DISCO Nets: DISsimilarity COefficient Networks

Diane Bouchacourt
University of Oxford
diane@robots.ox.ac.uk

M. Pawan Kumar
University of Oxford
pawan@robots.ox.ac.uk

Sebastian Nowozin
Microsoft Research Cambridge
sebastian.nowozin@microsoft.com

Abstract

We present a new type of probabilistic model which we call DISsimilarity COefficient Networks (DISCO Nets). DISCO Nets allow us to efficiently sample from a posterior distribution parametrised by a neural network. During training, DISCO Nets are learned by minimising the dissimilarity coefficient between the true distribution and the estimated distribution. This allows us to tailor the training to the loss related to the task at hand. We empirically show that (i) by modeling uncertainty on the output value, DISCO Nets outperform equivalent non-probabilistic predictive networks and (ii) DISCO Nets accurately model the uncertainty of the output, outperforming existing probabilistic models based on deep neural networks.

1 Introduction

We are interested in the class of problems that require the prediction of a structured output $\boldsymbol{y} \in \mathcal{Y}$ given an input $\boldsymbol{x} \in \mathcal{X}$. Complex applications often have large uncertainty on the correct value of $\boldsymbol{y}$. For example, consider the task of hand pose estimation from depth images, where one wants to accurately estimate the pose $\boldsymbol{y}$ of a hand given a depth image $\boldsymbol{x}$. The depth image often has some occlusions and missing depth values and this results in some uncertainty on the pose of the hand. It is, therefore, natural to use probabilistic models that are capable of representing this uncertainty. Often, the capacity of the model is restricted and cannot represent the true distribution perfectly. In this case, the choice of the learning objective influences final performance. Similar to Lacoste-Julien et al. [12], we argue that the learning objective should be tailored to the evaluation loss in order to obtain the best performance with respect to this loss. In details, we denote by Δ_{training} the loss function employed during model training, and by Δ_{task} the loss employed to evaluate the model's performance.

We present a simple example to illustrate the point made above. We consider a data distribution that is a mixture of two bidimensional Gaussians. We now consider two models to capture the data probability distribution. Each model is able to represent a bidimensional Gaussian distribution with diagonal covariance parametrised by $(\mu_1, \mu_2, \sigma_1, \sigma_2)$. In this case, neither of the models will be able to recover the true data distribution since they do not have the ability to represent a mixture of Gaussians. In other words, we cannot avoid model error, similarly to the real data scenario. Each model uses its own training loss Δ_{training}. Model A employs a loss that emphasises on the first dimension of the data, specified for $\boldsymbol{x} = (x_1, x_2), \boldsymbol{x'} = (x'_1, x'_2) \in \mathbb{R}^2$ by $\Delta_A(\boldsymbol{x} - \boldsymbol{x'}) = (10 \times (x_1 - x'_1)^2 + 0.1 \times (x_2 - x'_2)^2)^{1/2}$. Model B does the opposite and employs the loss function $\Delta_B(\boldsymbol{x} - \boldsymbol{x'}) = (0.1 \times (x_1 - x'_1)^2 + 10 \times (x_2 - x'_2)^2)^{1/2}$. Each model performs a grid search over the best parameters values for $(\mu_1, \mu_2, \sigma_1, \sigma_2)$. Figure 1 shows the contours of the Mixture of Gaussians distribution of the data (in black), and the contour of the Gaussian fitted by each model (in red and green). Detailed setting of this example is available in the supplementary material.

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

Table 1: $\Delta_{task}\pm$ *SEM (standard error of the mean) with respect to* $\Delta_{training}$ *employed. Evaluation is done the test set.*

Δ_{training} \ Δ_{task}	Δ_A	Δ_B
Δ_A	$\mathbf{11.6 \pm 0.287}$	13.7 ± 0.331
Δ_B	12.1 ± 0.305	$\mathbf{11.0 \pm 0.257}$

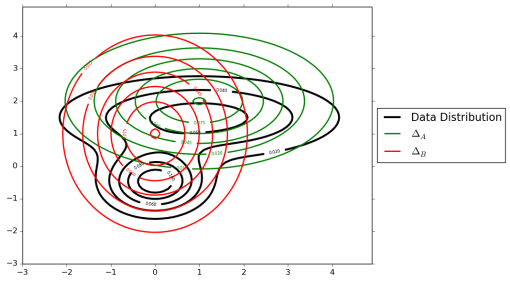


Figure 1: *Contour lines of the Gaussian distribution fitted by each model on the Mixture of Gaussians data distribution. Best viewed in color.*

As expected, the fitted Gaussian distributions differ according to Δ_{training} employed. Table 1 shows that the loss on the test set, evaluated with Δ_{task}, is minimised if $\Delta_{\text{training}} = \Delta_{\text{task}}$. This simple example illustrates the advantage to being able to tailor the model's training objective function to have $\Delta_{\text{training}} = \Delta_{\text{task}}$. This is in contrast to the commonly employed learning objectives we present in Section 2, that are agnostic of the evaluation loss.

In order to alleviate the aforementioned deficiency of the state-of-the-art, we introduce DISCO Nets, a new class of probabilistic model. DISCO Nets represent P, the true posterior distribution of the data, with a distribution Q parametrised by a neural network. We design a learning objective based on a dissimilarity coefficient between P and Q. The dissimilarity coefficient we employ was first introduced by Rao [23] and is defined for any non-negative symmetric loss function. Thus, any such loss can be incorporated in our setting, allowing the user to tailor DISCO Nets to his or her needs. Finally, contrarily to existing probabilistic models presented in Section 2, DISCO Nets do not require any specific architecture or training procedure, making them an efficient and easy-to-use class of model.

2 Related Work

Deep neural networks, and in particular, Convolutional Neural Networks (CNNs) are comprised of several convolutional layers, followed by one or more fully connected (dense) layers, interleaved by non-linear function(s) and (optionally) pooling. Recent probabilistic models use CNNs to represent non-linear functions of the data. We observe that such models separate into two types. The first type of model does not explicitly compute the probability distribution of interest. Rather, these models allow the user to sample from this distribution by feeding the CNN with some noise $\boldsymbol{z}$. Among such models, Generative Adversarial Networks (GAN) presented in Goodfellow et al. [7] are very popular and have been used in several computer vision applications, for example in Denton et al. [1], Radford et al. [22], Springenberg [25] and Yan et al. [28]. A GAN model consists of two networks, simultaneously trained in an adversarial manner. A generative model, referred as the *Generator* G, is trained to replicate the data from noise, while an adversarial discriminative model, referred as the *Discriminator* D, is trained to identify whether a sample comes from the true data or from G. The GAN training objective is based on a minimax game between the two networks and approximately optimizes a Jensen-Shannon divergence. However, as mentioned in Goodfellow et al. [7] and Radford et al. [22], GAN models require very careful design of the networks' architecture. Their training procedure is tedious and tends to oscillate. GAN models have been generalized to conditional GAN (cGAN) in Mirza and Osindero [16], where some additional input information can be fed to the *Generator* and the *Discriminator*. For example in Mirza and Osindero [16] a cGAN model generates tags corresponding to an image. Gauthier [4] applies cGAN to face generation. Reed et al. [24] propose to generate images of flowers with a cGAN model, where the conditional information is a word description of the flower to generate[1]. While the application of cGAN is very promising, little quantitative evaluation has been done. Furthermore, cGAN models suffer from the same difficulties we mentioned for GAN. Another line of work has developed towards the use of statistical hypothesis testing to learn probabilistic models. In Dziugaite et al. [2] and Li et al. [14], the authors propose to train generative deep networks with an objective function based on the Maximum Mean Discrepancy (MMD) criterion. The MMD method (see Gretton et al. [8, 9]) is a statistical hypothesis test assessing if two probabilistic distributions are similar. As mentioned in Dziugaite et al. [2], the MMD test can been seen as playing the role of an adversary.

[1] At the time writing, we do not have access to the full paper of Reed et al. [24] and therefore cannot take advantage of this work in our experimental comparison.

The second type of model approximates intractable posterior distributions with use of variational inference. The Variational Auto-Encoders (VAE) presented in Kingma and Welling [10] is composed of a *probabilistic encoder* and a *probabilistic decoder*. The *probabilistic encoder* is fed with the input $\boldsymbol{x} \in \mathcal{X}$ and produces a posterior distribution $P(\boldsymbol{z}|\boldsymbol{x})$ over the possible values of noise $\boldsymbol{z}$ that could have generated $\boldsymbol{x}$. The *probabilistic decoder* learns to map the noise $\boldsymbol{z}$ back to the data space $\mathcal{X}$. The training of VAE uses an objective function based on a Kullback-Leibler Divergence. VAE and GAN models have been combined in Makhzani et al. [15], where the authors propose to regularise autoencoders with an adversarial network. The adversarial network ensures that the posterior distribution $P(\boldsymbol{z}|\boldsymbol{x})$ matches an arbitrary prior $P(\boldsymbol{z})$.

In hand pose estimation, imagine the user wants to obtain accurate positions of the thumb and the index finger but does not need accurate locations of the other fingers. The task loss Δ_{task} might be based on a weighted L2-norm between the predicted and the ground-truth poses, with high weights on the thumb and the index. Existing probabilistic models cannot be tailored to task-specific losses and we propose the DISsimilarity COefficient Networks (DISCO Nets) to alleviate this deficiency.

3 DISCO Nets

We begin the description of our model by specifying how it can be used to generate samples from the posterior distribution, and how the samples can in turn be employed to provide a pointwise estimate. In the subsequent subsection, we describe how to estimate the parameters of the model.

3.1 Prediction

Sampling. A DISCO Net consists of several convolutional and dense layers (interleaved by non-linear function(s) and possibly pooling) and takes as input a pair $(\boldsymbol{x}, \boldsymbol{z}) \in \mathcal{X} \times \mathcal{Z}$, where $\boldsymbol{x}$ is input data and $\boldsymbol{z}$ is some random noise. Given one pair $(\boldsymbol{x}, \boldsymbol{z})$, the DISCO Net produces a value for the output $\boldsymbol{y}$. In the example of hand pose estimation, the input depth image $\boldsymbol{x}$ is fed to the convolutional layers. The output of the last convolutional layer is flattened and concatenated with a noise sample $\boldsymbol{z}$. The resulting vector is fed to several dense layers, and the last dense layer outputs a pose $\boldsymbol{y}$. From a single depth image $\boldsymbol{x}$, by using different noise samples, the DISCO Net produces different pose candidates for the depth image. This process is illustrated in Figure 2. Importantly, DISCO Nets are flexible in the choice of the architecture. For example, the noise could be concatenated at any stage of the network, including at the start.

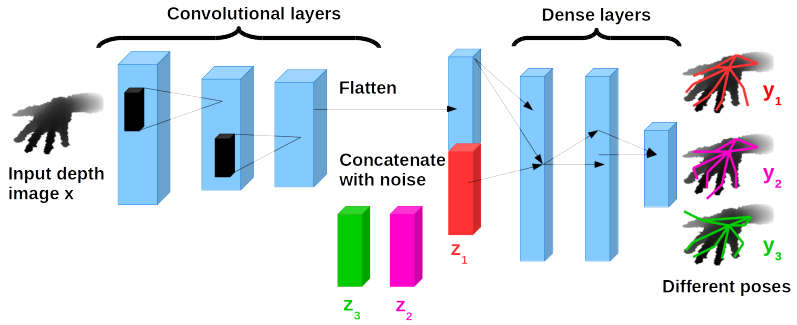


Figure 2: *For a single depth image $\boldsymbol{x}$, using 3 different noise samples $(\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3)$, DISCO Nets output 3 different candidate poses $(\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3)$ (shown superimposed on the depth image). The depth image is from the NYU Hand Pose Dataset of Tompson et al. [27], preprocessed as in Oberweger et al. [17]. Best viewed in color.*

We denote Q the distribution that is parametrised by the DISCO Net's neural network. For a given input $\boldsymbol{x}$, DISCO Nets provide the user with samples $\boldsymbol{y}$ drawn from $Q(\boldsymbol{y}|\boldsymbol{x})$ without requiring the expensive computation of the (often intractable) partition function. In the remainder of the paper we consider $\boldsymbol{x} \in \mathbb{R}^{d_x}$, $\boldsymbol{y} \in \mathbb{R}^{d_y}$ and $\boldsymbol{z} \in \mathbb{R}^{d_z}$.

Pointwise Prediction. In order to obtain a single prediction $\boldsymbol{y}$ for a given input $\boldsymbol{x}$, DISCO Nets use the principle of Maximum Expected Utility (MEU), similarly to Premachandran et al. [21]. The prediction $\boldsymbol{y}_{\Delta_{\text{task}}}$ maximises the expected utility, or rather minimises the expected task-specific loss Δ_{task}, estimated using the sampled candidates. Formally, the prediction is made as follows:

$$\boldsymbol{y}_{\Delta_{\text{task}}} = \underset{k\in[1,K]}{\operatorname{argmax}} \operatorname{EU}(\boldsymbol{y}_k) = \underset{k\in[1,K]}{\operatorname{argmin}} \sum_{k'=1}^{K} \Delta_{\text{task}}(\boldsymbol{y}_k, \boldsymbol{y}'_k) \tag{1}$$

where $(\boldsymbol{y_1}, ..., \boldsymbol{y}_K)$ are the candidate outputs sampled for the single input $\boldsymbol{x}$. Details on the MEU method are in the supplementary material.

3.2 Learning DISCO Nets

Objective Function. We want DISCO Nets to accurately model the true probability $P(\boldsymbol{y}|\boldsymbol{x})$ via $Q(\boldsymbol{y}|\boldsymbol{x})$. In other words, $Q(\boldsymbol{y}|\boldsymbol{x})$ should be as similar as possible to $P(\boldsymbol{y}|\boldsymbol{x})$. This similarity is evaluated with respect to the loss specific to the task at hand. Given any non-negative symmetric loss function between two outputs $\Delta(\boldsymbol{y}, \boldsymbol{y'})$ with $(\boldsymbol{y}, \boldsymbol{y'}) \in \mathcal{Y} \times \mathcal{Y}$, we employ a diversity coefficient that is the expected loss between two samples drawn randomly from the two distributions. Formally, the diversity coefficient is defined as:

$$\text{DIV}_\Delta(P, Q, D) = E_{\boldsymbol{x}\sim D(\boldsymbol{x})}[E_{\boldsymbol{y}\sim P(\boldsymbol{y}|\boldsymbol{x})}[E_{\boldsymbol{y'}\sim Q(\boldsymbol{y'}|\boldsymbol{x})}[\Delta(\boldsymbol{y}, \boldsymbol{y'})]]] \tag{2}$$

Intuitively, we should minimise $\text{DIV}_\Delta(P, Q, D)$ so that $Q(\boldsymbol{y}|\boldsymbol{x})$ is as similar as possible to $P(\boldsymbol{y}|\boldsymbol{x})$. However there is uncertainty on the output $\boldsymbol{y}$ to predict for a given $\boldsymbol{x}$. In other words, $P(\boldsymbol{y}|\boldsymbol{x})$ is diverse and $Q(\boldsymbol{y}|\boldsymbol{x})$ should be diverse as well. Thus we encourage $Q(\boldsymbol{y}|\boldsymbol{x})$ to provide sample outputs, for a given $\boldsymbol{x}$, that are diverse by minimising the following dissimilarity coefficient:

$$\text{DISC}_\Delta(P, Q, D) = \text{DIV}_\Delta(P, Q, D) - \gamma\text{DIV}_\Delta(Q, Q, D) - (1-\gamma)\text{DIV}_\Delta(P, P, D) \tag{3}$$

with $\gamma \in [0, 1]$. The dissimilarity $\text{DISC}_\Delta(P, Q, D)$ is the difference between the diversity between P and Q, and an affine combination of the diversity of each distribution, given $\boldsymbol{x} \sim D(\boldsymbol{x})$. These coefficients were introduced by Rao [23] with $\gamma = 1/2$ and used for latent variable models by Kumar et al. [11]. We do not need to consider the term $\text{DIV}_\Delta(P, P, D)$ as it is a constant in our problem, and thus the DISCO Nets objective function is defined as follows:

$$F = \text{DIV}_\Delta(P, Q, D) - \gamma\text{DIV}_\Delta(Q, Q, D) \tag{4}$$

When minimising F, the term $\gamma\text{DIV}_\Delta(Q, Q, D)$ encourages $Q(\boldsymbol{y}|\boldsymbol{x})$ to be diverse. The value of γ balances between the two goals of $Q(\boldsymbol{y}|\boldsymbol{x})$ that are providing accurate outputs while being diverse.

Optimisation. Let us consider a training dataset composed of N examples input-output pairs $D = \{(\boldsymbol{x}_n, \boldsymbol{y}_n), n = 1..N\}$. In order to train DISCO Nets, we need to compute the objective function of equation (4). We do not have knowledge of the true probability distributions $P(\boldsymbol{y}, \boldsymbol{x})$ and $P(\boldsymbol{x})$. To overcome this deficiency, we construct estimators of each diversity term $\text{DIV}_\Delta(P, Q)$ and $\text{DIV}_\Delta(Q, Q)$. First, we take an empirical distribution of the data, that is, taking ground-truth pairs $(\boldsymbol{x}_n, \boldsymbol{y}_n)$. We then estimate each distribution $Q(\boldsymbol{y}|\boldsymbol{x}_n)$ by sampling K outputs from our model for each $\boldsymbol{x}_n$. This gives us an unbiased estimate of each diversity term, defined as:

$$\begin{aligned}
\widehat{\text{DIV}}_\Delta(P, Q, D) &= \frac{1}{N}\sum_{n=1}^{N}\frac{1}{K}\sum_{k=1}^{K}\Delta(\boldsymbol{y}_n, G(\boldsymbol{z}_k, \boldsymbol{x}_n; \boldsymbol{\theta})) \\
\widehat{\text{DIV}}_\Delta(Q, Q, D) &= \frac{1}{N}\sum_{n=1}^{N}\frac{1}{K(K-1)}\sum_{k=1}^{K}\sum_{k'=1, k'\neq k}^{K}\Delta(G(\boldsymbol{z}_k, \boldsymbol{x}_n; \boldsymbol{\theta}), G(\boldsymbol{z}_{k'}, \boldsymbol{x}_n; \boldsymbol{\theta}))
\end{aligned} \tag{5}$$

We have an unbiased estimate of the DISCO Nets' objective function of equation (4):

$$\widehat{F}(\Delta, \boldsymbol{\theta}) = \widehat{\text{DIV}}_\Delta(P, Q, D) - \gamma\widehat{\text{DIV}}_\Delta(Q, Q, D) \tag{6}$$

where $\boldsymbol{y}_k = G(\boldsymbol{z}_k, \boldsymbol{x}_n; \boldsymbol{\theta})$ is a candidate output sampled from DISCO Nets for ($\boldsymbol{x}_n$,$\boldsymbol{z}_k$), and $\boldsymbol{\theta}$ are the parameters of DISCO Nets. It is important to note that the second term of equation (6) is summing over k and $k' \neq k$ to have an unbiased estimate, therefore we compute the loss between pairs of different samples $G(\boldsymbol{z}_k, \boldsymbol{x}_n; \boldsymbol{\theta})$ and $G(\boldsymbol{z}_{k'}, \boldsymbol{x}_n; \boldsymbol{\theta})$. The parameters $\boldsymbol{\theta}$ are learned by Gradient Descent. Algorithm 1 shows the training of DISCO Nets. In steps 4 and 5 of Algorithm 1, we draw K random noise vectors $(\boldsymbol{z}_{n,1}, ...\boldsymbol{z}_{n,k})$ per input example $\boldsymbol{x}_n$, and generate K candidate outputs $G(\boldsymbol{z}_{n,k}, \boldsymbol{x}_n; \boldsymbol{\theta})$ per input. This allow us to compute an unbiased estimate of the gradient in step 7. For clarity, in the remainder of the paper we do not explicitely write the parameters $\boldsymbol{\theta}$ and write $G(\boldsymbol{z}_k, \boldsymbol{x}_n)$.

Algorithm 1: DISCO Nets Training algorithm.

for *t=1...T epochs* **do**
  Sample minibatch of b training example pairs $\{(\boldsymbol{x}_1, \boldsymbol{y}_1)...(\boldsymbol{x}_b, \boldsymbol{y}_b)\}$.
  for *n=1...b* **do**
    Sample K random noise vectors $(\boldsymbol{z}_{n,1}, ...\boldsymbol{z}_{n,k})$ for training example $\boldsymbol{x}_n$
    Generate K candidate outputs $G(\boldsymbol{z}_{n,k}, \boldsymbol{x}_n; \boldsymbol{\theta}), k = 1..K$ for training example $\boldsymbol{x}_n$
  end
  Update parameters $\boldsymbol{\theta}^t \leftarrow \boldsymbol{\theta}^{t-1}$ by descending the gradient of equation (6) : $\nabla_{\boldsymbol{\theta}} \widehat{F}(\Delta, \boldsymbol{\theta})$.
end

3.3 Strictly Proper Scoring Rules.

Scoring Rule for Learning. A scoring rule $S(Q, P)$, as defined in Gneiting and Raftery [5], evaluates the quality of a predictive distribution Q with respect to a true distribution P. When using a scoring rule one should ensure that it is proper, which means it is maximised when $P = Q$. A scoring rule is said to be strictly proper if $P = Q$ is the unique maximiser of S. Hence maximising a proper scoring rule ensures that the model aims at predicting relevant forecast. Gneiting and Raftery [5] define score divergences corresponding to a proper scoring rule S:

$$d(Q, P) = S(P, P) - S(Q, P) \tag{7}$$

If S is proper, d is a valid non-negative divergence function, with value 0 if (and only if, in the case of strictly proper) $Q = P$. For example the MMD criterion (see Gretton et al. [8, 9]) mentioned in Section 2 is an example of this type of divergence. In our case, any loss Δ expressed with an universal kernel will define the DISCO Nets' objective function as such divergence (see Zawadzki and Lahaie [29]). For example, by Theorem 5 of Gneiting and Raftery [5], if we take as loss function $\Delta_\beta(\boldsymbol{y}, \boldsymbol{y'}) = ||\boldsymbol{y} - \boldsymbol{y'}||_2^\beta = \sum_{i=1}^{d_y} (|y^i - y'^i|^2)^{\beta/2}$ with $\beta \in [0, 2]$ excluding 0 and 2, our training objective is (the negative of) a strictly proper scoring rule, that is:

$$\widehat{F}(\Delta, \boldsymbol{\theta}) = \frac{1}{N} \sum_{n=1}^{N} \Big[\frac{1}{K} \sum_k ||\boldsymbol{y}_n - G(\boldsymbol{z}_k, \boldsymbol{x}_n)||_2^\beta - \frac{1}{2} \frac{1}{K(K-1)} \sum_k \sum_{k' \neq k} ||G(\boldsymbol{z}_{k'}, \boldsymbol{x}_n) - G(\boldsymbol{z}_k, \boldsymbol{x}_n)||_2^\beta \Big] \tag{8}$$

This score has been referred in the literature as the Energy Score in Gneiting and Raftery [5], Gneiting et al. [6], Pinson and Tastu [19].

By employing a (strictly) proper scoring rule we ensure that our objective function is (only) minimised at the true distribution P, and expect DISCO Nets to generalise better on unseen data. We show below that strictly proper scoring rules are also relevant to assess the quality of the distribution Q captured by the model.

Discriminative power of proper scoring rules. As observed in Fukumizu et al. [3], kernel density estimation (KDE) fails in high dimensional output spaces. Our goal is to compare the quality of the distribution captured between two models, Q_1 and Q_2. In our setting Q_1 better models P than Q_2 according to the scoring rule S and its associated divergence d if $d(Q_1, P) < d(Q_2, P)$. As noted in Pinson and Tastu [19], S being proper does not ensure $d(Q_1, \boldsymbol{y}) < d(Q_2, \boldsymbol{y})$ for all observations $\boldsymbol{y}$ drawn from P. However if the scoring rule is strictly proper scoring rule, this property should be ensured in the neighbourhood of the true distribution.

4 Experiments : Hand Pose Estimation

Given a depth image $\boldsymbol{x}$, which often contains occlusions and missing values, we wish to predict the hand pose $\boldsymbol{y}$. We use the NYU Hand Pose dataset of Tompson et al. [27] to estimate the efficiency of DISCO Nets for this task.

4.1 Experimental Setup

NYU Hand Pose Dataset. The NYU Hand Pose dataset of Tompson et al. [27] contains 8252 testing and 72,757 training frames of captured RGBD data with ground-truth hand pose information. The training set is composed of images of one person whereas the testing set gathers samples from two persons. For each frame, the RGBD data from 3 Kinects is provided: a frontal view and 2 side views. In our experiments we use only the depth data from the frontal view. While the ground truth

contains J = 36 annotated joints, we follow the evaluation protocol of Oberweger et al. [17, 18] and use the same subset of J = 14 joints. We also perform the same data preprocessing as in Oberweger et al. [17, 18], and extract a fixed-size metric cube around the hand from the depth image. We resize the depth values within the cube to a 128×128 patch and normalized them in $[-1, 1]$. Pixels deeper than the back of the cube and missing depth values are both set to a depth of 1.

Methods. We employ loss functions between two outputs of the form of the Energy score (8), that is, $\Delta_{\text{training}} = \Delta_\beta(\boldsymbol{y}, \boldsymbol{y'}) = ||\boldsymbol{y} - \boldsymbol{y'}||_2^\beta$. Our first goal is to assess the advantages of DISCO Nets with respect to non-probabilistic deep networks. One model, referred as DISCO$_{\beta,\gamma}$, is a DISCO Nets probabilistic model, with $\gamma \neq 0$ in the dissimilarity coefficient of equation (6). When taking $\gamma = 0$, noise is injected and the model capacity is the same as DISCO$_{\beta,\gamma\neq 0}$. The model BASE$_\beta$, is a non-probabilistic model, by taking $\gamma = 0$ in the objective function of equation (6) and no noise is concatenated. This corresponds to a classic deep network which for a given input $\boldsymbol{x}$ generates a single output $\boldsymbol{y} = G(\boldsymbol{x})$. Note that we write $G(\boldsymbol{x})$ and not $G(\boldsymbol{z}, \boldsymbol{x})$ since no noise is concatenated. f

Evaluation Metrics. We report classic non-probabilistic metrics for hand pose estimation employed in Oberweger et al. [17, 18] and Taylor et al. [26], that are, the Mean Joint Euclidean Error (MeJEE), the Max Joint Euclidean Error (MaJEE) and the Fraction of Frames within distance (FF). We refer the reader to the supplementary material for detailed expression of these metrics. These metrics use the Euclidean distance between the prediction and the ground-truth and require a single pointwise prediction. This pointwise prediction is chosen with the MEU method among K candidates. We added the probabilistic metric ProbLoss. ProbLoss is defined as in Equation 8 with the Euclidean norm and is the divergence associated with a strictly proper scoring rule. Thus, ProbLoss ranks the ability of the models to represent the true distribution. ProbLoss is computed using K candidate poses for a given depth image. For the non-probabilistic model BASE$_\beta$, only a single pointwise predicted output $\boldsymbol{y}$ is available. We construct the K candidates by adding some Gaussian random noise of mean 0 and diagonal covariance $\Sigma = \sigma \mathbb{1}$, with $\sigma \in \{1\text{mm}, 5\text{mm}, 10\text{mm}\}$ and refer to the model as BASE$_{\beta,\sigma}$. [2]

Loss functions. As we employ standard evaluation metrics based on the Euclidean norm, we train with the Euclidean norm (that is, $\Delta_{\text{training}}(\boldsymbol{y}, \boldsymbol{y'}) = ||\boldsymbol{y} - \boldsymbol{y'}||_2^\beta$ with $\beta = 1$). When $\gamma = \frac{1}{2}$ our objective function coincides with ProbLoss.

Architecture. The novelty of DISCO Nets resides in their objective function. They do not require the use of a specific network architecture. This allows us to design a simple network architecture inspired by Oberweger et al. [18]. The architecture is shown in Figure 2. The input depth image $\boldsymbol{x}$ is fed to 2 convolutional layers, each having 8 filters, with kernels of size 5×5, with stride 1, followed by Rectified Linear Units (ReLUs) and Max Pooling layers of kernel size 3×3. A third and last convolutional layer has 8 filters, with kernels of size 5×5, with stride 1, followed by a Rectified Linear Unit. The ouput of the convolution is concatenated to the random noise vector $\boldsymbol{z}$ of size $d_z = 200$, drawn from a uniform distribution in $[-1, 1]$. The result of the concatenation is fed to 2 dense layers of output size 1024, with ReLUs, and a third dense layer that outputs the candidate pose $\boldsymbol{y} \in \mathbb{R}^{3\times J}$. For the non-probabilistic BASE$_{\beta,\sigma}$ model no noise is concatenated as only a pointwise estimate is produced.

Training. We use 10,000 examples from the 72,757 training frames to construct a validation dataset and train only on 62,757 examples. Back-propagation is used with Stochastic Gradient Descent with a batchsize of 256. The learning rate is fixed to $\lambda = 0.01$ and we use a momentum of $m = 0.9$ (see Polyak [20]). We also add L2-regularisation controlled by the parameter C. We use $C = [0.0001, 0.001, 0.01]$ which is a relevant range as the comparative model BASE$_\beta$ is best performing for $C = 0.001$. Note that DISCO Nets report consistent performances across the different values C, contrarily to BASE$_\beta$. We use 3 different random seeds to initialize each model network parameters. We report the performance of each model with its best cross-validated seed and C. We train all models for 400 epochs as it results in a change of less than 3% in the value of the loss on the validation dataset for BASE$_\beta$. We refer the reader to the supplementary material for details on the setting.

[2] We also evaluate the non-probabilistic model BASE$_\beta$ using its pointwise prediction rather than the MEU method. Results are consistent and detailed in the supplementary material.

Table 2: *Metrics values on the test set ± SEM. Best performances in bold.*

Model	ProbLoss (mm)	MeJEE (mm)	MaJEE (mm)	FF (80mm)
$\text{BASE}_{\beta=1,\sigma=1}$	103.8±0.627	25.2±0.152	52.7±0.290	86.040
$\text{BASE}_{\beta=1,\sigma=5}$	99.3±0.620	25.5±0.151	52.9±0.289	85.773
$\text{BASE}_{\beta=1,\sigma=10}$	96.3±0.612	25.7±0.149	53.2±0.288	85.664
$\text{DISCO}_{\beta=1,\gamma=0}$	92.9±0.533	21.6±0.128	46.0±0.251	92.971
$\text{DISCO}_{\beta=1,\gamma=0.25}$	89.9±0.510	21.2±0.122	46.4±0.252	93.262
$\text{DISCO}_{\beta=1,\gamma=0.5}$	**83.8 ±0.503**	**20.9±0.124**	**45.1±0.246**	**94.438**

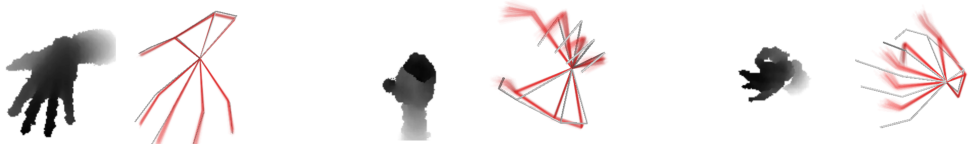
Table 3: *Metrics values on the test set ± SEM for cGAN.*

Model	ProbLoss (mm)	MeJEE (mm)	MaJEE (mm)	FF (80mm)
cGAN	442.7±0.513	109.8±0.128	201.4±0.320	0.000
$\text{cGAN}_{\text{init, fixed}}$	128.9±0.480	31.8±0.117	64.3±0.230	78.454

4.2 Results.

Quantitative Evaluation. Table 2 reports performances on the test dataset, with parameters cross-validated on the validation set. All versions of the DISCO Net model outperform the BASE_β model. Among the different values of γ, we see that $\gamma = 0.5$ better captures the true distribution (lower ProbLoss) while retaining accurate performance on the standard pointwise metrics. Interestingly, using an all-zero noise at test-time gives similar performances on pointwise metrics. We link this to the observation that both the MEAN and the MEU method perform equivalently on these metrics (see supplementary material).

Qualitative Evaluation. In Figure 3 we show candidate poses generated by $\text{DISCO}_{\beta=1,\gamma=0.5}$ for 3 testing examples. The left image shows the input depth image, and the right image shows the ground-truth pose (in grey) with 100 candidate outputs (superimposed in transparent red). The model predict the joint locations and we interpolate the joints with edges. If an edge is thinner and more opaque, it means the different predictions overlap and that the uncertainty on the location of the edge's joints is low. We can see that $\text{DISCO}_{\beta=1,\gamma=0.5}$ captures relevant information on the structure of the hand.



(a) *When there are no occlusions, DISCO Nets model low uncertainty on all joints.*

(b) *When the hand is half-fisted, DISCO Nets model the uncertainty on the location of the fingertips.*

(c) *Here the fingertips of all fingers but the forefinger are occluded and DISCO Nets model high uncertainty on them.*

Figure 3: *Visualisation of $\text{DISCO}_{\beta=1,\gamma=0.5}$ predictions for 3 examples from the testing dataset. The left image shows the input depth image, and the right image shows the ground-truth pose in grey with 100 candidate outputs superimposed in transparent red. Best viewed in color.*

Figure 4 shows the matrices of Pearson product-moment correlation coefficients between joints. We note that DISCO Net with $\gamma = 0.5$ better captures the correlation between the joints of a finger and between the fingers.

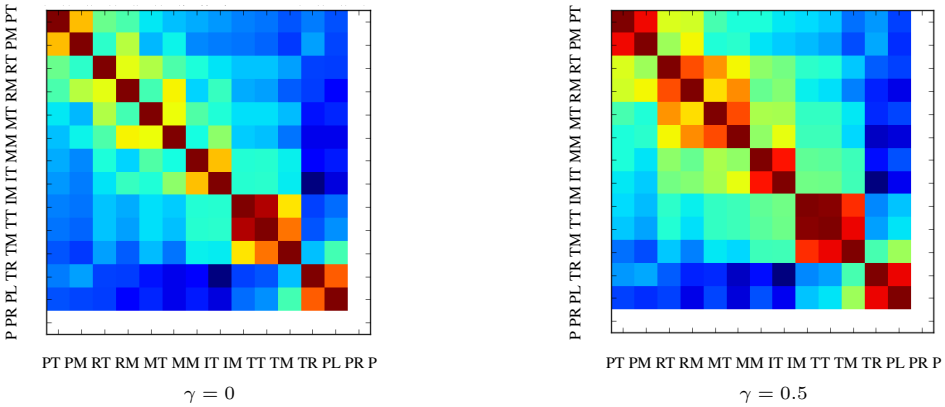


Figure 4: *Pearson coefficients matrices of the joints: Palm (no value as the empirical variance is null), Palm Right, Palm Left, Thumb Root, Thumb Mid, Index Mid, Index Tip, Middle Mid, Middle Tip, Ring Mid, Ring Tip, Pinky Mid, Pinky Tip.*

4.3 Comparison with existing probabilistic models.

To the best of our knowledge the conditional Generative Adversarial Networks (cGAN) from Mirza and Osindero [16] has not been applied to pose estimation. In order to compare cGAN to DISCO Nets, several issues must be overcome. First, we must design a network architecture for the Discriminator. This is a first disadvantage of cGAN compared to DISCO Nets which require no adversary. Second, as mentioned in Goodfellow et al. [7] and Radford et al. [22], GAN (and thus cGAN) require very careful design of the networks' architecture and training procedure. In order to do a fair comparison, we followed the work in Mirza and Osindero [16] and practical advice for GAN presented in Larsen and Sønderby [13]. We try (i) cGAN, initialising all layers of D and G randomly, and (ii) $\text{cGAN}_{\text{init, fixed}}$ initialising the convolutional layers of G and D with the trained best-performing $\text{DISCO}_{\beta=1,\gamma=0.5}$ of Section 4.2, and keeping these layers fixed. That is, the convolutional parts of G and D are fixed feature extractors for the depth image. This is a setting similar to the one employed for tag-annotation of images in Mirza and Osindero [16]. Details on the setting can be found in the supplementary material. Table 3 shows that the cGAN model obtains relevant results only when the convolutional layers of G and D are initialised with our trained model and kept fixed, that is $\text{cGAN}_{\text{init, fixed}}$. These results are still worse than DISCO Nets performances. While there may be a better architecture for cGAN, our experiments demonstrate the difficulty of training cGAN over DISCO Nets.

4.4 Reference state-of-the-art values.

We train the best-performing $\text{DISCO}_{\beta=1,\gamma=0.5}$ of Section 4.2 on the entire dataset, and compare performances with state-of-the-art methods in Table 4 and Figure 5. These state-of-the-art methods are specifically designed for hand pose estimation. In Oberweger et al. [17] a constrained prior hand model, referred as NYU-Prior, is refined on each hand joint position to increase accuracy, referred as NYU-Prior-Refined. In Oberweger et al. [18], the input depth image is fed to a first network NYU-Init, that outputs a pose used to synthesize an image with a second network. The synthesized image is used with the input depth image to derive a pose update. We refer to the whole model as NYU-Feedback. On the contrary, DISCO Nets uses a single network whose architecture is similar to NYU-Prior (without constraining on a pose prior). By accurately modeling the distribution of the pose given the depth image, DISCO Nets obtain comparable performances to NYU-Prior and NYU-Prior-Refined. Without any extra effort, DISCO Nets could be embedded in the presented refinement and feedback methods, possibly boosting state-of-the-art performances.

Table 4: *DISCO Nets compared to state-of-the-art performances* $\pm$ *SEM.*

Model	MeJEE (mm)	MaJEE (mm)	FF (80mm)
NYU-Prior	20.7±0.150	44.8±0.289	91.190
NYU-Prior-Refined	19.7±0.157	44.7±0.327	88.148
NYU-Init	27.4±0.152	55.4±0.265	86.537
NYU-Feedback	16.0±0.096	36.1±0.208	97.334
$\text{DISCO}_{\beta=1,\gamma=0.5}$	20.7±0.121	45.1±0.246	93.250

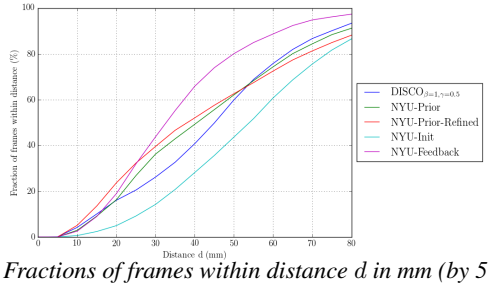


Figure 5: *Fractions of frames within distance* d *in mm (by 5 mm). Best viewed in color.*

5 Discussion.

We presented DISCO Nets, a new family of probabilistic model based on deep networks. DISCO Nets employ a prediction and training procedure based on the minimisation of a dissimilarity coefficient. Theoretically, this ensures that DISCO Nets accurately capture uncertainty on the correct output to predict given an input. Experimental results on the task of hand pose estimation consistently support our theoretical hypothesis as DISCO Nets outperform non-probabilistic equivalent models, and existing probabilistic models. Furthermore, DISCO Nets can be tailored to the task to perform. This allows a possible user to train them to tackle different problems of interest. As their novelty resides mainly in their objective function, DISCO Nets do not require any specific architecture and can be easily applied to new problems. We contemplate several directions for future work. First, we will apply DISCO Nets to other prediction problems where there is uncertainty on the output. Second, we would like to extend DISCO Nets to latent variables models, allowing us to apply DISCO Nets to diverse dataset where ground-truth annotations are missing or incomplete.

6 Acknowlegements.

This work is funded by the Microsoft Research PhD Scholarship Programme. We would like to thank Pankaj Pansari, Leonard Berrada and Ondra Miksik for their useful discussions and insights.

References.

[1] E.L. Denton, S. Chintala, A. Szlam, and R. Fergus. Deep generative image models using a Laplacian pyramid of adversarial networks. In *NIPS*. 2015.

[2] G. K. Dziugaite, D. M. Roy, and Z. Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *UAI*, 2015.

[3] K. Fukumizu, L. Song, and A. Gretton. Kernel Bayes' rule: Bayesian inference with positive definite kernels. *JMLR*, 2013.

[4] J. Gauthier. Conditional generative adversarial nets for convolutional face generation. *Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition*, 2014.

[5] T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 2007.

[6] Tilmann Gneiting, Larissa I. Stanberry, Eric P. Grimit, Leonhard Held, and Nicholas A. Johnson. Assessing probabilistic forecasts of multivariate quantities, with an application to ensemble predictions of surface winds. *TEST*, 2008.

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

[8] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Scholkopf, and A. J. Smola. A kernel method for the two-sample problem. In *NIPS*, 2007.

[9] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Scholkopf, and A. J. Smola. A kernel two-sample test. In *JMLR*, 2012.

[10] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *ICLR*, 2014.

[11] M. P. Kumar, B. Packer, and D. Koller. Modeling latent variable uncertainty for loss-based learning. In *ICML*, 2012.

[12] S. Lacoste-Julien, F. Huszar, and Z. Ghahramani. Approximate inference for the loss-calibrated Bayesian. In *AISTATS*, 2011.

[13] A. B. L. Larsen and S. K. Sønderby. URL `http://torch.ch/blog/2015/11/13/gan.html`.

[14] Y. Li, K. Swersky, and R. Zemel. Generative moment matching networks. In *ICML*, 2015.

[15] A. Makhzani, J. Shlens, N. Jaitly, and I. J. Goodfellow. Adversarial autoencoders. *ICLR Workshop*, 2015.

[16] M. Mirza and S. Osindero. Conditional generative adversarial nets. In *NIPS Deep Learning Workshop*, 2014.

[17] M. Oberweger, P. Wohlhart, and V. Lepetit. Hands deep in deep learning for hand pose estimation. In *Computer Vision Winter Workshop*, 2015.

[18] M. Oberweger, P. Wohlhart, and V. Lepetit. Training a Feedback Loop for Hand Pose Estimation. In *ICCV*, 2015.

[19] Pierre Pinson and Julija Tastu. Discrimination ability of the energy score. *Technical University of Denmark. (DTU Compute-Technical Report-2013; No. 15)*, 2013.

[20] B. T. Polyak. Some methods of speeding up the convergence of iteration methods. 1964.

[21] V. Premachandran, D. Tarlow, and D. Batra. Empirical minimum Bayes risk prediction: How to extract an extra few% performance from vision models with just three more parameters. In *CVPR*, 2014.

[22] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2015.

[23] C.R. Rao. Diversity and dissimilarity coefficients: A unified approach. *Theoretical Population Biology*, pages Vol. 21, No. 1, pp 24–43, 1982.

[24] S. Reed, Z. Akata, X. Yan, L. Logeswaran, H. Lee, and B. Schiele. Generative adversarial text to image synthesis. In *ICML*, 2016.

[25] J. T. Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *ICLR*, 2016.

[26] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The vitruvian Manifold: Inferring dense correspondences for oneshot human pose estimation. In *CVPR*, 2012.

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

[28] X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2image: Conditional image generation from visual attributes. 2016.

[29] E. Zawadzki and S. Lahaie. Nonparametric scoring rules. In *AAAI Conference on Artificial Intelligence*. 2015.