditions, hand appearances and self-occlusion arising from high degrees of freedom of the human hand. There are several ways to deal with it. The most straight forward way is to use more labeled data that spans a large diversity of lightning conditions, environments and poses. However, acquiring $3D$ labeled data is laborious and expensive, as it requires a large lab-like setting. The labeled data hence collected often does not translate well to in-the-wild imagery [20, 42]. The hand pose community has been relying increasingly more on methods that can efficiently use supplementary data with $2D$ joint annotations or no annotations. Such annotations are generally cheaper to acquire. Research has shown that the inclusion of such auxiliary data [4, 14, 31] leads to a better prediction accuracy. [31] shows that one can outperform many supervised approaches by using weakly-supervised data more effectively via kinematic priors, [14] exploits temporal information to improve accuracy and [4] uses weak supervision with depth maps. Other works has also explored using an off-the-shelf joint detector to automatically generate the 2D joint labels, like in [20] where the authors use OpenPose [5] to automatically generate $2D$ annotations. However, there is no guarantee that these poses are indeed correct and the accuracy one can achieve with such an approach is bounded by the performance of the

OpenPose model.

Alternatively, one could resort to using unlabeled data directly with the help of self-supervision. In a nutshell, these methods train for a pretext task that does not require labels. This means self-supervised methods do not suffer from the uncertainty introduced by noisy labels in the auxiliary data. Contrastive learning is one such self-supervised approach. In contrastive learning, data is encoded to a latent space with the help of a neural network. Its main objective is to minimize the similarity between dissimilar inputs while maximize it between similar inputs in the latent space representation. In this method, similar images are generated by applying image augmentations to an image and dissimilar inputs are generated by applying similar augmentations to a separate image.
Recently, works like [6, 7] have shown that with contrastive learning one can reach parity or even outperform supervised baseline models on an image classification task. This raises an interesting question: *Does the contrastive self-supervised approach extend to structured regression tasks as well?* We hypothesize that features learned during standard contrastive training may not readily transfer to regression-based tasks, as optimizing the standard contrastive objective will result in features being *invariant* to any augmentation used during training. This also leads to invariance in the feature representation where equivariance is desired. For example, rotating an image rotates the underlying pose equivariantly. However, the objective function of a standard contrastive learning framework will encourage invariance to this rotation and will not encode the desired equivariance. In this work, we propose a *pose equivariant contrastive learning (PeCLR)* framework that tackles this issue. PeCLR learns a feature representation that is equivariant to geometric augmentations but maintains invariance to appearance based augmentations. Therefore, PeCLR enables the use of a large amount of unlabeled data to learn a general feature representation that can be used for hand pose estimation with supervised fine-tuning.

In this work, we employ a two stage training. In the first stage, we learn a general feature representation from a large amount of unlabeled data with contrastive learning. In this stage, we apply geometric and appearance based augmentations to generate similar and dissimilar input images. These images are used to train an encoder via our proposed equivariant contrastive loss, where we revert the geometric augmentations in the feature space. In the second stage, the trained encoder from the first stage is fine-tuned on the task of 3D hand pose estimation using labeled data. We investigate this setup in a variety of experiments. We demonstrate an increased label efficiency and show that using more unlabeled data improves the 3D performance by 43% (PA MKP-3D), when only 10% of the labeled data is used for supervised training(cf. 5.5b). Furthermore, we show that our approach reaches the state-of-the-art performance with a standard ResNet-152, outperforming several specialized architectures. Lastly, we show that the benefits of pretraining with our approach also extend to cross-domain generalization, where we observe an improvement of 4.8% (PA MKP-3D) on a dataset not included in the supervised training.

In summary our contribution is as follows:

- To the best of our knowledge, we perform the first investigation of contrastive learning to efficiently leverage unlabeled data for hand pose estimation.
- We propose a novel contrastive learning objective that encourages equivariance to geometric augmentations and invariance to appearance based augmentations.

- We conduct controlled experiments to evaluate the quality of learned representation for several augmentations, compare it with SimCLR and empirically derive the best performing augmentations.
- We show that representations learned with our contrastive objective leads to a higher label efficiency and that adding more unlabeled data is beneficial.
- We demonstrate that our proposed method outperforms more specialized state-of-the-art methods, using a simple ResNet architecture.

The thesis is structured as follows. Chapter 2 touches on the prior work. Chapter 3 defines the task and sets up the theory. Chapter 4 delineates the methodology. Chapter 5 presents the experiment protocol and results. Chapter 6 summarizes the findings and avenues for further research and improvements.

Related work

2.1. Hand pose estimation

3D hand pose estimation involves predicting 3D joint skeletons or the MANO [30] mesh.[1] Works like [4, 11, 18, 26, 27, 31–33, 37, 41] predict the 3D keypoints, whereas some works [1–3, 14, 15, 40] predict the parameters of the parametric hand model to obtain the MANO mesh. Alternatively, works like [13, 21, 25] directly predict the full MANO mesh of the hand.

3D labeled data is scarce, therefore the research community has worked on ways to avoid overfitting and to improve the prediction accuracy. A staged approach, where the 3D keypoints are lifted from the 2D predictions is discussed in [41]. Mueller *et al*. [27] use General Adversarial Network and synthetic datasets to reduce synthetic/real discrepancy. Cai *et al*. [4] include supplementary depth supervision to augment the training set. Yang *et al*. [37] use disentangled latent space for hand pose estimation and image synthesis. Iqbal *et al*. [18] propose a more efficient 2.5D representation that can be used to predict the 3D pose. Weak supervision with the help of bio mechanical constraints is used to refine the pose prediction on 2D supervised data by [31], whereas [11] learns these constraints by using a graph based neural network in final layers to refine the predicted hand pose. Moon *et al*. [26] take the hand interaction into account to predict the pose of both hands. In the work of [33], action recognition as well as hand pose estimation is performed.

The use of MANO introduces a prior of hand poses and a mesh surface. Works like [1, 3, 40] estimate the MANO parameters directly from RGB images. [3, 40] use in-the-wild 2D annotations, whereas [1, 40] make use of hand masks for weak supervision to predict the MANO parameters. [15] predicts the object and MANO parameters jointly in a unified approach. [14]

[1] Section 3.2 discusses these representations in further detail.

further develops on the framework and employs a photometric loss on the partially labeled sequences. In contrast to regressing the parameters of the MANO model, some works directly regress the entire MANO mesh [13, 21, 25]. Spiral convolutions are used to predict the hand mesh in [21]. That being said, predicting MANO mesh/parameters suffers from the same problem the 3D keypoints face, *i.e*. scarcity of labeled data. Ge *et al*. [13] tackle this by introducing a fully mesh annotated synthetic dataset and perform noisy supervision for real data. An alternative to MANO is proposed in [25] by using a base hand model and predicting the pose and the subject dependent correctives.

Most of the work in recent years has been dedicated to custom, sometimes highly specialized, architectures. However, in this work we explore purely data driven approaches, utilizing unlabeled data and an equivariance inducing contrastive formulation to achieve a state-of-the-art performance with a standard ResNet model.

2.2. Self-supervised learning

Self-supervised learning is a paradigm that aims to learn the representation of data without any annotations, by optimizing a cost function that encodes a pre-text task. A pre-text task is a pre-designed task for a model to solve in order to learn meaningful features in a self-supervised way. A pre-text task could be: predicting the position of a second patch relative to the first [10], colorizing a gray scale image [39], solving a jigsaw puzzle [28], estimating the motion flow of pixels in a scene [35], predicting positive future samples in audio signals [29] or completing the next sentence based on the relation between two sentences [9]. The representations learned by solving a pretext task are usually used to help solve a main downstream task. For instance, the pretext task of colorizing a gray scale image could be used for object recognition as the image color is important for the object classification [12]. Not all pretext tasks are the same and solving one might not lead to any improvements in the downstream performance. For instance, intuitively speaking, colorizing a gray scale image may not help with the hand pose estimation as the color of the image is not an important feature for pose estimation. The selection of appropriate pre-text tasks is important because an optimal pre-text task can lead to improvements in terms of performance and generalizability.

In this work, we use a contrastive learning objective as a pretext task for self-supervision. This pre-text has shown a lot of promise in downstream tasks such as image classification, video classification, object detection and speech classification. However, contrastive learning has not yet been investigated for the task of hand pose estimation. Recent works in contrastive learning that are closest to the work presented in this thesis are Contrastive Predictive Coding (CPC) [17, 29], Contrastive Multiview Coding (CMC) [34], and SimCLR [6, 7]. CPC learns to extract representations by predicting the future in latent space with auto-regressive models. In CPC, authors show that the feature representations learned this way achieve strong performances on four distinct domains: speech, images, text and reinforcement learning in 3D environments. CMC learns view-agnostic representations by maximizing mutual information among different views of the same scene and shows that contrastive loss outperforms a loss based on cross-view prediction. SimCLR extends the contrastive loss used in CMC to a more simplified contrastive learning framework. SimCLR uses strong image augmentations to generate views of a scene

and shows that the representations learned this way are on par with supervised models on the image classification task. In this approach a latent space representation is generated for each data. The core idea being, if the data points are connected in a meaningful way, then the corresponding latent space representations should lie “close" to each other, whereas latent space representations of unrelated data are further apart. The “close"(ness) is measured by a suitable distance metric measured in the latent space. Not all augmentations described in SimCLR are suitable for structured regression tasks such as hand pose estimation. Hand pose estimation is equivariant to geometric augmentations and invariant to appearance based augmentations. For instance, rotating the input image rotates the underlying hand pose by the same amount whereas changing the color of the image doesn’t alter the underlying pose. We address this issue and extend contrastive learning to structured regression tasks by differentiating between appearance based and geometric augmentations. More specifically, while we keep the invariant contrastive objective, we require equivariance for any geometric transformations. This results in the feature representation learned by our approach to be more suited for pose estimation tasks.

Background

In this chapter, we introduce the necessary background knowledge and explain terms and concepts used throughout this thesis. We start by establishing the general notation used in this work in section 3.1. In section 3.2, we give a brief introduction of two common ways to represent a 3D hand pose. We then discuss the steps for converting the 3D representation used in this work to the target labels for supervised training. This is done with the help of a pinhole camera model and hence relies on the knowledge of the camera parameters. The camera parameters are not available for some of the data used in this work. Therefore, in section 3.4, we discuss a method to enable 3D pose estimation in this scenario. We follow it up with section 3.5, which discusses the metrics used to quantify the quality of the predicted pose. Lastly, in section 3.6, we describe the datasets used in this work.

3.1. Notation

We follow the notation most commonly used by the computer vision community. We use bold capital font for matrices "$\boldsymbol{X}$", bold lower case for vectors "$\boldsymbol{x}$" and roman font for scalars "x". Additionally, we assume the image to be monocular RGB that corresponds to a right hand.[1]

3.2. Pose representation

There are two commonly followed ways of representing a hand pose in the 3D space, *i.e*. with a skeleton based **keypoint approach** or through a mesh-based parametric **hand shape model**.

[1] An image of left hand can be horizontally flipped to make it image of right hand.

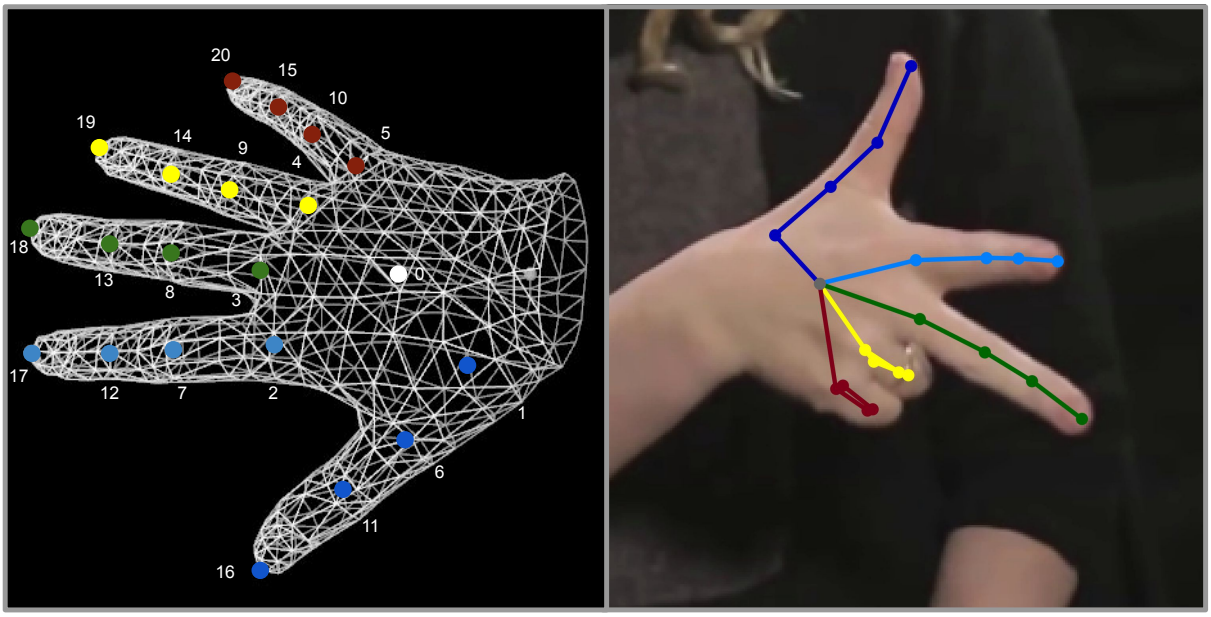


Figure 3.1.: **Pose representation**. We illustrate the MANO[30] (left) representation of a right hand with 21 joints of keypoint based approach overlapped. We show a monocular RGB image(right) from YouTube 3D Hands [20] overlapped with 2D projection of keypoint based pose representation in image plain.

The skeleton-based keypoint representation is the most straightforward way to represent the hand pose. It defines the structure of the hand with the location of a predetermined set of joints in 3D space. On the other hand, a parametric hand shape model usually defines a volumetric hand with the help of pose and shape parameters. In recent years, the most popular way of predicting hand pose has been using MANO [30]. Figure 3.1 (left) shows the MANO representation of a hand. The figure also shows the joints used in the keypoint based approach.

In this work, we focus on the keypoint based approach to represent a hand in the 3D space. During supervised training, 3D keypoints are mapped to a 2.5D representation introduced in [18]. The first two dimensions in the 2.5D representation are the 2D projections of the 3D pose on the image plane. The "half" dimension corresponds to the scale-normalized root relative depths. The normalization is achieved by scaling 3D keypoints with a hand-specific scalar s. In this work, we use the length of the bone between the index metacarpophalangeal joint and the palm as scale *i.e.* the euclidean distance between keypoint 2 and keypoint 0 in figure 3.1(left).

We explain the steps involved in converting the 3D pose from and to the 2.5D pose. We define $\boldsymbol{p_i}$ as the i^{th} joint in 3D space and $\boldsymbol{K}$ as the camera matrix. Then we can write the 2D projection on the image plane using the pinhole camera model as follows:

$$z_i \boldsymbol{q_i} = \boldsymbol{K} \boldsymbol{p_i}. \tag{3.1}$$

Where,

$$\boldsymbol{q_i} = \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} ; \boldsymbol{p_i} = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} ; \boldsymbol{K} = \begin{pmatrix} f & 0 & t_u \\ 0 & f & t_v \\ 0 & 0 & 1 \end{pmatrix} [\boldsymbol{R_c} | \boldsymbol{t_c}]. \tag{3.2}$$

In equation 3.2, f is the focal length of the camera, (t_u, t_v) is the shift in the image center, $\boldsymbol{R_c} \in \mathbb{R}$ is the rotation of the camera frame with respect to the world coordinate frame and $\boldsymbol{t_c}$ is the translation of the camera frame with respect to the world camera frame.

The normalized root relative depth is calculated with respect to a reference keypoint. Here we use palm as the reference. The depth of the reference keypoint is called the root depth z_o. The depths calculated with respect to the reference keypoint are then normalized with the scale to obtain the normalized root relative depth as follows:

$$\tilde{z}_i^r = \frac{1}{s}(z_i - z_0). \tag{3.3}$$

Where,

$$s = ||\boldsymbol{p}_2 - \boldsymbol{p}_0||_2. \tag{3.4}$$

The scaling of the 3D joints with the scalar s does not affect the 2D camera projection because,

$$z_i \boldsymbol{q_i} = \boldsymbol{K}\boldsymbol{p_i} \implies \frac{z_i}{s}\boldsymbol{q_i} = \frac{1}{s}\boldsymbol{K}\boldsymbol{p_i}. \tag{3.5}$$

In summary, the 2.5D representation $\boldsymbol{p}_i^{2.5D}$ of a point $\boldsymbol{p}_i$ is,

$$\boldsymbol{p}_i^{2.5D} = \begin{pmatrix} u_i \\ v_i \\ \tilde{z}_i^r \end{pmatrix}. \tag{3.6}$$

The 2.5D representation is an efficient way to represent a 3D hand pose. This representation can be directly used as training labels for supervised training. On the other hand, 3D representation by itself is an inefficient way to train the model directly, as estimating the 3D pose from a monocular image is a severely ill posed problem. This is due to the scale and the depth ambiguity in a 2D image projection. Estimating the 3D pose directly from a monocular image leads at best to over-fitting on a specific environment and subject.

Obtaining the 3D pose from the 2.5D pose is also very straightforward. It is obtained by first determining the depth of the palm as follows:

$$z_0 = \frac{s(-b + \sqrt{b^2 - 4ac})}{2a}. \tag{3.7}$$

Where,

$$a = (u_2 - u_0)^2 + (v_2 - v_0)^2, \tag{3.8}$$

$$b = 2(\tilde{z}_2^r(u_2^2 + v_2^2 - u_2u_0 - v_2v_0) + \tilde{z}_0^r(u_0^2 + v_0^2 - u_2u_0 - v_2v_0)), \tag{3.9}$$

and

$$c = (u_2\tilde{z}_2^r - u_0\tilde{z}_0^r)^2 + (v_2\tilde{z}_2^r - v_0\tilde{z}_0^r)^2 - (\tilde{z}_2^r - \tilde{z}_0^r)^2 - 1. \tag{3.10}$$

After the palm depth is known, the x_i, y_i are calculated from equation 3.1.

3.3. Depth refinement

Calculating the depth from the 2.5D representation involves the calculation of the root relative depth z_0. However, the calculation of z_0 is very sensitive to errors in 2.5D projections. Therefore, we refine the predicted root depth to increase the accuracy and improve the stability in the presence of outliers. This post process step is first introduced in [31]. Similar to [31], we use a MLP $\mathcal{M}_{ref}$ to refine the predicted root depth $\hat{z}_0$. $\mathcal{M}_{ref}$ refines the $\hat{z}_0$ as follows:

$$\hat{z}_0^{refined} = \hat{z}_0 + \mathcal{M}_{ref}(\hat{\boldsymbol{z}}_r, \boldsymbol{K}^{-1}\hat{\boldsymbol{J}}^{2D}, \hat{z}_0). \tag{3.11}$$

Here $\boldsymbol{K}$ is the camera matrix and $\hat{\boldsymbol{z}}^r$ is the vector of root relative depths of all coordinates. $\mathcal{M}_{ref}$ is trained by minimizing the following loss

$$\mathcal{L}_{ref} = |\hat{z}_0^{refined} - z_0|. \tag{3.12}$$

3.4. Procrustes alignment

In the section 3.2, we showed that one can obtain 3D pose, $\boldsymbol{J}^{3D} \in \mathbb{R}^{21\times3}$ from 2.5D pose, $\boldsymbol{J}^{2.5D}$. However, it relies on the knowledge of the scale s and the camera matrix $\boldsymbol{K}$. *What if we don't know $\boldsymbol{K}$ and s?* In this case, we assume $\boldsymbol{K}$ as identity, scale as 1 and calculate the 3D pose $\tilde{\boldsymbol{J}}^{3D}$ with these assumptions. The pose estimated with these assumptions is related to the original 3D pose $\boldsymbol{J}^{3D}$. The relation between the two can be written as follows:

$$\boldsymbol{J}^{3D} = k\tilde{\boldsymbol{J}}^{3D}\boldsymbol{R}^T + \boldsymbol{L}. \tag{3.13}$$

Where $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ is a 3D rotation matrix, $\boldsymbol{L} \in \mathbb{R}^{21\times3}$ is a translation matrix and $k \in \mathbb{R}$ is a scalar. All 21 rows of $\boldsymbol{L}$ correspond to the same 3D translation vector $(x \quad y \quad z)^T$. We can find $\boldsymbol{R}$, $\boldsymbol{L}$ and k by minimizing the frobenius norm between $\boldsymbol{J}^{3D}$ and $\tilde{\boldsymbol{J}}^{3D}$,

$$\min_{\boldsymbol{R},\boldsymbol{L},k} ||\tilde{\boldsymbol{J}}^{3D} - \boldsymbol{J}^{3D}||_F. \tag{3.14}$$

This process of aligning two point clouds in a N dimensional space by adjusting the scale, the rotation and the translation is called **Procrustes alignment**. The research community uses this method to estimate the quality of 3D predictions in scenarios where the camera matrix and the scale are either unreliable or unknown.

The relation in equation 3.13 comes from the assumptions of perspective projection in a pinhole camera model. In the following analysis we support this claim. We rewrite the equation 3.1 but expand[2] the camera matrix $\boldsymbol{K}$ in terms of the focal length f, the shift of the principal point(image center) from the origin t_u, t_v.

$$z_i \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & t_u \\ 0 & f & t_v \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}. \tag{3.15}$$

[2]We assume the 3D pose is measured with respect to the camera frame, *i.e.* the extrinsic camera matrix is identity.

The above relation can be decomposed as follows:

$$x_i = f\frac{x_i}{x_i} + t_u \implies x_i = u_i\frac{z_i}{f} - t_u\frac{z_i}{f}, \tag{3.16}$$

and

$$v_i = f\frac{y_i}{z_i} + t_v \implies y_i = v_i\frac{z_i}{f} - t_u\frac{z_i}{f}. \tag{3.17}$$

One can observe that scale normalizing the 3D coordinates does not change the camera projection u_i and v_i as

$$u_i = f\frac{x_i}{z_i} + t_u = f\frac{\frac{x_i}{s}}{\frac{z_i}{s}} + t_u, \tag{3.18}$$

and

$$v_i = f\frac{y_i}{z_i} + t_v = f\frac{\frac{y_i}{s}}{\frac{z_i}{s}} + t_v. \tag{3.19}$$

We can further write the above expression as

$$\hat{x_i} = \frac{x_i}{s} = u_i\hat{s} - \hat{t_u}, \tag{3.20}$$

and

$$\hat{y_i} = \frac{y_i}{s} = v_i\hat{s} - \hat{t_v}. \tag{3.21}$$

Where

$$\hat{s} = \frac{\frac{z_i}{s}}{f}, \hat{t_u} = t_u\frac{\frac{z_i}{s}}{f} \text{ and } \hat{t_v} = t_v\frac{\frac{z_i}{s}}{f}. \tag{3.22}$$

It is immediate from equation 3.20 and 3.22 that, provided we know u_i, v_i and z_i/s, we can obtain a scaled and shifted version of the original 3D pose. The 2.5D pose gives exactly those parameters. The scale and shift are adjusted with Procrustes alignment.

3.5. Metrics

In this section, we define the metrics that we use for quantitatively evaluating the performance of a model on the hand pose estimation task. We represent $\boldsymbol{J}^{3D} \in \mathbb{R}^{21\times3}$ as the ground truth and $\hat{\boldsymbol{J}}^{3D} \in \mathbb{R}^{21\times3}$ as the predictions. The 2.5D and 2D joints are represented by superscript 2.5D and 3D. Additionally, $\boldsymbol{J}_i^{3D(t)}$ represents the i^{th} keypoint for t^{th} ground truth sample.

Mean keypoint error 3D (MKP-3D) is the most common metric used to evaluate the 3D hand pose. It is the average euclidean distance between the predicted and the ground truth hand pose.

$$E_{\text{MKP-3D}} = \frac{1}{21N}\sum_{t=1}^{N}\sum_{i=1}^{21}||\boldsymbol{J}_i^{3D(t)} - \hat{\boldsymbol{J}}_i^{3D(t)}||_2, \tag{3.23}$$

where N is the size of the evaluation set. This metric sometimes suffers from outliers, as conversion from a 2.5D to a 3D pose can introduce some errors, especially when a in equation 3.7 approaches zero. This usually happens when the 2D projections of keypoints used for scale normalization lie very close to each other or overlap. This makes calculated z_0 very large and

it affects the metric disproportionately. The research community uses AUC to deal with such outliers.

Area under the curve (AUC) is used to get a better estimate of the performance by being less sensitive to outliers. In the case of the former, it represents the area under the Receiver Operating Characteristics (ROC), whereas in the latter it represents the area under the Percentage of Correct Keypoints (PCK) curve. PCK measures the mean percentage of predicted joint locations that fall under an error threshold. The error is measured in terms of euclidean distance. In our analysis we plot the PCK curve for the threshold, ranging from 0mm to 500mm in steps of 5mm.

In our analysis, we also evaluate the 2D performance of our models *i.e*. performance in terms of the 2D camera projections of the 3D pose. We use **Mean keypoint Error 2D (MKP-2D)** to quantify the 2D performance. Since the $2D$ predictions are not affected by the numerical instability arising from $2.5D$ to $3D$ conversion, we do not rely on other metrics for the 2D performance.

$$E_{\text{MKP-2D}} = \frac{1}{21N} \sum_{t=1}^{N} \sum_{i=1}^{21} ||\boldsymbol{J}_i^{2D(t)} - \hat{\boldsymbol{J}}_i^{2D(t)}||_2. \tag{3.24}$$

Lastly, we also measure the 3D metrics on the Procrustes aligned predictions and call those metrics **PA MKP-3D** and **PA-AUC**. These metrics are useful while evaluating datasets where the camera matrix $\boldsymbol{K}$ and normalizing scale s is not known.

3.6. Datasets

There are several hand pose datasets available. We focused on the most recent datasets for our analysis, namely **FreiHAND** (FH) [42] and **Youtube 3D Hands** (YT3D) [20].

FreiHAND consists of $32'560$ frames captured with a green screen background in the train set. Each sample in the train set is post processed in four different ways to remove the background. This inflates the training set to $4 * 32'560 = 130'240$ samples. The dataset doesn't contain a separate validation set. Therefore, we randomly sample 10% of the $32'560$ unique training samples and their corresponding augmented counterparts as our validation set. The test set contains $3'960$ samples. The 3D labels for the train set are not released publicly by the authors. Instead, to evaluate the performance on the test set, a competition is hosted by the authors on codaLab[3].

YouTube 3D Hands consists of in-the-wild images gathered from several youtube videos. It contains $47'125$, 1262 and 1262 samples in the train, validation and test set respectively. The 3D labels for each sample are automatically acquired via keypoint detection from OpenPose [5] and MANO [30] fitting. We use this data exclusively for self-supervision and evaluation, *i.e*. we don't use the 3D labels during training. In the context of the evaluation, the dataset doesn't contain the camera intrinsic matrix, therefore we report PA MKP-3D and MKP-2D.

[3]https://competitions.codalab.org/competitions/21238

Methods

In this chapter, we introduce the methods used for self-supervision with unlabeled data, followed by methods used for supervised training on the hand pose estimation task. In self-supervised methods, we first introduce the simplified contrastive learning framework from [6]. We identify an issue with this approach for structured regression tasks like hand pose estimation. We use the key issue as motivation to introduce our novel pose equivariant contrastive learning framework. In supervised methods, we describe the model used for pose estimation and describe the refinement steps detailed in [31], used to reduce the effect of outliers

4.1. Self-supervised contrastive learning

We use contrastive learning as a pre-training step in order to improve the downstream performance of hand pose estimation. It is a self-supervised learning method, hence does not use labels for training.

In a nutshell, a model trained with this approach generates an embedding for an input which satisfies two properties. Firstly, the embedding has a high similarity with the embeddings from similar inputs. Secondly, the embedding has a low similarity with the embeddings from dissimilar inputs. The end goal is that pre-training with this objective will result in the model learning generic visual representations that can further help improve the downstream performance upon task-specific finetuning. This approach has gained some traction in the computer vision community after the success of SimCLR [6] on image classification. The authors in [6] showed that the features learned by contrastive approaches achieve better performance than fully supervised approaches in classification. In our work we propose that this can be extended to structured regression based tasks as well, albeit with a few geometric modifications. We call our contrastive

framework *pose equivariant contrastive learning* or *PeCLR*. It builds upon SimCLR. In the following text, we describe the general setup and input processing shared by SimCLR and PeCLR. Next, we describe the loss formulation for SimCLR, followed by our proposed modifications. Further, we justify our modifications theoretically. In the end we summarize by describing the training algorithm of PeCLR.

4.1.1. Input

The input processing is the same for both SimCLR and PeCLR. We define $\boldsymbol{I}^n$ as the n^{th} raw image sampled from the training data. Let T be the set of random image augmentations such that the augmentation t sampled from this set can be decomposed into the appearance based augmentation t^a and the geometric augmentation t^g respectively. If $\boldsymbol{I}^n$ is the input to the model then positive samples are defined as,

$$t(\boldsymbol{I}^n), \forall t \in T, \tag{4.1}$$

whereas negative samples to x_i are,

$$t(\boldsymbol{I}^m), \forall t \in T; m \neq n. \tag{4.2}$$

The model gets the augmented views of the raw input image $\boldsymbol{I}$ as the input. In each iteration, the model samples both positive and negative pairs. For a given batch of N images, two random augmentations are applied to each raw image, resulting in 2N augmented images. Hence, for every augmented image $\boldsymbol{I}_i^n$, there is one positive sample $\boldsymbol{I}_j^n$ and $2(N-1)$ negative samples $\{\boldsymbol{I}_k^m\}_{m \neq n}$.

4.1.2. Model

The contrastive model consists of an encoder E and a projection head g. The similarity is measured for the 2D embedding $\boldsymbol{z} \in \mathbb{R}^{m \times 2}$ and generated by the projection head. The projection head in turn generates the embeddings from features $\boldsymbol{z}$, generated by the encoder from input $t(\boldsymbol{I})$. We represent the whole model as f. The relationships described can be expressed as follows:

$$\boldsymbol{z}_i^n = f(t_i(\boldsymbol{I}^n)) = g(E(t_i(\boldsymbol{I}^n))) = g(\boldsymbol{z_i^n}). \tag{4.3}$$

4.1.3. SimCLR loss function

The contrastive loss used by SimCLR ensures that negative samples are far apart and positive samples are close to each other in the latent space. The loss was termed as **normalized temperature scaled cross entropy (NT-Xent)** in [6]. Here we follow the same nomenclature. This loss function is also used in [17]. The (NT-Xent) loss for a input pair $\{\boldsymbol{I}_i^n, \boldsymbol{I}_j^n\}$ is,

$$\mathcal{L}_{i,j} = -\log \frac{\exp\left(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp\left(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau\right)}. \tag{4.4}$$

Where τ is the temperature parameter, $\text{sim}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^T\boldsymbol{v}/||\boldsymbol{u}||||\boldsymbol{v}||$ is the cosine similarity between $\boldsymbol{z}_i^n$, $\boldsymbol{z}_j^n$ and $\mathbb{1}_{[k\neq i]}$ is the indicator function. In the loss formulation, we abuse the notation a bit by not including the superscript in the notation for the projection $\boldsymbol{z}$. We assume that $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ are projections from augmented views of raw samples $\boldsymbol{I}^n$ and all other projections $\{\boldsymbol{z}_k\}_{k=1,k\neq i\neq j}^{2N}$ are obtained from augmented views of raw images $\{\boldsymbol{I}^m\}_{m=1,m\neq n}^{N}$.

4.1.4. Equivariant contrastive representation

Upon inspecting equation 4.4, we observe that SimCLR promotes invariance under all transformations. This can be explained from the following analysis. Given a sample $\boldsymbol{I}_j^n = t_j(\boldsymbol{I}^n)$ and its positive sample $\boldsymbol{I}_i^n = t_i(\boldsymbol{I}^n) = t_i(t_j^{-1}(\boldsymbol{I}_j^n)) = \tilde{t}_i(\boldsymbol{I}_j^n)$, the numerator in Eq. 4.4 is minimized if $f(\boldsymbol{I}_j^n) = \boldsymbol{z}_j^n = \boldsymbol{z}_i^n = f(\tilde{t}_i(\boldsymbol{I}_j^n))$. Hence, a model that satisfies Eq. 4.4 needs to be invariant to all transformations in T. However, hand pose estimation requires equivariance with respect to geometric transformations, as these change the displayed pose and at the same time retain invariance with respect to appearance based transformations. These two requirements can be expressed as follows:

$$t^g f(\boldsymbol{I}^n) = f(t^g(\boldsymbol{I}^n)) \tag{4.5}$$

and

$$f(\boldsymbol{I}^n) = f(t^a(\boldsymbol{I}^n)). \tag{4.6}$$

4.1.5. PeCLR loss function

Geometric equivariance can be ensured with the following modification to equation 4.5:

$$f(\boldsymbol{I}^n) = (t^g)^{-1} f(t^g(\boldsymbol{I}^n)). \tag{4.7}$$

This implies that if the projections are transformed with the inverse of the input geometric augmentation, followed by the NT-Xent loss optimization, equations 4.5 and 4.6 are satisfied. We define the transformed projection as follows:

$$\tilde{\boldsymbol{z}}_i = (t_i^g)^{-1}\boldsymbol{z}_i \tag{4.8}$$

and the modified equivariant loss as:

$$\mathcal{L}_{i,j} = -\log \frac{\exp\left(\text{sim}((\tilde{\boldsymbol{z}}_i, \tilde{\boldsymbol{z}}_j)/\tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{[k\neq i]} \exp\left(\text{sim}(\tilde{\boldsymbol{z}}_i, \tilde{\boldsymbol{z}}_k)/\tau\right)}. \tag{4.9}$$

The loss is minimized if the numerator is maximized, *i.e.* maximizing $\text{sim}(\tilde{\boldsymbol{z}}_i, \tilde{\boldsymbol{z}}_j)$. This leads to the desired geometric equivariance.

4.2. Inverting transformations in latent space

We use affine transformation to augment the input image geometrically. Therefore, the calculation of $(t^g)^{-1}$ is straightforward. Scale and rotation augmentations are performed relatively to

the magnitude. However, the translation is performed in an absolute quantity, *i.e.* if we translate an image $\boldsymbol{I}^n$ by $\boldsymbol{x}$ pixels, we need to translate its latent space representation $\boldsymbol{z}^n$ by a proportional quantity. Following this intuition, we translate z^n by a quantity proportional to its magnitude. This is achieved by scaling the absolute translation by the ratio of the range spanned by the projections with respect to the image size. The normalized translation $\hat{\boldsymbol{v}}$ can be expressed as follows:

$$\hat{\boldsymbol{v}} = \frac{\boldsymbol{v}}{L} L_z. \tag{4.10}$$

Here $\boldsymbol{v}$ is the absolute translation, $L_z = \max(z_i) - \min(z_i)$ is the range spanned by projection $\boldsymbol{z}_i$ and L is the image size. We emphasize that due to cosine similarity being used in equation 4.9, the effect of scaling is removed (*i.e.* $\text{sim}(a\boldsymbol{z}_i, b\boldsymbol{z}_j) = \text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)$, for $a, b \in \mathbb{R}$). Algorithm 1 describes the pose equivariant contrastive learning. We further describe it visually in figure 4.1

Algorithm 1: Equivariant contrastive learning

Input: batch size N, constant τ, f and T
for *sampled batch* $\{x_k\}_{k=1}^{N} \sim$ *Training Data* **do**
  `// Generate embeddings`
  for $k \in \{1, ..., N\}$ **do**
    Sample t and t' from T;
    $z_{2k-1} = f \circ t \circ x_k$;
    $z_{2k} = f \circ t' \circ x_k$;
    `// Revert geometric transforms`
    $\tilde{z}_{2k-1} = t_g^{-1} \circ z_{2k-1}$;
    $\tilde{z}_{2k} = t_g^{-1} \circ z_{2k}$;
  `// Compute similarity`
  for $i, j \in \{1, ..., 2N\}$ **do**
    $s_{i,j} = \tilde{z}_i^T \tilde{z}_j / (||\tilde{z}_i|| ||\tilde{z}_j||)$
  `// Calculate loss`
  for $i, j \in \{1, ..., 2N\}$ **do**
    $l_{i,j} = (s_{i,j}/\tau) - \log\left(\sum_{k=1}^{2N} 1_{[k \neq i]} \exp\left(s_{i,j}/\tau\right)\right)$
  $Loss = \frac{1}{2N} \sum_{k=1}^{N} [l_{2k-1,2k} + l_{2k,2k-1}]$;
  Update f to minimize $Loss$
Return trained model f

4.3. Supervised learning

In this section, we describe the input, output labels, model and loss used for supervised training on the 3D hand pose estimation task. We start by defining the input and output labels used to train the model. Next, we describe the model used and the loss function. Lastly, we describe a refinement step to improve the 3D predictions of the final model. The model is visually depicted in figure 4.2

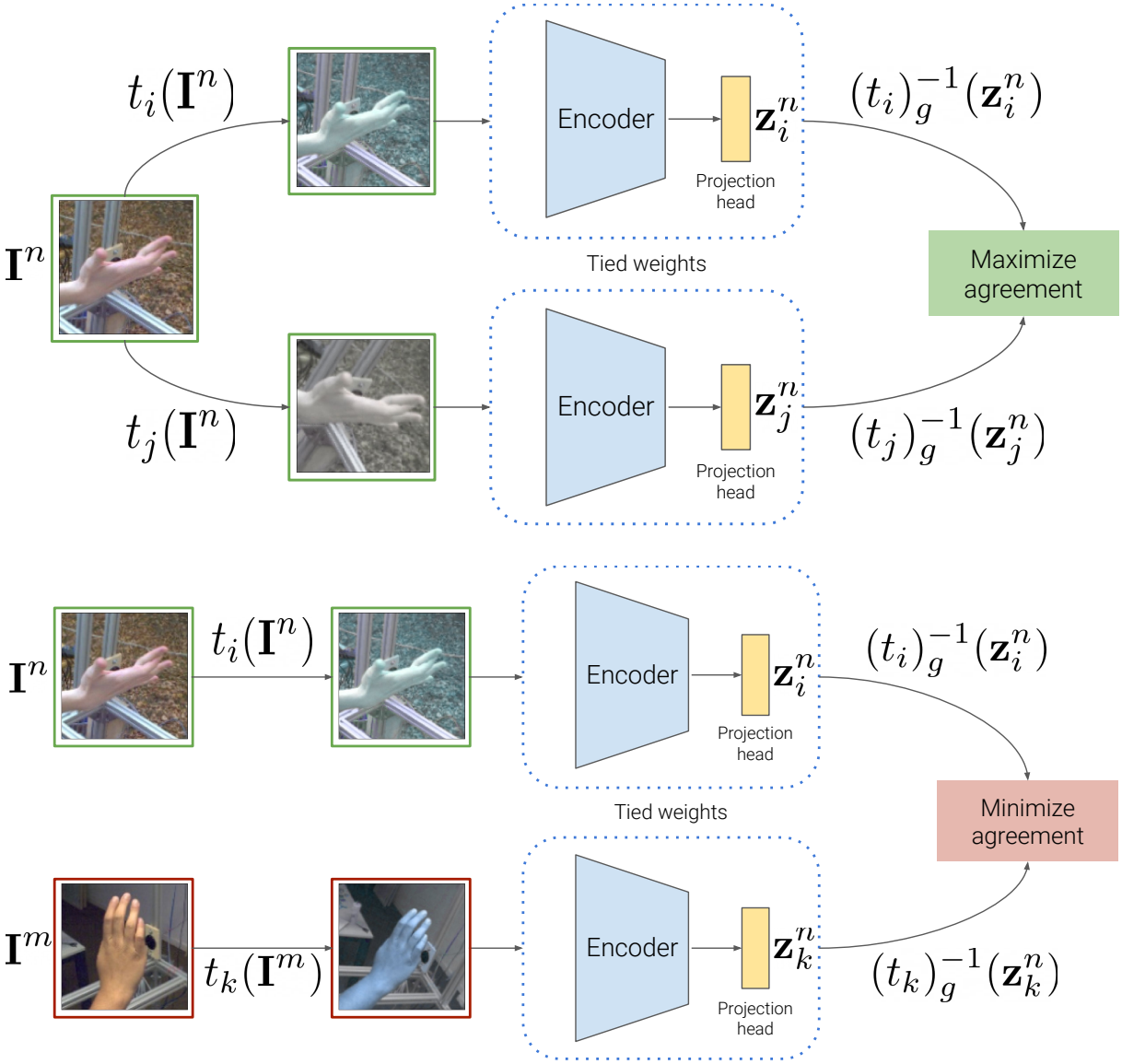


Figure 4.1.: **Method overview**. An augmentation $t = t_g \circ t_a$ is applied to input image $\boldsymbol{I}^n$. Here t_g and t_a denote the geometric and appearance components of the augmentation $t \in T$, respectively. The model then generates the projections $\boldsymbol{z}^n$ for each augmented input. Geometric augmentations are *reversed* in *projection space* before optimizing the contrastive objective. The agreement between projections from the same input image is maximized (top) and agreements amongst projections from different input images are minimized (bottom).

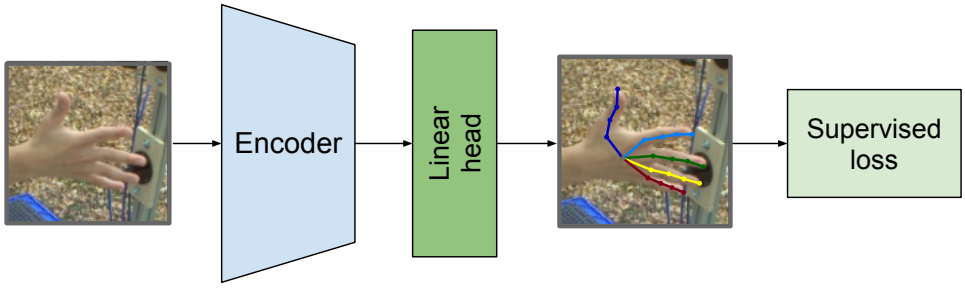


Figure 4.2.: **Supervised model**. We use an encoder and a linear layer to regress the 2.5D pose.

4.3.1. Input

Monocular RGB image $\boldsymbol{I}$ is used as the input to the model. The 2.5D representation $\boldsymbol{J}^{2.5D}$ of the 3D pose $\boldsymbol{J}^{3D}$ is used as the target. The conversion between the two representations is discussed in section 3.2.

4.3.2. Model

We use the encoder E appended with linear layer h as the supervised model. The architecture of the encoder is the same as that of the contrastive model's encoder. This is to ensure that the trained weights of the contrastive model's encoder can be used as initial weights for the supervised model. The model predicts the 2.5D pose from the RGB input image as follows:

$$\hat{\boldsymbol{J}}^{2.5D} = h \circ E(\boldsymbol{I}). \tag{4.11}$$

4.3.3. Loss

We minimize the mean absolute error of the camera projections of the pose in the image plane and the root relative depth. The two losses can be expressed as follows:

$$\mathcal{L}_{\boldsymbol{J}^{2D}} = \frac{1}{21}\sum_{i=1}^{21} |\hat{\boldsymbol{J}}_i^{2D} - \boldsymbol{J}_i^{2D}| \tag{4.12}$$

and

$$\mathcal{L}_{z^r} = \frac{1}{21}\sum_{i=1}^{21} |\hat{z}_i^r - z_i^r|. \tag{4.13}$$

Here, the subscript i denotes the keypoint index of the 21 key points in a hand pose. The overall loss is expressed as follows:

$$\mathcal{L}_{sup} = \mathcal{L}_{\boldsymbol{J}^{2D}} + \alpha\mathcal{L}_{z^r}. \tag{4.14}$$

Here α is a scalar factor to balance the two components of the $\mathcal{L}_{sup}$. We set it to 5 in our experiments. We further refine the 3D prediction by performing the depth refinement described in section 3.3.

4.4. Pre-training to fine-tuning

The contrastive learning is performed first and is called pre-training in this work. After the pre-training step, we fine-tune the encoder in the supervised manner. This is done after removing the projection layer g and replacing it with the linear layer h, similar to [6]. The entire model is trained end-to-end, using the supervised loss described in section 4.3.3. The overall setup is described in figure 4.3

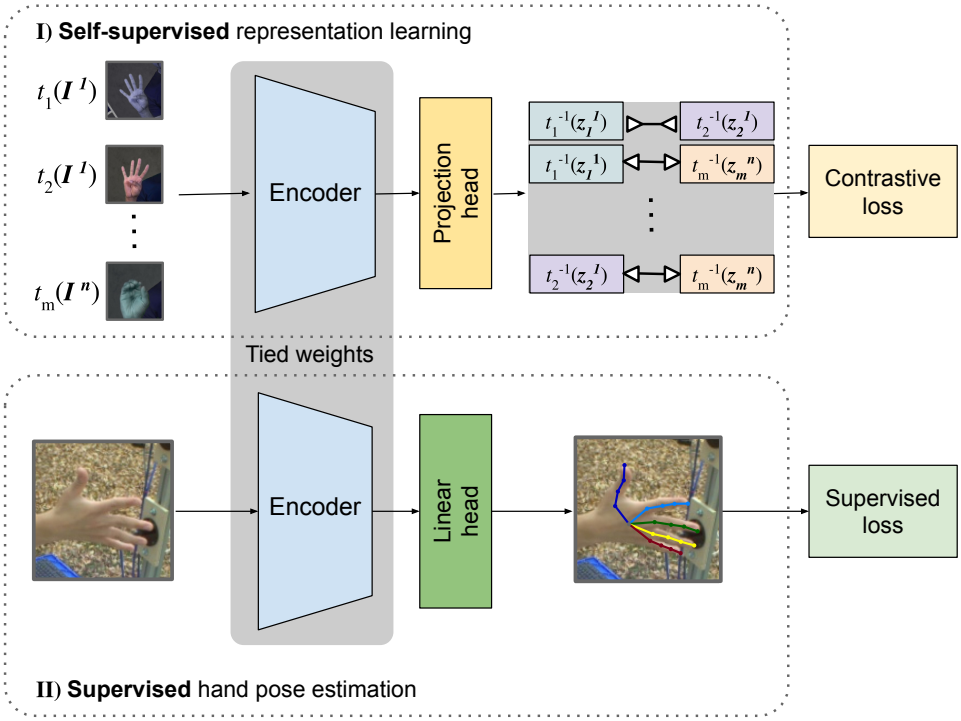


Figure 4.3.: **Overall setup**. In the first stage, we train the encoder in a self-supervised manner on a large unlabeled dataset. We follow it by the second stage, *i.e.* fine-tuning the hence trained encoder with a small amount of labeled data.

Experiments

This chapter details the experiments conducted to evaluate the effectiveness of the pose equivariant contrastive learning framework (PeCLR) for Hand pose estimation. The augmentations used during contrastive training is crucial and has an impact on the pose estimation. Therefore, we start by evaluating the quality and effectiveness of the features learned by contrastive learning on hand pose estimation in presence of one augmentation at a time. This experiment is followed by an exhaustive search for an optimal augmentation composition for the contrastive training, which leads to the optimal performance for hand pose estimation. The optimal contrastive training parameters are then compared with state of the art supervised methods on the FreiHAND dataset. Lastly, we conduct a cross data set evaluation to show the improvement offered by contrastive learning across different domain distributions.

5.1. Implementation

We compare two contrastive learning strategies, namely SimCLR[6] that uses NT-Xent, described in equation 4.4, and our proposed PeCLR that uses an equivariant contrastive loss, as described in equation 4.9 . The contrastive training is also referred as pre-training. We use ResNet [16] as the encoder. It takes RGB images of the size 128×128 as the input. A training batch spans 2048 samples. LARS [38] wrapped ADAM [19] with a learning rate of $4.5e$-3 is used as a pre-training optimizer. The pre-training consists of 100 epochs.

The supervised training with 3D hand pose labels is referred to as the fine-tuning step. During this step we use RGB images of size 128×128 and 256×256. We use ADAM with a learning rate of $5e$-4. Appendix A.1 contains the training parameters in more detail.

5.2. Evaluation of augmentation strategies

We study the performance of feature representations learned during pre-training. The pre-training is done with the different augmentations. The feature representation is then evaluated on the hand pose estimation task. The augmentations studied are classified into two categories, namely appearance based and geometric. Appearance based augmentations include color jitter, cut out, sobel filter, color drop, Gaussian blur and Gaussian noise. Whereas geometric augmentations include scale, rotate and translate. These augmentations are visually depicted in figure 5.1 . In this experiment, we train the encoder with the PeCLR and the SimCLR framework followed by supervised finetuning on the hand pose estimation task. The encoder is frozen during finetuning, instead an appended MLP is trained. This ensures that the features learned during pre-training are not changed during finetuning. We start by studying the effect of one augmentation at a time and follow up with an exhaustive search to find the best augmentation composition for SimCLR and PeCLR. We use ResNet-50 as the encoder and FreiHAND as the dataset for this study. We create our own train-val split where 90% of the data is used for the pre-training and 10% of the data is used for the evaluation. Same splits are used during the pre-training and the finetuning step to prevent information leakage between them. The pre-training spans 100 epochs and finetuning spans 50 epochs. The models are restored based on the contrastive loss and 3D loss measured on the validation split during pre-training and finetuning, respectively.

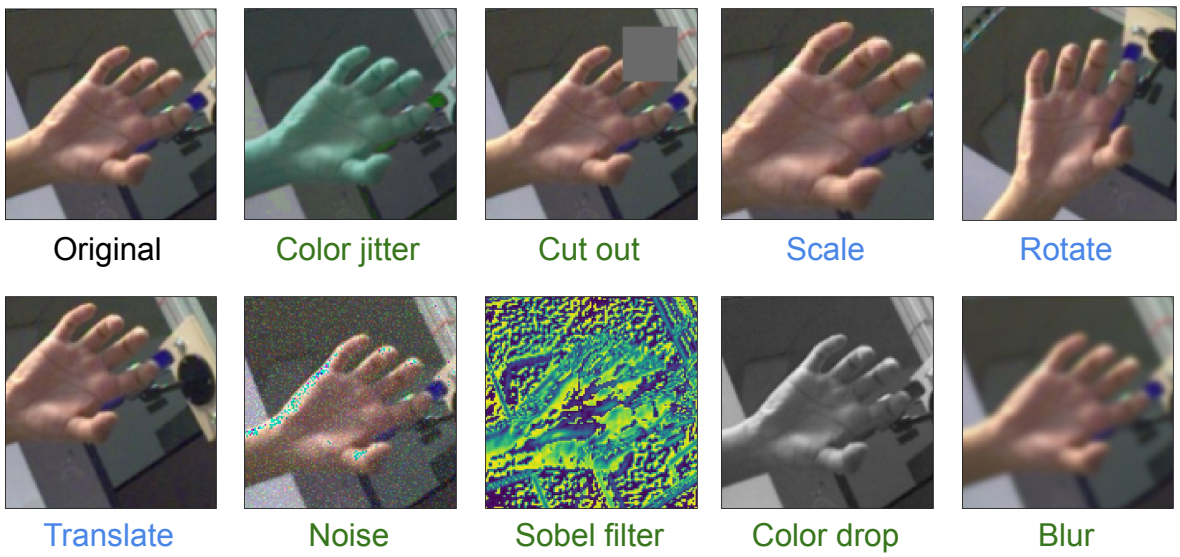
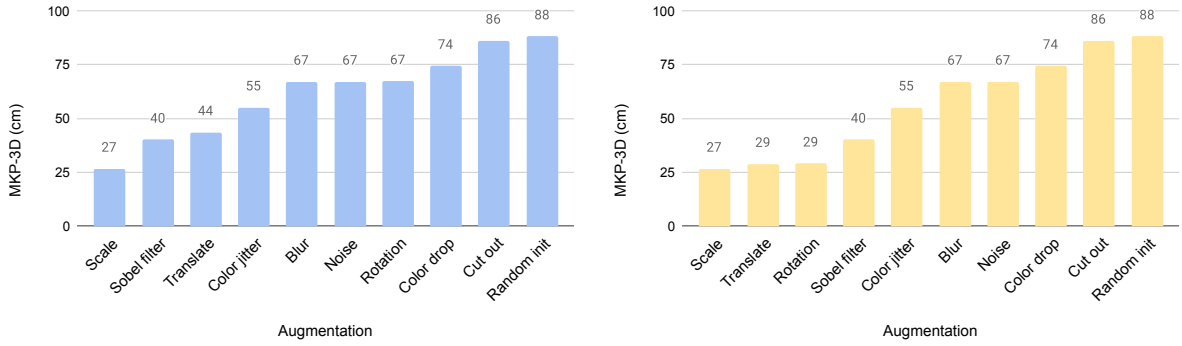


Figure 5.1.: **Augmentations.** Appearance(green) and geometric(blue) augmentations used during contrastive learning.

5.2.1. Performance of individual augmentation

The impact of each augmentation when trained with SimCLR and PeCLR is shown in figure 5.2. We observe that irrespective of the chosen augmentation, the encoders trained with contrastive objective perform better than the encoders which are randomly initialized. Additionally, we observe that for geometric transformations like rotation and translation PeCLR shows a significant

improvement of 34% and 56.7% with respect to SimCLR , respectively. We hypothesize that the poor performance stems from the fact that SimCLR promotes invariance under all augmentations where as PeCLR promotes equivariance under geometric and invariance under appearance based augmentations. Scale is also a geometric augmentation but no difference is observed between PeCLR and SimCLR. This observation is explained by the fact that the effect of scale is eliminated in cosine similarity. However, one observation is clear from figure 5.2 that scale, translate and rotations are far more important than the appearance based augmentations for the structured regression task like hand pose estimation. We emphasize here that since PeCLR as well as SimCLR promote invariance under appearance based augmentations, there is no difference in the performance of color jitter, cut out, sobel filter, color drop, Gaussian blur and Gaussian noise.



(a) SimCLR

(b) PeCLR

Figure 5.2.: **Performance of individual augmentation**. The performance of SimCLR(left) and PeCLR(right) on the hand pose estimation task. The encoders are pre-trained in the presence of one augmentation at a time, shown along the x-axis, followed by supervised finetuning of an additional MLP layer. *Translate* and emphrotate's performance improve by 34% and 56%, respectively with PeCLR.

5.2.2. Performance of composite augmentations

The quality of the features learned with contrastive pre-training improve significantly by using multiple augmentations. However, not every augmentation composition leads to an improvement for the hand pose estimation task. Therefore we perform an exhaustive study to find the best augmentation composition for SimCLR and PeCLR respectively. We narrow our search space to the top-4 performing augmentations from figure 5.2. For the exhaustive search for PeCLR, sobel filter is replaced with color jitter as the former didn't improve the performance in presence of other augmentations. We observe that for PeCLR scale, rotate, translate and color jitter perform the best, whereas for SimCLR scale and color jitter perform the best. In table 5.1, we compare the encoder pre-trained with SimCLR and the one pre-trained with PeCLR with their respective optimal augmentation composition. PeCLR improves MKP-3D by 3.4% and MKP-2D by 12.8% with respect to SimCLR. This demonstrates that the proposed equivariant contrastive loss leads to an effective representation learning approach for hand pose estimation.

Model	MKP-3D ↓ (cm)	AUC ↑	MKP-2D ↓ (px)
SimCLR	16.62	0.72	12.05
PeCLR (ours)	**16.05**	**0.74**	**10.51**

Table 5.1.: **Comparison with SimCLR**. PeCLR is compared wih SimCLR on the hand pose estimation task. The encoders are pre-trained with SimCLR or PeCLR, and are *frozen* during fine-tuning. Both methods use their optimal set of augmentations, as explained in section 5.2.2.

5.3. Inspecting equivariance of PeCLR and SimCLR

We investigate the equivariance of fine-tuned SimCLR and PeCLR models. The models are pre-trained on FreiHAND and YouTube 3D Hands combined with the optimal augmentation compositions and fine-tuned only on FreiHAND. We quantify the equivariance of a model by measuring deviations from predictions made on geometrically unaugmented inputs. Specifically, we report:

$$\mathcal{L}_{equiv}(\boldsymbol{I}^n) = ||t_i^g f(\boldsymbol{I}^n) - f(t_i^g(\boldsymbol{I}^n))||_2. \tag{5.1}$$

We do the analysis for both rotation and translation augmentations, since PeCLR reverts these augmentations in feature space. To quantify the difference in performance between PeCLR and SimCLR, we visualize the following measure of improvement:

$$\mathcal{L}_{improv}(I^n) = \frac{\mathcal{L}_{equiv}^{SimCLR}(I^n) - \mathcal{L}_{equiv}^{PeCLR}(I^n)}{\mathcal{L}_{equiv}^{SimCLR}(I^n)}. \tag{5.2}$$

This measure allows quantifying the improvement relative to the scale of the error. For a given augmentation, we sample points equidistantly on their respective parameter ranges. For rotation we sample points at the steps of 10° in the range $[-90^\circ, 90^\circ]$. For translation, we set the ranges at $[-20, 20]^2$. Each augmentation is evaluated on the entire evaluation split of YouTube 3D Hands and FreiHAND. We first visualize the results for the rotation augmentation as shown in Figure 5.3. For both datasets, we see that $\mathcal{L}_{improv}$ is positive for the entire range tested, indicating that PeCLR performs better on equivariance tasks. The amount of improvement declines as we enter more extreme ranges. The same trend can be observed for both datasets. Figure 5.4 shows the effect of translation on equivariance for both models. Similar to rotation, we observe an overall improvement of PeCLR over SimCLR across all ranges sampled.

This experiment demonstrates that the equivariance property holds, even after fine-tuning the network.

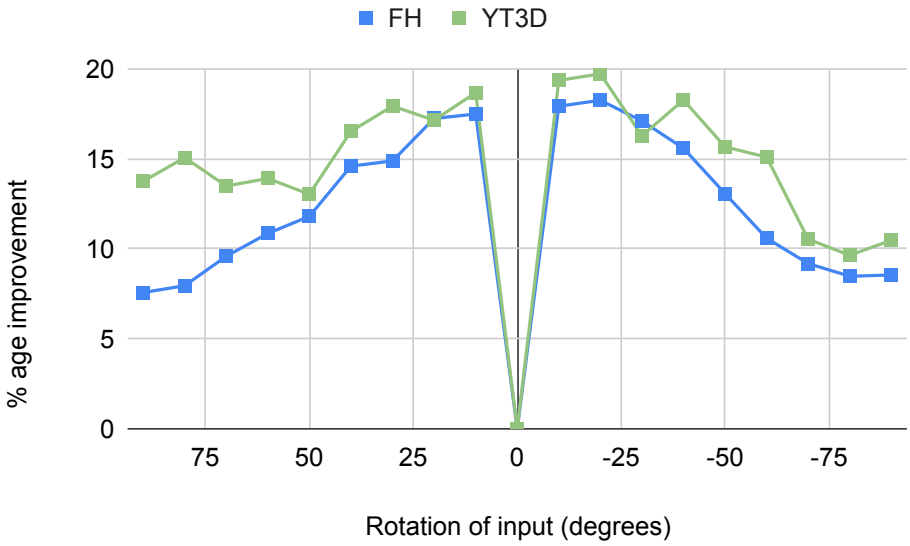
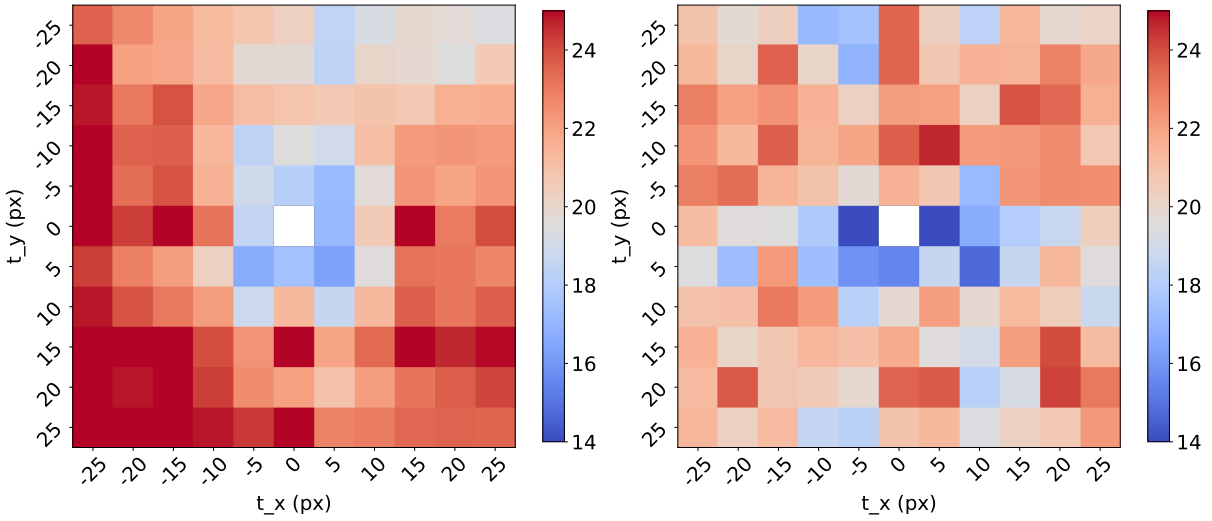


Figure 5.3.: **Quantitative analysis of rotational equivariance**. Each point denotes the improvement of PeCLR over SimCLR for rotational equivariance, as measured by MKP-2D. We see that across all sampled rotations, PeCLR leads to increased equivariance on both datasets. The models are fine-tuned on FreiHAND (FH) and pre-trained on YouTube 3D Hands (YT3D) and FreiHAND.



(a) FreiHAND (b) YouTube 3D Hands

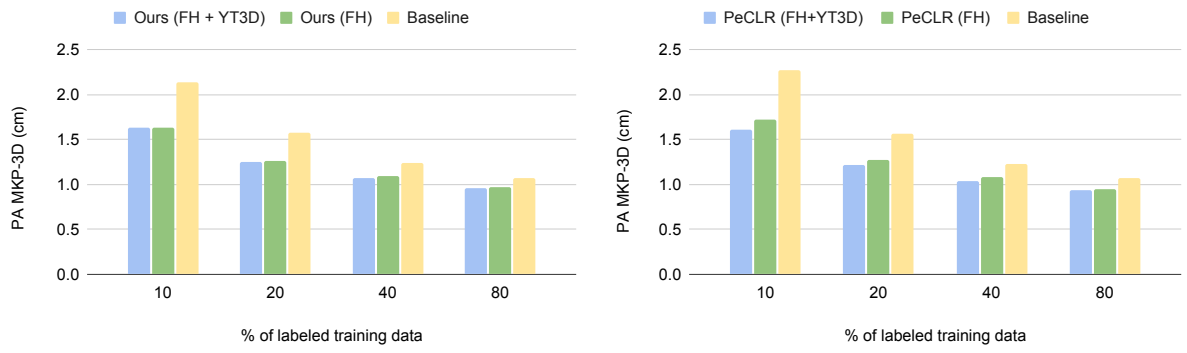
Figure 5.4.: **Quantitative analysis of translational equivariance**. Each point denotes the improvement of PeCLR over SimCLR for translational equivariance, as measured by MKP-2D. We see that across all sampled translation on the grid, PeCLR leads to increased equivariance on both the datasets. The models are fine-tuned on FreiHAND and pre-trained on YouTube 3D Hands.

5.4. Label efficiency in semi-supervised learning

We study the label efficiency of an encoder pre-trained with PeCLR in a semi-supervised setting. We use the optimal data augmentation composition described in section 5.2.2 to pre-train the encoder. The pre-trained encoder appended with a final linear layer is then fine-tuned on 10%, 20%, 40%, 80% of FreiHAND's labeled data. For this study, we consider three encoders, the first two are pre-trained on FreiHAND and FreiHAND and YouTube 3D Hands combined. The third encoder is randomly initialized. We refer to the models using these encoders as $M_{FH+YT3D}$, M_{FH} and M_b, respectively. ResNet-152 is used as the encoder because deeper neural networks benefit more from large amounts of training data [6]. We confirm the same with experiments on the smaller ResNet-50 encoder (Figure 5.5a).

In figure 5.5, we show the performance of $M_{FH+YT3D}$, M_{FH} and M_b with the ResNet-150 encoder on the hand pose estimation task. Figure 5.5 shows the same for the ResNet-50 style encoder. We observe that M_{FH}, $M_{FH+YT3D}$ outperform the baseline M_b regardless of the amount of used labels and of the encoder size. This result is in agreement with [6], confirming that pre-trained models can increase the label efficiency for the hand pose estimation task. Additionally, we observe that $M_{FH+YT3D}$ and M_b when fine-tuned with 20% of labeled data $M_{FH+YT3D}$ performs almost on par with M_b using 40% of the labeled data (PA MKP-3D of 1.21 cm vs 1.23 cm for ResNet-152 and 1.25 cm vs 1.24 cm for ResNet-50).

We further observe that for a bigger and deeper encoder like that of ResNet-152, increasing the training data during the pre-training phase further improves the performance. This trend is weaker for the smaller ResNet-50 encoder.



(a) ResNet-50

(b) ResNet-152

Figure 5.5.: **PeCLR in a semi-supervised setting**. ResNet-50(left) and ResNet-152(right) are used as the encoder. We observe that pre-training with PeCLR (green and blue), we achieve a higher training accuracy when compared to the supervised baseline (yellow).

5.5. Comparison with the state of the art

In this section, we use the PeCLR ResNet-152 encoder pre-trained on FreiHAND and YouTube 3D Hands with the optimal augmentation composition from section 5.2.2. We fine-tune the encoder with a linear layer on 100% of the FreiHAND training dataset and compare it with the state of the art methods. We also increase the image resolution during finetuning from 128×128 to 256×256. In addition to the state of the art methods, we also compare our model with a supervised baseline model trained on FreiHAND.
We report our results in table 5.2. We observe that a ResNet-152 model trained only on FreiHAND does not outperform the state of the art, despite its large model capacity. We hypothesize that this is due to the comparably small size of FreiHAND. However, by including YouTube 3D Hands during self supervised pre-training, we are able to close the gap. We observe an improvement of 7.6% in terms of PA MKP-3D with respect to our supervised baseline. This also results in our method achieving state of the art performance in a purely data driven way. We further emphasize that in contrast to our approach all other methods mentioned in table 5.2 use highly specialized architectures.

Method	PA MKP-3D ↓ (cm)	PA-AUC ↑
Spurr et al[31]	0.90	0.82
Kulon et al[22]	0.84	0.83
Li et al[23]	0.80	0.84
Pose2Mesh[8]	0.77	-
I2L-MeshNet[24]	0.74	-
RN152	0.79	0.84
+ PeCLR (**ours**)	**0.73**	**0.86**

Table 5.2.: **Comparison with SotA**. A standard RN152 model is unable to outperform state-of-the-art methods. By pre-training using PeCLR, we yield a 7.6% performance increase, resulting in state-of-the-art performance.

Fairness in comparison: The self-supervised-fine-tuned (SSFT) models often have more training time compared to the fully supervised baseline for a given data set. For instance, let us assume a SSFT model is pre-trained with a self-supervised objective with x iterations on a data set A. It is then fine-tuned for y iterations on the same data set. Now it is not fair to compare this SSFT model with a fully supervised model trained for y iterations on A. The comparison is only fair if the supervised model is trained for $x + y$ iterations on A. In our experiments protocol we ensure that the total number of iterations on a labeled data set remains the same. To put it in exact numbers we pre-train our self-supervised model for $139.6K$ iterations combined on YTB[20] and FH [42]. The samples from each dataset are sampled randomly and with replacement. This ensures that each data set has an equal number of training iterations, $69.8K$

each. The trained self supervised model is then fine-tuned on FH for $101.8K$ iterations. The supervised model used for comparison is trained for $69.8K + 101.8K$ iterations on FH. We assume that other state of the art methods are trained to optimality.

5.6. Cross-dataset analysis

In this section, we investigate the predictive power of a pre-trained encoder on the dataset that is not used for finetuning. We pre-train a ResNet-152 encoder on FreiHAND and YouTube 3D Hands. We fine-tune the trained encoder on FreiHAND. We then evaluate the performance on both FreiHAND and YouTube 3D Hands. A model trained only on FreiHAND in a supervised manner is used as a baseline. The results from this experiment setup shed light on how pre-trained models perform under a domain shift in comparison to their fully supervised counterparts. Generally speaking this is assumed to be a very challenging task for most existing methods in the hand pose community. However, it is important for real-world applications.

Table 5.3 shows that PeCLR outperforms the baseline with improvements of 4.8% in MKP-3D and 9.8% in MKP-2D when evaluated on YouTube 3D Hands. This improvement indicates that our approach is a promising way forward in using unlabeled data for representation learning and training a model that can be adapted to other data distributions.

	FreiHAND		YouTube 3D Hands	
Method	MKP-3D ↓ (cm)	AUC ↑	PA MKP-3D ↓ (cm)	MKP-2D ↓ (px)
Supervised	5.40	0.32	3.08	20.59
PeCLR (Ours)	**5.09**	**0.34**	**2.93**	**18.70**
Improvement	5.74 %	6.25 %	4.84 %	9.18 %

Table 5.3.: **Cross-dataset analysis**. PeCLR model with the ResNet-152 architecture is pre-trained on YouTube 3D Hands (YT3D) and FreiHAND (FH) and then fine-tuned on FH. The model is then evaluated on both FH (top) and YT3D (bottom) test sets. We observe that similar improvements are gained across both datasets.

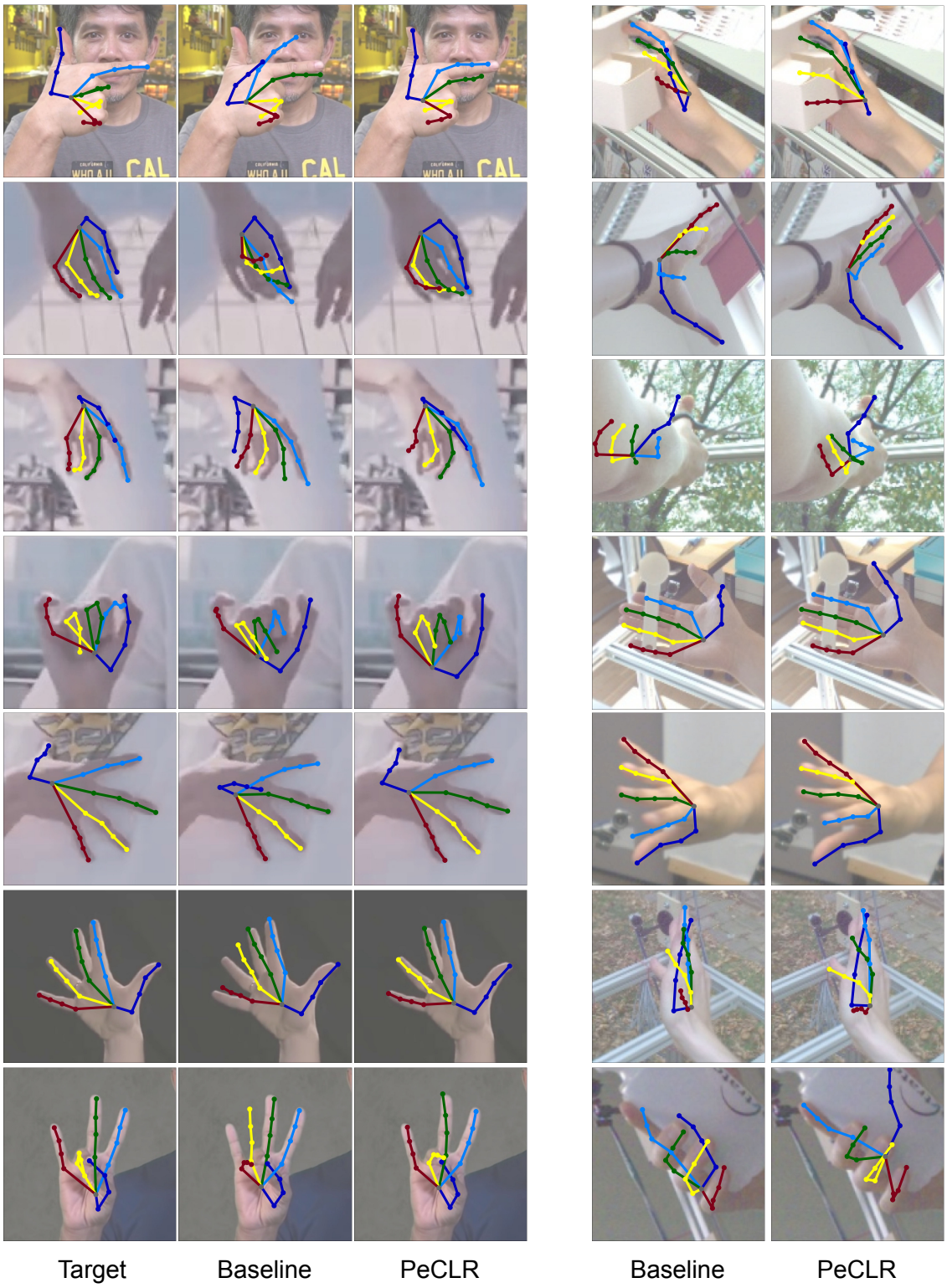


Figure 5.6.: **Qualitative keypoint predictions** are shown for YouTube 3D Hands (left) and FreiHAND (right) test sets. Results from RN152 (Baseline) and RN152 + PeCLR are shown in each column. The ground truth data is not publicly available for FreiHAND, therefore, only the predictions are shown on the right.

Conclusion and Outlook

The task of 3D hand pose estimation from monocular RGB images is a challenging task, due to large diversity in environmental conditions and hand appearances. Over the years, the research community has developed specialized architectures, artificial data synthesis and much more to address this challenge. In this work we propose a novel contrastive framework that uses unlabeled data efficiently to improve the prediction accuracy of 3D hand pose.

We approach the problem by using the state-of-the-art contrastive framework [6], which has shown promise for classification. We then identify a key issue in the approach, *i.e*. augmentation agnostic invariance. We show that this is detrimental in the presence of geometric augmentations with standard contrastive loss formulation, *i.e*. invariance to geometric augmentations learned during contrastive learning is detrimental to the hand pose estimation task. We further propose modifications to make the contrastive learning objective equivariant to geometric augmentations and invariant to appearance based augmentations. This modification leads to an improvement over the state-of-the-art contrastive framework. Later, we perform an exhaustive search for the best augmentation composition for the pose related downstream task. This is especially useful to the pose community, as the knowledge of augmentations that lead to improvement on downstream tasks is not immediate. Additionally, we show that the proposed novel contrastive learning framework with an optimal augmentation composition can help achieve the state-of-the-art performance without any special architecture. We repeat this experiment with a cross-dataset setup, *i.e*. we pre-train with a contrastive objective on data from two modalities (FreiHAND and YouTube 3D Hands) without labels, followed by finetuning on labeled data from one of the modality(FreiHAND) . We evaluate the performance on the modality used during pre-training but not during fine-tuning (YouTube 3D Hands). This setup leads to improvement in the YouTube 3D Hands predictions with respect to a baseline trained only on FreiHAND. With this, we show that our approach to contrastive learning (PeCLR) provides a feasible solution to improve generalizabilty across datasets.

6.1. Future work

Distilling from larger models. This work shows the promise of PeCLR on smaller models with a relatively small amount of unlabeled data [1]. However, contrastive learning's prowess is best utilized with larger models and a large amount of pre-training data. To put it in numbers, the largest model used in this work has 50 million parameters, compared to a model with 400 million parameters in [6]. Similarly, the data used by us is also relatively small, *i.e.* approx 0.2 million training samples from FreiHAND and YouTube 3D Hands, compared to 14 million ImageNet samples used in [6]. We believe that when used to its full potential, PeCLR can achieve an even better performance. In [7] authors use large pretrained SimCLR models to transfer knowledge to smaller models. This is indeed a promising direction for reducing the inference time. This direction is not explored in the scope of this work and is a promising direction with large application based consequences.

3D geometric augmentations. In this work, we discussed the improvement in the 3D hand pose estimation task by using 2D affine transformations as geometric augmentations during PeCLR pre-training. This hints that 3D transformations might improve the performance even more. However, applying 3D transformations on monocular RGB images is not straightforward. A possible way around this is using RGB images of a hand pose from different views by using a multi-camera setup or synthesizing a hand pose dataset from a MANO model. The exact effect and implementation of 3D geometric augmentations is a valid avenue for future work.

Catastrophic forgetting. There is a risk of forgetting features learned during pre-training while fine-tuning on a smaller dataset. One possible solution is using bigger models which take longer to unlearn the features learned during pre-training. Another solution is freezing the pre-trained weights. These two solution come with their own challenges. For instance, larger models take significantly more time and data[2] to train. On the other hand, freezing the pre-trained weights removes the possibility of learning new features altogether during fine-tuning. Simultaneous pre-training and fine-tuning could also be a promising solution to catastrophic forgetting. These solutions for catastrophic forgetting in the context of this work are yet to be explored.

Extension to heatmap based approaches. Heatmap-based approaches introduced in [36] offer state of the art performances across all pose related tasks with higher parameter efficiency. Pre-training heatmap based models with a contrastive objective is not straightforward. One of the challenges is that the use of cosine similarity in the heatmap space is not very intuitive when compared to a vector. Additionally, measuring the similarity between heatmaps is computationally intensive and requires large GPU memory. This further limits the batch size that can be used for one iteration[3] One possible solution is using spatial-softargmax to obtain keypoints from the output heatmaps and computing the similarity on these keypoints. However, this results in sacrificing the rich information encoded by the heatmaps. Future work in this direction could be very beneficial to all of the pose estimation community.

[1] This is an intentional choice because of limited computational resources.
[2] This protects against over-fitting.
[3] Large batch sizes are important if random negative sampling is used.

Appendix

A.1. Training details

Here, we describe the training details of our pre-training and finetuning steps in more detail.

Self-supervised pre-training. We train our model with 100 epochs with contrastive objective as empirically it performed the best. We use ADAM wrapped LARS as the optimizer, inspired by [6]. It indeed resulted in a better downstream performance compared to a ADAM optimizer alone. We use an effective batch size of 2048 samples consisting of 16 mini-batches of 128 samples each. This was done to train a batch on a Nvidia RTX 2080 Ti GPU. We accumulate the gradients across these 16 mini-batches before updating the model weights. The learning rate is linearly increased until $1e-4 \times \sqrt{batch}$ for 10 epochs, followed by cosine annealing for the rest of the training. When using FreiHAND and YouTube 3D Hands together to pre-train the model, we ensure that a batch contains an equal amount of samples from both datasets, by using weighted sampling. The geometric augmentation parameters are randomly sampled from a sensible range, determined empirically. The rotation $r \in [-45°, 45°]$, translation $t \in [-15, 15]^2$ and scaling $s \in [0.6, 2.0]$. The color jitter augmentation involves adjusting the hue, the saturation and the brightness of an image. Hue and saturation are changed by scaling them randomly with a scale in the range $[0.01, 1.0]$, whereas the brightness is adjusted by scaling original the value randomly in the range $[0.5, 1.0]$ and adding a random bias from the range $[5, 20]$.

Supervised fine-tuning. We train the supervised model with a learning rate of $5e-4$ in conjunction with cosine annealing. We use ADAM as the optimizer and the batch size is set to 128.The input images are augmented using rotations in the range $[-90°, 90°]$, translation in the range $[-20, 20]^2$ and scaling in the range $[0.7, 1.3]$. These ranges empirically performed the best.

Bibliography

[1] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019.

[2] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Weakly-supervised domain adaptation via GAN and mesh model for estimating 3d hand poses interacting objects. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 2020.

[3] Adnane Boukhayma, Rodrigo de Bem, and Philip H. S. Torr. 3d hand shape and pose from images in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019.

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

[5] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields, 2019.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, 2020.

[7] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E. Hinton. Big self-supervised models are strong semi-supervised learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[8] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose, 2020.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL-HLT*, Minneapolis, Minnesota, 2019.

[10] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, 2015.

[11] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David J. Crandall. Hope-net: A graph-based model for hand-object pose estimation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 2020.

[12] B. Funt and Ligeng Zhu. Does colour really matter? evaluation via object classification. 2018.

[13] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single RGB image. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019.

[14] Yana Hasson, Bugra Tekin, Federica Bogo, Ivan Laptev, Marc Pollefeys, and Cordelia Schmid. Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 2020.

[15] Yana Hasson, Gül Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016.

[17] Olivier J. Hénaff. Data-efficient image recognition with contrastive predictive coding. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, 2020.

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

[19] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. of ICLR*, 2015.

[20] Dominik Kulon, Riza Alp Güler, Iasonas Kokkinos, Michael M. Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 2020.

[21] Dominik Kulon, Riza Alp Güler, Iasonas Kokkinos, Michael M. Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 2020.

[22] Dominik Kulon, Riza Alp Güler, Iasonas Kokkinos, Michael M. Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 2020.

[23] Moran Li, Yuan Gao, and Nong Sang. Exploiting learnable joint groups for hand pose estimation. *arXiv preprint arXiv:2012.09496*, 2020.

[24] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image, 2020.

[25] Gyeongsik Moon, Takaaki Shiratori, and Kyoung Mu Lee. Deephandmesh: A weakly-supervised deep encoder-decoder framework for high-fidelity hand mesh modeling. *arXiv preprint arXiv:2008.08213*, 2020.

[26] 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. *arXiv preprint arXiv:2008.09309*, 2020.

[27] 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 *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018.

[28] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*. Springer, 2016.

[29] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[30] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 2017.

[31] Adrian Spurr, Umar Iqbal, Pavlo Molchanov, Otmar Hilliges, and Jan Kautz. Weakly supervised 3d hand pose estimation via biomechanical constraints, 2020.

[32] Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018.

[33] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+O: unified egocentric recognition of 3d hand-object poses and interactions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019.

[34] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[35] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*. Springer, 2016.

[36] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *CoRR*, abs/1908.07919, 2019.

[37] Linlin Yang and Angela Yao. Disentangling latent hands for image synthesis and pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019.

[38] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks, 2017.

[39] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*. Springer, 2016.

[40] Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular RGB image. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*,

2019.

[41] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single RGB images. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017.

[42] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan C. Russell, Max J. Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single RGB images. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, 2019.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor .

Title of work (in block letters):

Exploring self-supervised learning techniques for hand pose estimation

Authored by (in block letters):

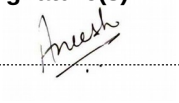
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Dahiya	Aneesh

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work .

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zurich, 6 April 2021	

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.