Supplementary Materials: Weakly But Deeply Supervised Occlusion-Reasoned Parametric Road Layouts

Buyu Liu[1] Bingbing Zhuang[1] Manmohan Chandraker[1,2]
[1]NEC Laboratories America [2]UC San Diego

In this supplementary material, we include further details on the following:

- The network structure, that is, the Perspective Semantics (PS) , Top-view Semantics (TS) and Top-view Parametric Prediction (TPP) modules.
- The training protocol.
- The dataset and definitions of layout parameters, as well as the annotation process and costs.
- Pseudo-code for rendering process.
- Illustration for baselines and ablation studies.
- More results and analysis.
- Limitations and potential negative social impact

1. Network Structure

Our model consists of three modules. The first Perspective Semantics (PS) module inputs the RGB image and outputs the Occlusion-reasoned Semantics in Perspective view (OSP). The second Top-view Semantics (TS) module projects OSP into top-view and learns to predict Hallucinated Semantics in Top-view (HST). The last Top-view Parametric Prediction (TPP) module parses the HST and provides predictions on road layout related attributes in top-view. In this section, we will provide more details on network structures for each of them.

1.1. Perspective Semantics Module

Given an $I \in \mathbb{R}^{H\times W\times 3}$, the PS module would output a OSP $x^p \in \mathbb{R}^{H\times W\times (C+1)}$, which denotes the probability of each pixel belongs to specific category. Again, H and W is the height and width of perspective RGB image and C is the number of interested background categories. To achieve that, we then follow the structure of [9] as our semantic segmentation backbone. Specifically, we follow the HRNetV2-W18 structure that re-scales the low-resolution representations through bilinear upsampling without changing the number of channels to the high resolution, and concatenates the four representations, followed by a 1×1 convolution to mix the four representations. 18 indicates the width of the high-resolution convolution.

1.2. Top-view Semantics Module

As discussed in the main paper, our TS module consists of a transformation module and a hallucination module. As for the transformation module, it first maps the OSP to top-view with the assumption that our camera/ego car is located at the bottom center. We are interested in the region 60m in front of the given camera and 15m to either side. To this end, we represent the 60m $\times$ 30m semantic space in top-view with a 256×128×(C+1) image. For completeness, we detail below the geometric transformation more formally, though the reader is reminded that this step itself is standard and does not contain new contributions. Denote the camera pose w.r.t the world coordinate as $[\boldsymbol{R}, \boldsymbol{t}]$, such that a 3D point $\boldsymbol{X}$ is projected to a 2D point $\boldsymbol{x}$ via

$$\bar{\boldsymbol{x}} \simeq \boldsymbol{K}(\boldsymbol{R}\boldsymbol{X} + \boldsymbol{t}), \tag{1}$$

where $\simeq$ denotes equality up to a scale, and $\bar{\boldsymbol{x}}$ denotes the homogeneous form of $\boldsymbol{x}$. Suppose all the 3D points lie on a plane $(\boldsymbol{n}, d)$, i.e.

$$\boldsymbol{n}^\top \boldsymbol{X} = d, \tag{2}$$

we have

$$\bar{\boldsymbol{x}} \simeq \boldsymbol{K}(\boldsymbol{R} + \frac{\boldsymbol{n}^\top \boldsymbol{t}}{d})\boldsymbol{X}, \tag{3}$$

where $(\boldsymbol{R} + \frac{\boldsymbol{n}^\top \boldsymbol{t}}{d})$ is a 3D-2D homography mapping. In this work, we assume that the extrinsincs $[\boldsymbol{R}, \boldsymbol{t}]$ and the ground plane parameters $(\boldsymbol{n}, d)$ could be obtained by calibration in advance. Thus, we could choose the world coordinate to be identical to the camera coordinate, and hence Eq. 2 and $\bar{\boldsymbol{x}} \simeq \boldsymbol{k}\boldsymbol{X}$ fully determine the mapping between $\boldsymbol{x}$ and $\boldsymbol{X}$ which is required in our task.

As for hallucination module, we borrow the structure from [8] and utilize a 5-layer encoder and decoder U-Net. Our input and output of hallucination module is of the same

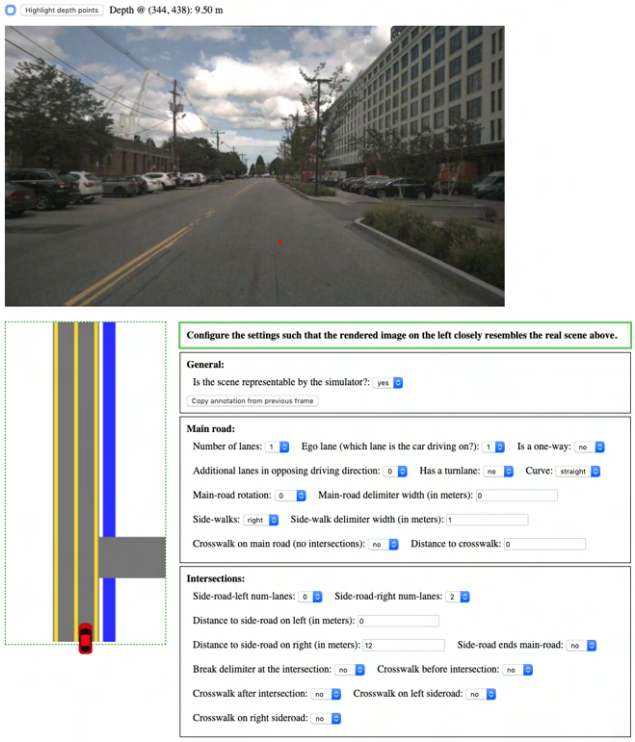

Figure 1. The annotation tool for scene attributes in [10]: The user sees the RGB image as well as the sparse depth points when hovering over the image (red dot, all points can be highlighted too). Note that the depth points can obtained either from LiDAR or stereo images. The authors unitize LiDAR images provided in original In KITTI and NuScenes dataset in practice. They ask annotators to fill out the form below the image with all attributes that describe the scene. As soon as the annotators change any value in the form, they exploit their render to generate an abstract semantic top-view (left to the form), which gives immediate feedback.

size, or 256×128 ×(C+1). The encoder of U-Net downsamples the feature maps and the decoder upsamples features, applies feature concatenation with the correspondingly cropped feature map from encoder and makes predictions with concatenated features. We refer the final output of hallucination module as HST.

1.3. Top-view Parametric Prediction Module

Our last module, TPP module, takes the HST as input and outputs three separate predictions η_{b}, η_{m} and η_{c} for each of the parameter groups Θ_{b}, Θ_{m} and Θ_{c} of the scene model Θ. To this end, we introduce a multi-layer perceptron (h) and convolutional neural networks (g). Specifically, g is a convolutional neural network (CNN) that converts top-view semantics HST into a 1-dimensional feature vector $f_x \in \mathbb{R}^D$. Receiving the f_x, h then outputs the prediction for each group. Note that we h consists of three branches, one for each prediction group. We refer the readers to Sec. 3 for more details on the definition of each attribute group.

2. Training Protocol

Instead of training the full model in an end-to-end manner from scratch in the very beginning, we propose a multi-stage training protocol. Given the full model f^{full}:

$$\Theta = f^{\text{full}}(I) = (f \circ f^{\text{ts}} \circ f^{\text{ps}})(I) \ , \tag{4}$$

where $\circ$ defines a function composition. f^{ps}, f^{ts} and f correspond to three modules. We first train f^{ps} and fix the parameters and then move on the training process of f^{ts}. After finishing the first two modules, we further learn TPP module f, or parameters in g and h. After finishing all three modules, we then relax all modules and learn the full model in an end-to-end manner. We adapt the ADAM and initiate learning rate to 1e-4 during training. We set the epoch number to 100, 40 and 60 for f^{ps}, f^{ts} and f, respectively. As for the last stage of training, or end-to-end stage, we train f^{full} 50 epochs.

We observe that this protocol firstly guarantees that all intermediate modules (PS and TS) can achieve meaningful and high quality performance (OSP and HST). In the meantime, it also enables efficient training.

3. Dataset and Layout Parameters

Dataset: We utilize the annotated KITTI and NuScenes data in [3, 10, 11], including around 17000 annotations in terms of scene layout annotation. To avoid overlapping, annotations are split into training and testing depending on the video sequences. Specifically, among all about 40 sequences, 8 of them are selected as testing set, or about 2k images in total. Please note that the parametric annotations are all from [10] and we summarize their annotation process here for paper completeness.

Fig. 1 is from [10] and shows their web-based annotation tool to collect the scene attributes ground-truth. The figure caption explains how the annotation tool works.

Specifically, they ask annotators to describe the scene as closely as possible with the available set of attributes and to use the depth estimates *only* for distance-related attributes, e.g. starting point of left or right sideroad. If a scene is not representable with the tool, the user should indicate that in the form too. Annotating binary or multiple class attributes, such as existence of sidewalk or number of lanes on the left, is fairly simple and can be done within seconds. Other continuous attributes, such as road rotation or curvature, are also easy to obtain. Specifically, annotators have some discrete options and they select the one that fits the current scene best. As soon as the annotators change any value in the form, the render will provide semantic top-view

(left in Fig. 1), which gives immediate feedback. Perhaps the most time-consuming part of the annotation process is from starting point of sideroads, where the annotators move mouse in the information from depth image (top figure in Fig. 1), stop at where they believe the starting point is, read the feedback from depth image (9.5m in this example) and then type in the number in the form (right form in Fig. 1). Since most of the data comes from a video sequence, they let annotators process the data in order and the tool copies all attributes from the previous frame automatically. Since many attributes, e.g. number of lanes and existence of sidewalks, stay constant over a long time, this feature reduces annotation cost significantly.

Compared to pixel-level semantic annotation that requires labor-intense human annotation, e.g. about half an hour to annotate only visible regions of an image in perspective view, the annotation process of [10] is far less painful. In their experiments, it only takes a few seconds when the scene is simple and less than a minute when facing complicated scenarios, e.g. approaching intersections while the distance to left and right side-road are not the same. On average, it takes about 20s to annotate top parametric for a single image.

As described in our paper, as long as we obtain the parametric annotation, we are able to render the top-view semantics as well as semantics in perspective automatically, which corresponds to the HST and OSP ground-truth we used for training. We will release our generated/rendered ground-truth for OSP and HST.

Layout Parameters: We follow the definition in [3]. Tab. 1 is a detailed table for the scene parameters as well as their prediction space.

Note that during training process, we normalize the Θ_c to -0.5 to 0.5 and then discretize the prediction of each attribute into 100 bins by convolving a dirac delta function centered at Θ_c with a Gaussian of fixed variance. With the help of multi-modal predictions, we can easily extend our model with graphical models.

4. Pseudo-code for Renderer

We provide a simplified version of pseudo-code of our renderer in Alg. 1. It is implemented with python code. We firstly take the scene parameters, either come from ground truth or predictions, and convert them into polygons for each of the classes separately, i.e., road or sidewalk. Then we draw these polygons with PIL. Please note that during the drawing process, everything is relative to the ego car. Specifically, we put the ego-car in the bottom center and draw everything relative from there. For instance, if we have two lanes to the right, we compute 2.5 times the lane width (half the lane width for the ego lane itself) and the road starts on the right side of the car. As our 256×128 space representing the 60m by 30m space in real world, we have the hyper-parameter $c = 4.27$ that converts the meter space to pixel space. Similar things apply for all other scene parameters.

Algorithm 1 Pseudo code for rendering semantic map in BEV

Input: Hyper-parameters such as default width of lane w_l, sidewalk w_s and crosswalk w_c; Scene parameters; Hyper-parameter c that converts meter space to pixel space
Output: Semantic map in BEV

if We have N lanes on our right **then**

Prepare a rectangle whose top left corner is [64,0] and bottom right corner is [64+(0.5+N)*c*w_l,256];

end if

if Right sideroad exists **then**

if We have M lanes and distance to right sideroad is X meter **then**

Prepare a rectangle whose top left corner is [64+(0.5+N)*c*w_l,(60-X-w_l*M)*c] and bottom right corner is [128,(60-X)*c];

end if

end if

if Crosswalk on the right sideroad exists **then**

Prepare a rectangle whose top left corner is [64+(0.5+N)*c*w_l,(60-X-w_l*M)*c] and bottom right corner is [64+((0.5+N)*w_l+w_c)*c,(60-X)*c];

end if

if Sidewalk exist on the right handside **then**

if Delimeter width between mainroad and right sideroad is Y meter **then**

Prepare a rectangle whose top left corner is [64+((0.5+N)*w_l+Y)*c,0] and bottom right corner is [64+((0.5+N)*w_l+w_s+Y)*c,256];

end if

end if

Draw all rectangle in PIL

5. Illustration for baselines and ablation studies

In this section, we demonstrate various baselines and our detailed design for ablation studies. Fig. 2 provides visual examples for supervision, inputs and outputs of each baseline. For instance, BEV [11] and BEV-J-O [3] take semantics in top-view as input and outputs parametric layout predictions. And such top-view semantics are obtained with

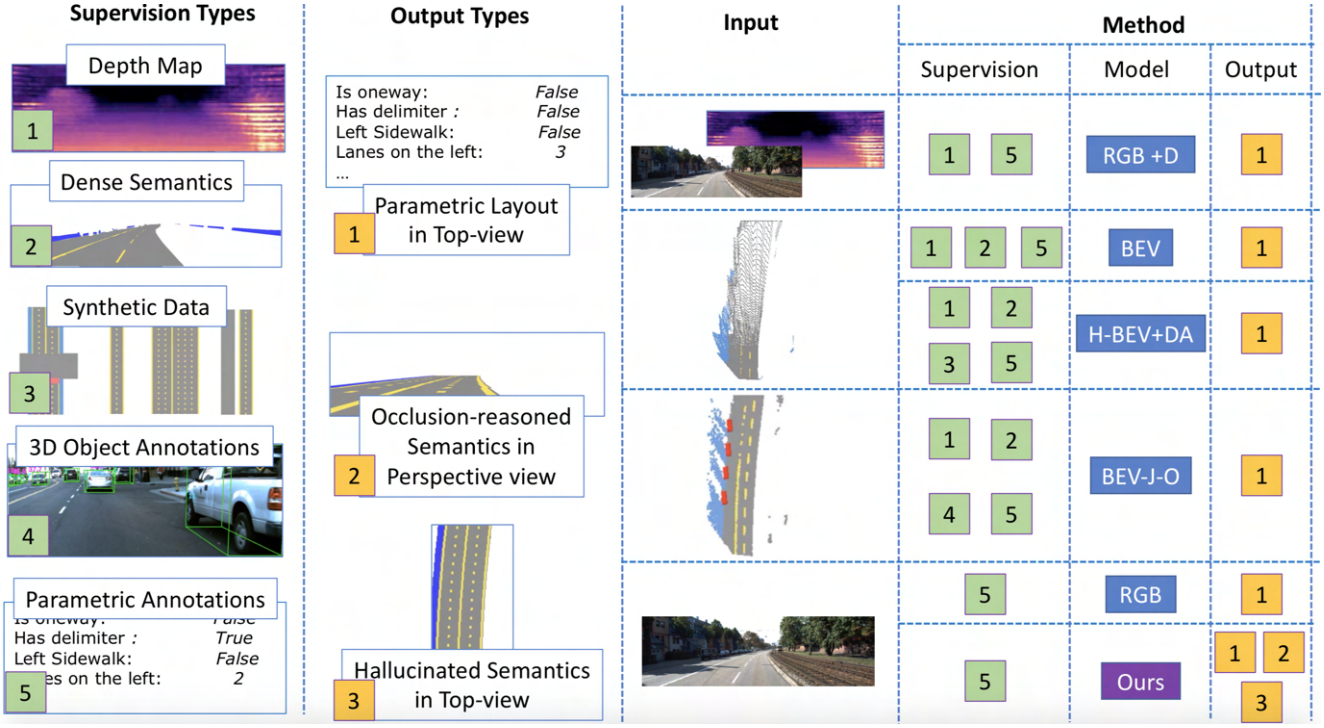

Figure 2. We provide more details for various types of supervisions, inputs and outputs of each baseline method.

Group	Attribute Definition	Prediction
Binary Θ_b	Main road is curved or not	Binary
	Main road is one-way or not	Binary
	Main road delimiter existence	Binary
	Delimiter between main road and sidewalk	Binary
	Existence of left sidewalk	Binary
	Existence of right sidewalk	Binary
	Existence of crosswalk before intersection	Binary
	Existence of crosswalk after intersection	Binary
	Existence of crosswalk on the left side-road	Binary
	Existence of crosswalk on the right side-road	Binary
	Existence of crosswalk on the main road (no intersection)	Binary
	Existence of left side-road	Binary
	Existence of right side-road	Binary
	Main road ends the side-road	Binary
Multiclass Θ_m	Number of lanes on the right of ego car	0-11
	Number of lanes on the left of ego car	0-11
Continuous Θ_c	Main road rotation	-1/8 π - 1/8 π
	Width of left side-road	0-30 m
	Width of right side-road	0-30 m
	Main road delimiter width	0-10 m
	Starting point of left side-road	0-60 m
	Starting point of right side-road	0-60 m
	Distance to crosswalk on the main road	0-60 m
	Main road sidewalk width	0-10 m
	Main road sidewalk delimiter width	0-10 m
	Main road curvature	1-Inf

Table 1. Attribute definition and their prediction space.

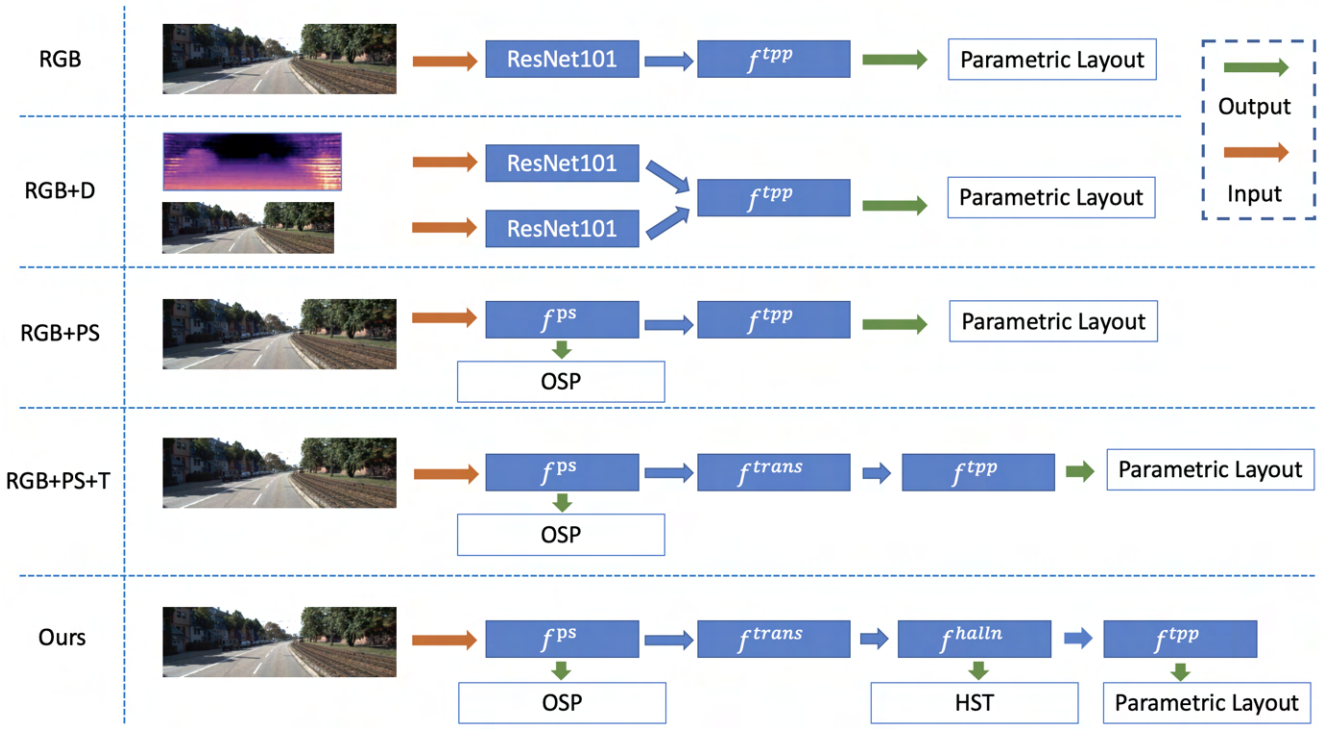

Figure 3. Model structures for baselines and ablation studies. Best view in color.

dense semantic and depth predictions thus corresponding human annotations/supervisions are required. We would like to note that although existing methods [4–7, 12] that output top-view semantics with perspective image seem to be potential alternative of our baselines for BEV and BEV-J-O, these methods do not include important semantics such as crosswalk or lane boundary due to their definition and annotation limitation in semantics, leading to incomparable situation. In addition, all these method requires dense semantic annotations in either perspective or top-view, which still validate our claim that our proposed method requires far cheaper compared to existing methods. Another baseline, RGB+D [11], takes both RGB and dense depth map as input and outputs also parametric predictions, avoiding semantic supervisions in perspective view. The only baseline that exploits the same supervision type with our method is RGB. Together with our quantitative results, we can see that our proposed method exploits the least human supervision and is able to achieve the SOTA performances w.r.t. existing methods that requires additional labor-intense human annotations.

We further provide details of model structures for our ablation studies in Fig. 3. As can be seen in this figure, RGB exploits ResNet101 [2] as backbone while RGB+D baseline is a two-head network where each branch uses ResNet101 [2] to extract features. Instead of using f^{ps} as backbone, we found that when loss in PS Module is not introduced, ResNet101 extracts more useful features thus leading to better performance. RGB+PS contains the PS module and directly predicts parametric predictions with perspective outputs. Similarly, RGB+PS+T further introduces transformation module as another intermediate representation. Ours is the full model that outputs all three types of outputs with single perspective image as input.

6. More Results

More Ablations on Intermediate Representations In our main paper, we did not compare with previous methods as they either care predictions for visible regions [6, 7] or have only one or two semantic classes for layout [4, 6, 12]. However, as a reference, we report here the semantics in perspective and top-view in [3, 11]. Specifically, their averaged IoU are 33.4% and 33.4% for OSP and 20.1% and 23.6% for HST, indicating the effectiveness of our method. Please note that these numbers are not directly comparable as [3,11] require additional depth and semantic supervisions for visible regions.

To further demonstrate that the performance improvements among RGB, $RGB + PS$ and our full model comes from deep supervision and our intermediate representations rather than deeper model, we include two more baselines on KITTI [1], B1 and B1-p, which share the same model architecture with our full model. B1 is trained with para-

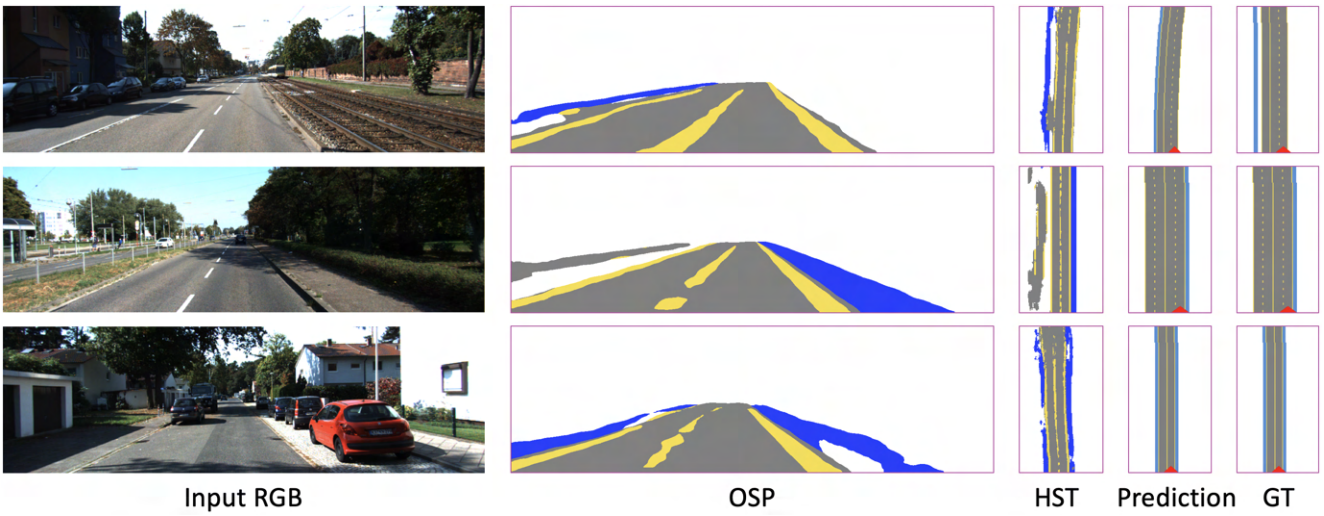

Figure 4. Full predictions of our propose model. From left to right: input RGB, OSP, HST, image rendered from parametric predictions and image rendered from ground-truth attributes.

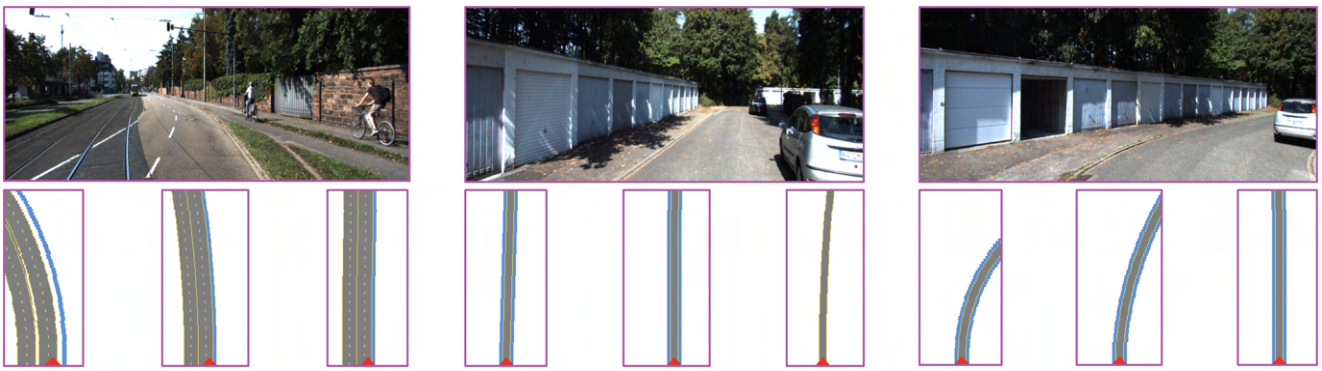

Figure 5. First row: Input perspective RGB image. Second row: Ground-truth annotation, our prediction and SOTA. Please note that both methods provide parametric layout predictions and we visualize rendered BEV semantics in the 2nd row.

metric supervision while B1-p is with additional perspective view supervision. Scores are 81.2%, 77.9% and .223 for binary, multi-class and continuous variables for B1 and 82.0%, 82.8% and .161 for B1-p, which showcases the effectiveness of deep supervision.

As emphasized in our paper, our model does not depend on the specific details of these sub-modules but is generally applicable if this three-stage architecture holds. Existing architecture, such as Monolayout [4], can be also exploited. We did not perform such ablations as they are orthogonal to our contributions.

More Qualitative Results: We provide visual examples in Fig. 4. As can be seen in this figure, our model is able to output satisfactory results on all three representations. We are able to handle complex road layout such as arbitrary number of lanes, curved road and with heavy occlusions. Again, please note that OSP and HST are obtained without per-pixel human annotations. We also provide qualitative comparison with SOTA [3]. As can be found in Fig. 5, we are able to provide more accurate layout predictions in BEV. Specifically, we perform better when encountering curved road or have sidewalks.

Evaluations on Intermediate Representations for Occlusion Cases: We further provided some visual examples for demonstrating the effectiveness of our method in terms of handling occlusions. As can be found in these examples, our model can provide occlusion-aware semantic predictions on both perspective and top view. Visual examples are shown in Fig. 6 showcasing our capability of handling various occlusions, curved road that beyond view or cars that occlude road or sidewalk, in OSP and HST. And we highlight the occlusions in red. More visual results can be further found in Fig. 7 to demonstrate our ability in handling complex road layout such as arbitrary number of lanes, curved

road and with heavy occlusions.

7. Limitations and Potential Negative Social Impacts

Limitations Our representation is limited to the design of road attributes. We are not able to handle arbitrary road layouts, e.g. round-about, due to missing such attributes in our design. It is our future work to extend our model to handle more complex layouts.

Potential Negative Impact Though we aim to provide road topology in BEV for path planning and decision making in intelligent driving systems, drivers' complete reliance on such systems might cause severe traffic accidents under complicated circumstances.

References

[1] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets Robotics: The KITTI Dataset. *IJRR*, 2013. 5

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[3] Buyu Liu, Bingbing Zhuang, Samuel Schulter, Pan Ji, and Manmohan Chandraker. Understanding road layout from videos as a whole. In *CVPR*, 2020. 2, 3, 5, 6

[4] Kaustubh Mani, Swapnil Daga, Shubhika Garg, Sai Shankar Narasimhan, Madhava Krishna, and Krishna Murthy Jatavallabhula. Monolayout: Amodal scene layout from a single image. In *WACV*, 2020. 5, 6

[5] B. Pan, J. Sun, H. Y. T. Leung, A. Andonian, and B. Zhou. Cross-view semantic segmentation for sensing surroundings. *IEEE Robotics and Automation Letters*, 2020. 5

[6] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. *arXiv preprint arXiv:2008.05711*, 2020. 5

[7] Thomas Roddick and Roberto Cipolla. Predicting semantic map representations from images using pyramid occupancy networks. In *CVPR*, 2020. 5

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015. 1

[9] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *PAMI*, 2020. 1

[10] Ziyan Wang, Buyu Liu, Samuel Schulter, and Manmohan Chandraker. A dataset for high-level 3d scene understanding of complex road scenes in the top-view. In *CVPR Workshop*, 2019. 2, 3

[11] Ziyan Wang, Buyu Liu, Samuel Schulter, and Manmohan Chandraker. A parametric top-view representation of complex road scenes. In *CVPR*, 2019. 2, 3, 5

[12] Weixiang Yang, Qi Li, Wenxi Liu, Yuanlong Yu, Yuexin Ma, Shengfeng He, and Jia Pan. Projecting your view attentively: Monocular road scene layout estimation via cross-view transformation. In *CVPR*, 2021. 5

Figure 6. First row: RGB. Second row: predicted OSP and HST. We can see that our model is able to handle various occlusions, e.g. occluded curved road or road occluded by cars, very in both perspective and top view.

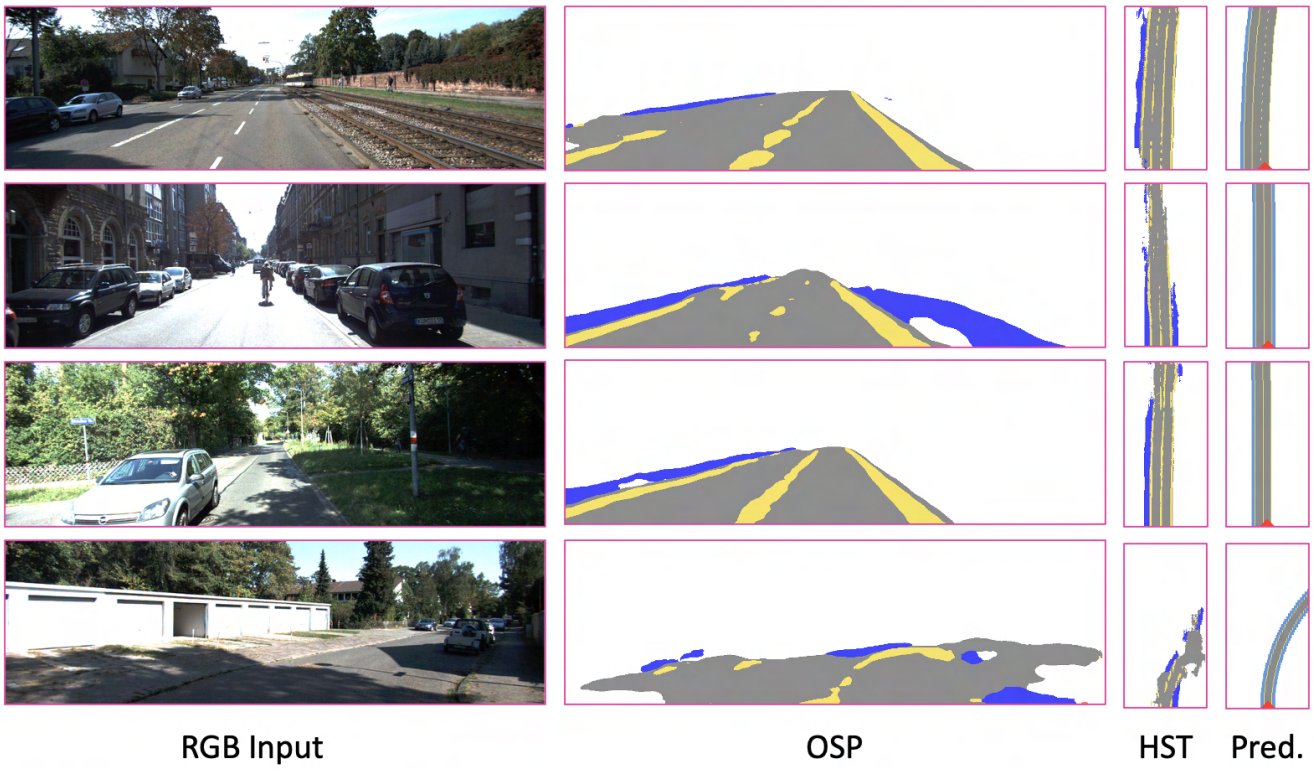

Figure 7. Full predictions of our propose model. From left to right: input RGB, OSP, HST and image rendered from parametric predictions.