VL4Pose: Supplementary material

Megh Shukla ✉ [1]
megh.shukla@epfl.ch
Roshan Roy *[3]
roshan.roy@lmco.com
Pankaj Singh *[2]
pankaj.singh@mercedes-benz.com
Shuaib Ahmed [2]
shuaib.ahmed@mercedes-benz.com
Alexandre Alahi [1]
alexandre.alahi@epfl.ch

[1] Visual Intelligence for Transportation Lab
École Polytechnique Fédérale de Lausanne
Lausanne, Switzerland

[2] Mercedes-Benz Research and Development India
Bengaluru, India

[3] Lockheed Martin Corporation
New Jersey, USA

Abstract

This supplementary material covers the following: 1) Deriving our expected likelihood formulation 2) Short discussion of aleatoric uncertainty in pose estimation 3) Description of the method to visualize our conditional distribution 4) Additional images for likelihood estimation and pose refinement 5) Algorithm implementation: VL4Pose

1 Likelihood Estimation

Our skeleton formulation allows us to describe the distribution over joints as:

$$q_{BN}(y_1, y_2 \dots y_N | x, \theta) = \left[\prod_{i=1}^{N-1} q(y_i | y_{i+1}, x, \theta) \right] q(y_N | x, \theta) \tag{1}$$

Here 'N' represents the number of joints with $q(y_N|x,\theta)$ representing the distribution of the root node, such as the head joint. We have also defined the pose estimator's distribution over the joints:

$$p_{pose}(Y) = p_{pose}(y_1, y_2 \dots y_N) = \prod_{i=1}^{N} p(y_i) \tag{2}$$

As previously noted, our assumption of independence is in line with the training objective of popular pose estimators [1, 5, 6]. We also note that Y denotes the set of random variables $y_1 \dots y_N$. The expected log-likelihood *w.r.t* the set of keypoints is:

$$\mathbb{E}_Y \left[\texttt{log}\; q_{BN}(y_1, y_2 \dots y_N | x, \theta) \right] \tag{3}$$

Substituting Eq: 1 in Eq: 3 and expanding, we get:

$$\mathbb{E}_Y\left[\texttt{log}\, q_{BN}(y_N|x,\theta) + \sum_{i}^{N-1} \texttt{log}\, q_{BN}(y_i|y_{i+1},X,\theta)\right] \tag{4}$$

For human pose, the domain of $p(Y)$ represents all possible positions in the heatmap across all joints, which is intractable to compute. Hence, we limit the domain to local maxima in the heatmap for all joints. To ensure that the resultant distribution is valid, we normalize the local maxima within a heatmap to sum upto one. For hand pose, the network does not predict a distribution but provides a point estimate of y, limiting the domain to one pose configuration only. Therefore we can represent Eq: 4 as:

$$\sum_{Y}\left[\texttt{log}\, q_{BN}(y_N|x,\theta)\prod_{i=1}^{N} p(y_i) + \sum_{i}^{N-1} \texttt{log}\, q_{BN}(y_i|y_{i+1},X,\theta)\prod_{i=1}^{N} p(y_i)\right] \tag{5}$$

We make one important note: $q(y_i|\ldots)$ depends only on $p(y_i)$, therefore $\sum_Y \prod_{j=1}^{N} p(y_j) = 1$ where $j \neq i$. Hence, we arrive at our final formulation:

$$\sum_{Y}\left[p_{pose}(y_N)\texttt{log}\, q_{BN}(y_N|x,\theta) + \sum_{i}^{N-1} p_{pose}(y_i)\texttt{log}\, q_{BN}(y_i|y_{i+1},X,\theta)\right] \tag{6}$$

2 Aleatoric Uncertainty For Pose Estimation

Since our approach is similar to that in [1, 2, 4], one might be tempted to brand this method as aleatoric uncertainty. However, we consciously refrain from doing so. Aleatoric uncertainty represents the noise inherent in our data which cannot be reduced by increasing the samples drawn. While [2, 4] and [1] further define aleatoric uncertainty for human and hand pose respectively, we believe that aleatoric uncertainty for keypoints is incorrectly represented in this literature.

Caramalau *et al.* [1] directly extends [3] to hand keypoint estimation, thereby solving $p(\texttt{joints}|X,\Theta) = \prod_i p(y_i|X,\theta)$. The assumption that all joints are independent of each other is incorrect when computing aleatoric uncertainty. Observing any one of these variable results in drastic uncertainty reduction for the unobserved counterpart, going against the principle of aleatoric uncertainty.

Works such as [2, 4] model aleatoric uncertainty as a multivariate normal distribution over joints Y. However, this uncertainty is reducible by observing more data and thus not aleatoric in a strict sense. For instance, rare poses are difficult to learn and hence the any network estimates a covariance matrix that reflects the uncertainty in the model's predictions. However, if we sample and train the model on more of such rare poses, it is expected that the model performance improves on these poses thereby reducing the associated uncertainty. This is in conflict with the definition of aleatoric uncertainty. Hence, we refrain from categorizing our uncertainty measures as aleatoric or epistemic since more investigation is required into sources of uncertainty for human pose.

3 Visualizing the conditional distribution

Visualizing the offset based conditional distribution for hand pose estimation is trivial. The normal distribution for the child joint is centred around the point determined by the parent joint adjusted with the predicted offset. Instead of visualizing in 3D, we visualize the distribution in 2D which is better suited for print media. This requires marginalizing over the depth d since we wish to preserve the spatial representation for the multivariate normal distribution. Fortunately for us, marginalizing over a multivariate normal distribution is equivalent to dropping the variable being marginalized from the mean and covariance matrix of the distribution. Therefore, the resultant spatial normal distribution obtained by marginalizing the depth is straightforward to visualize in 2D.

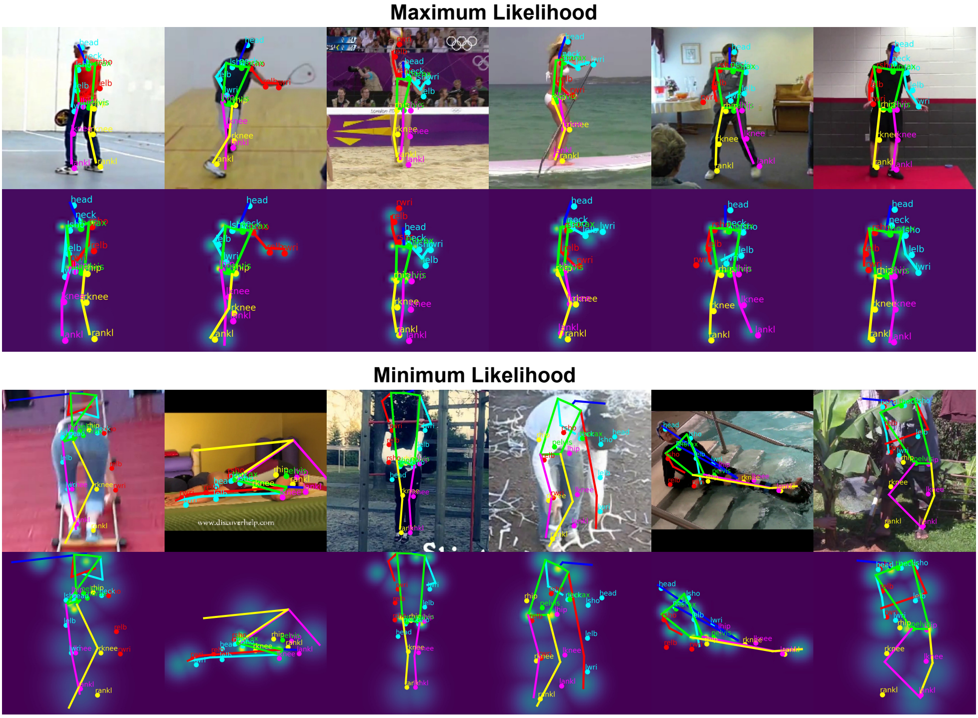
In contrast, visualizing the distance based conditional distribution for human pose estimation is tricky. Viewing the distribution as a ring for all the skeletal links with radius, thickness as per the predicted mean, variance soon results in an overlapping non-informative visualization. Instead, for each link we plot a univariate gaussian in 2D with its centre located at the predicted distance along the line joining the parent and the child. Our reasoning for following this approach is based on the triangle inequality, where the difference between the predicted and actual distance is the lowest when the predicted gaussian lies along the same axis as the parent-child joints. Fortunately, this approach is easier for the visualization and analysis of multiple joints simultaneously as shown in the paper.

4 Algorithm Implementation: VL4Pose

Algorithm : 1 provides a pseudo-code for implementing VL4Pose. The essence of the pseudo-code lies in depth first search to evaluate the likelihood for various poses. The directed acyclic graph is represented as a tree with each node representing the joint. Each joint is associated with peaks and locations (obtained from joint heatmap) as well as parameters associated with the parent-child distribution. The pseudo-code recursively evaluates the combination of joints which results in the highest expected likelihood. The pseudo-code can be easily tweaked to obtain the highest likelihood as well as pose from the resultant heatmaps and conditional distributions.

5 Visualizations

We present more images depicting likelihood estimation (Fig: 1) and pose refinement (Fig: 2) using VL4Pose.


Maximum Likelihood

Minimum Likelihood

Figure 1: [Please zoom in] *Visualizing* $q_{BN}(y_i|y_{i+1},x,\theta)$: The skeleton represents the pose estimator's predictions $\hat{Y} = f(x,\Theta)$ and filled circles are the ground truth Y. We highlight the correlation between pose uncertainty and likelihood, and likelihood with actual model performance.

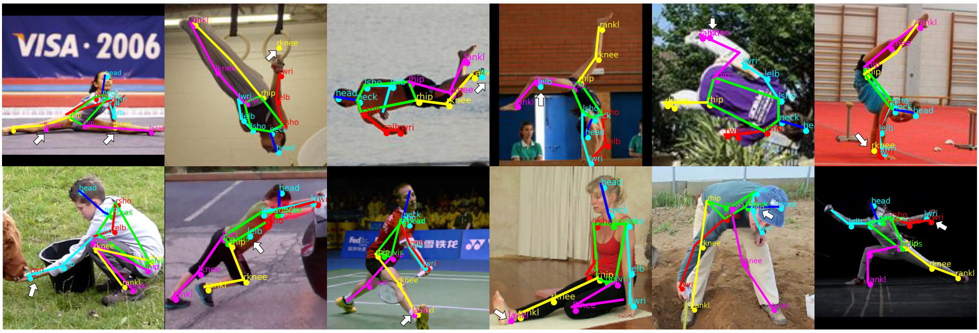


Figure 2: [Please zoom in] *Pose refinement*: The skeleton represents the optimal pose configuration Y^* that maximizes the likelihood, and filled circles are the the pose estimator's predictions $\hat{Y} = f(x,\Theta)$. We highlight VL4Pose's potential to identify the correct pose Y^* even when $\hat{Y}$ has <u>minor</u> errors (marked in arrows).

Algorithm 1: ***VL4Key***

```
Input: Human pose estimator ($f_\Theta$), Auxiliary network ($g_\theta$), Budget ($\mathcal{B}$)
Output: Unlabelled images ($x^*_\mathcal{U}$) for annotation
Data: Unlabelled ($x_\mathcal{U}$) images

skeleton = [ head $\rightarrow$ neck ; neck $\rightarrow$ thorax ... ]        // length: num_links

class Keypoint:
    /* Default initialization for each joint */
    function __init__ () :
        string name
        list locations, peaks, children, parameters

    /* DFS: Depth First Search likelihood evaluation */
    function compute_likelihood (parent_loc, link_params) :
        empty list max_likelihood_per_location
        for i, loc in enumerate(self.locations) do
            if len (self.children) == 0 then
                /* Leaf node reached: recursion exit condition */
                return 0
            else
                /* Evaluate position of self given parent location */
                log_ll = log $\mathcal{N}$ (dist (parent_loc , loc) ; link_params)
            /* log prob: log $\hat{p}_i(y_i[u,v] = loc)$ where $y_i$ is heatmap $i$ */
            log_ll += log peaks[i]
            for child in self.children do
                log_ll += child.compute_likelihood (loc, parameters[i])
            max_likelihood_per_location.append(log_ll)
        return max ( max_likelihood_per_location )

initialize likelihoods = empty_array(size = $x_\mathcal{U}$.shape[0])

/* GPU parallel since each image is independent of the other */
for i, x in enumerate($x_\mathcal{U}$) do
    $\bar{y} = f(x, \Theta)$                    // heatmaps of size: num_joints × 64 × 64
    params $= g(x, \theta)$                     // Gaussian parameters: num_links × 2
    locations, peaks = local_maxima ($\bar{y}$)
    keypoints_holder = dict()

    for j, joint in enumerate(joints) do
        keypoints_holder [joint] = Keypoint (name=joint, locations[j], peaks[j])

    for j, link in enumerate(skeleton) do
        parent = link [0]
        child = link [1]
        keypoints_holder [parent].children.append (keypoints_holder [child])
        keypoints_holder [parent] = params [j]
    likelihoods [i] = keypoint_holder [‘head’].compute_likelihood()
return $x^*_\mathcal{U}$: Return samples corresponding to bottom - $\mathcal{B}$ likelihoods
```

References

[1] Razvan Caramalau, Binod Bhattarai, and Tae-Kyun Kim. Active learning for bayesian 3d hand pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3419–3428, 2020.

[2] Nitesh B Gundavarapu, Divyansh Srivastava, Rahul Mitra, Abhishek Sharma, and Arjun Jain. Structured aleatoric uncertainty in human pose estimation. In *CVPR Workshops*, volume 2, 2019.

[3] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584, 2017.

[4] Changsheng Lu and Piotr Koniusz. Few-shot keypoint detection with uncertainty learning for unseen species. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19416–19426, 2022.

[5] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 483–499, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46484-8.

[6] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.