MIXTURE DENSITY NETWORKS FIND VIEWPOINT THE DOMINANT FACTOR FOR ACCURATE SPATIAL OFFSET REGRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Offset regression is a standard method for spatial localization in many vision tasks, including human pose estimation, object detection, and instance segmentation. However, if high localization accuracy is crucial for a task, convolutional neural networks will offset regression usually struggle to deliver. This can be attributed to the locality of the convolution operation, exacerbated by variance in scale, clutter, and viewpoint. An even more fundamental issue is the multi-modality of real-world images. As a consequence, they cannot be approximated adequately using a single mode model. Instead, we propose to use mixture density networks (MDN) for offset regression, allowing the model to manage various modes efficiently and learning to predict full conditional density of the outputs given the input. On 2D human pose estimation in the wild, which requires accurate localisation of body keypoints, we show that this yields significant improvement in localization accuracy. In particular, our experiments reveal viewpoint variation as the dominant multi-modal factor. Further, by carefully initializing MDN parameters, we do not face any instabilities in training, which is known to be a big obstacle for widespread deployment of MDN. The method can be readily applied to any task with a spatial regression component. Our findings highlight the multi-modal nature of real-world vision, and the significance of explicitly accounting for viewpoint variation, at least when spatial localization is concerned.

1 INTRODUCTION

Training deep neural networks is a non-trivial task in many ways. Properly initializing the weights, carefully tuning the learning rate, normalization of weights or targets, or using the right activation function can all be vital for getting a network to converge at all. From another perspective, it is crucial to carefully formulate the prediction task and loss on top of a rich representation to efficiently leverage all the features learned. For example, combining representations at various network depths has been shown to be important to deal with objects at different scales Newell et al. (2016); Lin et al. (2017); Liu et al. (2016).

For some issues, it is relatively straightforward to come up with a network architecture or loss formulation to address them – see e.g. techniques used for multi-scale training and inference. In other cases it is not easy to manually devise a solution. For example, offset regression is extensively used in human pose estimation and instance segmentation, but it lacks high spatial precision. Fundamental limitations imposed by the convolution operation and downsampling in networks, as well as various other factors contribute to this – think of scale variation, variation in appearance, clutter, occlusion, and viewpoint. When analyzing a standard convolutional neural network (CNN) with offset regression, it seems the network knows roughly where a spatial target is located and moves towards it, but cannot get precise enough. How can we make them more accurate? That's the question we address in this paper, in the context of human pose estimation. Mixture density models offer a versatile framework to tackle such challenging, multi-modal settings. They allow for the data to speak for itself, revealing the most important modes and disentangling them.

To the best of our knowledge, mixture density models have not been successfully integrated in 2D human pose estimation to date. In fact, our work has only become possible thanks to recent work

of Zhou et al. (2019a) proposing an offset based method to do dense human pose estimation, object detection, depth estimation, and orientation estimation in a single forward pass. Essentially, in a dense fashion they classify some central region of an instance to decide if it belongs to a particular category, and then from that central location regress offsets to spatial points of interest belonging to the instance. In human pose estimation this would be keypoints; in instance segmentation it could be extreme points; and in tracking moving objects in a video this could be used to localize an object in a future frame Zhou et al. (2019b); Neven et al. (2019); Novotny et al. (2018); Cui et al. (2019). This eliminates the need for a two stage top-down model or for an ad hoc post processing step in bottom-up models. The former would make it very slow to integrate a density estimation method, while for the latter it is unclear how to do so – if possible at all.

In particular, we propose to use mixture density networks (MDN) to help a network disentangle the underlying modes that, when taken together, force it to converge to an average regression of a target. We conduct experiments on the MS COCO human pose estimation task Lin et al. (2014), because its metric is very sensitive to spatial localization: if the ground truth labels are displaced by just a few pixels, the scores already drop significantly, as shown in top three rows of Table 4. This makes the dataset suitable for analyzing how well different models perform on high precision localization. Any application demanding high precision localization can benefit from our approach. For example, spotting extremely small broken elements on an electronic board or identifying surface defects on a steel sheet using computer vision are among such applications.

In summary, our contributions are as follows:

- We propose a new solution for offset regression problems in 2D using MDNs. To the best of our knowledge this is the first work to propose a full conditional density estimation model for 2D human pose estimation on a large unconstrained dataset. The method is general and we expect it to yield significant gains in any spatial dense prediction task.
- We show that using MDN we can have a deeper understanding of what modes actually make a dataset challenging. Here we observe that viewpoint is the most challenging mode that forces a single mode model to settle down for a sub-optimal solution.

2 Related Work

Multi-person human pose estimation solutions usually work either top-down or bottom-up. In the top-down approach, a detector finds person instances to be processed by a single person pose estimator He et al. (2017); Newell et al. (2016); Chen et al. (2018); Li et al. (2019). When region-based detectors Girshick et al. (2014) are deployed, top-down methods are robust to scale variation. But they are slower compared to bottom-up models. In the bottom-up approach, all keypoints are localized by means of heatmaps Cao et al. (2017), and for each keypoint an embedding is learned in order to later group them into different instances. Offset based geometric Cao et al. (2018; 2017); Papandreou et al. (2018) and associative Newell et al. (2017) embeddings are the most successful models. However, they lead to inferior accuracy and need an ad hoc post-processing step for grouping.

To overcome these limitations, recently Zhou et al. (2019a) proposed a solution that classifies each spatial location as corresponding to (the center of) a person instance or not and at the same location generates offsets for each keypoint. This method is very fast and eliminates the need for a detector or post-processing to group keypoints. In spirit, it is similar to YOLO and SSD models developed for object detection Redmon et al. (2016); Liu et al. (2016). However, offset regression does not deliver high spatial precision and the authors still rely on heatmaps to further refine the predictions. Overcoming this lack of accuracy is the main motivation for this work.

As for the superiority of having multiple choice solutions for vision tasks, Guzman-Rivera et al. (2012); Lee et al. (2015; 2016); Rupprecht et al. (2017) have shown that having multiple prediction heads while enforcing them to have diverse predictions, works better than a single head or an ensemble of models. However, they depend on an oracle to choose the best prediction for a given input. The underlying motivation is that the system later will be used by another application that can assess and choose the right head for an input. Clearly this is a big obstacle in making such models practical. And, of course, these methods do not have a mechanism to learn conditional density of outputs for a given input. This is a key feature of mixture models.

Mixture density networks Bishop (1994) have attracted a lot of attention in the very recent years. In particular, it has been applied to 3D human pose estimation Li & Lee (2019), and 3D hand pose estimation Ye & Kim (2018). Both works are applied to relatively controlled environments. In 2D human pose estimation, Rupprecht et al. (2017) have reported unsuccessful application of MDNs, caused by numerical instabilities originating from the variance values. Here, we show that properly initializing the variance values easily avoids such instabilities.

3 FORMULATION

We first review the mixture density networks and then show how we adapt it for offset regression.

3.1 MIXTURE DENSITY NETWORKS

Mixture models are theoretically very powerful tools to estimate the density of any distribution McLachlan & Basford (1988). They recover different modes that contribute to the generation of a dataset, and are straightforward to interpret. Mixture density networks (MDN) Bishop (1994) is a technique that enables us to use neural networks to estimate the parameters of a mixture density model. MDNs estimate the probability density of a target conditioned on the input. This is a key technique to avoid converging to an average target value given an input. For example, if a 1D distribution consists of two Gaussians with two different means, trying to estimate its density using a single Gaussian will result in a mean squashed in between the two actual means, and will fail to estimate any of them. This effect is well illustrated in *Figure 1* of the original paper by Bishop Bishop (1994).

In a regression task, given a dataset containing a set of input vectors as $\{\mathbf{x}_0 \dots \mathbf{x}_n\}$ and the associated target vectors $\{\mathbf{t}_0 \dots \mathbf{t}_n\}$, MDN will fit the weigths of a neural network such that it maximizes the likelihood of the training data. The key formulation then is the representation of the probability density of the target values conditioned on the input, as shown in equation 1:

$$p(\mathbf{t}_i|\mathbf{x}_i) = \sum_{m=1}^{M} \alpha_m(\mathbf{x}_i)\phi_m(\mathbf{t}_i|\mathbf{x}_i) \tag{1}$$

Here M is a hyper-parameter and denotes the number of components constituting the mixture model. $\alpha_m(\mathbf{x}_i)$ is called mixing coefficient and indicates the probability of component m being responsible for generation of the sample $\mathbf{x}_i$. ϕ_m denotes the probability density function of component m for $\mathbf{t}_i|\mathbf{x}_i$. The conditional density function is not restricted to be Gaussian, but that is the most common choice and works well in practice. It is given in equation 2:

$$\phi_m(\mathbf{t}_i|\mathbf{x}_i) = \frac{1}{(2\pi)^{c/2}\sigma_m(\mathbf{x}_i)^c} exp\Big\{ - \frac{||\mathbf{t}_i - \boldsymbol{\mu}_m(\mathbf{x}_i)||^2}{2\sigma_m(\mathbf{x}_i)^2} \Big\} \tag{2}$$

In equation 2, c is the dimension of the target vector, $\boldsymbol{\mu}_m$ is the component mean and σ_m is the common variance for the elements of the target. The variance term does not have to be shared between dimensions of target space, and can be replaced with a diagonal or full co-variance matrix if necessary Bishop (1994).

3.2 MDNS APPLIED TO HUMAN POSE ESTIMATION

Given an image with an unspecified number of people in uncontrolled poses, the goal of human pose estimation is to localize a predefined set of keypoints for each person and have them grouped together per person. We approach this problem using a mixed bottom-up and top-down formulation very recently proposed in Zhou et al. (2019a). In this formulation, unlike the top-down methods there is no need to use an object detector to localize the person instance first. And unlike bottom-up methods, the grouping is not left as a post-processing step. Rather, at a given spatial location, the model predicts if it is the central pixel of a person, and at the same location, for each keypoint it generates an offsets vector to the keypoint location.

This formulation takes the best of both approaches: it is fast like a bottom-up method, and post-processing free as in a top-down model. At least equally important is the fact that it enables applying

many advanced techniques in an end-to-end manner. As a case in point, in this paper it allows us to perform density estimation for human pose in a dense fashion. That is, in a single forward pass through the network, we estimate the parameters of a density estimation model. In particular, we use the mixture density model to learn the probability density of poses conditioned on an input image.

Formally, Zhou et al. (2019a) start from an input RGB image I of size $H * W * 3$, and a CNN that receives I and generates an output with height H', width W', and C' channels. If we indicate the downsampling factor of the network with D, then we have $H = D * H'$, and similarly for width. We refer to the set of output pixels as P'. Given the input, the network generates a dense 2D classification map C to determine instance centers, i.e. $C_{p'}$ indicates the probability of location $p' \in P'$ corresponding to the center of a person instance. Simultaneously, at p', the network predicts K 2D offset vectors $O = [o^0_{p,x}, o^0_{p,y}, \ldots, o^{K-1}_{p,x}, o^{K-1}_{p,y}]$, where K is the number of keypoints that should be localized. Once the network classifies p' as a person's central pixel, the location of each keypoint is directly given by the offset vectors O.

In the literature, it is common to train for offset regression O using L_1 loss Papandreou et al. (2018); Kreiss et al. (2019); Cao et al. (2017); Zhou et al. (2019a). However, spatial regression is a multi-modal task and having a single set of outputs will lead to a sub-optimal prediction, in particular when precise localization is important. With this in mind, we use mixture density networks to model the offset regression task. In this case, $\boldsymbol{\mu}_m$ from equation 2 would be used to represent offsets predicted by different components. Then the density of the ground truth offset vectors G conditioned on image I is given by equation 3, where the density ϕ_m for each component is given by equation 4. Here $O_m(I)$ is the input dependent network output function for component m that generates offsets and $G = [g^0_{p,x}, g^0_{p,y}, \ldots, g^{K-1}_{p,x}, g^{K-1}_{p,y}]$ indicates the ground truth offsets. $\sigma_m(I)$ is the standard deviation of the component m in two dimensions, X and Y. It is shared by all keypoints of an instance. However, in order to account for scale differences of keypoints, in equation 4 for each keypoint we divided $\sigma_m(I)$ by its scale factor provided in COCO dataset. In this framework, the keypoints are independent within each component, but the full model does not assume such independence.

$$p(G|I) = \sum_{m=1}^{M} \alpha_m(I)\phi_m(G|I) \tag{3}$$

$$\phi_m(G|I) = \frac{1}{(2\pi)^{c/2}\sigma_m(I)^c} exp\Big\{ - \frac{||G - O_m(I)||^2}{2\sigma_m(I)^2} \Big\} \tag{4}$$

Given the conditionally probability density of the ground truth in equation 3, we can define the loss using the negative log likelihood formulation and minimize it using stochastic gradient descent. The loss for MDN is given in equation 5. Here N is the number of samples in the dataset. Practically, this loss term replaces the popular L_1 loss. Please note that MDN is implemented in a dense fashion, that density estimation is done independently at each spatial location $p' \in P'$. A schematic overview of the model is shown in Figure 1.

$$L_{MDN} = \sum_{i=1}^{N} -\ln \sum_{m=1}^{M} \alpha_m(I_i)\phi_m(G_i|I_i) \tag{5}$$

We do not modify the other loss terms used in Zhou et al. (2019a). This includes a binary classification loss L_C, a keypoint heatmap loss L_{HM} (used for refinement), a small offset regression loss to compensate for lost spatial precision due to downsampling for both center and keypoints $L_{C_{off}}$ and $L_{KP_{off}}$, and a loss for instance size regression L_{wh}. The total loss is given in equation 6:

$$L_{total} = L_C + 0.1L_{MDN} + L_{HM} + L_{C_{off}} + L_{KP_{off}} + 0.1L_{wh} \tag{6}$$

3.3 Inference

Once the network is trained, at each spatial location, C will determine if that is the center of a person (the bounding box center is used for training). Each MDN component at that location will generate

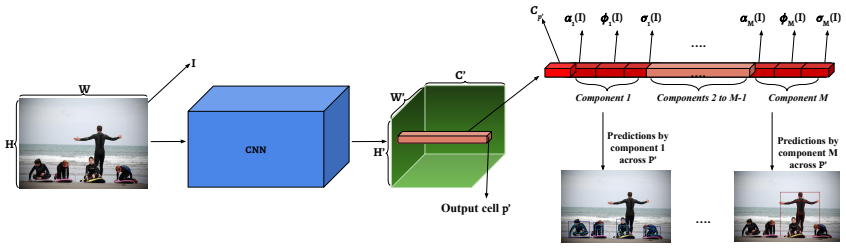

Figure 1: Schematic overview of our proposed solution using mixture density networks.

offsets conditioned on the input. To obtain the final offset vectors, we can use either the mixture of the components or the component with the highest probability. We do experiments with both and using the maximum component leads to slightly better results. Once we visually investigate what modes the components have learned, they seem to have very small overlap. Hence, it is not surprising that both approaches have similar performance Bishop (1994).

4 Experiments

We train and test our model on MS COCO 2017 dataset Lin et al. (2014). For training, we use 64k images that have at least one person instance (*coco-train*). For ablation studies we use the standard validation set containing 5k images (*coco-val*). At training, for computing the likelihood, we ignore the keypoints that are not annotated. Proper initialization of the variance is very important for successful training of the network. When initialized just to be positive, it leads to instability and mode collapse Cui et al. (2019); Rupprecht et al. (2017); Li & Lee (2019). So we initialize the variance such that the smallest value is 10. This is decided in relation to the dataset. To generate variance values, we use the exponential linear units (ELU) Clevert et al. (2015), but modify it such that minimum values is 10. We did experiment with smaller and larger values for minimum, but did not observe any significant difference. To avoid numerical issues, we have implemented the log likelihood using the LogSumExp function.

Our implementation is on top of the code base published by Zhou et al. (2019a), and we use their model as the base in our comparisons. The network architecture is based on a version of stacked hourglass Newell et al. (2016) presented in Law & Deng (2018). We refer to this architecture as LargeHG. To analyse effect of model capacity, we also conduct experiments with smaller variants. The SmallHG architecture is obtained by replacing the residual layers with convolutional layers, and XSmallHG is obtained by further removing one layer from each hourglass level.

Unless stated otherwise, all models are trained for 50 epochs (1X schedule) using batch size 12 and ADAM optimizer Kingma & Ba (2014) with learning rate 2.5e-4. Only for visualization and comparison to state-of-the-art we use a version of our model trained for 150 epochs (3X). Except for comparison with the state-of-the-art, we have re-trained the base model to assure fair comparison.

4.1 Number of components

To analyse effect of number of components, we train on XSmallHG and SmallHG architectures with up to 5 components, and on LargeHG architecture with up to 3 components. Table 1 shows the evaluation results for various models on the *coco-val*. The table also shows the evaluation for MDN models when we ignore the predictions by particular components. We report predictions with and without using the extra heatmap based refinement deployed in Zhou et al. (2019a). This refinement is a post-processing step, which tries to remedy the lack of enough precision in offset regression by pushing the detected keypoints towards the nearest detection from the keypoint heatmaps.

It is clear that MDN leads to a significant improvement. Interestingly, only two modes will be retrieved, no matter how many components we train and how big the network is. Having more than two components results in slightly better recall, but it will not improve precision. Only when the network capacity is very low more than two component seems to have significant contribution

Table 1: Evaluations of models trained for 50 epochs. MDN_X, indicates MDN with X components, "w/o C_X" indicates average precision when predictions by component X are ignored.

		Average Precision (AP)/ Average Recall(AR)						
Architecture	**Model**		w/ refinement	w/o C_1	w/o C_2	w/o C_3	w/o C_4	w/o C_5
XSmallHG	Base	40.3/50.2	48.5/58.5	-	-	-	-	-
XSmallHG	MDN_1	41.1/50.6	49.1/58.3	-	-	-	-	-
XSmallHG	MDN_2	43.7/53.3	50.3/59.9	33.9/40.5	11.3/13.3	-	-	-
XSmallHG	MDN_3	44.4/54.1	50.8/60.5	44.5/54.2	12.0/16.4	33.5/39.9	-	-
XSmallHG	MDN_5	42.9/53.3	50.1/60.1	37.8/47.4	18.6/30.3	33.2/44.9	40.9/50.4	43.1/53.3
SmallHG	Base	41.2/50.9	48.6/58.6	-	-	-	-	-
SmallHG	MDN_1	43.4/53.7	51.3/60.9	-	-	-	-	-
SmallHG	MDN_2	47.9/58.0	54.3/64.0	11.8/14.2	36.9/66.1	-	-	-
SmallHG	MDN_3	48.9/59.5	54.6/64.4	12.8/17.1	36.6/44.5	49.3/59.5	-	-
SmallHG	MDN_5	48.8/59.4	54.8/64.9	48.5/58.7	48.8/48.9	36.7/44.4	49.2/59.4	13.2/18.0
LaregHG	Base	46.4/56.1	54.0/63.2	-	-	-	-	-
LaregHG	MDN_1	47.7/57.6	55.1/64.2	-	-	-	-	-
LaregHG	MDN_2	52.3/62.0	57.7/66.8	40.6/48.1	12.4/14.4	-	-	-
LaregHG	MDN_3	52.3/62.7	57.2/67.3	12.9/17.3	52.4/62.8	40.7/47.8	-	-

to the predictions. On the XSmallHG with MDN_5, four of the five components have significant contribution to the final predictions. Although, two of them have much more significant effect. Visualizing prediction by various models, makes it clear that one of the modes focuses on frontal view instances, and the other one on the instance with backward view. Figure 2 shows sample visualisation from MDN_3 model trained with 3X schedule.

We further evaluate the MDN_2 trained on the LargeHG on various subsets of the COCO validation split by ignoring predictions by each of components or forcing all predictions to be made by a particular component. The detailed evaluations are presented in table 2. The results show that the components correlate well with face viability, confirming the conclusion we make by visualising predictions. It is worth noting that although we use annotation of nose as indicator of face visibility, it is noisy, as in some case the person is view from side such that nose is just barely visible and the side view is very close to back view (like the first image in the third row of Figure 2). But, even this noisy split is enough to show that two modes are chose based on viewpoint.

Figure 2: Sample predictions by the MDN_3 on *coco-val*. Instance scale variance (in X and Y dimensions) is shown by an ellipse around the predicted center. Blue and red color of boxes and ellipses represent the two different modes (the third mode is redundant as shown in Table 1).

Table 2: AP and AR on different subsets of *coco-val* for MDN_2 trained on LargeHG.

Components used for evaluation	***coco-val* subset** All	Visible Keypoints	Occluded Keypoints	Visible Face	Occluded Face
Full model	53.2/62.0	56.3/65.5	38.0/50.4	63.4/70.8	52.8/60.9
Keep C_1 predictions only	12.5/14.4	13.1/15.1	9.6/12.3	4.0/4.3	37.0/40.8
Keep C_2 predictions only	41.4/48.1	44.0/51.1	29.6/39.7	60.5/67.0	16.4/20.7
All with C_1	38.0/49.9	40.8/53.4	27.8/41.3	43.9/55.0	45.1/54.2
All with C_2	38.4/52.1	41.1/55.5	29.0/43.2	60.6/69.0	16.7/30.0

Subset name	**Description**
All	The full coco validation split
Visible Keypoints	All keypoints that are occluded and annotated are ignored
Occluded Keypoints	All keypoints that are visible and annotated are ignored
Visible Face	Instances with at least 5 annotated keypoints where nose is visible and annotated
Occluded Face	Instances with at least 5 annotated keypoints where nose is occluded or not annotated

Table 3: Statistics for *coco-val* subsets and MDN max component. For face visibility, instances with more than 5 annotated keypoints (in parentheses for minimum of 10) are used. For components, predictions with score at least .5 are considered (in parentheses for minimum of .7).

Occluded Keypoints	**Visible Keypoints**	**Occluded Face**	**Visible Face**	***comp*$_1$ (back view)**	***comp*$_2$ (front view)**
12.3	87.7	30.3 (22.1)	69.7 (77.9)	25.5 (27.0)	74.5 (70.0)

Table 3 compares portion of the dataset each subset comprises against portion of predictions made by each component of MDN_2. Obviously, the component statistics correlates well with the face visibility, which in fact is an indicator of viewpoint in 2D. Majority of instances in the dataset are in frontal view, and similarly the front view component makes majority of the prediction.

Related to our results, Belagiannis & Zisserman (2017) have shown that excluding occluded keypoints from training (by treating them as background) leads to improved performance. More recently, Ye & Kim (2018) achieves more accurate 3D habd pose estimation by proposing a model that directly predicts occlusion of a keypoint in order to use it for selecting a downstream model. And, here we illustrate that occlusion caused by viewpoint imposes more challenge to spatial regression models, than other possible factors, like variation in pose itself.

4.2 L1 vs Normalized L2

It is common to train offset regression targets with L1 loss Zhou et al. (2019a); Papandreou et al. (2018); Law & Deng (2018). In contrast, the single component version of our model is technically equal to L2 loss normalized by the instance scale learned via MDN variance terms. This is in fact equal to directly optimizing the MS COCO OKS scoring metric for human pose estimation. Comparing the performance of the two losses in Table 3, we see normalized L2 yields superior results. That is, for any capacity, MDN_1 outperforms the base model which is trained using L1.

4.3 Fine grained evaluation

For a deeper insight on what body parts gain the most from the MDN, we do fine grained evaluation for various keypoint subsets. In doing so, we modify the COCO evaluation script such that it only considers set of keypoints we are interested in. Table 4 shows the results. For the facial keypoints where the metric is the most sensitive, the improvement is higher. nevertheless, the highest improvement comes for the wrists, which have the highest freedom to move. On the other hand, for torso keypoints (shoulders and hips) which are the most rigid, there is almost no different in comparison to base model.

4.4 Explicit binary classification

Given that MDN revels two modes, we build a hierarchical model by doing a binary classification and using it to choose from two separate full MDN models. The goal is to see, if binary classification

Table 4: Fine grained evaluation on *coco-val* in terms of average precision.

	All	Facial	None Facial	Nose	Eyes	Ears	Shoulders	Elbows	Wrists	Hips	Knees	Ankles
Ground truth displaced by 1 pixel	96.0	83.7	99.6	77.9	76.5	88.7	99.4	99.0	97.1	99.2	97.4	95.2
Ground truth displaced by 2 pixels	80.4	47.4	93.4	42.3	39.8	59.8	93.4	91.0	86.8	97.2	93.1	91.7
Ground truth displaced by 3 pixels	63.0	25.6	82.7	22.6	21.0	35.8	82.0	78.3	71.2	91.3	84.6	83.3
Base model	46.4	44.3	45.7	42.3	42.2	43.9	59.5	43.6	31.7	58.5	44.1	38.2
$\boldsymbol{MDN_2}$	52.3	54.1	49.9	50.3	50.5	54.0	60.3	50.8	41.7	58.3	47.6	42.6
Relative improvement	12.7 %	22.1%	9.2 %	18.9%	19.7%	23.0%	1.3%	16.5%	31.5%	-0.3%	7.9%	11.5%

works as well as MDN_2. Note that the classification is not directly supervised, as we do not have any such annotation. This model also divides the data analogous to MDN_2. However, on the SmallHG architecture, MDN_2 reaches AP of 47.9 and the hierarchical version achieves 46.5 (with refinement and flip test AP is 55.6 vs 54.7). Therefore, a two component MDN that learns full conditional probability density and assumes dependence between target dimensions delivers higher performance.

$$p(G|I) = v_1 * \phi_{m_1}(G|I) + v_2 * \phi_{m_2}(G|I)$$
$$s.t.\ v_1 + v_2 = 1,\ v_1 \geq 0,\ v_2 \geq 0 \tag{7}$$

4.5 Comparison to the state-of-the-art

For comparison to the state-of-the-art in human pose estimation, we train MDN_1 and MDN_3 for 150 epochs using the LaregHG architecture, and test it on COCO test-dev split. The results are presentetd in Table 5. Using MDN significantly improves the offset regression accuracy (row 6 vs row 10 of the table). When refined, both models achieve similar performance.

In contrast to all other state-of-the-art models, MDNs performance drops if we deploy the ad-hoc left-right flip augmentation at inference time. This is a direct consequence of using a multi-modal prediction model which learns to deal with viewpoint. It is important to note that left-right flip is used widely for increasing accuracy at test time for object detection and segmentation tasks as well. Therefore, we expect our method to improve performance for those tasks as well.

MDN_1 with refinement gives slightly lower accuracy than the base model. Our investigation attributes this discrepancy to a difference in the training batch size. The official base model is trained with batch size 24, but we train all models with batch size 12, due to limited resources. Under the same training setting, MDN_1 will outperform the base model, as shown in Table 1.

Table 5: Performance on *COCO test-dev*. Unless stated otherwise, all results correspond to single-scale inference with left-right flip augmentation. The top rows belong to top-down models, and the middle rows to bottom-up models.

model	AP	AP_{50}	AP_{75}	AP_M	AP_L
Mask R-CNN He et al. (2017)	63.1	87.3	68.7	57.8	71.4
Simple baselines Xiao et al. (2018)	73.7	91.9	81.1	70.3	80.0
PersonLab Papandreou et al. (2018)	66.5	88.0	72.6	62.4	72.3
OpenPose Cao et al. (2018)	64.2	86.2	70.1	61.0	68.8
AssociativeEmbedding Newell et al. (2017)	62.8	84.6	69.2	57.5	70.6
Base model Zhou et al. (2019a)	55.0	83.5	59.7	49.4	64.0
Base model refined Zhou et al. (2019a)	63.0	86.8	69.6	58.9	70.4
MDN_1 w/o left-right flip (ours)	55.1	80.3	60.7	53.0	61.7
MDN_1 refined (ours)	61.1	83.7	67.6	57.2	69.7
MDN_3 (ours)	57.9	82.7	63.7	52.3	67.8
MDN_3 w/o left-right flip (ours)	59.0	82.7	65.3	56.4	65.9
MDN_3 refined (ours)	62.9	85.1	69.4	58.8	71.4

5 Discussion

We have shown mixture density models significantly improve spatial offset regression accuracy. Further, we have demonstrate that MDNs can be deployed on real world data for conditional density

estimation without facing mode collapse. Analyzing the ground truth data and revealed modes, we have observe that in fact MDN picks up on a mode, that significantly contributes to achieving higher accuracy and it can not be incorporated in a single mode model. In the case of human pose estimation, it is surprising that viewpoint is the dominant factor, and not the pose variation. This stresses the fact that real world data is multi-modal, but not necessarily in the way we expect. Without a principled approach like MDNs, it is difficult to determine the most dominant factors in a data distribution.

A stark difference between our work and others who have used mixture models is the training data. Most of the works reporting mode collapse rely on small and controlled datasets for training. But here we show that when there is a large and diverse dataset, just by careful initialization of parameters, MDNs can be trained without any major instability issues. We have made it clear that one can actually use a fully standalone multi-hypothesis model in a real-world scenario without the need to rely on an oracle or postponing model selection to a downstream task.

We think there is potential to learn more finer modes from the dataset, maybe on the pose variance, but this needs further research. Specially, it will be very helpful if the role of training data diversity could be analysed theoretically. At the same time, the sparsity of revealed modes also reminds us of the sparsity of latent representations in generative models Xu et al. (2019). We attribute this to the fact that deep models, even without advanced special prediction mechanism, are powerful enough to deliver fairly high quality results on the current datasets. Perhaps, a much needed future direction is applying density estimation models to fundamentally more challenging tasks like the very recent large vocabulary instance segmentation task Gupta et al. (2019).

REFERENCES

Vasileios Belagiannis and Andrew Zisserman. Recurrent human pose estimation. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pp. 468–475. IEEE, 2017.

Christopher M Bishop. Mixture density networks. 1994.

Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7291–7299, 2017.

Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018.

Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7103–7112, 2018.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

Henggang Cui, Vladan Radosavljevic, Fang-Chieh Chou, Tsung-Han Lin, Thi Nguyen, Tzu-Kuo Huang, Jeff Schneider, and Nemanja Djuric. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 2090–2096. IEEE, 2019.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5356–5364, 2019.

Abner Guzman-Rivera, Dhruv Batra, and Pushmeet Kohli. Multiple choice learning: Learning to produce multiple structured outputs. In *Advances in Neural Information Processing Systems*, pp. 1799–1807, 2012.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

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

Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Pifpaf: Composite fields for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11977–11986, 2019.

Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 734–750, 2018.

Stefan Lee, Senthil Purushwalkam, Michael Cogswell, David Crandall, and Dhruv Batra. Why m heads are better than one: Training a diverse ensemble of deep networks. *arXiv preprint arXiv:1511.06314*, 2015.

Stefan Lee, Senthil Purushwalkam Shiva Prakash, Michael Cogswell, Viresh Ranjan, David Crandall, and Dhruv Batra. Stochastic multiple choice learning for training diverse deep ensembles. In *Advances in Neural Information Processing Systems*, pp. 2119–2127, 2016.

Chen Li and Gim Hee Lee. Generating multiple hypotheses for 3d human pose estimation with mixture density network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9887–9895, 2019.

Wenbo Li, Zhicheng Wang, Binyi Yin, Qixiang Peng, Yuming Du, Tianzi Xiao, Gang Yu, Hongtao Lu, Yichen Wei, and Jian Sun. Rethinking on multi-stage networks for human pose estimation. *arXiv preprint arXiv:1901.00148*, 2019.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pp. 21–37. Springer, 2016.

Geoffrey J McLachlan and Kaye E Basford. *Mixture models: Inference and applications to clustering*, volume 84. M. Dekker New York, 1988.

Davy Neven, Bert De Brabandere, Marc Proesmans, and Luc Van Gool. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8837–8845, 2019.

Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pp. 483–499. Springer, 2016.

Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Advances in Neural Information Processing Systems*, pp. 2277–2287, 2017.

David Novotny, Samuel Albanie, Diane Larlus, and Andrea Vedaldi. Semi-convolutional operators for instance segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 86–102, 2018.

George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 269–286, 2018.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

Christian Rupprecht, Iro Laina, Robert DiPietro, Maximilian Baust, Federico Tombari, Nassir Navab, and Gregory D Hager. Learning in an uncertain world: Representing ambiguity through multiple hypotheses. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3591–3600, 2017.

Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 466–481, 2018.

Zhenjia Xu, Zhijian Liu, Chen Sun, Kevin Murphy, William T Freeman, Joshua B Tenenbaum, and Jiajun Wu. Unsupervised discovery of parts, structure, and dynamics. *arXiv preprint arXiv:1903.05136*, 2019.

Qi Ye and Tae-Kyun Kim. Occlusion-aware hand pose estimation using hierarchical mixture density network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 801–817, 2018.

Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019a.

Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-up object detection by grouping extreme and center points. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 850–859, 2019b.