Learning Human Poses from Actions - Supplementary Material

Aditya Arun[1]
aditya.arun@research.iiit.ac.in
C.V. Jawahar[1]
jawahar@iiit.ac.in
M. Pawan Kumar[2]
pawan@robots.ox.ac.uk

[1] IIIT Hyderabad
[2] University of Oxford &
The Alan Turing Institute

In this supplementary material, we provide additional details on optimization of our learning objective, implementation details, and visualization of the learning process. We also provide additional results of training a different architecture for human pose estimation on two data sets.

1 Optimization

In this section, we provide details of optimization presented in section 3.5 of the paper.

1.1 Learning Objective

We represent the prediction distribution using a DISCO Net, which we denote by $\Pr_{\mathbf{w}}$, $\mathbf{w}$ being the parameter of the network. Similarly, we represent the conditional distribution using a set of DISCO Nets, which we denote by $\Pr_{\boldsymbol{\theta}}$. The set of parameters for the conditional networks is denoted by $\boldsymbol{\theta}$. We compute samples from the prediction network as $\{\mathbf{h}_k^{\mathbf{w}}, k = 1, \cdots, K\}$, and samples from conditional network as $\{\mathbf{h}_k'^{\boldsymbol{\theta}}, k = 1, \cdots, K\}$ for a given training sample. The unbiased estimated value of the learning objective (Equation (5) of the main paper) can be written as follows:

$$\operatorname*{arg\,min}_{\mathbf{w},\theta} F(\mathbf{w},\boldsymbol{\theta}) = \frac{1}{NK^2}\sum_{i=1}^{N}\left(\sum_{k,k'}\Delta(\mathbf{h}_k^{\mathbf{w}},\mathbf{h}_{k'}^{\theta}) - \gamma\sum_{k,k'}\Delta(\mathbf{h}_k^{\mathbf{w}},\mathbf{h}_{k'}^{\mathbf{w}}) - (1-\gamma)\sum_{k,k'}\Delta(\mathbf{h}_k^{\theta},\mathbf{h}_{k'}^{\theta})\right) \tag{1}$$

In order to minimize the dissimilarity coefficient between the parameters of the prediction and the conditional distributions, we employ stochastic gradient descent. We note that jointly optimizing the objective function over the parameters of the prediction and the conditional distribution networks is expensive in terms of memory and time, as it involves optimizing two networks together. Therefore, first, we initialize the two networks by training them with the small amount of fully annotated pose data. We then perform iterative optimization using block coordinate descent to first train the parameters of the prediction and conditional distribution and then proceed with more expensive joint optimization. Algorithm for optimizing these two sets of parameters are shown in the following subsections. Using this hybrid training strategy, we reduce the training complexity without compromising on the accuracy.

1.2 Iterative Optimization

The coordinate descent optimization proceeds by iteratively fixing the prediction network and estimating the conditional networks, followed by updating the prediction network for fixed conditional networks. The parameters of both the set of networks are initialized using the small amount of fully supervised samples available in the data set. The main advantage of the iterative strategy is that it results in a problem similar to the fully supervised learning of DISCO Nets at each iteration. This, in turn, allows us to readily use the algorithm developed in [3]. Furthermore, it also reduces the memory complexity of learning, thereby allowing us to learn a large network. The two steps of the iterative algorithm are described below.

Optimization over Conditional Network For fixed $\mathbf{w}$, the learning objective corresponds to the following:

$$\underset{\boldsymbol{\theta}}{\arg\min} \sum_{i} DIV(\Pr_{\mathbf{w}}, \Pr_{\boldsymbol{\theta}}) - (1-\gamma) DIV(\Pr_{\boldsymbol{\theta}}, \Pr_{\boldsymbol{\theta}}) \tag{2}$$

The above equation can be expanded as,

$$\min_{\boldsymbol{\theta}} F(\boldsymbol{\theta}) = \frac{1}{NK^2} \sum_{i=1}^{N} \left(\sum_{k,k'} \Delta(\mathbf{h}_k^{\mathbf{w}}, \mathbf{h}_{k'}^{\theta}) - (1-\gamma) \sum_{k,k'} \Delta(\mathbf{h}_k^{\boldsymbol{\theta}}, \mathbf{h}_{k'}^{\boldsymbol{\theta}}) \right) \tag{3}$$

The above objective function is similar to the one used in [3] for fully supervised learning. Similar to [3], we solve it via stochastic gradient descent. Note that since it is possible to generate samples from both the prediction and the conditional network, we can obtain an unbiased estimate of the gradient of the objective function (3). As observed in [3], this is sufficient to minimize the learning objective in order to estimate the DISCO Net parameters.

The above objective function is solved via stochastic gradient descent, as shown in Algorithm 1.

Algorithm 1 Optimization over θ

```
Input: Data set $\mathcal{D}$ and initial estimate $\theta^0$
  for $t = 1 \ldots T$ epochs do
      Sample mini-batch of $b$ training example pairs
      for $n = 1 \ldots b$ do
          Sample $K$ random noise vectors $\mathbf{z}_k$
          Generate $K$ candidate output from $\Pr_{\mathbf{w}}(\mathbf{x}, \mathbf{z}_k)$ and $\Pr_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}_k)$
      end for
      Compute $F(\theta)$ as given here in equation (3) here.
      Update parameters $\theta$ via SGD with momentum
  end for
```

Optimization over Prediction Network For fixed $\boldsymbol{\theta}$, the learning objective corresponds to the following:

$$\min_{\mathbf{w}} \sum_{i} DIV(\Pr_{\mathbf{w}}, \Pr_{\boldsymbol{\theta}}) - \gamma DIV(\Pr_{\mathbf{w}}, \Pr_{\mathbf{w}}) \tag{4}$$

The above equation can be expanded as,

$$\min_{\mathbf{w}} F(\mathbf{w}) = \frac{1}{NK^2} \sum_{i=1}^{N} \left(\sum_{k,k'} \Delta(\mathbf{h}_k^{\mathbf{w}}, \mathbf{h}_{k'}^{\boldsymbol{\theta}}) - \gamma \sum_{k,k'} \Delta(\mathbf{h}_k^{\mathbf{w}}, \mathbf{h}_{k'}^{\mathbf{w}}) \right) \tag{5}$$

Once again, using the fact that it is possible to obtain unbiased estimates of the gradients of the above objective function, we employ stochastic gradient descent to update the parameters of the prediction network.

Similar to the conditional network, the above objective function is optimized by using stochastic gradient descent as shown in Algorithm 2.

Algorithm 2 Optimization over $\mathbf{w}$

```
Input: Data set $\mathcal{D}$ and initial estimate $\mathbf{w}^0$
    for $t = 1 \dots T$ epochs do
        Sample mini-batch of $b$ training example pairs
        for $n = 1 \dots b$ do
            Sample $K$ random noise vectors $\mathbf{z}_k$
            Generate $K$ candidate output from $\Pr_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}_k)$ and $\Pr_{\mathbf{w}}(\mathbf{x}, \mathbf{z}_k)$.
        end for
        Compute $F(\mathbf{w})$ as given in equation (5) here.
        Update parameters $\mathbf{w}$ via SGD with momentum
    end for
```

1.3 Joint Optimization

Although the iterative optimization provides for faster convergence of our objective function, this approach of finding a local minima along one coordinate direction at the current point, in each iteration, often leads to an approximate solution with respect to the optimization problem at hand. To address this problem and find accurate local minima of our non-convex objective (equation (5) of main paper), we perform joint optimization of our objective function by employing stochastic gradient descent to update the parameters of both conditional and prediction distribution networks. We obtain the gradients by computing the unbiased estimate of our objective function and update the two networks using stochastic gradient descent as shown in Algorithm 3. Additionally, we initialize our parameters of the networks corresponding to the two distributions with the values obtained after the iterative optimization. This initialization strategy also reduces the number of iterations required for convergence, thus reducing the training time complexity.

2 Visualization of the Learning Process

We provide visualization of the iterative learning procedure as discussed in the optimization section of the main paper. We show a hundred different pose estimates of two examples, of

Algorithm 3 Joint Optimization over $\mathbf{w}, \boldsymbol{\theta}$

```
Input: Data set $\mathcal{D}$, learning rate $\eta$, momentum $m$,
    and initial estimate $\mathbf{w}^0, \boldsymbol{\theta}^0$
    for $t = 1 \dots T$ epochs do
        Sample mini-batch of $b$ training example pairs
        for $n = 1 \dots b$ do
            Sample $K$ random noise vectors $\mathbf{z}_k$
            Generate $K$ candidate output from $\Pr_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}_k)$ and $\Pr_{\mathbf{w}}(\mathbf{x}, \mathbf{z}_k)$.
        end for
        Compute $F(\mathbf{w}, \boldsymbol{\theta})$ as given in equation (1) here.
        Update parameters $\mathbf{w}$ via SGD with momentum
    end for
```

varying difficulty, over the iterations of the optimization algorithm. The pose estimates are superimposed on the image. Hence, if all the pose estimates agree with each other, the lines depicting the samples will be thin and opaque. In order to represent the low uncertainty in the pose estimates of this image, we will draw a green bounding box around the image. For such images, the expected loss is less than 3. In contrast, if the pose estimates vary significantly from each other, then the lines depicting the samples will be spread out and less opaque. In order to represent the high uncertainty in the pose estimates of this image, we will draw a blue bounding box around the image. For these samples, the expected loss is more than 3.

The first case shown in figure 1 represents an easy case where the initial prediction and conditional networks, $\Pr_{\mathbf{w}}$ and $\Pr_{\boldsymbol{\theta}}$, trained only on the fully annotated training set, have low uncertainty for the predicted pose. In these images, there are no occlusions of any human part, and the person present in the image is in the standard pose for the particular action he is performing. For such cases, the fully annotated training data set is enough to train the prediction network such that it has high confidence in the estimated pose, and they do not require weakly supervised training. However, even in such cases, we see a minor improvement in the estimated pose over the iterations of the optimization algorithm.

Figure 2 represents a moderately difficult example. Typically, such examples are those where a person is performing commonly occurring actions, like exercising, riding a bike or skate board, or running. In such examples, some joints are occluded and the person in the image is in some variation of the standard pose for a particular action he is performing. The majority of the data set are comprised of moderately difficult examples. In such cases, the prediction network $\Pr_{\mathbf{w}}$ has high uncertainty over the predicted pose, but conditional network $\Pr_{\boldsymbol{\theta}}$ has high confidence and therefore low uncertainty over the predicted pose. Here we observe that over the iterations, the prediction network gains confidence as the information present in the conditional network is successfully transferred to it.

The final case, shown in figure 3 represents a difficult example, where the person is performing an unusual or rare action, like underwater swimming or a person kicking a ball in the air. The rarity of such poses in the supervised training set means that both prediction and conditional networks, $\Pr_{\mathbf{w}}$ and $\Pr_{\boldsymbol{\theta}}$, have high uncertainty in the predicted pose. However, over the iterations, by using the information gained from other simpler examples in the weakly supervised data set, the accuracy for such cases improves significantly.

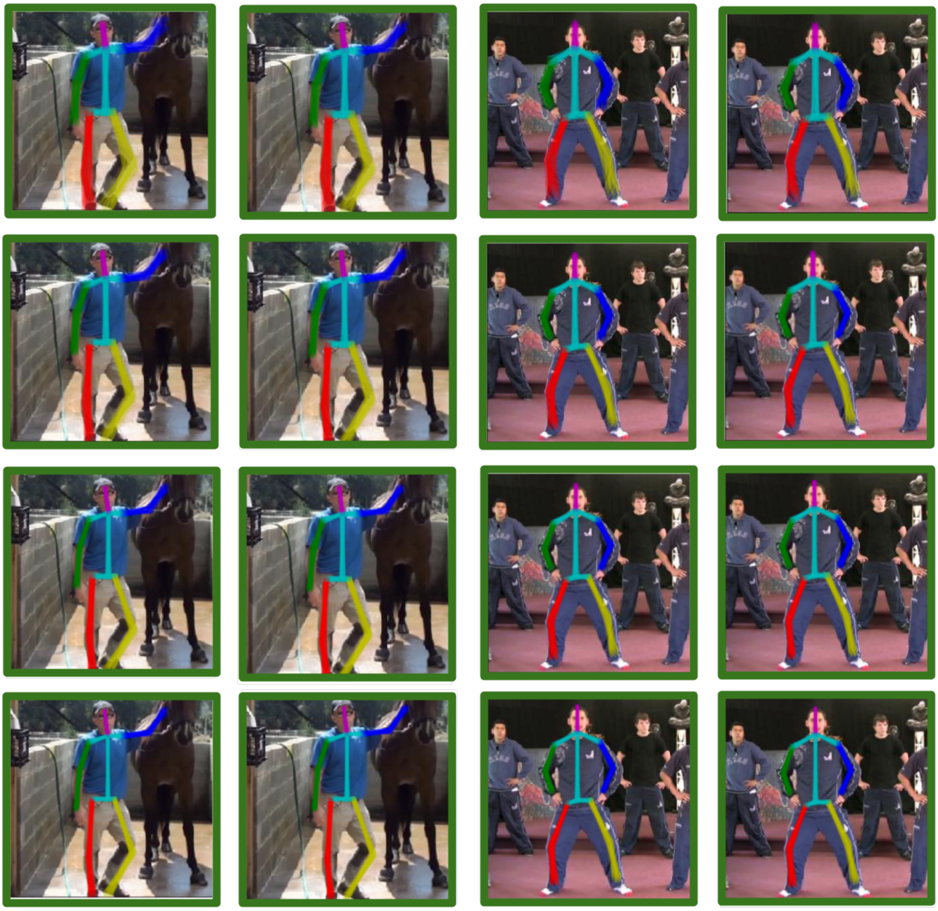

Figure 1: *Example of superimposed pose predictions by* Disco *Nets illustrating the uncertainty in the pose across training iterations for an easy case. The blue box around the images represent a high diversity coefficient value, and the green box around them represents low diversity coefficient value. Columns* 1 *and* 3 *are outputs of the prediction network and columns* 2 *and* 4 *are outputs of conditional network. Row* 1 *represents initial prediction of networks; rows* 2 *and* 3 *represents prediction of networks in second and fifth iteration respectively and last row represents prediction of networks when they have converged. The images in the first and second column show an easy example of a person standing straight with his one hand held out and the third and fourth columns show a person standing in relaxed upright pose. where both the conditional network and the prediction network performs well from the beginning of the optimization procedure. For each example, the first column shows estimated pose from prediction network and the second column shows estimated pose from conditional network. Best viewed in color.*

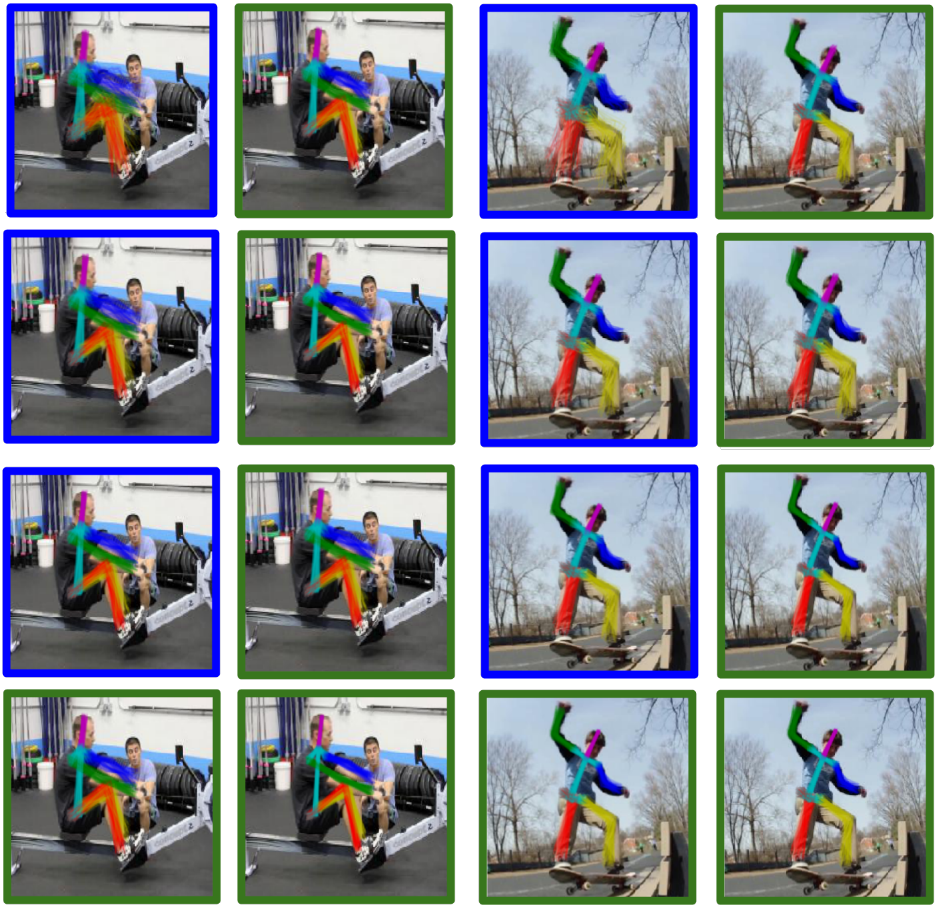

Figure 2: *Example of superimposed pose predictions by* DISCO *Nets illustrating the uncertainty in the pose across training iterations for examples with moderate difficulty. The blue box around the images represent a high diversity coefficient value, and the green box around them represents low diversity coefficient value. Columns* 1 *and* 3 *are outputs of the prediction network and columns* 2 *and* 4 *are outputs of conditional network. Row* 1 *represents initial prediction of networks; rows* 2 *and* 3 *represents prediction of networks in second and fifth iteration respectively and last row represents prediction of networks when they have converged. The images in the first and second column show a common action of a person exercising and the third and fourth column shows a person riding a skate board. In these cases, the conditional network performs well from the beginning of the optimization procedure. At convergence, both the prediction network provides accurate pose estimates for such moderately difficult images by transferring information from conditional network. For each example, the first column shows estimated pose from prediction network and the second column shows estimated pose from conditional network. Best viewed in color.*

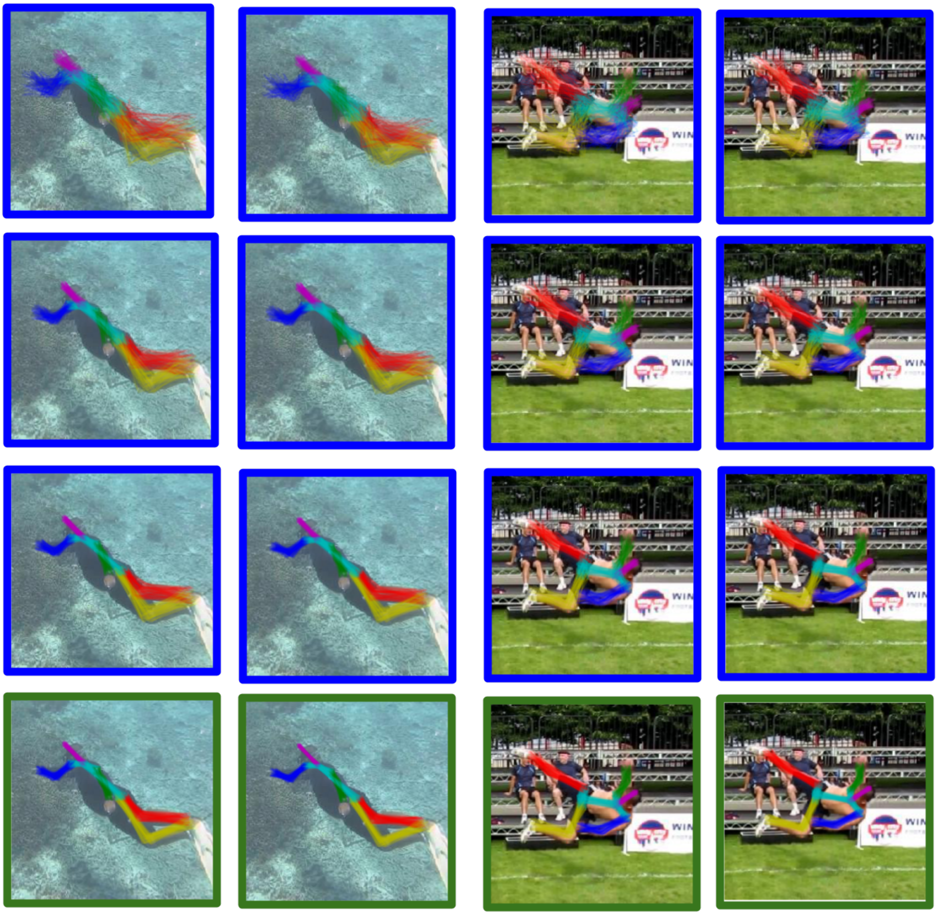

Figure 3: *Example of superimposed pose predictions by* DISCO *Nets illustrating the uncertainty in the pose across training iterations for difficult examples. The blue box around the images represent a high diversity coefficient value, and the green box around them represents low diversity coefficient value. Columns* 1 *and* 3 *are outputs of the prediction network and columns* 2 *and* 4 *are outputs of conditional network. Row* 1 *represents initial prediction of networks; rows 2 and 3 represents prediction of networks in second and fifth iteration respectively and last row represents prediction of networks when they have converged. The images in the first and second column show a rare action of person swimming underwater, and the third and fourth columns show a person in an unusual pose, where he is kicking the ball in air. Such rarity in pose leads to high uncertainty in both the networks initially. At convergence, both the networks provided accurate pose estimates for the difficult image by learning from the easier images. For each example, the first column shows estimated pose from prediction network and the second column shows estimated pose from conditional network. Best viewed in color.*

3 Implementation Details

In this section, we provide the details of our experimental setup. We construct $\mathrm{Pr}_{\mathbf{w}}$ by taking a standard architecture for human pose estimation, namely, the stacked hourglass network [5]. A noise filter of size 64×64 is added to the output of the penultimate hourglass module, which itself consists of 256 64×64 filters. The 257 channels are convolved with a 1×1 filter to bring the number of channels back to 256. This is followed by a final hourglass module as shown in figure 2 (closely following stacking approach of Stacked Hourglass network [5]. We note that all parameters remain differentiable and hence can be trained via backpropagation as discussed in Section 1 of the supplementary.

The conditional network $\mathrm{Pr}_{\boldsymbol{\theta}}$ is modeled exactly as the prediction network $\mathrm{Pr}_{\mathbf{w}}$, except that there are $\mathbf{a}$ different output branches (consisting of 1 hourglass module), one for each possible action class, stacked on top of penultimate hourglass module. Note that for each action class, we have a unique set of noise filters. During forward and backward propagation of the conditional network given an image from a particular action class, we mask the output from every other branch not corresponding to that particular action class.

The non probabilistic pointwise network is a DISCO Net that uses the architecture shown in figure 2 (of the main paper), but discards the last two self-diversities terms in the learning objective (Equation (5) of main paper), and whose pointwise prediction is computed by principle of maximum expected utility (MEU) (Equation (1) of main paper). We refer this pointwise network as PW Net.

For the given data set $\mathcal{D}$, as given in section 4 of the paper, we train our three networks, FS, $\mathrm{PW}_{\mathbf{w}}$ and $\mathrm{Pr}_{\mathbf{w}}$ on the fully annotated training set. We note that after data augmentation, our training set (fully annotated data and the weakly annotated data) for each split, becomes $4\times$ larger, and for the FS network, we additionally perform random crops such that the number of training samples for all three networks are the same. Networks $\mathrm{PW}_{\boldsymbol{\theta}}$ and $\mathrm{Pr}_{\boldsymbol{\theta}}$ are first initialized by the weights of $\mathrm{PW}_{\mathbf{w}}$ and $\mathrm{Pr}_{\mathbf{w}}$ respectively, then they are fine tuned using action specific samples from the fully annotated training set. For training, we used $\eta = 0.025$ and momentum $m = 0.9$. We cross validated weight decay regularization parameter C in the range $[0.1, 0.01, 0.001, 0.0001]$ for our baseline networks FS and PW and found that values 0.001 and 0.0001 works best for FS and PW respectively. We chose $C = 0.01$ for training our probabilistic networks. Moreover, for our probabilistic network, $\mathrm{Pr}_{\mathbf{w}}$, we choose $K = 100$ samples. However, for a different task, it has been observed that results hold even for $K = 2$ [3].

While training the baseline non probabilistic point wise prediction network PW using diverse data using self paced learning, we only backpropagate when the loss computed is within some threshold t. For such network, the loss would be high when predicted pose from $\mathrm{PW}_{\mathbf{w}}$ and $\mathrm{PW}_{\boldsymbol{\theta}}$ are very different from each other. Applying threshold on the loss for backpropagation ensures that these networks are only updated when both of them agree and therefore, they do not learn from erroneous or less confident predictions.

For our probabilistic network, $\mathrm{Pr}_{\mathbf{w}}$, we do not require such threshold as the diversity coefficient term in our objective function ensures that our network learns only from confident predictions and not from samples when the network has low confidence. In other words, our method has fewer parameters than the baseline.

We train all of these networks for 100 epochs and monitor the training and validation accuracies for each epoch. We employ an early stopping strategy based on validation accuracy to avoid over-fitting the data set. We save the network parameters corresponding to the best validation accuracy and report our result on the held out test set.

4 Results

In this section, we provide additional results of training the three network (FS, PW and $\mathrm{Pr}_{\mathbf{w}}$) described in section 4 of the main paper.

4.1 Results on MPII data set

The detailed PcKh graphs on MPII data set by training an 8-stack hourglass network on various setting described in the paper are presented in figure 4.

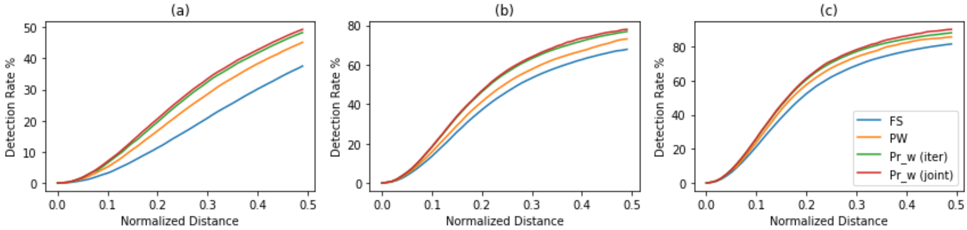

Figure 4: *Total PcKh comparison on MPII when trained on (a)* $25-75$ *split, (b)* $50-50$ *split; and (c)* $75-25$ *split.*

In the figure, we can see that we consistently outperform the baseline FS and PW networks across all normalized distances. The networks trained on diverse data set (the PW and the $\mathrm{Pr}_{\mathbf{w}}$ network) performs significantly better on lower normalized scores than the FS net which does not utilize the action annotations when there are only a few strong pose annotations available. This shows the utility of using action annotations when pose annotations are missing. The importance of using the probabilistic framework can be seen for lower normalized distance for all three splits, where the $\mathrm{Pr}_{\mathbf{w}}$ network effectively captures the uncertainty present in the data set. We observe that as the number of supervised samples in our diverse data set increase, the accuracy of all the networks improves for smaller normalized distance. The joint training of the $\mathrm{Pr}_{\mathbf{w}}$ network also improves the results over the iterative optimization of $\mathrm{Pr}_{\mathbf{w}}$ network.

4.2 Results on JHMDB data set

In this subsection, we provide additional results of training our various models based on 8-stack hourglass network [5] on the JHMDB data set [4] for $50-50$ split.

The JHMDB data set, which consists of 33183 frames from 21 action class, have 13 annotated joint locations. We split the frames from each action class into $\{70, 15, 15\}\%$ training, validation and test sets, which corresponds to 22883 frames in the training set, and 4150 frames in the validation and the test set. To create a diverse data set with $50-50$ split, we randomly drop pose annotations from 50% from the frames of the training set, similar to those described in Section 4 of the main paper.

The result for training the FS, PW and $\mathrm{Pr}_{\mathbf{w}}$ networks for the $50-50$ split on the JHMDB data set are summarized in table 1.

We observe that the accuracies of the three networks (FS, PW and $\mathrm{Pr}_{\mathbf{w}}$) holds similar trends as we had seen for the MPII data set.

Method	FS	PW	$\text{Pr}_\mathbf{w}$ (iter)	$\text{Pr}_\mathbf{w}$ (joint)
Total Accuracy	80.01	85.77	89.90	91.25

Table 1: *Results on JHMDB data set (*PCKh@0.5*), where* FS *is trained using* 50% *percentage of fully annotated data and* PW *and* Pr_w *are trained on* $50-50$ *split of fully annotated and weakly annotated training data. Here* FS *and* PW *are the fully supervised and the point-wise networks respectively, and* Pr_w *(iterative) and* Pr_w *(joint) is our proposed probabilistic network trained with block coordinate optimization and joint optimization respectively.*

5 Additional Results

To prove the generality of our method, we provide additional results using a different architecture, as proposed by Belagiannis *et al.* [2]. The authors pose the problem of estimating human poses as regression and propose to minimize a novel Tukey's biweight function as loss function for their ConvNet. They empirically show that their method outperforms the simple $L2$ loss. The point-wise architecture, consisting of five convolutional layers and two fully connected layers is modified to a DISCO Net as shown in the figure 5 below. A 1024 dimensional noise vector, sampled from a uniform distribution, is appended to the flattened CNN features, before applying fully connected layers.

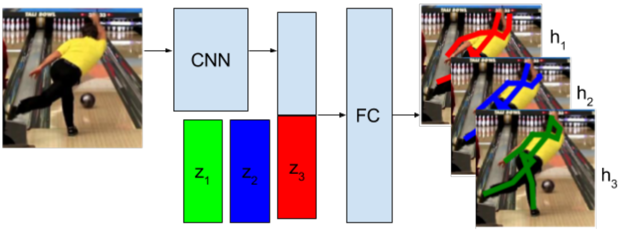

Figure 5: Modified architecture, as proposed by Belagiannis *et al.* [2]. The figure shows the sampling process of DISCO Net. The block CNN consists of 5 convolution layers. The middle block is the flattened feature vector obtained after convolution. The block FC consists of two fully connected layers.

We evaluate the performance of the FS, PW and our proposed probabilistic network $\text{Pr}_\mathbf{w}$ on $50-50$ split of two data sets, namely (i) MPII Human Pose data set [1], and (ii) JHMDB data set [4]. The various splits of MPII Human Pose are similar to the ones described in Section 4 of the main paper. The MPII and the JHMDB data set is split exactly as it was done for the stacked hourglass network.

The results are summarized in Table 2. .

We observe that the results shown in Table 2 on both the data sets are consistent with our observations on the stacked hourglass network. Networks PW and $\text{Pr}_\mathbf{w}$ trained on the diverse data, outperforms the FS Net, which is trained only using the fully supervised annotations. This demonstrates the advantage of using diverse learning over a fully supervised method. Moreover, our proposed probabilistic net $\text{Pr}_\mathbf{w}$ outperforms the pointwise network PW, this signifies the importance of modeling uncertainty over pose. We also note that performing joint optimization, after iterative optimization step, further increases our accuracy by 1.2% on MPII Human Pose data set and by 1.4% on JHMDB data set.

Method	MPII	JHMDB
FS	41.89	54.31
PW	54.37	66.19
$\mathrm{Pr}_{\mathbf{w}}$ (iterative)	56.09	71.02
$\mathrm{Pr}_{\mathbf{w}}$ (joint)	**57.28**	**72.61**

Table 2: *Results on MPII Human Pose data set and JHMDB data set (*PCKh@0.5*), where* FS *is trained using* 50% *percentage of fully annotated data and* PW *and* Pr_w *are trained on* $50-50$ *split of fully annotated and weakly annotated training data. Here* FS *and* PW *are the fully supervised and the pointwise networks respectively, and* Pr_w *(iterative) and* Pr_w *(joint) is our proposed probabilistic network trained with block coordinate optimization and joint optimization respectively.*

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014.

[2] Vasileios Belagiannis, Christian Rupprecht, Gustavo Carneiro, and Nassir Navab. Robust optimization for deep regression. In *ICCV*, 2015.

[3] Diane Bouchacourt, M. P. Kumar, and Sebastian Nowozin. Disco nets: Dissimilarity coefficients networks. In *NIPS*, 2016.

[4] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *ICCV*, 2013.

[5] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.