Crossing Nets: Combining GANs and VAEs with a Shared Latent Space for Hand Pose Estimation

Chengde Wan[1], Thomas Probst[1], Luc Van Gool[1,3], and Angela Yao[2]

[1]ETH Zürich [2]University of Bonn [3]KU Leuven

Abstract

State-of-the-art methods for 3D hand pose estimation from depth images require large amounts of annotated training data. We propose to model the statistical relationships of 3D hand poses and corresponding depth images using two deep generative models with a shared latent space. By design, our architecture allows for learning from unlabeled image data in a semi-supervised manner. Assuming a one-to-one mapping between a pose and a depth map, any given point in the shared latent space can be projected into both a hand pose and a corresponding depth map. Regressing the hand pose can then be done by learning a discriminator to estimate the posterior of the latent pose given some depth map. To improve generalization and to better exploit unlabeled depth maps, we jointly train a generator and a discriminator. At each iteration, the generator is updated with the back-propagated gradient from the discriminator to synthesize realistic depth maps of the articulated hand, while the discriminator benefits from an augmented training set of synthesized and unlabeled samples. The proposed discriminator network architecture is highly efficient and runs at 90ḞPS on the CPU with accuracies comparable or better than state-of-art on 3 publicly available benchmarks.

1. Introduction

We address the problem of estimating 3D hand pose from single depth images. Accurate estimation of the 3D pose in real-time has many challenges, including the presence of local self-similarity and self-occlusions.Since the availability of low-cost depth sensors, the progress made in developing fast and accurate hand trackers have relied heavily on having a large corpus of depth images annotated with hand joints. This is especially true for the recent success of deep learning-based methods [46, 20, 21, 32, 9, 51, 45, 48] which are all fully-supervised.

Accurately annotating 3D hand joints on a depth map is both difficult and time-consuming. While it is possible to synthesize data with physical renderers, there are usually discrepancies between the real and synthesized data. Generated hand poses are not always natural nor reflective of poses seen in real-life applications. More importantly, it is very difficult to accurately model and render depth sensor noise in a realistic way.

Figure 1. **Random walk in the learned shared latent space.** A set of points is sampled on the connecting line between two points in the shared latent space. The pose and corresponding depth map are then reconstructed through our network. Our method generates meaningful and realistic interpolations in both pose and appearance space.

On the other "hand", it is very simple to collect an unlabeled dataset of real human hands with standard consumer depth cameras. This begs the question: how can one use these unlabeled samples for training? To date, there has been virtually no work presented on semi-supervised learning for hand pose estimation. The one notable exception [41] is a discriminative approach using transductive random forests and largely ignores high-order pixel correlations of unlabeled depth maps.

Previous works from neuroscience [29], robotics [1] and hand motion capture [15] have demonstrated that hand movements exhibit strong correlations between joints. We therefore conclude that the space of realistic hand posescan be represented by a manifold in a lower-dimensional subspace. We further intuit that depth maps of the hand can be similarly encoded in a low-dimensional manifold, and be

faithfully reconstructed with an appropriate generator.

In this paper, we propose a dual generative model that captures the latent spaces of hand poses and corresponding depth images for estimating 3D hand pose. We use the variational autoencoder (VAE) and the generative adversarial network (GAN) for modelling the generation process of hand poses and depth maps respectively. We assume a one-to-one mapping between a depth map and a hand pose; in this way, one can consider the latent hand pose space and latent depth map space to be shared. Having a shared space is highly beneficial, since a point sampled in either latent space can be expressed both as a 3D pose, via the VAE's decoder, or as a depth map, through the GAN's generator.

Fig. 2 gives an overview of our proposed framework. Our core idea is to learn a bi-directional mapping that relates the two latent spaces of hand poses and depth maps and therefore link together the pose encoder network with the generative models for hand poses and depth maps. A very efficient discriminator network then regresses the pose from the generated depth image. We argue that end-to-end learning of the "crossed" networks is highly beneficial for pose estimation for several reasons. First, this architecture implicitly encodes skeleton constraints as learned from the pose data distribution. Second, the generator network effectively serves to augment the training set and improves generalization by encouraging the discovery of general representations of the observed depth data in the discriminator network. Finally, the architecture naturally allows for exploiting also unlabeled data in a semi-supervised manner.

We learn our discriminator in a multi-task setting. First the discriminator must be able to measure the difference between two given depth maps in the latent space. For the generator, synthesized images from random noise are encouraged to have a desired difference to some labeled reference depth maps as measured by the discriminator. This results in a generator that produces smoother results w.r.t. the latent space. The second task of the discriminator is the standard GAN task of disambiguating real and synthesized depth maps. The posterior estimation of the hand pose, which is at the core of our method, is the third task for the discriminator. All three tasks share the same input features, *i.e*. the first several layers of the network and enables the posterior estimation to benefit from unlabeled and synthesized samples.

The resulting estimation framework is evaluated on 3 challenging benchmarks. Due to its simple network architecture, our method can run in real-time on the CPU and achieves results comparable or better than state-of-art with more sophisticated models. Our contributions can be summarized as follows:

- We extend the GAN to a semi-supervised setting for real-valued structured prediction. Previous semi-supervised adaptations of the GAN [22, 18, 34, 30] have only focused on classification and are based on the fundamental assumption that the latent distribution is multi-modal with each mode corresponding to one class. This assumption does not hold for the continuous pose regression task, since the underlying distribution of the depth map latent space does not necessarily feature multiple distinct modes.
- We tackle posterior estimation within a multi-task learning framework. We take advantage of the GAN to synthesize highly realistic and accurate depth maps of the articulated hand during training. Compared to a baseline which estimates the posterior directly, the multitask setting estimates more accurate poses, with the difference becoming especially prominent when training data is scarce.
- The learned generator synthesizes realistic depth maps of highly articulated hand poses under dramatic viewpoint changes while remaining well-behaved w.r.t. the latent space. Our novel distance constraint enforces smoothness in the learned latent space so that performing a random walk in the latent space corresponds to synthesizing a sequence of realistically interpolated poses and depth maps (see Fig. 1).

2. Related Works

Deep Generative Models The generative adversarial network (GAN) [10] and the variational autoencoder (VAE) [14] are two recently proposed deep generative models. Typically, determining the underlying data distribution of unlabeled images can be highly challenging and inference on such distributions is highly computationally expensive and or intractable except in the simplest of cases. GANs and VAEs provide efficient approximations, making it possible to learn tractable generative models of unlabeled images. We provide a more detailed description in Section 3 and refer the reader to [10, 14] for a more exhaustive treatment.

Recent works have extended the VAE [13, 33, 27] and the GAN [18, 34, 22, 30] from unsupervised to semi-supervised settings, though only for classification tasks. These works assume a multi-modal distribution in the latent space; while fitting for classification, this assumption does not hold for real-valued structured prediction, as is the case for hand pose estimation. Other works [11, 4, 25, 50, 30] modify the generation model to improve synthesis. For example, the methodology in [25, 30] stabilized the training process of the GAN, resulting in higher quality synthetic samples. We use the fully convolutional network as proposed in [25] as the GAN architecture and the feature matching strategy proposed in [30].

Since it is not possible to estimate the posterior on the GAN, [6, 7, 2] have extended the GAN to be bidirectional.

Our proposed network most resembles [2], which also formulates posterior estimation as multi-task learning. However, instead of only estimating a subvector of the latent variable and leaving the rest as random noise as in [2], we learn the entire posterior. Some other works extend the GAN to cover multiple domains, and synthesize images from text [26, 17] or from another image domain [16, 38]. We tackle a far more challenging case of synthesizing depth maps from given poses. The synthesized depth map need to be very accurate to correspond to the given pose parameters and indeed they are, as we are even able to use synthesized images for training.

Hand pose estimation Hand pose estimation generally falls into two camps, *i.e*. model-based tracking and frame-wise discriminative estimation. Conventional methods need either manually designed energy functions to measure the difference between synthesized samples and observations in model-based tracking [23, 24, 31, 40, 35, 47, 42] or hand-crafted local [41, 39, 36, 49, 40] or holistic [3] features for discriminative estimation.

Most recent works [46, 20, 21, 32, 9, 51, 45, 48] apply convolutional neural networks (CNNs), and combine feature extraction and discriminative estimation into an end-to-end learning framework. CNNs need lots of labeled training data and few works have considered utilizing more easily accessible unlabeled depth maps to learn better representations. In that sense, our work resembles [41] which tries to correlate unlabelled depth maps. While [41] takes a discriminative approach to learn a transductive random forest, our generative approach is able to capture the distribution of unlabeled depth maps.

Our work is inspired by [8, 19], which learned a shared manifold for observations and pose parameters based on the Gaussian process latent variable model (GPLVM). Another similar line of works are [5, 52], which try to learn a shared latent space between pose and gait also based on GPLVM. The GPLVM is a non-parametric model, whereas our generative model is in the format of neural network, which makes it possible to learn the generative models together with the posterior estimation in an end-to-end manner.

3. Preliminaries

Let o represent some observation (either the hand pose or the depth map). We wish to estimate a prior $p(o)$ by modeling the generation process of o by sampling some z from an arbitrary low-dimensional distribution $p(z)$ as $p(o) = \int_z p(o|z)p(z)dz$. Fitting $p(o)$ directly is intractable and usually involves expensive inference. We therefore approximate $p(o)$ using two recently developed and very powerful deep generative models: the variational autoencoder (VAE) and the generative adversarial network (GAN).

In the remainder of this section, we provide a brief introduction of the VAE and GAN which we use to model the prior of hand poses and depth maps. Notation-wise, we refer to a given depth map as x and a hand pose as y. We denote the latent variable as z and further distinguish as z_x and z_y indicate the latent depth map and pose respectively when the distinction is necessary. $\bar{x}$ refers to the synthesized depth map from GAN generator and $\bar{y}$ to the reconstructed pose parameter from VAE decoder.

3.1. Pose Variational Autoencoder (Pose VAE)

A VAE comprises an *encoder* which estimates the posterior of latent variable and a *decoder* generates sample from latent variable as follows,

$$z_y \sim \text{Enc}(y) = q(z_y|y),\ \bar{y} \sim \text{Dec}(z_y) = p(y|z_y). \quad (1)$$

The VAE regularizes the encoder by imposing a prior over the latent distribution on $p(z_y)$ while at the same time reconstructing $\bar{y}$ to be as close as possible to the original y. Typically, a Gaussian prior is used, *i.e*. $z_y \sim \mathcal{N}(0, I)$, and is incorporated into the loss as the Kullback-Leibler divergence D_{KL} between the encoded distribution $q(z_y|y)$ and the prior $p(z_y)$. The VAE loss is then the sum of the reconstruction error and latent prior:

$$\mathcal{L}_{\text{vae}} = \mathcal{L}^{pose}_{\text{recons}} + \mathcal{L}_{\text{prior}}, \quad (2)$$

where

$$\mathcal{L}^{pose}_{\text{recons}} = -\mathbb{E}_{q(z_y|y)}[\log p(y|z_y)] \quad (3)$$

and

$$\mathcal{L}_{\text{prior}} = D_{\text{KL}}(q(z_y|y)||p(z_y)). \quad (4)$$

We use the VAE to model a prior distribution on hand pose configurations. The encoder-decoder structure allows us to learn a mapping from high dimensional hand poses to a low-dimensional representation while ensuring a high reconstruction accuracy through the decoder. Furthermore, the constraint on the latent distribution simplifies the learning of a shared latent space of between the depth map and the pose (see Section 4.2 for details).

3.2. Depth Map Generative Adversarial Network (Depth GAN)

A GAN consists of a *generator* and a *discriminator*. The generator synthesizes samples by mapping a random noise sample z_x, from an arbitrary distribution, to a sample in the data space $\bar{x}$.The discriminator tries to distinguish between real data samples x and synthesized samples $\bar{x}$ from the generator. The loss function for the GAN can be formulated as a binary entropy loss as follows:

$$\mathcal{L}_{\text{gan}} = \log(\text{Dis}(x)) + \log(1 - \text{Dis}(\text{Gen}(z_x))), \quad (5)$$

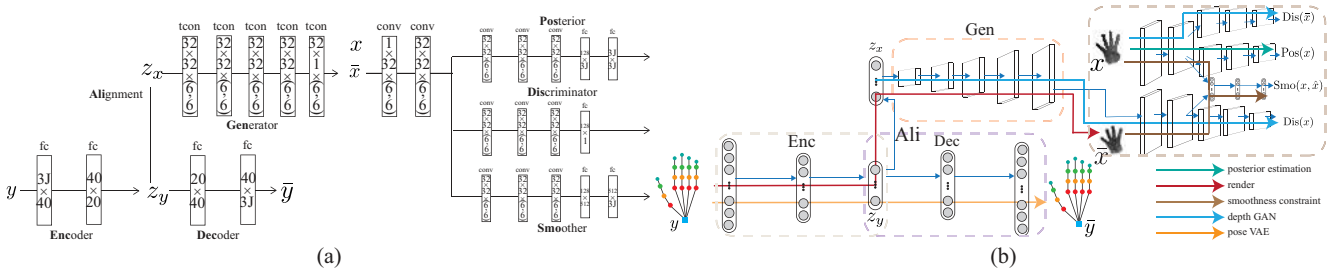


Figure 2. **Overview of the proposed system.** (a) shows the network architecture and a sketch of the variable relationships. **fc** stands for fully connected layers, **tcon** stands for transposed convlutional layers with dialation factor of 2, and **conv** stands for convolutional layers with stride of 2. Numbers inside the boxes denote the parameter size. (b) depicts the data flows within the network used in our work. Arrows with different colours indicate data flows associated with a specific task as shown in the legend. See Section 4.1 for details. The figure is best viewed in colour.

where $\mathrm{Dis}(x)$ is the discriminator output and is a measure of the probability of x being a real data sample. Training alternates between minimizing $\mathcal{L}_{\text{gan}}$ w.r.t. parameters of the generator while maximizing $\mathcal{L}_{\text{gan}}$ w.r.t. parameters of the discriminator. The generator tries to minimize the loss to generate more realistic samples to fool the discriminator while the discriminator tries to maximize the loss.

The GAN does not explicitly model reconstruction loss of the generator; instead, network parameters are updated by back-propagating gradients only from the discriminator. This effectively avoids pixel-wise loss functions that tend to produce overly smoothed results and enables realistic modeling of noise as present in the training set. The GAN can therefore generate depth images with high realism and learn latent representations with linear semantics, *i.e*. simple arithmetic operations in the latent space can result in semantic transformations in the data space [25, 30]. As such, the GAN is well-suited to model the generation process of depth maps and can be used, together with the shared latent space, for synthesizing samples to augment the training corpus. In this work, we adopt a deep convolutional GAN network architecture of [25] and a feature matching strategy [30] for stable and fast-converging training. The noise is sampled from a uniform distribution as $z_x \sim \mathcal{U}(-1, 1)$.

4. Method

4.1. System Overview - Crossing Nets

We formulate hand pose estimation as a statistical learning problem: given a corpus of depth maps, we aim to learn a posterior distribution over the corresponding hand poses. We approach this by combining two generative neural networks, one for pose, and one for depth appearance. First, we pre-train each network separately to capture statistics of the individual domains. We then learn a mapping between the two latent spaces z_x and z_y. The complete network is then further trained end-to-end for the pose estimation task. Fig. 2 gives an overview of our architecture.

In Fig. 2, the blue and yellow routes represent the forward paths of the VAE and the GAN for pose and depth map respectively. The blue route, *i.e*. the render route, links the VAE and the GAN together through the mapping *Ali*. Given any pose, the data is forwarded through the blue route and the network can synthesize a depth map with the corresponding pose. Details of training the render route is given in Section 4.2. The green route estimates the posterior of shared latent variable given the depth map, while the brown route places a smoothness constraint on the generator of GAN. Both the green and the brown routes share the parameters with the discriminator of GAN, with details described in Section 4.3.

Neglecting sensor noise, we assume that there is a one-to-one mapping between the depth map and the hand pose for the free moving hand. As such, we can arbitrarily choose either the pose or the depth map latent space as the reference shared space and then learn a mapping to the other latent space to link the two generative models together. We show how this mapping is learned in Section 4.2.

To prevent from over-fitting, we formulate the posterior estimation as a multi-task learning in which all tasks share the first several convolutional layers. In addition to latent variable regression or *posterior task*, we also consider a *smoothness task* and the *GAN task*. By jointly training the generator and discriminator, as explained in Section 4.3, our method can benefit from the unlabeled samples as well as synthesized samples from the generator.

4.2. Learning the Shared Latent Space

It is not possible with the machinery of the depth GAN alone to estimate the latent variable posterior. As such, we must first learn a mapping from one latent space to the other. We choose the latent space of hand pose parameter as the reference space and learn a mapping to the depth map latent space, *i.e*. $z_y = \mathrm{Ali}(z_x)$. Note that we do not have training pairs of corresponding (z_x, z_y). What we do have, how-

ever, are corresponding pairs (x, y), so it is possible instead to compare observed depth images x with synthesized images $\bar{x}$ that are projected to z_y and then mapped to z_x. As such, we introduce a proxy loss $\mathcal{L}_{\text{recons}}$, based on the reconstruction error of the rendered depth map given a latent input $z_y^{(i)} = \text{Enc}(y^{(i)})$ which is mapped to the GAN latent space:

$$\mathcal{L}_{\text{recons}} = \frac{1}{N}\sum_{i}^{N} \max(\|x^{(i)} - \text{Gen}(\text{Ali}(z_y^{(i)}))\|^2,\ \tau), \quad (6)$$

We model $\text{Ali}(\cdot)$ as a single fully connected neuron with a $\tanh$ activation. The forward pass corresponds to the purple route in Fig. 2. Similar to the golden energy used in [31], we use a clipped mean squared error for our loss function, to remain robust to depth sensor noise. Since the depth map is normalized to $[-1, 1]$, we set the clip threshold $\tau = 1$.

Parameters of the mapping θ_{Ali} are optimized through back-propagation. Since both the pose VAE and the depth GAN are able to learn low-dimensional representations (our z_x and z_y are both 23 dimensions respectively), we are able to fit the alignment and generate realistic samples with very few labeled (x, y) pairs.

After the mapping $\text{Ali}(\cdot)$ is learned, any point in the latent pose space can then be projected into both a hand pose (through the pose VAE) or into a corresponding depth map (through the depth map GAN). We can therefore regard the two latent spaces synonymously as a common shared latent pose. The composite function $\text{Gen}(\text{Ali}(\cdot))$ acts as the new generator for the depth latent space.

Since we impose a Gaussian prior $\mathcal{N}(0, I)$ on z_y, ideally, any random noise sampled from the standard normal distribution can be mapped both to a hand pose or a corresponding depth map. Note that $\text{Ali}(\cdot)$ is implicitly learning a mapping from a normal distribution (z_y) to a uniform distribution (z_x).

4.3. Learning the posterior of shared latent variable

There are three types of data we can use to learn the latent posterior: labeled samples (X_l, Y_l), synthesized samples from random noise $(Z_r, \bar{X}_r = \text{Gen}(\text{Ali}(Z_r)))$ and unlabeled depth maps X_u. In this section, we overload our notation and use the capital letters to indicate mini-batch data matrices of N columns, where each column vector is a sample. For any given matrix A, we use $\|A\|_*$ to indicate the sum of Euclidean norms for each column vector, *i.e.* $\|A\|_* = \sum_{j=1}^{n}(\sum_{i=1}^{m}|a_{ij}|^2)^{\frac{1}{2}}$.

Although it is theoretically sufficient to use only (X_l, Y_l) pairs for learning the posterior, one does not fully exploit the learned priors from the depth GAN. To allow the posterior estimate to benefit from also synthesized and unlabeled samples and therefore increase generalization power, we add two more tasks, *i.e.* a smoothness task and a GAN disambiguation task. All three tasks share the first several convolutional layers, taking synthesized and unlabeled samples as input to exploit the benefits of the depth GAN.

To encourage the generator to synthesize more accurate and realistic samples, parameters θ_{Ali} and θ_{Gen} of the composite generation function $\text{Gen}(\text{Ali}(\cdot))$ are updated together with the aforementioned multitasks. For simplicity, we use *generator* to indicate the composite function of $\text{Gen}(\text{Ali}(\cdot))$ which takes noise from the shared latent space as input and generates a depth map. We use *discriminator* to indicate the multitask learning as a whole, taking depth maps as input. In each iteration, both the generator and the discriminator are updated jointly. The discriminator is updated with labeled, unlabeled and synthesized samples; at the same time, the generator is updated through back-propagated gradients from discriminator. The joint update ensures that the generator synthesizes progressively more realistic samples for the discriminator. We define the joint generator and discriminator loss as

$$\mathcal{L}_G = \mathcal{L}_{\text{recons}} + \mathcal{L}_{\text{smo}} - \mathcal{L}_{\text{gan}}, \quad (7)$$

$$\mathcal{L}_D = \mathcal{L}_{\text{pos}} + \mathcal{L}_{\text{smo}} + \mathcal{L}_{\text{gan}}, \quad (8)$$

where $\mathcal{L}_G$ represents the generator loss and $\mathcal{L}_D$ the discriminator loss.

Smoothness task. To encourage the underlying latent space to be smooth, we define a $\mathcal{L}_{\text{smo}}$ for both the generator and the discriminator. Given two depth maps $x_1,\ x_2$ and their corresponding underlying latent variables $z_1,\ z_2$, the smoothness $\text{smo}(x_1, x_2)$ task takes x_1 and x_2 as input and estimates the corresponding latent variable difference $z_1 - z_2$. The estimated difference is then compared to the actual difference. To make $\mathcal{L}_{\text{smo}}$ regularize both discriminator and generator, we substitute one of the latent-observation pairs with random noise z_r and the corresponding synthesized image $\bar{x}_r$, as indicated by d_{comb} in Eq. 9. At the same time, we want the projected z_l of the labeled sample to synthesize into an image as close as possible to the original so we add the term d_{self}, resulting in the following smoothness loss:

$$\begin{aligned}\mathcal{L}_{\text{smo}} =& d_{\text{comb}} + d_{\text{self}} \\ =& \frac{1}{N}\|\text{smo}(\bar{X}_r, X_l) - (Z_r - Z_l)\|_*^2 \\ &+ \frac{1}{N}\|\text{smo}(\bar{X}_l, X_l)\|_*^2.\end{aligned} \quad (9)$$

Here, X_l is a set of labeled depth maps, $Z_l = \text{Enc}(Y_l)$ is their corresponding latent variable and $\bar{X}_l = \text{Gen}(\text{Ali}(Z_l))$ the depth maps reconstructed through the generator. $\bar{X}_l$ is also compared to the depth maps $\bar{X}_r = \text{Gen}(\text{Ali}(Z_r))$, synthesized from a set of random noise vectors Z_r in the latent space. In practice, the $\text{smo}(\cdot, \cdot)$ operation is implemented as a Siamese network as depicted in Fig 2.

GAN task. Although disambiguating real from synthetic samples is not directly linked to posterior estimation, it has been shown in several previous works [50, 22, 18, 25] having such a loss encourages the hidden activations of the discriminator to learn, as the name implies, inherently discriminative features without additional supervision. We therefore add the following GAN loss term

$$\mathcal{L}_{\text{gan}} = \frac{1}{N} \| \log(\text{Dis}(X)) + \log(1 - \text{Dis}(\text{Gen}(Z))) \|_*^2, \tag{10}$$

where $X = X_l \cup X_u$ is the union of labeled and unlabeled depth maps and $Z = Z_l \cup Z_r$ is the union of synthesized depth maps from latent variables of labeled samples and randomly sampled ones from prior distribution.

Posterior task. Given an input depth map, we formulate a loss for the shared latent variable posterior as

$$\mathcal{L}_{\text{pos}} = \frac{1}{N} \| \text{pos}(X_l) - Z_l \|_*^2, \tag{11}$$

where $\text{pos}(X)$ maps the training set of depth maps X to the corresponding shared latent variable vector Z. Z_l is the set of target positions in the latent space, as obtained by the VAE.

Multi-task Training. We additively combine the three loss functions into a single loss function, using equal weights. In each training iteration, both the generator and the discriminator network parameters are updated once. The detailed training procedure is shown in Algorithm 1.

Algorithm 1 Training the posterior via multitask learning

$\theta_{\text{Ali}}, \theta_{\text{Gen}}, \theta_{\text{Dis}} \leftarrow$ initialized through pretraining
$\theta_{\text{smo}}, \theta_{\text{pos}} \leftarrow$ randomly initialized
$\theta_G := \theta_{\text{Ali}} \cup \theta_{\text{Gen}}$
$\theta_D := \theta_{\text{smo}} \cup \theta_{\text{pos}} \cup \theta_{\text{Dis}}$
for number of training epoch **do**
  $X_l, Y_l \leftarrow$ random minibatch labeled pairs
  $X_u \leftarrow$ random minibatch unlabeled depth map
  $Z_r \leftarrow$ random noises sampled from $p(z)$
  $Z_l, \bar{X}_l, \bar{X}_r \leftarrow \text{Enc}(X_l), \text{Gen}(\text{Ali}(Z_l)), \text{Gen}(\text{Ali}(Z_r))$
  $X_1, X_2, Z_1, Z_2 \leftarrow$ random equal split of X and Z
  $X, Z \leftarrow X_l \cup X_u, Z_l \cup Z_r$
  $d_{\text{comb}} := \frac{1}{N} \| \text{smo}(\bar{X}_r, X_l) - (Z_r - Z_l) \|_*^2$
  $d_{\text{self}} := \frac{1}{N} \| \text{smo}(\bar{X}_l, X_l) \|_*^2$
  $\mathcal{L}_{\text{smo}} \leftarrow d_{comb} + d_{self}$
  $\mathcal{L}_{\text{recons}} \leftarrow \| \max(\|X_l - \bar{X}\|, \tau) \|_*^2$
  $\mathcal{L}_{\text{pos}} \leftarrow \| \text{pos}(X_l) - Z_l \|_*^2$
  $\mathcal{L}_{\text{gan}} \leftarrow \frac{1}{N} \| \log(\text{Dis}(X)) + \log(1 - \text{Dis}(\text{Gen}(Z))) \|_*^2$
  $\theta_D \leftarrow \theta_D - \nabla_{\theta_D} (\mathcal{L}_{\text{pos}} + \mathcal{L}_{\text{smo}} - \mathcal{L}_{\text{gan}})$
  $\theta_G \leftarrow \theta_G - \nabla_{\theta_G} (\mathcal{L}_{\text{recons}} + \mathcal{L}_{\text{smo}} + \mathcal{L}_{\text{gan}})$
end for

4.4. Implementation details

The first 2 convolutional layers of the discriminator network is shared by the three tasks (smoothness task, GAN task and posterior estimation). To stabilize training, we use batch normalization on every hidden layer.Instead of sampling noise from the prior distribution, we generate random noise as the convex combination with random weights from the labeled latent variables. We use the Adam [12] method to update network parameters. To make the generator and discriminator more robust, we injected random Gaussian noise with 0.05 standard deviation to the latent variable after VAE encoder $\text{Enc}(\cdot)$ during training. We set the learning rate as 0.001 and train the complete network for 100 epochs. It takes about 10 hours for training with around 70k samples on one Nvidia TITAN X GPU.

5. Experiments

We performed experiments on 3 publicly available datasets. As each dataset has its own set of challenges, we briefly summarize their characteristics in Table 1. NYU is quite noisy and has a wide range of poses with continuous movements, while MSRA is limited to 17 gestures but has many viewpoint changes. ICVL has large discrepancies between training and testing; test sequences are with fast and abrupt finger movements whereas training sequences have continuous palm movement with little finger movement.

While we estimate all 36 annotated joints on NYU, we only evaluate on a subset of 14 joints as in [46, 20, 21] to make a fair comparison.

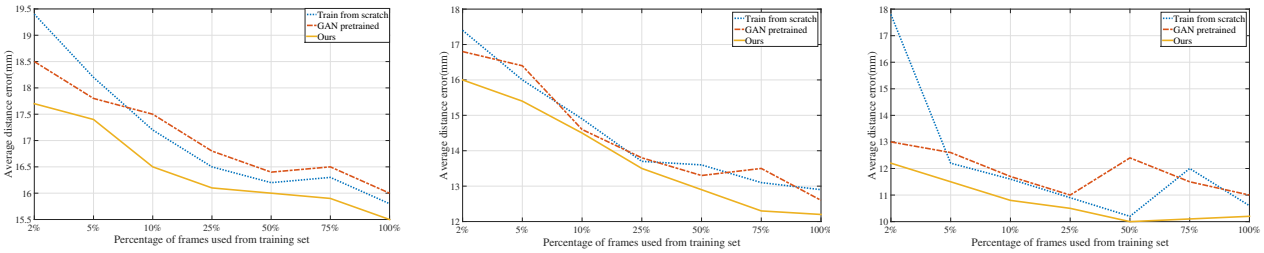
We quantitatively evaluate our method with two metrics: mean joint error (in mm) averaged over all joints and all frames, and percentage of frames in which all joints are below a certain threshold [43]. Qualitative results are shown in Fig. 5 for estimation results and Fig. 1 for the synthesized images from the neural network. We encourage the reader to watch the supplementary videos for a closer qualitative look. The networks were implemented with the Theano package[44]; on an Intel 3.40 GHz i7 machine, the average run time is 11ṁs per image (90.9ḞPS).

5.1. Semi-supervised learning

To explore how our method performs in the semi-supervised setting, we uniformly sample $m\%$ of frames from the training set as labeled data and use the remaining frames unlabeled. We then vary m from 2% to 100% and evaluate the mean joint error averaged over all joints and all frames. We compare against two baseline posterior estimation methods: one network trained from scratch (using randomly initialized parameters), and one network where

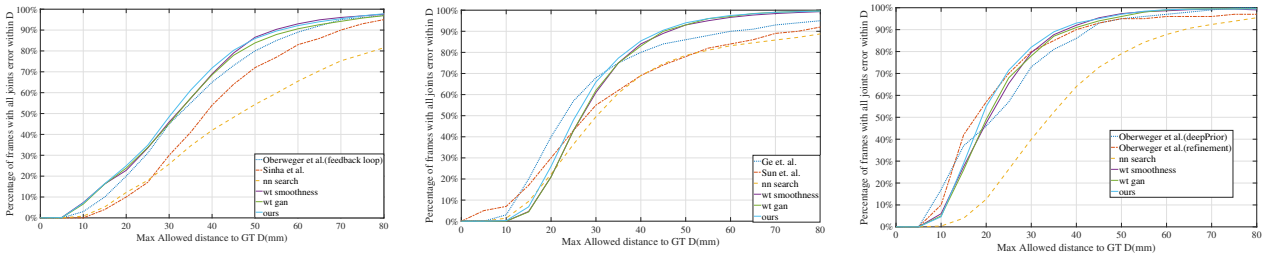
Dataset	Depth Sensor	Train/Test	Noise
NYU [46]	PrimeSense	72.7k / 8.2k	high
MSRA [36]	Intel RealSense	76.5k, 9 users / leave-user-out	low
ICVL [39]	Intel RealSense	20k (160k) / 1.6k	low

Table 1. **Hand pose estimation benchmarks.**



(a) Comparing on NYU (b) Comparing on MSRA (c) Comparing on ICVL

Figure 3. **Semi-supervised learning.** Comparison of our approach and two baseline methods when using $m\%$ of frames from the training set as labeled data, and discarting the labels of the other images.



(a) Comparing on NYU (b) Comparing on MSRA (c) Comparing on ICVL

Figure 4. **Comparison of our approach with state-of-the-art methods.** We compare our approach with of previous method on three challenging datasets.

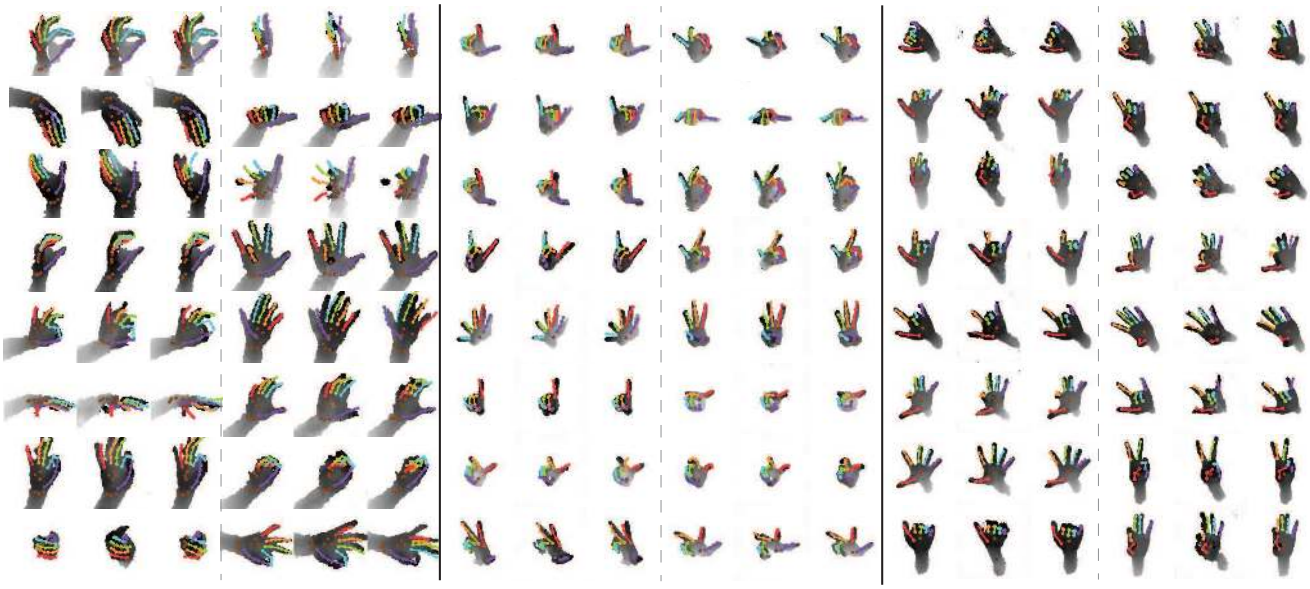


Figure 5. **Qualitative hand pose estimation results.** Left: NYU[46], middle: MSRA[36], right: ICVL[39]. For each sample triplet, left is ground-truth, middle is reconstructed depth map and pose from shared latent space, right is estimated result.

the first two convolutional layers are initialized with parameters from a GAN pretrained on the entire training set.

Unsurprisingly, when $m=2\%$, both the GAN-pretrained baseline and our semi-supervised setup achieve better results than training from scratch. This validates that our depth GAN is effective at learning good representations in an unsupervised way. However, the GAN-pretrained method does not give better results than training from scratch when $m \geq 5\%$. A more unexpected finding was that using more training samples did not lead to a monotonous decrease in the average joint error on both baselines. We attribute this to two causes. First, the labeled frames are uniformly sampled. Since in all three datasets there is slow continuous movement, there is a high correlation between the frames; a 5% sampling may already cover a large portion of distinct hand poses and more samples does not add substantially more information. Secondly, as we evaluate based on the number of training epochs, having more training samples effectively results in more gradient updates and may lead the network to over-fitting. Nevertheless, our method always outperforms the two baselines, showing that using synthesized and unlabeled samples does help with network generalization and preventing overfitting.

5.2. Contribution of multi-task learning

The comparison against the baselines described in Section 5.1 demonstrates that our multi-task learning outperforms direct posterior estimation, both in a semi-supervised and fully supervised setting. To investigate the independent contributions of each energy term in detail, we introduce two more baselines: one trained without the smoothness loss $\mathcal{L}_{\text{smo}}$ and another one without the GAN loss $\mathcal{L}_{\text{gan}}$. The results (plotted as solid lines in Fig. 4) evince that our multi-task approach consistently outperforms both baselines, validating the effectiveness of $\mathcal{L}_{\text{smo}}$ and $\mathcal{L}_{\text{gan}}$ terms.

5.3. Comparison with State-of-the-Art

We compare the accuracy of our method with 6 previous state-of-art methods. In general, our results show that our method is either on par with competing methods or even outperforming them. Compared to hierarchical methods[36, 20, 9], our results are slightly worse at low error thresholds. This demonstrates a general pattern: holistic methods that estimate the hand as a whole tend to be more robust but not very accurate at estimating the finger pose. Hierarchical methods on the other hand estimate the finger pose conditioned on the estimated palm pose and are therefore more accurate, but are also sensitive to the noisy estimation of palm pose. Meanwhile, inspired by [28, 37] we also compare against a nearest neighbour searching based baseline(indicated as nn-search in Fig. 4), in which PCA is used reduce the input depth map into a 512 dimensional feature vector followed by nearest neighbour searching. Given the training and testing samples are similar, the nn-search baseline works reasonably well as on MSRA and vice versa on NYU and ICVL.

On NYU, we compare with Sinha *et al.* [32] and Oberweger *et al.*(feedback loop) [21]. As shown in Fig. 4 (a), we outperform [32, 21] by a large margin.

On MSRA, we compare with Ge *et al.* [9] and Sun *et al.* [36]. Since our approach is holistic, it is not as accurate as the hierarchical methods of [9, 36] on error threshold from 10-30mm. However, we outperform these two when the error threshold is larger than 35mm, which we attribute to our method being more robust to large viewpoint changes.

On ICVL, we compare with thet two variations (deepPrior) and (refinement) of Oberweger *et al.*[20]. We outperform (deepPrior) when the error threshold is $\geq$ 20mm with a large margin. Compared to the much more sophisticated (refinement) variation, which refines the estimate of each joint via a cascaded network, our method is better by 2% with error thresholds when error threshold is $\geq$ 30mm.

6. Conclusion

In this paper, we propose a hand pose estimation method by estimating the posterior of the shared latent space of depth map and hand pose parameters. We formulate the problem as a multi-task learning problem on a network architecture that crosses two deep generative networks: a variational auto encoder (VAE) for hand poses and a generative adversarial network (GAN) for modeling the distributions of depth images. By learning a mapping between the two latent spaces, we can train the complete network end-to-end. In our experiments we demonstrate that this has a number of advantages: we can exploit the generalization properties of the GAN as well as the pose constraints implicitly learned by the VAE to improve discriminative pose estimation. Moreover, our architecture naturally allows for learning from unlabeled data, which is very valuable for the problem of hand pose estimation, where annotated training data is sparse. Our approach therefore extends the semi-supervised setting of GAN to making real valued structured predictions. We evaluated our method on 3 publicly available datasets and demonstrate that our approach consistently achieves better performance over previous state-of-art methods. Due to a very efficient design of the discriminator network our approach is capable of running in real-time on the CPU.

Acknowledgment The authors gratefully acknowledge support by armasuisse, KTI project with Faswhell and Chinese Scholarship Council.

References

[1] M. Bianchi, N. Carbonaro, E. Battaglia, F. Lorussi, A. Bicchi, D. D. Rossi, and A. Tognetti. Exploiting hand kinematic synergies and wearable under-sensing for hand functional grasp recognition. In *Wireless Mobile Communication and Healthcare*, 2014.

[2] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *arXiv preprint arXiv:1606.03657*, 2016.

[3] C. Choi, A. Sinha, J. H. Choi, S. Jang, and K. Ramani. A collaborative filtering approach to Real-Time hand pose estimation: Supplementary material. In *ICCV*, 2015.

[4] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. 2015.

[5] M. Ding and G. Fan. Multilayer joint gait-pose manifolds for human gait motion modeling. *IEEE Transactions on Cybernetics*, 2015.

[6] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[7] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[8] C. H. Ek, P. H. S. Torr, and N. D. Lawrence. Gaussian process latent variable models for human pose estimation. In *International workshop on machine learning for multimodal interaction*. 2007.

[9] L. Ge, H. Liang, J. Yuan, and D. Thalmann. Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In *CVPR*, 2016.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

[11] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.

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

[13] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *NIPS*, 2014.

[14] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[15] J. Lin, Y. Wu, and T. S. Huang. Modeling the constraints of human hand motion. In *Human Motion, 2000. Proceedings. Workshop on*, 2000.

[16] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. *arXiv preprint arXiv:1606.07536*, 2016.

[17] V. Madhavan, P. Cerles, and N. Desai. Image generation from captions using Dual-Loss generative adversarial networks. *vashishtm.net*, 2016.

[18] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[19] R. Navaratnam, A. W. Fitzgibbon, and R. Cipolla. The joint manifold model for semi-supervised multi-valued regression. In *ICCV*, 2007.

[20] M. Oberweger, P. Wohlhart, and V. Lepetit. Hands deep in deep learning for hand pose estimation. *arXiv preprint arXiv:1502.06807*, 2015.

[21] M. Oberweger, P. Wohlhart, and V. Lepetit. Training a feedback loop for hand pose estimation. In *ICCV*, 2015.

[22] A. Odena. Semi-Supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016.

[23] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *BMVC*, 2011.

[24] C. Qian, Q. Chen, S. Xiao, W. Yichen, T. Xiaoou, and S. Jian. Realtime and robust hand tracking from depth. In *CVPR*, 2014.

[25] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[26] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

[27] D. J. Rezende, S. Eslami, S. Mohamed, P. Battaglia, M. Jaderberg, and N. Heess. Unsupervised learning of 3d structure from images. *arXiv preprint arXiv:1607.00662*, 2016.

[28] G. Rogez, J. S. Supancic, and D. Ramanan. Understanding everyday hands in action from RGB-D images. In *ICCV*, 2015.

[29] A. G. Rouse and M. H. Schieber. Advancing brain-machine interfaces: moving beyond linear state space models. *Front. Syst. Neurosci.*, 2015.

[30] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*, 2016.

[31] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. K. C. R. I. Leichter, A. V. Y. Wei, D. F. P. K. E. Krupka, A. Fitzgibbon, and S. Izadi. Accurate, robust, and flexible real-time hand tracking. In *ACM Conference on Human Factors in Computing Systems*, 2015.

[32] A. Sinha, C. Choi, and K. Ramani. DeepHand: Robust hand pose estimation by completing a matrix imputed with deep features. In *CVPR*, 2016.

[33] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In *NIPS*. 2015.

[34] J. T. Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390*, 2015.

[35] S. Sridhar, F. Mueller, M. Zollhoefer, D. Casas, A. Oulasvirta, and C. Theobalt. Real-time joint tracking of a hand manipulating an object from rgb-d input. In *ECCV*, 2016.

[36] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun. Cascaded hand pose regression. In *CVPR*, 2015.

[37] J. S. Supancic, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan. Depth-Based hand pose estimation: Data, methods, and challenges. In *ICCV*, 2015.

[38] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016.

[39] D. Tang, H. J. Chang, A. Tejani, and T.-K. Kim. Latent regression forest: Structured estimation of 3D articulated hand posture. In *CVPR*, 2014.

[40] D. Tang, J. Taylor, P. Kohli, C. Keskin, T.-K. Kim, and J. Shotton. Opening the black box: Hierarchical sampling optimization for estimating human hand pose. In *ICCV*, 2015.

[41] D. Tang, T.-H. Yu, and T.-K. Kim. Real-Time articulated hand pose estimation using semi-supervised transductive regression forests. In *ICCV*, 2013.

[42] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, B. Luff, et al. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Transactions on Graphics (TOG)*, 2016.

[43] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *CVPR*, 2012.

[44] T. T. D. Team, R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov, et al. Theano: A python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*, 2016.

[45] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua. Structured prediction of 3d human pose with deep neural networks. *arXiv preprint arXiv:1605.05180*, 2016.

[46] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (TOG)*, 2014.

[47] D. Tzionas, L. Ballan, A. Srikantha, P. Aponte, M. Pollefeys, and J. Gall. Capturing hands in action using discriminative salient points and physics simulation. *International Journal of Computer Vision (IJCV)*, 2016.

[48] T. Vodopivec, V. Lepetit, and P. Peer. Fine hand segmentation using convolutional neural networks. *arXiv preprint arXiv:1608.07454*, 2016.

[49] C. Wan, A. Yao, and L. Van Gool. Hand pose estimation from local surface normals. In *ECCV*, 2016.

[50] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In *ECCV*, 2016.

[51] Q. Ye, S. Yuan, and T.-K. Kim. Spatial attention deep net with partial PSO for hierarchical hybrid hand pose estimation. In *ECCV*, 2016.

[52] X. Zhang, M. Ding, and G. Fan. Video-based human walking estimation by using joint gait and pose manifolds. *IEEE Trans. Circuits Syst. Video Technol.*, 2016.