2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020) | 978-1-7281-3079-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/FG47880.2020.00013

Generative Model-Based Loss to the Rescue: A Method to Overcome Annotation Errors for Depth-Based Hand Pose Estimation

Jiayi Wang Franziska Mueller Florian Bernard Christian Theobalt
Max Planck Institute for Informatics, Saarbrcken, Germany

***Abstract*— We propose to use a model-based generative loss for training hand pose estimators on depth images based on a volumetric hand model. This additional loss allows training of a hand pose estimator that accurately infers the entire set of 21 hand keypoints while only using supervision for 6 easy-to-annotate keypoints (fingertips and wrist). We show that our partially-supervised method achieves results that are comparable to those of fully-supervised methods which enforce articulation consistency. Moreover, for the first time we demonstrate that such an approach can be used to train on datasets that have erroneous annotations, i.e. "ground truth" with notable measurement errors, while obtaining predictions that explain the depth images better than the given "ground truth".**

I. Introduction

Accurate hand-pose estimation from monocular depth images is vital for applications such as fine-grained control in human–computer interaction, or virtual and augmented reality [25]. However, it is a challenging task due to e.g. complex poses, self-similarities, and self-occlusions. Many existing methods address these challenges with powerful learning-based tools. Such methods dominate the benchmarks on large public datasets such as NYU [36], and Hands in the Million Challenge (HIM) [41]. Most of these approaches are trained in a fully supervised manner to predict the full set of 21 hand keypoint positions in 3D. However, the current lack of large-scale training datasets that are accurate and diverse causes such methods to overfit. This makes it difficult to generalize well to new settings, or even across benchmarks [41]. Retraining these methods on different data requires the full set of 21 (3D) keypoint annotations, which are tedious to obtain. More importantly, this process is prone to errors in the data annotations, either due to measurement errors, or due to human errors. Additionally, methods that learn a direct mapping from depth image to keypoints often ignore the inherent geometry of the hands, such as constant bone lengths or joint angle limits. As such, albeit their general good performance, these methods may produce bio-mechanically implausible poses [38]. An alternative to learning-based approaches are model-based hand tracking methods, such as [15], [27], [32], [35], among others. These methods use generative hand models to recover the pose that best explains the image through an analysis-by-synthesis strategy. While not suffering from anatomical inconsistencies, and generalizing better to yet-unseen scenarios, they require good initialization of the model parameters in order to minimize the non-convex energy function.

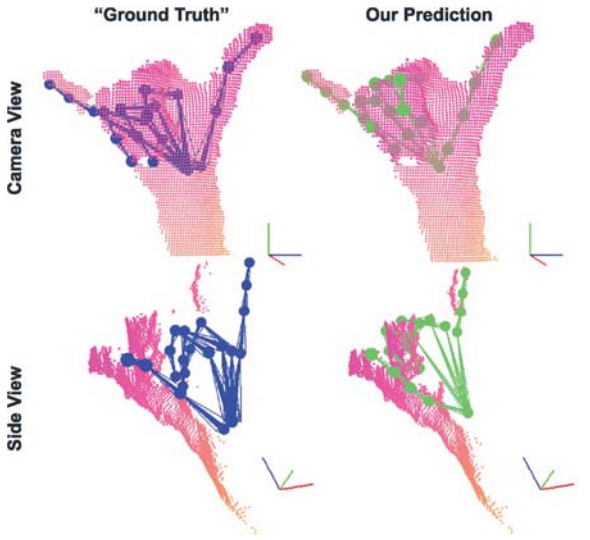


Fig. 1: Our method uses self-supervision to compensate for erroneous "ground truths" (Blue), resulting in predictions (Green) that better fit the observed depth image.

Our method addresses the shortcomings of both approaches with a generative model-based loss embedded into a learning-based method. Based on a volumetric Gaussian hand model, this loss incorporates additional annotation-free self-supervision from the depth image. When combined with anatomical priors, this supervision can take the place of the majority of joint annotations for resolving hand pose and bone length ambiguities. In total, our approach reduces the number of required annotations from 21 to 6, a 71% decrease. At the same time, the learning-based framework enables accurate and efficient inference during test time without requiring initialization. This effectively combines the main advantages of the two popular categories.

Most existing methods that utilize a model-based loss [13], [14], [38], [43] do not explain the input images in a generative manner. As such, they still require the full set of 21 annotated keypoints per frame. Additionally, due to the reliance on the annotations as the only source of supervision, these methods can overfit to errors and biases in the annotations. We demonstrate that our method can overcome such errors through the use of our additional generative loss.

We summarize our main contributions as follows:

- Compared to classical fully supervised methods, our generative loss significantly reduces the amount of

978-1-7281-3079-8/20/$31.00 ©2020 IEEE
DOI 10.1109/FG47880.2020.00013

annotations need to accurately infer the full hand pose.

- Despite ambiguities resulting from the reduced annotations, our method can simultaneously infer pose and bone lengths at test time.
- We provide a new dataset, HANDID, which includes fingertips and wrist annotations for 7 users to address the lack of hand shape variations in existing datasets.
- Most importantly, for the first time we demonstrate that such an approach can produce hand pose predictions that better fit to the depth image than the "ground truth" annotations it is trained on.

II. Related Work

Existing approaches for hand pose estimation can be broadly categorized into learning-based approaches, model-based approaches, and hybrid approaches.

Discriminative, learning-based approaches. These methods regress the pose parameters directly from image and annotation pairings. Tompson et al. [36] first used a Convolutional Neural Network (CNN) for the task of hand pose estimation. From this foundation, many methods [17], [24] develop novel architectures and training procedures to better model the nonlinear manifold of hand poses. Recent methods investigate the use of different input representations such as multi-view, voxels, and point clouds, [5], [6], [7] to take advantage of known camera intrinsics.

Generative, model-based approaches. These methods iteratively refine an estimated pose by fitting a 3D hand model to the input depth image. Previous work demonstrated that energies based on articulated, rigid, part-based models of the hand can be optimized to provide good tracking [20], [15]. Additional 3D hand representations, including continuous subdivision surfaces [31], collection of Gaussians [26], [28], sphere meshes [34], and articulated signed distance functions [32], have been proposed with the goal of creating detailed models that are still fast to optimize.

Hybrid approaches. These methods combine learning-based and model-based approaches into one framework to combine the strengths of both. One class of hybrid methods uses learning-based components in a tracking framework to initialize, update, or otherwise guide the tracker's convergence to the correct pose [18], [23], [27], [29], [30], [12]. These methods are more robust than the traditional model-based trackers, but must trade-off model and solver efficiency with accuracy during runtime. Another class of hybrid methods uses the learning-based framework and incorporates a model-based loss, usually based on a kinematic skeleton [13], [14], [38], [40], [43]. These methods can better enforce anatomically plausible pose predictions by including pose priors losses in the model space. However, since the model is not generative, they still rely on difficult-to-acquire annotations and overfit to annotation errors if present.

Our proposed hybrid method incorporates a loss that is both *generative* and *model-based*, into the learning framework. Unlike other hybrid approaches, the generative model provides supervision from the input depth image. With that, we are able to reduce the requirements on the quantity and accuracy of annotations needed for training, thereby reducing the necessary human effort for data annotation.

Model-based Autoencoder. Autoencoders are used for obtaining compressed representations from a distribution of inputs. They consist of an encoder that maps the input to a compact code, and a decoder that maps the code back to the (approximate) input. Although the encoder and decoder are usually trained jointly, the encoder can learn to invert a generative model being used as the decoder in an self-supervised manner [16]. As a learning objective, the model-based decoder can draw upon the entire training corpus as regularizer to overcome local minima that arise from noise or ambiguities present in a single image. Tewari et al. [33] use such an autoencoder with a face model to estimate and disentangle face shape, expression, reflectance, and illumination. Recently, such approaches have also been proposed for hand pose estimation in RGB images [2], [3], [8]. These methods have in common that they use geometric cues (e.g. annotated silhouettes and paired depth map) as supervision for training. Dibra et al. [4] and Wan et al. [37] use autoencoders for inverting a hand model to solve the hand pose estimation problem from depth images without additional cues. In contrast to [4], our use of a volumetric Gaussian hand model [27] as a decoder provides a stronger shape prior than their unconstrained articulating point cloud. This allows our method to solve the much harder problem of combined pose and shape estimation, while their method cannot adapt the hand shape at test time. Although conceptually our method has similarities with the (concurrently developed) work [37], our method uses a smooth hand representation compared to their spherical representation. More importantly, we extensively study the effect of a model-based generative loss when training with erroneous annotations (e.g. as present in the HIM [41] dataset), and hence we believe both works can be seen as complementary.

III. Method

The main idea of our approach is to explain a depth image of a hand based on a generative hand model, cf. Fig. 2. Given a depth image as input, we use a CNN-based encoder to obtain a low-dimensional embedding of the depth image. Our parametric model-based decoder is build upon a parametric hand model that produces a volumetric representation of the hand from a given code vector. Since the code vector from the encoder initializes a parametric model, this enforces a semantically meaningful code vector. By using a suitable representation of the input depth image, we are able to efficiently and analytically compute the overlap between the "rendered" volumetric hand representation generated by the decoder and the input depth image. To be more specific, we approximate the surface of the hand with a collection of 3D Gaussians rigidly attached to a kinematic hand skeleton model. The corresponding Gaussians in image space can be obtained by projecting the 3D Gaussians using the camera intrinsics. Moreover, the depth image is also represented with image space Gaussians by quadtree-decomposing the

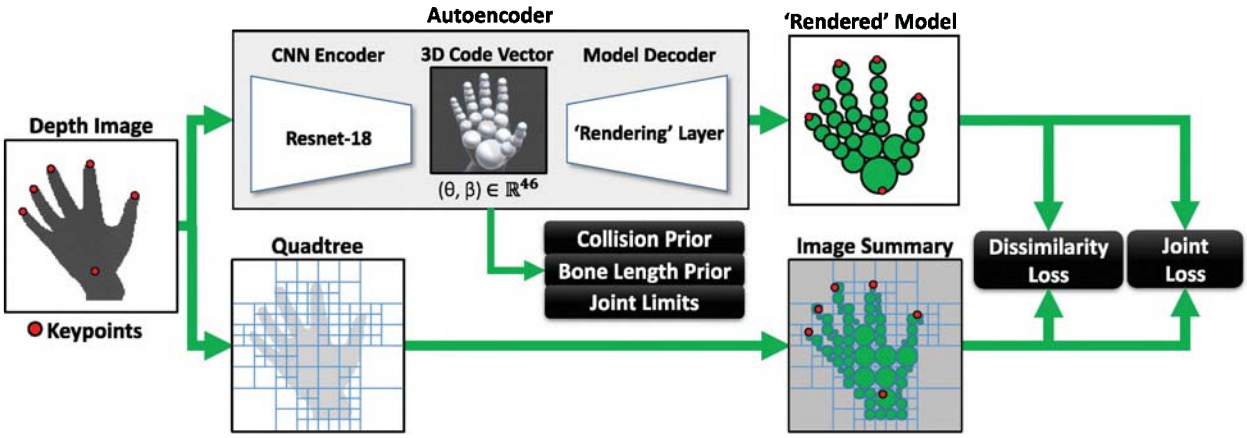


Fig. 2: **Framework Overview.** During training, an encoder is used to regress a code vector that parameterizes the bone lengths and pose in 3D. A model-based generative decoder "renders" the 3D volumetric Gaussian hand into Gaussians in the image space. The original depth image is also summarized as Gaussians in image space through a Quadtree encoding. The dissimilarity between the two sets of Gaussians provides an unsupervised generative loss for training the encoder. Additionally, bone lengths and pose prior losses are used to regularize the encoding, and a partial supervision defined on a subset of the keypoints helps to overcome bad local optima in the dissimiliarity loss.

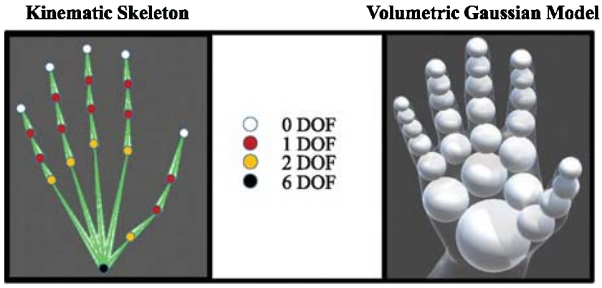


Fig. 3: **Left:** Our skeleton which comprises 20 bones and 15 articulating joints with varying degrees of freedom (DOF). In total, there are 26 joint parameters, and 20 bone length parameters. **Right**: Our volumetric Gaussian model.

image into regions of homogeneous depth and fitting an image Gaussian to each region. The similarity between the model and the image can then be described as the depth-weighted overlap of all pairs of model and image Gaussians. This overlap serves as generative model-based loss during network training and ensures that the predicted hand faithfully represents the observed data. To enforce plausible poses and bone lengths, we add additional prior losses to avoid inter-penetrations of hand parts, violations of joint limits, and unphysiological combinations of bone lengths. Lastly, supervision for a small subset of keypoints is provided as a way to mitigate the multiple minima present in the non-convex energy. At test time, the so-trained encoder is able to directly regress the hand pose and bone length parameters.

A. Hand Model

Kinematic Skeleton. Our kinematic skeleton parameterizes hand shape in terms of bone lengths, and pose as articulation angles with respect to the joint axes. It comprises 20 bones with lengths $b \in \mathbb{R}^{20}$ and 26 degrees of freedom (DOF) $\theta \in \mathbb{R}^{26}$ (20 angles of articulation and 6 additional DOF for global rotation and translation), see Fig. 3.

To ensure that the predicted bone length vector is plausible, b is parameterized by an affine model constructed using 20 PCA basis vectors, i.e.

$$b = b_{\text{avg}} + M_{\text{pca}}\beta \quad . \tag{1}$$

Here, $b_{\text{avg}} \in \mathbb{R}^{20}$ is the average bone length vector and $M_{\text{pca}} \in \mathbb{R}^{20\times 20}$ are the linear PCA basis vectors of the bone length variations scaled by their standard deviations. By scaling the basis vectors, β follows an isotropic standard normal distribution, and deviations along each basis are penalized inversely to how much natural variation exists in that direction. Both b_{avg} and M_{pca} are obtained from bone length vectors computed from 10,000 hand meshes sampled from the linear PCA parameters of the MANO model [21].

The pose parameter vector θ controls the angles of articulation with respect to the joint axes in the forward kinematics chain, as well as the global translation and rotation of the entire hand, where the latter is is parameterized using Euler angles. Given the bone length parameters β and pose θ, we can obtain the N_j joint positions by applying forward kinematics $F(\theta, \beta) \in \mathbb{R}^{N_j \times 3}$.

Volumetric Gaussian Model. Similar to [27], [28], we model the hand volume with a mixture of N_m 3D Gaussians, i.e.

$$G_{3D}(x) = \sum_{h=1}^{N_m} g_{\mu_h(\theta,\beta),\sigma_h}(x)\,, \tag{2}$$

where g is an isotropic Gaussian with mean $\mu_h(\theta, \beta)$ and standard deviation σ_h. Each Gaussian is attached to a bone on the kinematic skeleton and articulates with that bone.

B. Depth Image Representation

The depth image is represented by a collection of 2D image Gaussian and depth value pairs $\{(g_{\mu_i,\sigma_i}(x), z_i)\}_{i=1}^{N_i}$.

Each Gaussian and depth value pair summarizes a roughly homogeneous region with a single depth. To obtain these regions, we use quadtree clustering to recursively divide the image into sub-quadrants until the depth difference within each region is below a threshold c (we used $c = 20$mm for our experiments). The Gaussian $g_{\mu_i,\sigma_i}(x)$, is chosen so that μ_i is the center and σ_i is half the side length of the region. The associated depth value z_i is then the average depth value of the quadrant.

C. Model-based Decoder

To measure the quality of the predicted hand pose and bone length parameters for a given input depth image, we incorporate a decoder layer that "renders" the 3D model representation to a 2.5D representation similar to the image representation. The camera-facing surface of the h-th 3D Gaussian is approximated by a projected 2D Gaussian $g_{\mu_p,\sigma_p}(x) = \Pi_K(g_{\mu_h,\sigma_h}(x))$ using the intrinsic camera matrix K and an associated depth value z_p. For details please refer to the supplemental document.

D. Loss Layer

For training the network, the loss is decomposed into an unsupervised dissimilarity term E_{dissim} for measuring the discrepancy between depth image and hand model, $E_{\text{collision}}$ to prevent self intersection, E_{bone} for regularizing the bone length parameters β, E_{lim} for regularizing the joint angles θ, and a supervised E_{joint} term for explaining the provided joint locations. The relative importance of each term is balanced with scaling factors λ. With that, the total energy reads

$$\begin{aligned} E(\theta,\beta) =& \lambda_{\text{dissim}} E_{\text{dissim}}(\theta,\beta) + \lambda_{\text{collision}} E_{\text{collision}}(\theta,\beta) + \\ & \lambda_{\text{bone}} E_{\text{bone}}(\beta) + \lambda_{\text{lim}} E_{\text{lim}}(\theta) + \lambda_{\text{joint}} E_{\text{joint}}(\theta,\beta)\,. \end{aligned} \tag{3}$$

In the following we describe the individual energy terms.

1) Dissimilarity Measure: To measure the overall similarity between two given (2D Gaussian, depth) tuples, we weight the similarity $S_{i,p}$ between the two Gaussians by their distance in depth values $\Delta(i,p)$. The pairwise similarity between image Gaussian g_{μ_i,σ_i} and projected model Gaussian g_{μ_p,σ_p} is defined using the integral over the product of the two functions. Since in our case the model Gaussian directly depends on the hand pose vector θ and bone length vector β, $S_{i,p}$ is a function of these parameters and is given by

$$S_{i,p}(\theta,\beta) = \int_{\mathbb{R}^2} g_{\mu_i,\sigma_i}(x) g_{\mu_p(\theta,\beta),\sigma_p}(x)\, dx\,. \tag{4}$$

Since $S_{i,p}(\theta,\beta)$ only measures the 2D overlap of the two Gaussians, we weight it based on the depth difference

$$\Delta(i,p) = \begin{cases} 0, & \text{if } |z_i - z_p| \geq 2\sigma_h \\ 1 - \frac{|z_i - z_p|}{2\sigma_h}, & \text{if } |z_i - z_p| < 2\sigma_h \end{cases}, \tag{5}$$

where σ_h is the standard deviation of the unprojected Gaussian g_{μ_h,σ_h} associated with g_{μ_p,σ_p}. This decreases the similarity score between two tuples whenever the depth values are far apart, and thereby forces the model to not only match the area of the hand in the depth image, but also the observed depth values.

The overall similarity S_{sim} is defined as the sum over all possible pairings between the model and the image Gaussians, and is given by

$$S_{\text{sim}} = \frac{\sum_{i=1}^{N_i}\sum_{p=1}^{N_m} \Delta(i,p) S_{i,p}}{\sum_{i=1}^{N_i}\sum_{k=1}^{N_i} S_{i,k}}\,, \tag{6}$$

where the denominator is the self-similarity of the image Gaussians used for normalization. We use $E_{\text{dissim}} = -S_{\text{sim}}$ since minimizing the loss maximizes the similarity.

2) Collision Prior: To ensure that the surface represented by the 1σ isosurface of the 3D Gaussians does not (self-)interpenetrate, a repulsive term based on the 3D overlap of the model Gaussians is used. Overloading the notation for the Gaussian overlap $S_{i,j}$ (cf. Eq. (4)) to denote the similarity between two different model Gaussian components, we analogously define

$$E_{\text{collision}} = \sum_{j=1}^{N_m}\sum_{k=j+1}^{N_m} S_{j,k}\,, \tag{7}$$

so that Gaussians of the model do not overlap in 3D.

3) Bone Length Prior: To keep the bone lengths β plausible, we impose the loss

$$E_{\text{bone}} = ||\beta||_2^2\,, \tag{8}$$

which penalizes the deviation of the predicted bone length parameters from the mean parameter. With that, this term helps to keep the predictions in the high probability region of the normal distribution used in the PCA prior.

4) Joint Limits: To keep joint articulations within mechanically and anatomically plausible limits, a joint limit penalty is imposed using

$$E_{\text{lim}} = \sum_{\theta_j \in \theta} \begin{cases} 0, & \text{if } \theta_j^l \leq \theta_j \leq \theta_j^h \\ (\theta_j^l - \theta_j)^2, & \text{if } \theta_j < \theta_j^l \\ (\theta_j - \theta_j^h)^2, & \text{if } \theta_j > \theta_j^h \end{cases}, \tag{9}$$

where θ_j^l and θ_j^h are the lower and upper limits of θ_j, which are defined based on anatomical studies of the hand [22].

5) Joint Location Supervision: We impose an additional supervision loss E_{joint} on a small subset of joint positions $J_1, \ldots, J_{N_s}$ in order to help the optimizer converge to a good minimum in the overall generative loss function. We use a combination of 2D and 3D joint location supervisions (depending on availability). If for a given joint with index j a full 3D supervision is provided, the distance Φ_j between the annotation $J_j \in \mathbb{R}^3$ and the model joint F_j is given by their ℓ_2 distance. If only 2D supervision is provided, Φ_j is the closest ℓ_2 distance between F_j and the ray $\overline{J}_j$ to which the annotation is projected using the camera intrinsics. Hence, Φ_j is defined as

$$\Phi_j = \begin{cases} ||F_j - \langle F_j, \overline{J}_j\rangle \overline{J}_j||_2\,, & \text{if } J_j \in \mathbb{R}^2 \\ ||F_j - J_j||_2\,, & \text{if } J_j \in \mathbb{R}^3 \end{cases}, \tag{10}$$

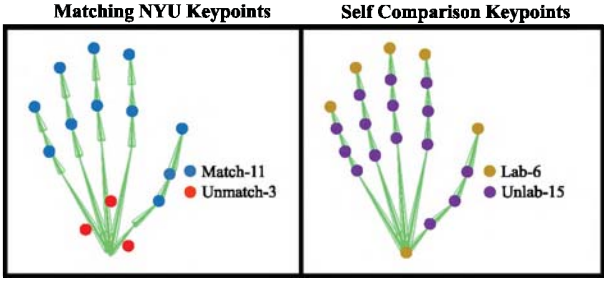


Fig. 4: **Left:** For comparisons against the state of the art, our model is evaluated on a subset of NYU keypoints (**Match-11**) due to mismatches to our skeleton. **Right:** For self-comparison, we evaluate on 21 keypoints (**All-21**), 6 of which have supervision (**Lab-6**), and 15 keypoints without supervision (**Unlab-15**).

where $F_j = F(\theta, \beta)_j$ is the j-th joint obtained from applying forward kinematics with the model parameters.

Due to inaccuracies in the annotation, the ground truth may conflict with the observed image. Hence, we modify the joint loss to account for annotation uncertainty by introducing a "slack" radius $s \in \mathbb{R}_+$ that models the expected uncertainty in millimeters. All predictions within this radius of the ground truth will not be penalized. This allows the encoder to be more robust to erroneous annotations. Together, the joint loss for the subset of N_s joints E_{joint} is defined as

$$E_{\text{joint}} = \sum_{j=1}^{N_s} \begin{cases} 0, & \text{if } \Phi_j \leq s \\ (\Phi_j - s)^2, & \text{if } \Phi_j > s \end{cases} . \qquad (11)$$

IV. Experiments

We evaluate the impact of our generative model-based loss on pose accuracy and bone length consistency when trained with a reduced set of keypoints. Additionally, we show qualitative results of our predictions and the erroneous "ground truth" on existing datasets to demonstrate the regularizing effect of our loss against annotation errors.

A. Architecture and Training

We use Resnet-18 [9] pre-trained on ImageNet as our encoder, as it is fast to use and refine, and achieves good accuracy. The encoder is trained with the Adam optimizer [11], using a learning rate of 10^{-5} and a batch size of 16. Our pipeline runs in Caffe [10], where we implemented the decoder and other losses as custom layers. During training, a forward-backward pass with batch size 16 takes 89ms (for comparison: ResNet-50 architecture takes 100ms). A forward pass at test time takes only 5ms.

B. Datasets

We evaluate on two common benchmarks, the NYU Hand Pose dataset [36] and the Hands in the Million Challenge dataset (HIM) [42]. We additionally introduce our own HANDID dataset for training to address the lack of hand shape variation in the NYU training data.

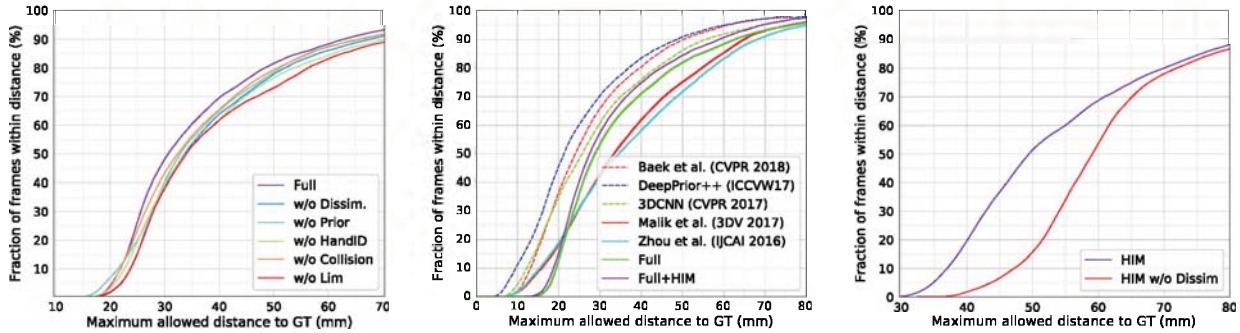
NYU Hand Pose Dataset. The NYU Hand Pose dataset [36] is collected using Microsoft Kinect sensors. It contains 72,757 depth images from a single subject in the training set, and 8,252 depth images from two subjects in the test set.

Our HANDID Dataset. Since the NYU training data only contains a single subject, we introduce additional training data with more hand shape variations to enable our method to learn this variation and hence adapt to different users at test time. We captured a dataset of 3,601 frames (640 x 480) from 7 subjects with the Intel SR300 sensor, which we call HANDID. A total of 6 pixels that correspond to the fingertips and wrist are annotated per frame. Occluded keypoints were indicated as such. During training, a batch contains examples from both HANDID and the NYU dataset with a mixing ratio of $1:3$.

To emphasize that it is significantly easier to obtain just the fingertips and wrist keypoints, we asked 5 users to annotate all 21 keypoints for a set of 10 depth images. We observed that additional keypoints take longer to annotate (each joint annotation takes 1.4 times longer) and are less consistent across users (with average distance to mean of 10.4 pixels vs 7.3 pixels). In total, the full annotation of 21 joints for 10 images requires 21.2 minutes, while our subset only needs 4.7 minutes.

Hands in the Million Challenge (HIM) Dataset. We evaluated our method on the Hands in the Million Challenge (HIM) dataset [42], where we discovered a systematic error in the "ground truth" annotations. Although the 2D projection of the keypoints into the image plane looks plausible, the 3D keypoint locations do not match the anatomical locations of hand joints (see Fig. 6). To quantitatively show this, we use the minimum-distance-to-point-cloud (MDPC) per joint to approximately quantify how well the joint predictions agree with the observed depth image. The NYU annotations and the erroneous HIM annotations have median MDPCs of 9.10mm (avg 10.99mm) and 21.54mm (avg 23.98mm), respectively. By assuming that the physical joint is located roughly at the center of the finger, the HIM annotations would imply an implausible finger thickness of ≈43mm, while the NYU annotations estimates a more reasonable thickness of ≈18mm. We hypothesize that there is a systematic pose-dependent error in corresponding the 3D magnetic sensor positions to the depth camera coordinate (see Fig. 4 of the Supplementary Document). Using our generative model-based loss, we are able to obtain predictions that are significantly more consistent with the observed depth images. The detailed experiment is presented in Section IV-D.

Pre-processing. Similar to established procedures [1], [18], we first localize the hand by using the ground truth joint locations and crop the image to a fixed-size cube with 300mm side length. Once localized, the image is re-cropped using the same cube, but centered at the average depth. We then scale it to 128 x 128 with a scaled depth range between $[-1, 1]$. During training, in-image-plane translations and rotations, as well as depth augmentations, are applied. This pre-processing step is used for all datasets.



(a) **Ablation Study:** All components of our method need to work together to resolve ambiguities from the reduced keypoint supervision (all keypoints (**All-21**) evaluated).

(b) **Comparison to state of the art:** Our method (**Full**) outperforms competing hybrid methods, even with less supervision. This is further improved by incorporating the HIM dataset, which is not possible without the dissimilarity loss.

(c) **Cross Benchmark Test:** We evaluate our method on the NYU dataset after training *only on the HIM dataset*. Without the dissimilarity loss, the mismatch in annotation results in worse generalization.

Fig. 5: Quantitative evaluation on the NYU dataset (in percentage of frames with maximum joint error below a threshold).

Method	Unlab-15	Lab-6	All-21
Full	**16.13**	**20.72**	**17.45**
w/o Dissim.	19.06	21.47	19.75
w/o Prior	18.53	22.03	19.53
w/o HANDID	17.01	23.20	18.78
w/o Collision	16.80	22.20	18.34
w/o Lim.	18.72	22.24	19.73

(a) **Ablation study** with keypoints (see Fig. 4) of the NYU dataset [36]. Dissimilarity loss, and the pose and shape priors help resolve ambiguities for unlabeled keypoints. The HANDID dataset helps on labeled keypoints by allowing adaptions to unseen users.

Method	Match-11
Full	18.50
Full+HIM w/o Dissim.	20.01
Full+HIM	**17.73**
Zhou et al. [43]	19.21
Malik et al. [13]	18.35
Baek et al. [1]	14.71
DeepPrior++ [19]	13.10
3DCNN [6]	15.09

(b) **Comparison to state of the art methods:** kinematic model-based (top, middle) enforces kinematic consistency and direct joint position regression (bottom) do not.

Method	S1	S2
Ground Truth	1.00	1.00
Full+HIM	**0.70**	**0.80**
Full	0.63	0.70
w/o Dissim.	0.57	0.59
w/o Prior	0.52	0.42
w/o HANDID.	0.55	0.54
w/o Collision	0.62	0.68
w/o Lim.	0.6	0.42

(c) F1 score of k-means clustering of bone lengths vectors for the two subjects in the test set.

TABLE I: Evaluations on NYU. (a-b) Comparisons of 3D mean per-joint error (in mm). (c) Evaluation of bone lengths learning.

Model Mismatch. Due to different joint locations in the NYU hand model and ours, only 11 of the commonly evaluated keypoints have a rough equivalence to our model (Fig. 4, left). Hence, we compare our predictions with the state-of-the-art predictions on this subset (**Match-11**). To better demonstrate that our method can infer the positions of unsupervised keypoints, we evaluate our algorithm for self comparison on an expanded set of 21 NYU keypoints (**All-21**) which roughly correspond to anatomical joints of our kinematic skeleton (Fig. 4, right). The results are further broken down for the 6 supervised keypoints (**Lab-6**) and the 15 unsupervised keypoints (**Unlab-15**).

C. Ablation Studies

For the ablation study, we perform quantitative evaluations on the NYU dataset.

Keypoint Accuracy. Removing components from our full method (**Full**) reduces accuracy. See Table Ia for the average per-joint error in millimeters, and Fig. 5a for the percentage of correct frames curve.

Bone Lengths. For bone length evaluation, we cannot directly compare the ground-truth bone lengths to our predicted bone lengths due to the mismatch in model definitions (cf. Fig. 4, left). Instead, we treat the 20 bone lengths of the hand as a 20-dimensional vector and use k-means clustering with $k = 2$ to separate the bone length vectors of the two subjects in the test set of the NYU dataset. In Table Ic, we show the F1 scores (defined as $\frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$) of the two clusters. k-means is meaningful for this task as clustering bone lengths of the annotations (**Ground Truth**) results in perfect F1 scores for both subjects. Note that poses with high self-occlusion result in depth images with very little information to help disambiguate hand shapes. Thus, one cannot expect methods that perform per-frame estimation to attain a perfect F1 score from the given supervision.

Discussion. Given the reduced supervision, it is ambiguous whether the loss is minimized by deforming the bone lengths or updating the hand pose. Consequently, the method without bone length prior can arbitrarily distort the bone lengths as long as the fingertips are correctly estimated (**w/o Prior**, see Table Ia). This results in a significant drop in accuracy for keypoints without direct supervision (**Unlab-15**). Correspondingly, k-means clustering fails to find consistent clusters for the two subjects.

However, the bone length prior alone is not enough to

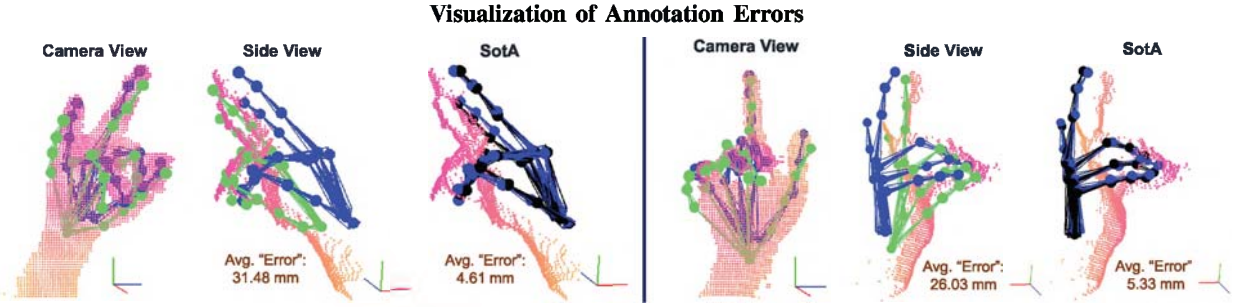


Fig. 6: **Annotation Errors in HIM:** Both the "ground truth" (Blue) and our predictions (Green) are consistent with the input in the camera view. However, as can be seen from the side view, the "ground truth" is erroneous and our prediction is more consistent. State-of-the-art (SotA) method [39] (black) learns to replicate the systematic error. This result is representative of the test set.

resolve the ambiguity in hand shape. A similar drop in accuracy on unsupervised keypoints (**Unlab-15**) occurs when the dissimilarity loss is removed (**w/o Dissim.**, see Table Ia). This is because statistically plausible bone lengths can still vary wildly to accommodate the fingertip annotations, without being constrained to explain the image. Pose priors in the form of joint limits (**w/o Lim.**) and collision prior (**w/o Collision**) additionally constrain the articulations, which improve the keypoint accuracy.

Due to the NYU training data containing only one hand shape, it is sufficient for the method to consistently regress this particular set of bone lengths when HANDID is not present (**w/o HANDID**, see Table Ia). As a result, the method cannot learn to discriminate between hand shapes of different users, leading to F1 scores that are close to random. Hence, for the unseen hand shape in the test set, the method cannot minimize the joint loss (see Eq. (10)) of the supervised keypoints, which leads to greatly reduced accuracy on supervised keypoints (**Lab-6**). This mode of failure can be accounted for if hand shape variations are present in the training data. The result of this can be seen in our full method (**Full**, see Table Ia).

D. Comparison to the State of the Art (SotA)

Although state-of-the-art methods obtain mean per-joint errors lower than 10mm (e.g. [6], [39]) on the HIM dataset, we emphasize that this is against the erroneous "ground truth". We train our method using a "slack" radius of 25 mm to account for the error and show better fitting pose predictions than even the "ground truth" (see Fig. 6 and Fig. 4 of Supplemental Material for more qualitative evaluation).

For a more fair quantitative evaluation, we instead use minimum-distance-to-point-cloud (MDPC) to approximate how well the predictions fit the input. On the HIM test set of [39] comprising of 95,540 images, our method achieves median MDPCs of 11.74mm (avg 13.87mm), while [39] achieves 21.97mm (avg 24.16mm). Our predictions better match the NYU annotations with median MDPCs of 9.10 mm (avg 10.99 mm). This suggests that our method better fits the observed input while most state-of-the-art methods learn to replicate the errors in the training data.

We further show that the dissimilarity loss helps to overcome annotation errors by testing the method trained on HIM data on the NYU data (See Fig. 5c). Without the dissimilarity loss, the method performs significantly worse.

On the NYU dataset (see Table Ib and Fig. 5b), our method outperforms the other kinematic model-based methods of Zhou et al. [43] and Malik et al. [13] while requiring less keypoint annotations. Although methods that directly predict 3D joint positions perform better [1], [6], [19], we emphasize that these methods without a model-based generative loss are liable to learning the annotation errors as shown.

We compare our method to Dibra et al. [4] and Wan et al. [37]. Although we were unable to obtain their predictions on the subset of **Match-11** keypoints, we note that Dibra et al. [4] have a similar "uncorrected" percentage of correct frames curve on all 14 keypoints to Zhou et al. [43], which we greatly outperform, and we achieve similar performance to Wan et al. [37]'s method with single view training.

While their methods do not require any annotation, our method additionally solves the more ambiguous and harder problem of adapting to the hand shapes of the user during test time, while their methods can only fit to the average hand shape of the training data or to preset bone lengths.

E. Adaptation to a New Domain

Despite the aforementioned annotation errors, the HIM dataset contains a large variety of views, poses, and hand shapes that could be used to supplement the NYU training data to help improve generalization. We show that our method can still benefit from data with erroneous annotations (see Table Ib and Fig. 5b). We trained our method by mixing the NYU, HIM, and HANDID datasets in a single batch with a ratio of 3:3:2. When HIM data is used without the dissimilarity loss (**Full + HIM w/o Dissim.**), the annotation errors cause the overall performance to degrade. With our dissimilarity loss enabled (**Full + HIM**), the self-supervision ignores the annotation errors and improves the results.

V. LIMITATIONS & DISCUSSION

Although our method outperforms other kinematic model-based methods, even with less annotations, there is still a

gap to recent learning-based methods that regress 3D joint positions. However, these methods

- are not explicitly penalized for producing anatomically implausible shapes due to the lack of an underlying kinematic hand model, and
- are prone to overfit to errors in the training annotations, as well as to errors in the annotation collection method.

Additionally, for poses with heavy self-occlusions, the monocular depth data is not sufficient to resolve ambiguities with the reduced annotation set used by our method. Extra supervision, such as from temporal consistency, or from multi-view constraints (as done in [37]), is needed to estimate the pose and shape in these cases.

VI. Conclusion

We have shown that a generative model-based loss can reduce the amount of supervision needed to learn both the pose and shape of hands. This greatly reduces the amount of annotations needed to adapt a method to data obtained in a new domain. Furthermore, we show that the generative model-based loss helps to regularize against annotation errors, for example on the HIM dataset, while existing methods overfit to these errors. This demonstrates the importance of ensuring that the model predictions explain not only the annotations but also the image itself.

References

[1] S. Baek et al. Augmented skeleton space transfer for depth-based hand pose estimation. In *CVPR*, 2018.

[2] S. Baek et al. Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In *CVPR*, 2019.

[3] A. Boukhayma et al. 3d hand shape and pose from images in the wild. In *CVPR*, 2019.

[4] E. Dibra et al. How to refine 3d hand pose estimation from unlabelled depth data? In *3DV*, pages 135–144, 2017.

[5] L. Ge et al. Robust 3d hand pose estimation in single depth images: From single-view cnn to multi-view cnns. *CVPR*, pages 3593–3601, 2016.

[6] L. Ge et al. 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images. In *CVPR*, pages 5679–5688, 2017.

[7] L. Ge et al. Point-to-point regression pointnet for 3d hand pose estimation. In *ECCV*, 2018.

[8] L. Ge et al. 3d hand shape and pose estimation from a single rgb image. In *CVPR*, 2019.

[9] K. He et al. Deep residual learning for image recognition. In *CVPR*, 2016.

[10] Y. Jia et al. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[11] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[12] M. Madadi et al. Top-down model fitting for hand pose recovery in sequences of depth images. *Image and Vision Computing*, 79, 2018.

[13] J. Malik et al. Simultaneous hand pose and skeleton bone-lengths estimation from a single depth image. In *3DV*, pages 557–565, 10 2017.

[14] J. Malik et al. Structure-aware 3d hand pose regression from a single depth image. In *EuroVR*, pages 3–17, 2018.

[15] S. Melax et al. Dynamics based 3d skeletal hand tracking. In *GI*, GI '13, pages 63–70. Canadian Information Processing Society, 2013.

[16] V. Nair et al. Analysis-by-synthesis by learning to invert generative black boxes. In *ICANN*, 2008.

[17] M. Oberweger et al. Hands deep in deep learning for hand pose estimation. In *CVWW*, pages 1–10, 2015.

[18] M. Oberweger et al. Training a feedback loop for hand pose estimation. In *ICCV*, pages 3316–3324, Washington, DC, USA, 2015. IEEE Computer Society.

[19] M. Oberweger et al. Deepprior++: Improving fast and accurate 3d hand pose estimation. In *ICCVW*, pages 585–594, 2017.

[20] I. Oikonomidis et al. Efficient model-based 3d tracking of hand articulations using kinect. In *BMVC*, pages 101.1–101.11. BMVA Press, 2011.

[21] J. Romero et al. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):245:1–245:17, Nov. 2017.

[22] E. S. Serra. Kinematic model of the hand using computer vision. *PhD thesis*, 2011.

[23] T. Sharp et al. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 3633–3642, New York, NY, USA, 2015. ACM.

[24] A. Sinha et al. Deephand: Robust hand pose estimation by completing a matrix imputed with deep features. In *CVPR*, pages 4150–4158, 2016.

[25] M. Soliman et al. Fingerinput: Capturing expressive single-hand thumb-to-finger microgestures. In *ISS*, pages 177–187. ACM, 2018.

[26] S. Sridhar et al. Real-time hand tracking using a sum of anisotropic gaussians model. In *3DV*, 2014.

[27] S. Sridhar et al. Fast and robust hand tracking using detection-guided optimization. In *CVPR*, 2015.

[28] C. Stoll et al. Fast articulated motion tracking using a sums of gaussians body model. In *ICCV*, pages 951–958, 2011.

[29] X. Sun et al. Cascaded hand pose regression. In *CVPR*, pages 824–832, 2015.

[30] D. Tang et al. Opening the black box: Hierarchical sampling optimization for estimating human hand pose. In *ICCV*, pages 3325–3333, 2015.

[31] J. Taylor et al. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. In *TOG*, volume 35. ACM, July 2016.

[32] J. Taylor et al. Articulated distance fields for ultra-fast tracking of hands interacting. *ACM Trans. Graph.*, 36(6):244:1–244:12, Nov. 2017.

[33] A. Tewari et al. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *ICCV*, 2017.

[34] A. Tkach et al. Sphere-meshes for real-time hand modeling and tracking. *ACM ToG*, 35:1–11, 11 2016.

[35] A. Tkach et al. Online generative model personalization for hand tracking. *ACM Trans. Graph.*, 36(6):243:1–243:11, Nov. 2017.

[36] J. Tompson et al. Real-time continuous pose recovery of human hands using convolutional networks. In *ToG*, volume 33, pages 169:1–169:10, New York, NY, USA, Sept. 2014. ACM.

[37] C. Wan et al. Self-supervised 3d hand pose estimation through training by fitting. In *CVPR*, 2019.

[38] J. Wohlke et al. Model-based hand pose estimation for generalized hand shape with appearance normalization. In *arXiv*, 2018.

[39] X. Wu et al. Handmap: Robust hand pose estimation via intermediate dense guidance map supervision. In *ECCV*, 2018.

[40] Q. Ye et al. Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation. In *ECCV*, 2016.

[41] S. Yuan et al. Bighand2.2m benchmark: Hand pose dataset and state of the art analysis. In *CVPR*, 2017.

[42] S. Yuan et al. Depth-based 3d hand pose estimation: From current achievements to future goals. In *CVPR*, 2018.

[43] X. Zhou et al. Model-based deep hand pose estimation. In *IJCAI*, IJCAI'16, pages 2421–2427. AAAI Press, 2016.