3D Hand Pose Estimation using Simulation and Partial-Supervision with a Shared Latent Space

Masoud Abdi[1]
mabdi@deakin.edu.au

Ehsan Abbasnejad[2]
ehsan.abbasnejad@adelaide.edu.au

Chee Peng Lim[1]
chee.lim@deakin.edu.au

Saeid Nahavandi[1]
saeid.nahavandi@deakin.edu.au

[1] Institute for Intelligent System Research and Innovation (IISRI)
Deakin University
Waurn Ponds, VIC 3216, Australia

[2] Australian Centre for Visual Technologies (ACVT)
University of Adelaide
South Australia 5005, Australia

Abstract

Tremendous amounts of expensive annotated data are a vital ingredient for state-of-the-art 3d hand pose estimation. Therefore, synthetic data has been popularized as annotations are automatically available. However, models trained only with synthetic samples do not generalize to real data, mainly due to the gap between the distribution of synthetic and real data. In this paper, we propose a novel method that seeks to predict the 3d position of the hand using both synthetic and partially-labeled real data. Accordingly, we form a shared latent space between three modalities: synthetic depth image, real depth image, and pose. We demonstrate that by carefully learning the shared latent space, we can find a regression model that is able to generalize to real data. As such, we show that our method produces accurate predictions in both semi-supervised and unsupervised settings. Additionally, the proposed model is capable of generating novel, meaningful, and consistent samples from all of the three domains. We evaluate our method qualitatively and quantitively on two highly competitive benchmarks (*i.e.*, NYU and ICVL) and demonstrate its superiority over the state-of-the-art methods. The source code will be made available at https://github.com/masabdi/LSPS.

1 Introduction

Recovering the 3d configuration of the human hand has many applications, including augmented/virtual reality, human-computer interaction, sign language recognition, and robotics. Deep neural networks have obtained significant success in 3d hand pose estimation over the past few years [20, 51]. These achievements, however, are highly dependent on the existence of massive amounts of training data, supervised by (usually) human-annotated targets, which are expensive and costly to acquire. As such, two extra sources of information have been extensively used to mitigate the need for the expensive annotations: synthetic data and unlabeled real data [16, 25, 26, 32, 33, 36].

To this end, three main research directions have been proposed to tackle this shortcoming. The first line of work uses unlabeled real samples to learn representations that are useful for the semi-supervised learning task. As a result, they use the unlabeled samples along with a few labeled examples to obtain a more accurate regression model [26, 32]. A second way to avoid the labeling burden is to use synthetic data, since annotations are automatically available [2, 33, 36]. However, learning from synthetic data usually results in a sub-optimal solution. This is due to the *domain-gap* between the distribution of synthetic and real data. Thus, the third direction of research incorporates both synthetic and unlabeled real data. It uses unlabeled real samples to map the synthetic data to a distribution close to that of the real data, and uses the annotations of the synthetic data to obtain a more accurate estimation of the hand pose [16, 25].

In this paper, we introduce a novel 3d hand pose estimation method that uses both synthetic and partially-labeled real data. We dub it Learning from Simulation and Partial-Supervision (LSPS). We formulate the problem as a generative modeling problem using a shared latent space. More specifically, we form a shared latent representation between: (1) real depth image domain, (2) synthetic depth image domain, and (3) pose domain. Firstly, we show that by carefully learning the shared latent space using synthetic and unlabeled real samples, we can train a regressor that is able to generalize to real data. We then extend the proposed method to a semi-supervised setting, where we use real annotations to further enhance the performance of the model. Additionally, we show that samples from each of the three domains can be mapped to the latent space and back to the original form, and that they can be translated to other domains in a coherent and meaningful manner. The proposed model allows us to generate novel and consistent samples from all of the three domains.

Our model based is on variational autoencoder (VAE) [13] and generative adversarial network (GAN) [9]. To obtain the shared latent representation, we use two VAE-GAN hybrids [14] to model the real and synthetic depth image domains, and a VAE for the pose domain. Since we do not generally have access to the corresponding real/synthetic image pairs, we exploit the weight-sharing and cycle-consistency constraints [15] to learn the mapping between the real and synthetic depth data. We also learn an auxiliary mapping function (M) and a posterior estimation function (P) that relate the latent space of the depth and that of the pose. This results in a single shared latent representation between the three domains. Extensive experimental results on two real-world datasets, *i.e.*, NYU [31] and ICVL [29], demonstrate that the proposed method performs better than the existing models.

In summary, the **main contributions** of this work are as follows: **(1)** Presenting the first deep model (to the best of our knowledge) that learns from both synthetic and partially-labeled real data. **(2)** Proposing a novel shared latent space between three modalities, *i.e.*, synthetic depth, real depth, and pose. **(3)** Performing extensive experiments on two highly competitive datasets and showing the superiority of our model over the state-of-the-art methods.

2   Related Work

3d hand pose estimation is a long-lasting problem in computer vision and related areas. It has received much attention recently due to its widespread application and affordable depth sensors [19, 20, 21, 27, 29, 31, 35, 36]. Approaches based on deep-learning, especially convolutional neural networks, are shown to be efficient and accurate in estimating the 3d position of the hand [19, 20, 21, 31, 35, 36]. In this section, we outline some of the recent
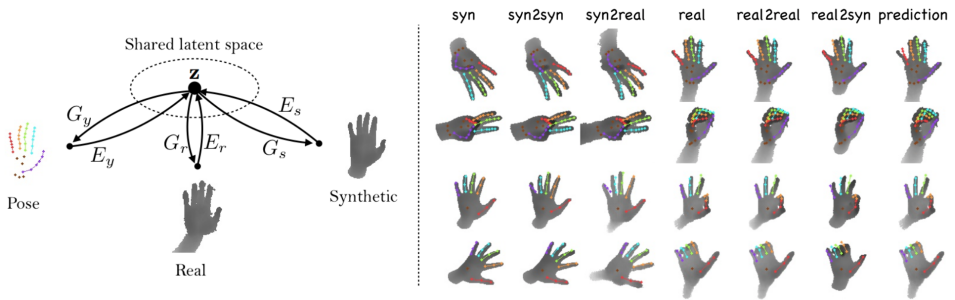
Figure 1: **Left** depicts our shared latent variable assumption. The assumption implies the existence of a latent code $\mathbf{z}$ for every given $(\mathbf{x}_r, \mathbf{x}_s, \mathbf{y})$, such that samples from each domain can be encoded (decoded) to (from) $\mathbf{z}$ using proper encoding (decoding) functions $(E_r, E_s, E_y, G_r, G_s, G_y)$. **Right** shows transformations of the samples from one domain to other domains using the learned latent space.

work surrounding our research.

Synthetic Data. Synthetic data has been a popular choice for pose estimation [25, 28, 33, 36]. To reduce the domain-gap between synthetic and real data, Tang *et al*. [28] propose a transductive regression forest that uses unlabeled and synthetic data to estimate the 3d hand pose. Shrivastava *et al*. [25] incorporate a refiner network that aims to improve the realism of the synthetic data through an adversarial training. Recent work of Mueller *et al*. [16] translates synthetic images to real data using a geometrically consistent image-to-image translation network, and predicts the position of the hand from RGB images. Rad *et al*. [23] use million-scale synthetic data, real data, and a feature matching strategy that is shown to produce high accuracy predictions.

Generative Modeling. Generative models, in particular VAEs [13] and GANs [9], have recently shown to be very effective in many applications such as: image generation [9, 13, 24], representation learning [3, 5], and image-to-image translation [11, 15]. Hybrid models combine multiple GANs and VAEs in order to take advantage of both frameworks [5, 6, 14, 15]. Generative models have also proven to be efficient in 3d pose estimation from a single image [16, 26].

Shared Embedding. The concept of shared embedding has been previously discussed in the literature, where it is assumed that samples from different modalities (*i.e.* RGB image, depth image, pose, etc.) can be mapped to a shared embedding. Ngiam *et al*. [18] form a shared representation between audio and video. Ek *et al*. [7] and Navaratnam *et al*. [17] use gaussian process latent variable models and form a shared latent representation between the image observations and human poses. A cross-modal variational model derives a variational lower-bound based on VAE, that can be used to learn a shared latent space between different modalities [26]. Wan *et al*. [32] model the shared latent variable using a combination of VAE and GAN and use a mapping function to relate the two latent spaces.

Our work is inspired by [15, 32]. Liu *et al*. [15] form a shared latent space between two image domains, and learn the mapping between the *unpaired* samples using weight-sharing and cycle-consistency constraints. Wan *et al*. [32] create a shared latent variable between depth image and pose domains using a combination of VAE and GAN. In this work, we use synthetic and partially-labeled real data to form a shared latent space between the two depth
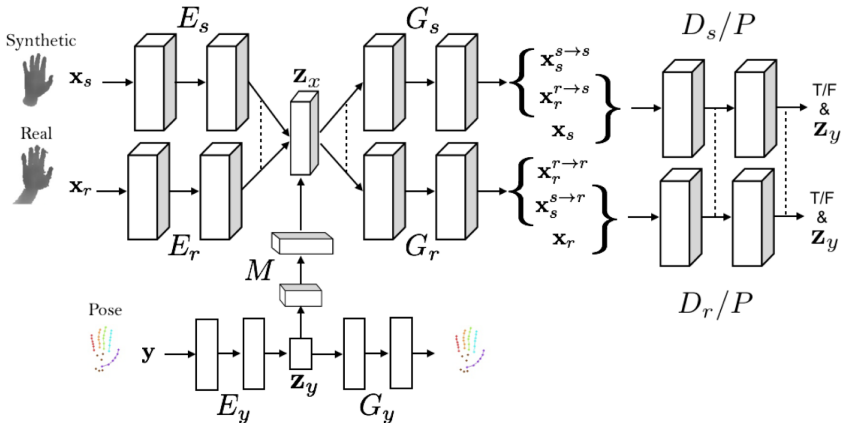
Figure 2: **Visualization of the proposed model.** Cubes and rectangles represent convolutional and linear layers, respectively. Dotted lines denote weight-sharing. $\mathbf{z}_y$ is the latent variable of the pose domain, while $\mathbf{z}_x$ is the latent variable of the depth domain.

image domains (synthetic and real), and the pose domain. This allows us to find a regressor that performs efficiently on real data, even without any real annotations.

3 Learning from Simulation and Partial-Supervision (LSPS)

Let $\mathcal{X}_r, \mathcal{X}_s$, and $\mathcal{Y}$ be the domain of real depth images, synthetic depth images, and poses, respectively. Using the shared latent space assumption, we assume that there exists a shared latent representation $\mathbf{z} \in \mathbb{R}^d$ for every triplet $(\mathbf{x}_r, \mathbf{x}_s, \mathbf{y})$ drawn from the joint distribution of the three domains $P_{\mathcal{X}_r, \mathcal{X}_s, \mathcal{Y}}(\mathbf{x}_r, \mathbf{x}_s, \mathbf{y})$. As such, the latent code $\mathbf{z}$ can be converted to (or recovered from) any of these three domains using proper encoding (or decoding) functions, see Figure 1. That is, we assume that we can find encoding (E_r, E_s, E_y) and decoding (G_r, G_s, G_y) functions such that all of these conditions hold:

$$E_r(\mathbf{x}_r) = E_s(\mathbf{x}_s) = E_y(\mathbf{y}) = \mathbf{z}, \quad G_r(\mathbf{z}) = \mathbf{x}_r, G_s(\mathbf{z}) = \mathbf{x}_s, G_y(\mathbf{z}) = \mathbf{y}.$$

Learning such latent representation is useful as it creates a shared understanding of the data in three domains, and enables us to transform samples from one domain to either of the other two domains. Furthermore, we can render the realistic depth map that corresponds to any arbitrarily hand pose $\mathbf{y}$ using the composition function $R_r = G_r(E_y(\mathbf{y}))$, or we can use the composite function of $P_r = G_y(E_r(\mathbf{x}_r))$ to predict the 3d position of the hand from a real depth image.

In order to form this shared latent space, we use three components to model each of the three domains $(\mathcal{X}_r, \mathcal{X}_s, \mathcal{Y})$. The pose domain is modeled using a VAE as its low-dimensional embedding is shown to impose pose constraints that improve the reliability of the predictions [20, 52]. The depth image domains are modeled through VAE-GAN hybrids [14], see Figure 2 for a visualization of the model.

3.1 Pose Domain

We first model the pose domain (VAE_y) using the variational upper bound as follows:

$$\mathcal{L}_{\text{VAE}_y}(E_y, G_y) = \lambda_0 \text{KL}(q_y(\mathbf{z}_y|\mathbf{y})||p(\mathbf{z}_y)) - \lambda_1 \mathbb{E}_{\mathbf{z}_y \sim q_y(\mathbf{z}_y|\mathbf{y})}[\log p_{G_y}(\mathbf{y}|\mathbf{z}_y)] \tag{1}$$

where $q_y(\mathbf{z}_y|\mathbf{y})$ is our encoding distribution described by a multivariate Gaussian distribution $\mathcal{N}(E_{y;\mu}(\mathbf{y}), E_{y;\sigma^2}(\mathbf{y}))$ with mean $E_{y;\mu}(\mathbf{y})$ and variance $E_{y;\sigma^2}(\mathbf{y})$, and $p(\mathbf{z}_y)$ is a Gaussian distribution with zero mean and unit variance $\mathcal{N}(0, I)$. We call this the *pose latent space*. Throughout this paper, λ_is are hyper-parameters that trade-off the relative importance of the terms. For example, here λ_0 and λ_1 control the relative importance of the KL and the negative log-likelihood term.

3.2 The Shared Latent Space

We then relate the two image domains by forming a shared latent space between them. Since we do not generally have access to the paired real/synthetic depth images, we use the weight-sharing and cycle-consistency constraints [15] to learn the shared latent space between the two image domains. We call this the *depth latent space*. This is not sufficient for our formulation of the shared latent space, as we need to relate the pose latent space to the depth latent space. To this end, we define an auxiliary mapping function $M(\mathbf{z}_y)$ that maps the samples of the pose latent space to those from the depth latent space [32].

To train the mapping function (M) we need the corresponding latent codes in the pose and the depth latent spaces. We obtain these codes using the synthetic pairs $(\mathbf{x}_s, \mathbf{y})$ and their respective encoding functions (E_s, E_y). Therefore, we propose the following cost function to learn our shared latent space:

$$\min_{M,P,E_s,E_r,G_s,G_r} \max_{D_s,D_r} \mathcal{L}_{\text{VAE}_s}(E_s, G_s) + \mathcal{L}_{\text{GAN}_s}(E_s, G_s, D_s) + \mathcal{L}_{\text{CC}_s}(E_s, G_s, E_r, G_r)$$
$$+ \mathcal{L}_{\text{VAE}_r}(E_r, G_r) + \mathcal{L}_{\text{GAN}_r}(E_r, G_r, D_r) + \mathcal{L}_{\text{CC}_r}(E_r, G_r, E_s, G_s)$$
$$+ \mathcal{L}_{\text{MAP}_s}(M, E_s, G_s, D_s) + \mathcal{L}_{\text{POS}}(D_s, D_r, P), \tag{2}$$

where $\mathcal{L}_{\text{VAE}_s}, \mathcal{L}_{\text{VAE}_r}, \mathcal{L}_{\text{GAN}_s}, \mathcal{L}_{\text{GAN}_r}, \mathcal{L}_{\text{CC}_s}$ and $\mathcal{L}_{\text{CC}_r}$ are responsible for learning a shared latent space between the two depth domains, and $\mathcal{L}_{\text{MAP}_s}$ and $\mathcal{L}_{\text{POS}}$ align the latent space of the pose to that of the depth. We will discuss each of these terms in details, separately.

The VAE terms in Equation 2 optimize a variational upper bound using the encoding functions, that output the mean $E_{s;\mu}(\mathbf{x}_s)$ of a multivariate Gaussian distribution with unit variance $(q_s(\mathbf{z}_s|\mathbf{x}_s) \equiv \mathcal{N}(E_{s;\mu}(\mathbf{x}_s), I))$ as follows:

$$\mathcal{L}_{\text{VAE}_s}(E_s, G_s) = \lambda_2 \text{KL}(q_s(\mathbf{z}_s|\mathbf{x}_s)||p(\mathbf{z})) - \lambda_3 \mathbb{E}_{\mathbf{z}_s \sim q_s(\mathbf{z}_s|\mathbf{x}_s)}[\log p_{G_s}(\mathbf{x}_s|\mathbf{z}_s)], \tag{3}$$

where $p(\mathbf{z}) \equiv \mathcal{N}(0, I)$, and $\mathcal{L}_{\text{VAE}_r}$ is defined in a similar way. The GAN terms in Equation 2 make sure that the generated samples of G_r and G_s are realistic looking and indistinguishable from their domain, using the corresponding discriminators D_r and D_s. Therefore,

$$\mathcal{L}_{\text{GAN}_s}(E_s, G_s, D_s) = \lambda_4 \mathbb{E}_{\mathbf{x}_s \sim P_{\mathcal{X}_s}}[\log D_s(\mathbf{x}_s)] + \lambda_4 \mathbb{E}_{\mathbf{z}_r \sim q_r(\mathbf{z}_r|\mathbf{x}_r)}[\log(1 - D_s(G_s(\mathbf{z}_r)))]. \tag{4}$$

Note that it suffices to apply the GAN cost function only to the translated images, as the reconstruction network is trained with the reconstruction term in Equation 3. The cycle-consistency term $\mathcal{L}_{\text{CC}_s}$ in Equation 2 ensures that a twice-translated image resembles the
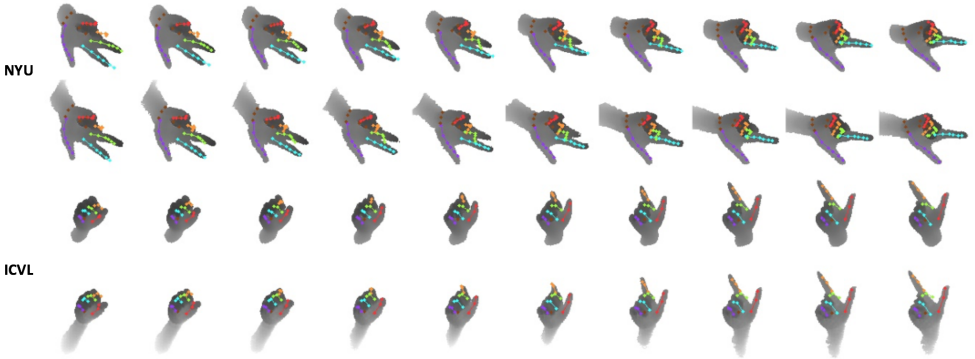
Figure 3: **A random walk in the learned latent space.** First and second pairs of rows correspond to NYU and ICVL datasets, respectively. In each pair, the top row shows the generated synthetic samples and the bottom row shows the generated real samples. One can see that samples generated on the connecting line of two latent codes have a meaningful interpolation in all three (real, synthetic, and pose) domains.

initial image [15]. This is done using a VAE-like cost function as follows:

$$\mathcal{L}_{\text{CC}_s}(E_s, G_s, E_r, G_r) = \lambda_2 \text{KL}(q_r(\mathbf{z}_r|\mathbf{x}_s^{s\rightarrow r}))||p(\mathbf{z})) - \lambda_3 \mathbb{E}_{\mathbf{z}_r \sim q_r(\mathbf{z}_r|\mathbf{x}_s^{s\rightarrow r})}[\log p_{G_s}(\mathbf{x}_s|\mathbf{z}_r)]. \quad (5)$$

where $\mathbf{x}_s^{s\rightarrow r}$ is the translated image from synthetic to real, which is obtained via $G_r(E_s(\mathbf{x}_s))$.

We now relate the latent space of the pose and the latent space of the depth. We take the latent variable of the pose to be the reference variable, and learn a mapping function M to the latent space of the depth as follows:

$$\begin{aligned}
\mathcal{L}_{\text{MAP}_s}(M, E_s, G_s, D_s) = &\lambda_5 \mathbb{E}_{\mathbf{z}_y \sim q_y(\mathbf{z}_y|\mathbf{y}), \mathbf{z}_s \sim q_s(\mathbf{z}_s|\mathbf{x}_s)}||M(\mathbf{z}_y) - \mathbf{z}_s||_2 \\
&+ \lambda_6 \mathbb{E}_{\mathbf{z}_y \sim q_y(\mathbf{z}_y|\mathbf{y})}[\log p_{G_s}(\mathbf{x}_s|M(\mathbf{z}_y))] \\
&+ \lambda_4 \mathbb{E}_{\mathbf{z}_y \sim q_y(\mathbf{z}_y|\mathbf{y})}[\log(1 - D_s(G_s(M(\mathbf{z}_y))))]. \quad (6)
\end{aligned}$$

where $||.||_2$ is the 2-norm operator. The first term in Equation 6 ensures that samples from the latent variable of the pose are close to their corresponding latent codes when mapped through the mapping function M. The latent codes are obtained using synthetic data $(\mathbf{x}_s, \mathbf{y})$. The second and the third terms ensure that generation results of the mapped latent codes resemble their corresponding depth data.

Since the mapping function only operates in one direction (from the pose latent space to the depth latent space), we propose to estimate the posterior of the pose using a new function P. The posterior estimation function $P : \mathcal{X}_s \rightarrow \mathbf{z}_y$ aims to find the latent code for a given image in the pose latent space. Therefore,

$$\mathcal{L}_{\text{POS}_s}(P) = \lambda_7 \mathbb{E}_{\mathbf{z}_y \sim q_y(\mathbf{z}_y|\mathbf{y})}||P(\mathbf{x}_s) - \mathbf{z}_y||_2. \quad (7)$$

In practice, P has the same architecture as the discriminators, thus it shares all of its layers with the discriminators, except for the last layer. Furthermore, the last term of our cost

function is described as follows:

$$\mathcal{L}_{\text{POS}}(D_s, D_r, P) = \mathcal{L}_{\text{POS}_s}(P) + \mathcal{L}_{\text{FM}}(D_s, D_r), \tag{8}$$

$$\mathcal{L}_{\text{FM}}(D_s, D_r) = \lambda_8 ||D_{s,\phi}(\mathbf{x}_s^{s \to s}) - D_{r,\phi}(\mathbf{x}_s^{s \to r})||_1$$
$$+ \lambda_8 ||D_{s,\phi}(\mathbf{x}_r^{r \to s}) - D_{r,\phi}(\mathbf{x}_r^{r \to r})||_1, \tag{9}$$

where $D_{s,\phi}$ is the activations of the penultimate layer of the discriminator (D_s), $\mathbf{x}_s^{s \to s}$ and $\mathbf{x}_r^{r \to r}$ are the reconstructed samples, while $\mathbf{x}_s^{s \to r}$ and $\mathbf{x}_r^{r \to s}$ are the translated ones. The $\mathcal{L}_{\text{FM}}$ term is added to make sure that discriminators, D_s and D_r, similarly interpret the two corresponding samples generated using G_s and G_r.

3.3 Extension to Semi-Supervised Learning

The flexibility of our model allows us to use any number of annotated real data to further enhance the overall performance of the model, when (partial-)supervision is available. This extra source of supervision is used to guide the mapping function M to better align the two latent spaces of pose and depth. This is done by defining $\mathcal{L}_{\text{MAP}_r}(M, E_r, G_r, D_r)$ term similar to $\mathcal{L}_{\text{MAP}_s}$, and adding it to Equation 2. More importantly, we use the supervision of real data to approximate the posterior more accurately, by adding an extra term $\mathcal{L}_{\text{POS}_r}(P)$ to Equation 8, where $\mathcal{L}_{\text{POS}_r}$ is also defined in a similar way to $\mathcal{L}_{\text{POS}_s}$.

4 Implementation Details

Our model consists of several sub-networks, some of which share a few layers (see Figure 2). The image encoders and decoders (E_s, E_r, G_s, G_r) have 3 convolutional layers and 4 residual blocks, where the first (last) residual block of the decoders (encoders) are shared. We use residual blocks with multiple residual connections [1, 34]. The discriminators (D_s, D_r) have 6 convolutional layers with 4 layers being shared between them. The pose model has one hidden layer with 30 units in both encoder and decoder (E_y, G_y), and a 20 dimensional latent space. The mapping function (M) has four transposed convolutional layers. The posterior estimation function (P) is a network with 6 convolutional layers that shares all of its layers with the discriminators, except for the last layer. More details about the network architectures can be found in appendix.

We use Adam [12] to train our model with a learning rate of 0.0001. First, the pose model is trained for 200k iterations with a mini-batch size of 128 using Equation 1. The depth model and the mapping function are then trained (Equation 2) using the learned parameters of the pose model. Training continues for 500k iterations using one sample from each domain. Finally, the posterior estimation model is trained for 50k iterations with a mini-batch size of 32 (Equation 8), resulting in our shared latent space. We note our design choice regarding the two latent spaces of depth and pose domains $(\mathbf{z}_x, \mathbf{z}_y)$, see Figure 2. Although it is theoretically possible to have a single multi-modal latent space [26], we believe that having two connected latent spaces is of importance to the success of our method. This is because learning the mapping between synthetic and real data will be made easier and would require a less complicated network. As a result, generalization from synthetic to real data will be more effective by avoiding possible artifacts in down-sampling and/or up-sampling operations [22].
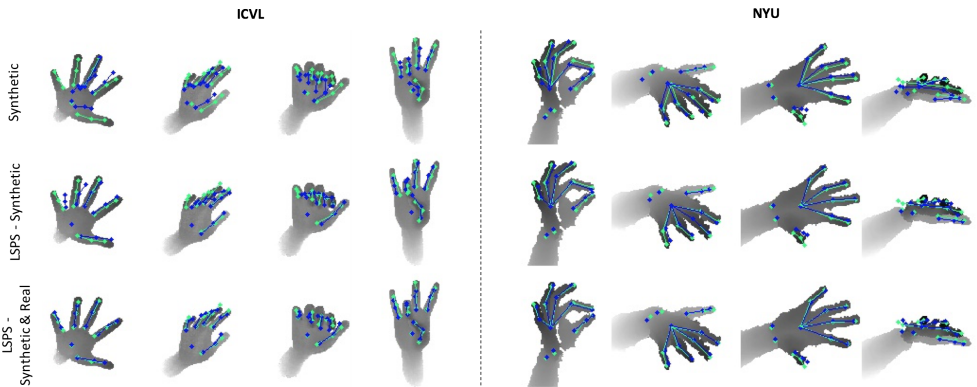
Figure 4: **Qualitative evaluation of the proposed method.** Predictions and ground-truth annotations are depicted as blue and green, respectively. First row shows the predictions of training only on the synthetic data. The second row depicts the predictions of LSPS, when only synthetic and unlabeled real data are used. The third row shows the predictions of LSPS using both synthetic and real data.

5 Experiments

We evaluate our method qualitatively and quantitatively on two publicly-available datasets, *i.e.*, NYU and ICVL. The NYU dataset [31] contains very noisy depth images and has a wide range of poses, making it challenging for most pose estimation methods. It has around 72k samples for training and 8k for testing. The ICVL dataset [29] consists of 22k training and 1.6k testing samples with a large discrepancy between the training and testing sequences. We use the synthetic data of NYU with 72k depth images and their corresponding 3d poses [31].

We follow the pre-processing pipeline of Oberweger *et al.* [19], where a fixed-size 3d cube centered at the center of the hand is cropped and resized to 128×128 depth images, normalized to $[-1,1]$. Since ICVL dataset contains left hand images, we flip the depth maps to resemble the right hand, as in the synthetic data. We augment the data with $180°$ bidirectional rotation and $10mm$ translation. We observed that our model works with many choices of hyper-parameters. We set these hyper-parameters to be $\lambda_0 = \lambda_2 = 0.1$, $\lambda_1 = \lambda_3 = \lambda_5 = 100$, $\lambda_4 = \lambda_7 = 10$, $\lambda_6 = 10000$, and $\lambda_8 = 1$. We also consider Equation 9 with $\lambda_8 = 0.0001$ when training the depth model using Equation 2.

	Joint mean error (mm)		% of frames within 40mm	
	NYU	**ICVL**	**NYU**	**ICVL**
Synthetic-Only	25.24	23.96	36.57	51.07
Synthetic-Only*	23.71	23.91	43.60	48.37
LSPS-Synthetic [Ours]	**17.84**	**14.09**	**62.57**	**86.78**
Real data*	15.83	7.09	73.62	96.18

Table 1: **Comparison of our method with two baselines on NYU and ICVL**. One can see that our model performs significantly better than the models trained only on the synthetic data, by incorporating unlabeled real samples. The rows with * use our pre-trained model.
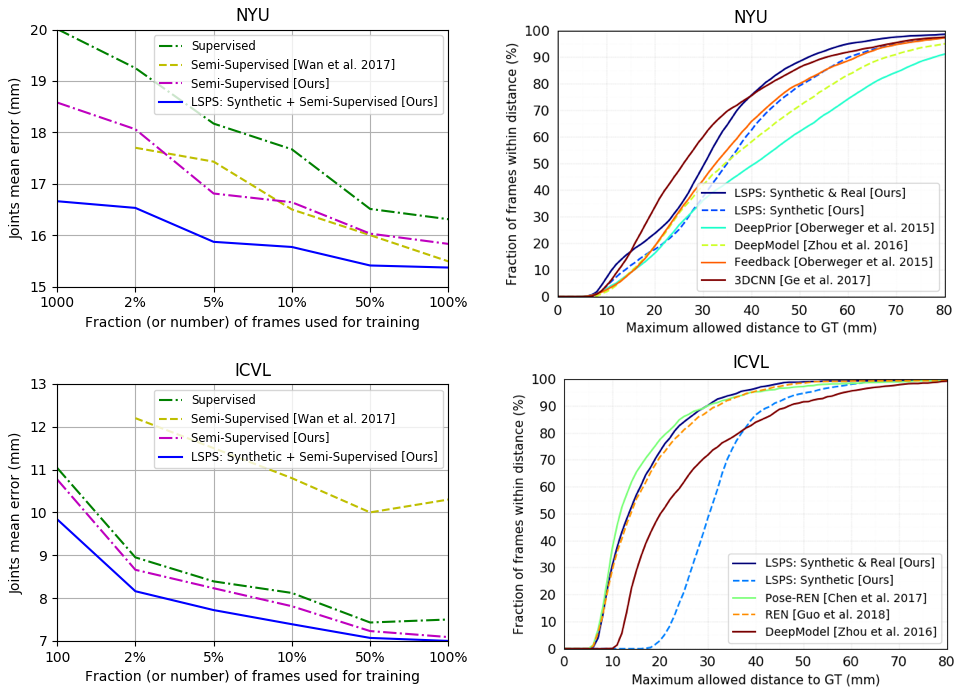
Figure 5: **Comparison to the state-of-the-art hand pose estimation models.** **Left** compares against the existing semi-supervised method. Our model uses synthetic and partially-labeled real data to predict the hand pose more accurately. This is particularly more apparent when fewer real annotations are available. Note that *x-axis* is not linear in scale. **Right** compares our model against existing fully-supervised methods. The proposed model performs comparable or better than the existing methods. It can be seen that our method outperforms some of the most recent state-of-the-art methods in an unsupervised setting.

Learning from simulation and unlabeled real data. We demonstrate that effective learning from unlabeled real data and synthetic data can lead to enhanced generalization. We compare our method with a model trained only on the synthetic data. We also use our pre-trained model and train a second baseline on the synthetic data to have a fair comparison. We use two metrics in our evaluation: joint mean error (in mm) averaged over all joints and all frames, and percentage of frames in which all joints are below a certain threshold (d) [30]. Following the previous work [20, 21], we only evaluate 14 joints from the NYU dataset.

Table 1 shows the results of the experiments against the two baselines. The proposed method achieves a mean error of **17.84mm** on NYU dataset, whereas training only on the synthetic data obtains an error rate of **23.71mm**. It can be seen that our result is quite close to training on the real data with **15.83mm** error. This demonstrates the benefits of learning from synthetic and unlabeled real data. On ICVL dataset, our method obtains **14.09mm** mean error, while training on the synthetic data obtains a poor generalization performance of **23.91mm**. Nevertheless, our method outperforms the baselines with about **19%** and **35%** absolute improvement in terms of the number of frames within 40mm on NYU and ICVL, respectively. Additionally, experiments on ICVL dataset demonstrate that our method is capable of learn-

ing from very different domains, where there exists a large discrepancy between the synthetic and real data.

Using partial supervision. We show that using a few labeled examples can significantly improve the performance of our proposed model. We uniformly sample $m\%$ of the real data and use their corresponding annotations during the training. As such, we also consider the cost functions described in Section 3.3 when training the models. We compare our results with the state-of-the-art hand pose estimation methods (see Figure 5). On the left we compare against crossingNet [32]. We also train our version of [32] using the same network to have a fair comparison. Our method shows up to 17% and 10% relative improvement on NYU as compared to the supervised baseline and the semi-supervised method of crossingNet [32], respectively. On ICVL dataset 11% and 9% improvement was obtained relative to the two other methods. The proposed method consistently outperforms the other two methods, especially when fewer labels are available. On the right (Figure 5), we compare our method with the existing fully-supervised methods *i.e.*, DeepPrior[20], DeepModel[35], Feedback[21], 3DCNN[8], REN[10], and Pose-REN[4]. Our method produces accuracies comparable or better than the existing methods in the fully-supervised experiments.

Generative capabilities. As outlined earlier, our method is capable of generating meaningful samples from all of the three domains. Figure 1 (right) shows that samples can be encoded to the latent space and decoded to any of the three domains, including their original domain. We also show the smoothness of the shared latent space by synthesizing samples along the connecting line of two latent codes (see Figure 3). This demonstrates that the learned latent space represents a valid statistical *multi-modal* model of the human hand. Interestingly, samples from the prior distribution $\mathcal{N}(0,I)$ can be used to generate valid data in all three domains. The generative capabilities of the proposed model can potentially be used to produce labelled data to further enhance the performance of existing models.

Qualitative results. We show the discriminative results of our proposed model in Figure 4. The first row shows the predictions of a model trained only on the synthetic data. The second row shows our predictions when only unlabeled real data and synthetic data are used, and the last row uses both synthetic and real data. One can see that our method produces more accurate predictions of the hand pose as compared to the model trained only on the synthetic data. See appendix for more qualitative results and some failure cases. We noticed that in some parts of the real distribution, LSPS-synthetic does not significantly improve the predictions compared to the synthetic-only case. We believe that this is due to the sparsity of the real and/or synthetic data in those parts of the distribution. Nevertheless, using real annotations along with synthetic samples consistently improves the quality of the predictions.

6   Conclusion

In this paper, we tackle the problem of 3d hand pose estimation using synthetic and partially-labeled real data. We form a shared latent space between real depth, synthetic depth, and pose domains. We use several techniques including the cycle-consistency and weight-sharing constraints to learn the shared latent space using unpaired real/synthetic samples. We show that the shared latent space facilitates a very accurate model that is able to generalize from synthetic data to real data. We also demonstrate that synthetic data can enhance the robustness of the model when partial supervision is available. The shared latent space allows us to generate samples consistently from all of the three domains.

References

[1] Masoud Abdi and Saeid Nahavandi. Multi-residual networks: Improving the speed and accuracy of residual networks. *arXiv preprint arXiv:1609.05672*, 2016.

[2] Ahmed Abobakr, Darius Nahavandi, Julie Iskander, Mohammed Hossny, Saeid Nahavandi, and Marty Smets. Rgb-d human posture analysis for ergonomie studies using deep convolutional neural network. In *Systems, Man, and Cybernetics (SMC), 2017 IEEE International Conference on*, pages 2885–2890. IEEE, 2017.

[3] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2172–2180, 2016.

[4] Xinghao Chen, Guijin Wang, Hengkai Guo, and Cairong Zhang. Pose guided structured region ensemble network for cascaded hand pose estimation. *arXiv preprint arXiv:1708.03416*, 2017.

[5] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[6] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[7] Carl Henrik Ek, Philip HS Torr, and Neil D Lawrence. Gaussian process latent variable models for human pose estimation. In *International workshop on machine learning for multimodal interaction*, pages 132–143. Springer, 2007.

[8] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 5, 2017.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[10] Hengkai Guo, Guijin Wang, Xinghao Chen, and Cairong Zhang. Towards good practices for deep 3d hand pose estimation. *arXiv preprint arXiv:1707.07248*, 2017.

[11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[12] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[14] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

[15] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, pages 700–708, 2017.

[16] 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. *arXiv preprint arXiv:1712.01057*, 2017.

[17] Ramanan Navaratnam, Andrew W Fitzgibbon, and Roberto Cipolla. The joint manifold model for semi-supervised multi-valued regression. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[18] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696, 2011.

[19] Markus Oberweger and Vincent Lepetit. Deepprior++: Improving fast and accurate 3d hand pose estimation. In *ICCV workshop*, volume 840, page 2, 2017.

[20] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Hands deep in deep learning for hand pose estimation. *arXiv preprint arXiv:1502.06807*, 2015.

[21] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Training a feedback loop for hand pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3316–3324, 2015.

[22] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 1(10):e3, 2016.

[23] Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Feature mapping for learning fast and accurate 3d pose inference from synthetic images. *arXiv preprint arXiv:1712.03904*, 2017.

[24] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *CoRR*, abs/1606.03498, 2016. URL http://arxiv.org/abs/1606.03498.

[25] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. Learning from simulated and unsupervised images through adversarial training. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 3, page 6, 2017.

[26] Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation. *arXiv preprint arXiv:1803.11404*, 2018.

[27] Xiao Sun, Yichen Wei, Shuang Liang, Xiaoou Tang, and Jian Sun. Cascaded hand pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 824–832, 2015.

[28] Danhang Tang, Tsz-Ho Yu, and Tae-Kyun Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3224–3231. IEEE, 2013.

[29] Danhang Tang, Hyung Jin Chang, Alykhan Tejani, and Tae-Kyun Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3786–3793, 2014.

[30] Jonathan Taylor, Jamie Shotton, Toby Sharp, and Andrew Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 103–110. IEEE, 2012.

[31] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG)*, 33(5):169, 2014.

[32] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[33] Erroll Wood, Tadas Baltrušaitis, Louis-Philippe Morency, Peter Robinson, and Andreas Bulling. Learning an appearance-based gaze estimator from one million synthesised images. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pages 131–138. ACM, 2016.

[34] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5987–5995. IEEE, 2017.

[35] Xingyi Zhou, Qingfu Wan, Wei Zhang, Xiangyang Xue, and Yichen Wei. Model-based deep hand pose estimation. *arXiv preprint arXiv:1606.06854*, 2016.

[36] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *International Conference on Computer Vision*, 2017.

7 APPENDIX

7.1 Network Architectures

The architecture of the pose model is represented in Table 2. The input to the pose encoder is a vector of $3J$ real numbers corresponding to J coordinates ($J = 36$ for NYU, and $J = 16$ for ICVL). The depth model uses convolutional layers and residual blocks [1, 34] with instance normalization. LeakyReLU activation function is used in almost of the layers of the model (see Table 4). The mapping function M consists of four transposed-convolutional layers that converts a 20 dimensional vector to the latent space of the depth with 32×32 spatial dimensions, see Table 3.

Layer	Encoder: E_y	Layer	Decoder: G_y
0	INPUT-(3J)	0	INPUT-(20)
1	FC-(30N), LeakyReLU	1	FC-(30N), LeakyReLU
μ	FC-(20N)	2	FC-(3J)
σ^2	FC-(20N)		

Table 2: Network architecture of the pose model.

Layer	Mapping: M
0	INPUT-(3J)
1	TCONV-(N1024,K4,S1), LeakyReLU
2	TCONV-(N1024,K4,S2), LeakyReLU
3	TCONV-(N512,K4,S2), LeakyReLU
4	TCONV-(N256,K4,S2)

Table 3: Network architecture of the mapping function.

Layer	Encoders: E_s, E_r	Shared?
1	CONV-(N64,K7,S1), LeakyReLU	✗
2	CONV-(N128,K3,S2), LeakyReLU	✗
3	CONV-(N256,K3,S2), LeakyReLU	✗
4	RESBLK-(N512,C4,K3,S1)	✗
5	RESBLK-(N512,C4,K3,S1)	✗
6	RESBLK-(N512,C4,K3,S1)	✗
μ	RESBLK-(N512,C4,K3,S1)	✓
Layer	**Generators: G_s, G_r**	**Shared?**
1	RESBLK-(N512,C4,K3,S1)	✓
2	RESBLK-(N512,C4,K3,S1)	✗
3	RESBLK-(N512,C4,K3,S1)	✗
4	RESBLK-(N512,C4,K3,S1)	✗
5	TCONV-(N256,K3,S2), LeakyReLU	✗
6	TCONV-(N128,K3,S2), LeakyReLU	✗
7	TCONV-(N3,K1,S1), TanH	✗
Layer	**Discriminators: D_s, D_r**	**Shared?**
1	CONV-(N64,K3,S2), LeakyReLU	✗
2	CONV-(N128,K3,S2), LeakyReLU	✗
3	CONV-(N256,K3,S2), LeakyReLU	✓
4	CONV-(N512,K3,S2), LeakyReLU	✓
5	CONV-(N1024,K3,S2), LeakyReLU	✓
6	CONV-(N1,K2,S1), Sigmoid	✓
Layer	**Posterior: P**	**Shared with D_s, D_r?**
1	CONV-(N64,K3,S2), LeakyReLU	✓
2	CONV-(N128,K3,S2), LeakyReLU	✓
3	CONV-(N256,K3,S2), LeakyReLU	✓
4	CONV-(N512,K3,S2), LeakyReLU	✓
5	CONV-(N1024,K3,S2), LeakyReLU	✓
6	CONV-(N20,K2,S1)	✗

Table 4: Network architecture of the depth model.

7.2 More Qualitative Results and Failure Cases

Figure 6 depicts the discriminative results and Figure 8 show the generative results. Also, some failure cases are shown in Figure 7.
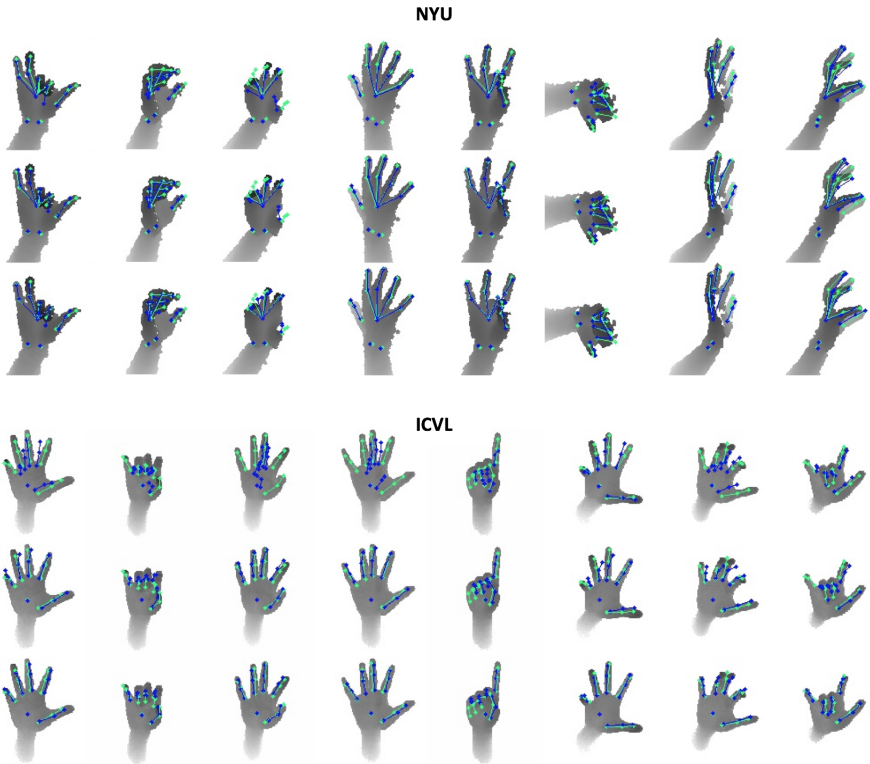
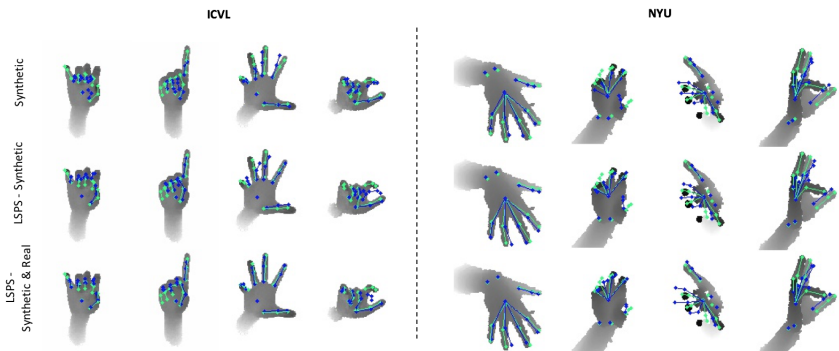Figure 6: More discriminative results on NYU and ICVL.

Figure 7: Some failure cases. In some parts of the real distribution, LSPS-synthetic does not necessarily improve the predictions as compared to the synthetic-only case. However, using real annotations along with the synthetic samples consistently improve the quality of the predictions.
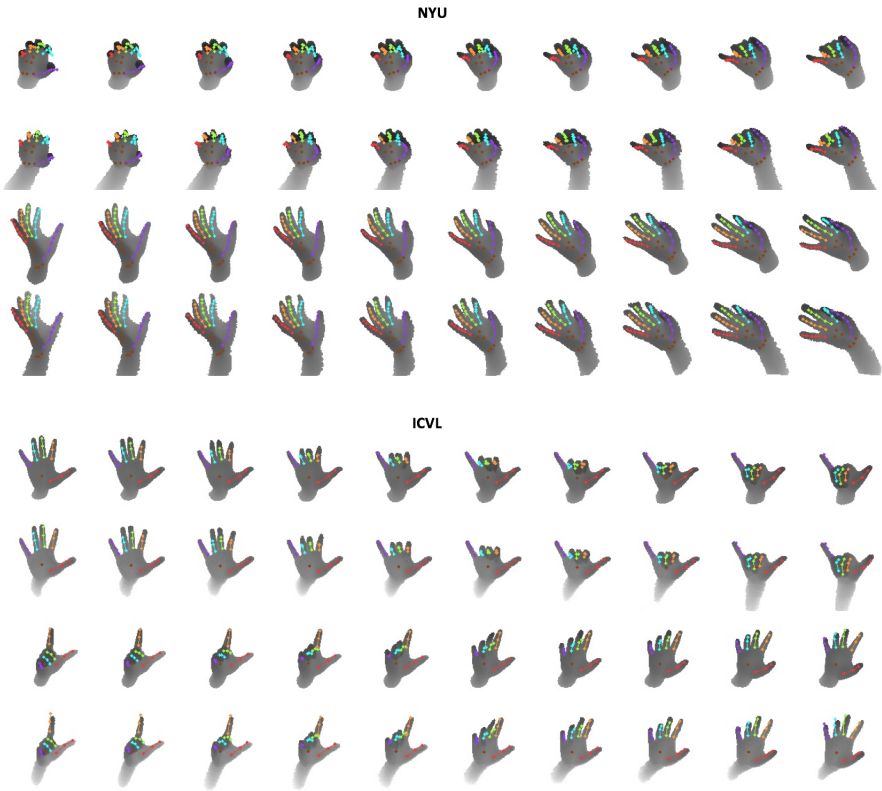
Figure 8: More generative results on NYU and ICVL.