Supplementary Material for Multi-Modal 3D Human Pose Estimation with 2D Weak Supervision in Autonomous Driving

Jingxiao Zheng[1] Xinwei Shi[1] Alexander Gorban[1] Junhua Mao[1] Yang Song[1]
Charles R. Qi[1] Ting Liu[2] Visesh Chari[1] Andre Cornman[1] Yin Zhou[1]
Congcong Li[1] Dragomir Anguelov[1]
[1] Waymo LLC [2] Google Research
{jingxiaozheng, xinweis, gorban, junhuamao, yangsong, rqi}@waymo.com,
liuti@google.com, {visesh, cornman, yinzhou, congcongli, dragomir}@waymo.com

1. Training Losses for Section 3.5

Point Network: The training loss for the regression branch is a weighted Huber loss defined as

$$L_{\text{reg}} = \frac{1}{K}\sum_{k=1}^{K} v_k r_k L_{\text{Huber}}\left(\frac{\hat{\mathbf{y}}_k^{(3)} - \tilde{\mathbf{y}}_k^{(3)}}{s_k}\right) \quad (1)$$

where $\hat{\mathbf{y}}_k^{(3)}$ is the 3D prediction from the model, $\tilde{\mathbf{y}}_k^{(3)}$ is the pseudo 3D label by label generation in Equation (1) in the main paper, v_k is the visibility label of keypoint k (0-1 valued), r_k is the label reliability (Section 3.4.1 in the main paper), and s_k is the scaling factor of keypoint k. The loss is only applied on visible keypoints with v_k being 1. Keypoints that need to be more accurately localized have smaller scales s_k (therefore larger weights) during training.

The loss for the segmentation branch is a weighted cross-entropy loss defined as

$$L_{\text{seg}} = \frac{1}{K}\sum_{i=1}^{N}\sum_{k=1}^{K} v_k \{w_{\text{pos}} l_{ik}\log p_{ik} + w_{\text{neg}}(1-l_{ik})\log(1-p_{ik})\} \quad (2)$$

where v_k is the visibility label of keypoint k, and w_{pos} and w_{neg} are the weights balancing the positive and negative samples. w_{pos} is usually much larger than w_{neg} since for each keypoint there are much more points with negative labels than points with positive labels.

The overall loss for the point network is

$$L = L_{\text{reg}} + \lambda L_{\text{seg}} \quad (3)$$

where λ is used to weigh the auxiliary segmentation loss.

Camera Network: Similar to [2], the camera network is trained on a mean-squared-error loss with ground truth 2D heatmap as

$$L_{\text{cam}} = \frac{1}{H'W'K}\sum_{i,j=1}^{H',W'}\sum_{k=1}^{K} v_k (h_{i,j,k} - g_{i,j,k})^2 \quad (4)$$

where v_k is the visibility label of keypoint k, and $g_{i,j,k}$ is the ground truth heatmap generated by Gaussian functions centered at 2D ground truth keypoints. We train the camera network independently, then freeze it during point network training.

2. Metrics for Section 4.1

2.1. OKS/ACC Metric

This paper focuses on pose estimation instead of keypoint detection by assuming that the person has been successfully detected and there is exactly one estimated pose for each ground-truth pose. Therefore, instead of using the OKS/AP metric defined in COCO keypoint challenge [1], we introduce a modified OKS/ACC metric for evaluation:

$$ACC^{OKS=t} = \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}\{\text{OKS}_n \geq t\} \quad (5)$$

where t is the threshold on OKS, N is the total number of samples in the test set, and OKS_n is the OKS of prediction on sample n. In our experiments we averaged OKS/ACC over t from 0.5 to 0.95 with a step-size of 0.05.

2.2. Per-keypoint OKS

Per-keypoint OKS is defined as OKS [1] for one keypoint type. For keypoint type i,

$$OKS = \frac{\exp(-d_i^2/2s^2k_i^2)\delta(v_i > 0)}{\delta(v_i > 0) + \epsilon} \quad (6)$$

where d_i is the distance between ground truth and prediction, v_i is visibility of the ground truth, s is the object scale, k_i is a per-keypoint constant, and ϵ is a small number here to prevent zero denominator.

3. Implementation Details for Section 4.1

The following are some details on training. For the camera network, we resize all input images to 256×256. The output heatmap size is $64 \times 64 \times 13$ (13 keypoints are predicted). The camera network is trained with an Adam optimizer and batch size 32×32 for 40000 iterations. The initial learning rate is 1×10^{-4} and is decayed by 0.1 at 20000 and 30000 iterations. Random augmentation is applied during training, so each input image is randomly rotated, scaled or flipped. The heatmap is furthered smoothed by a 7×7 Gaussian kernel with $\sigma = 3$.

For the point network, we sub-sample the input point cloud to a fixed size of 256 points. We only use the 3D coordinates of points as point feature, which is concatenated with the 13-dimensional camera feature from the camera network to perform modality fusion. We set $\lambda = 0.1$ for the segmentation task (Equation (3) in the main paper). The network is trained for 100000 iterations, with an SGD optimizer and batch size 128. The initial learning rate is 1×10^{-3} and is decayed by cosine decay. During training, input point clouds are rotated in the X-Y plane by a random angle in $[0, 2\pi)$ as data augmentation.

4. Qualitative Results

Figures 1, 2 , and 3 show qualitative results from the Waymo Open Dataset. Figure 1 compares different model architectures corresponding to Table 3 in the main paper. Row 1 shows the input camera image and LiDAR point cloud. Starting from Row 2, Columns 1 and 3 show the 2D projections of 3D predictions overlaid on camera images, and Columns 2 and 4 show 3D predictions. Rows 2 to 5 correspond to rows in Table 3 in the main paper, which are LiDAR-only model without segmentation branch, LiDAR-only model with segmentation branch, multi-modal model without segmentation branch and multi-modal model with segmentation branch, respectively.

Due to the objects (e.g., backpack, scooter, bike) attached to the pedestrian and the pose of the legs, the input LiDAR point cloud looks different from a regular pedestrian, which poses challenges for LiDAR-only 3D HPE. From the results, we can see that it is difficult to predict accurate keypoints (especially lower body keypoints) from LiDAR point cloud only (Rows 2 and 3). By utilizing texture information from the camera image, multi-modal architectures show much better performance (Rows 4 and 5) on all keypoints. On the other hand, comparing Rows 2, 4 with Rows 3, 5 respectively, we see that adding segmentation branch refines the predictions for both LiDAR-only and multi-modal architectures. Similar to the observations from Table 3 in the main paper, by adding key features to the model, the prediction accuracy improves consistently.

Figure 2 gives additional qualitative results in cases of occlusion. In each of these cases, we can see how adding camera information to LiDAR provides a big boost, especially for identifying the individual limbs of the subject in question. This is understandable, since with occlusion, it is often difficult to isolate LiDAR point clouds of a person from their immediate surrounding. Figure 3 shows more qualitative results, highlighting differences between various architectures. Note that even when the human is heavily occluded (last row), our approach can get reasonable results from the learned priors.

References

[1] Keypoint evaluation metrics used by coco. https://cocodataset.org/#keypoints-eval. 1

[2] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *European Conference on Computer Vision (ECCV)*, 2018. 1

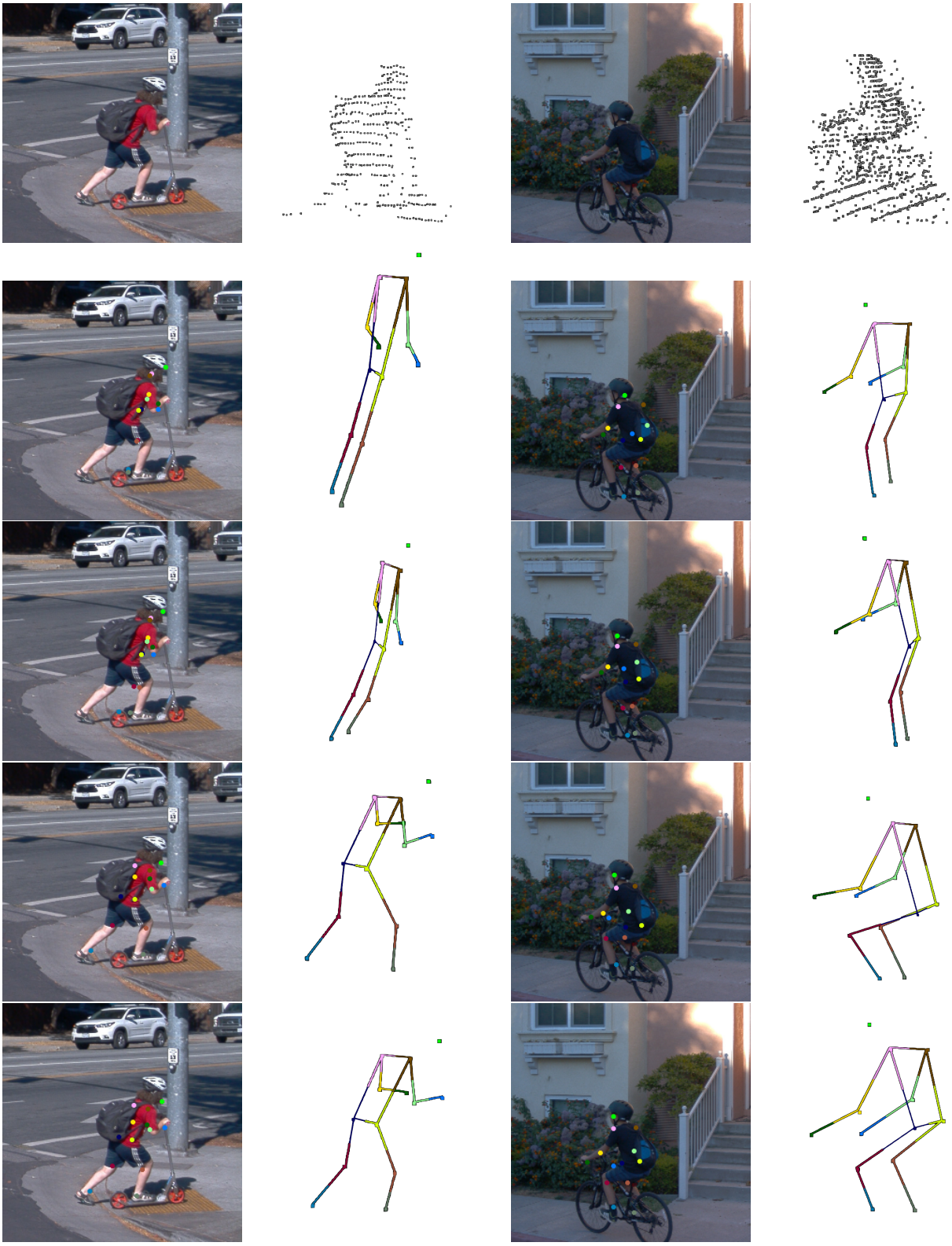

Figure 1. Qualitative results from Waymo Open Dataset, comparing different model architectures similar to Table 3 in the main paper. Row 1 is the input camera image and LiDAR point cloud. Starting from Row 2, Columns 1 and 3 show the 2D projections of 3D predictions overlaid on camera images; Columns 2 and 4 show 3D predictions. Row 2 to 5 correspond to LiDAR-only model without segmentation branch, LiDAR-only model with segmentation branch, multi-modal model without segmentation branch and multi-modal model with segmentation branch, respectively. Similar to the observations from Table 3 in the main paper, by adding key features to the model, the prediction accuracy improves consistently. Best viewed in color.

Figure 2. Additional qualitative results on the Waymo Open Dataset, showing the improvement that our approach brings over LiDAR-only model. The columns in each row show: a) LiDAR-only model without segmentation branch; b) LiDAR-only model with segmentation branch; c) multi-modal model without segmentation branch; and d) multi-modal model with segmentation branch. Note that in each case there is either self- or other forms of occlusion that deteriorates LiDAR only results. While segmentation and camera each can provide some additional clue, combining everything produces the best result. Best viewed in color.

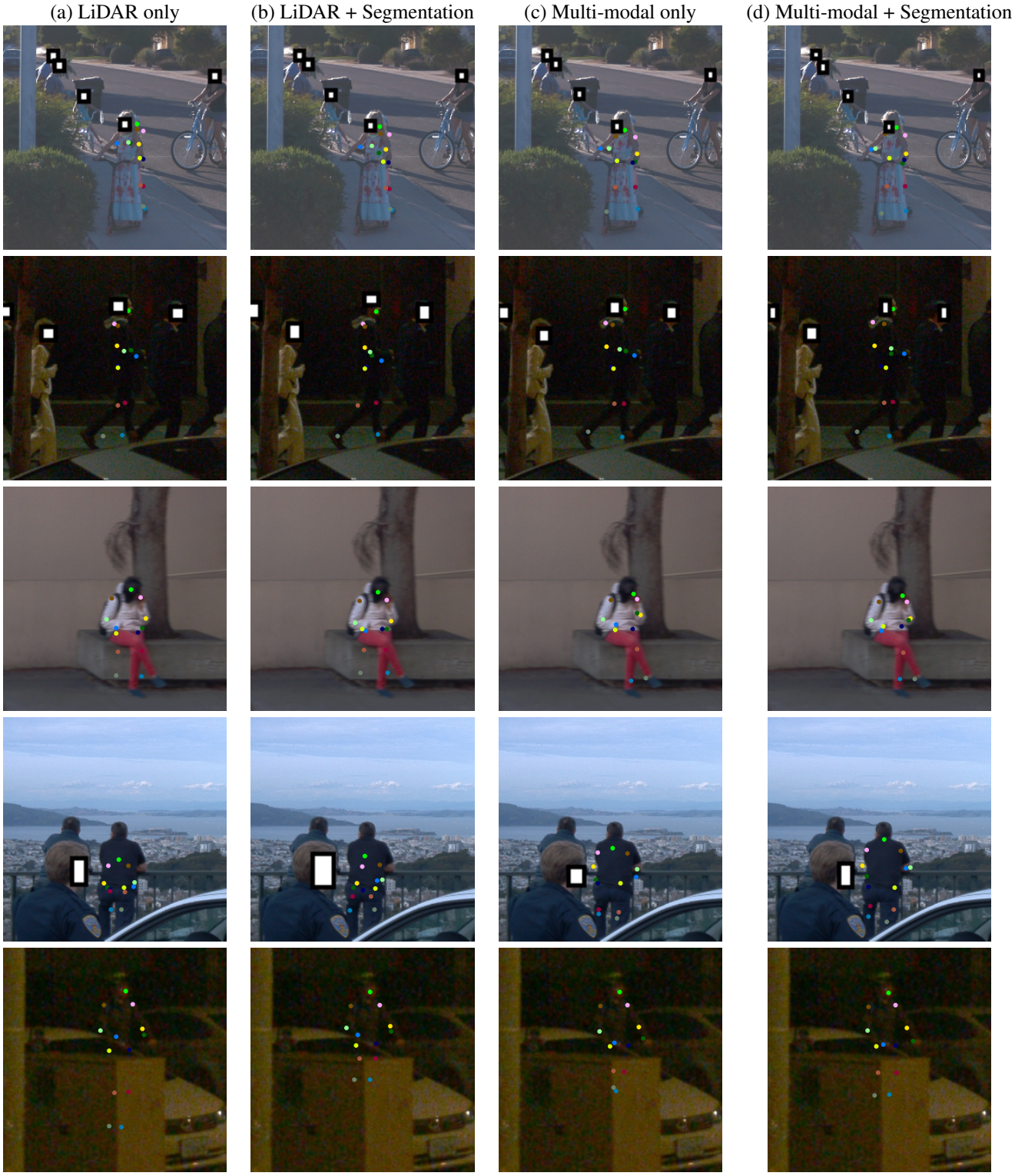

Figure 3. More qualitative results from the Waymo Open Dataset, highlighting differences between various architectures. Note that even when the human is heavily occluded (last row), our approach can get reasonable results from the learned priors (in this case, riding a bike). Best viewed in color.