A Parametric Top-View Representation of Complex Road Scenes Supplemental Material

Ziyan Wang[1*] Buyu Liu[2] Samuel Schulter[2] Manmohan Chandraker[2,3]
[1]Carnegie Mellon University [2]NEC Laboratories America [3]UC San Diego

The supplemental material contains the following details that we could not include in the main paper due to space restrictions:

- **Sec. 1:** Details of the proposed scene model, the corresponding dependencies among attributes, the annotation process and the statistics of our newly collected annotations.
- **Sec. 2:** A detailed description of our graphical model for predicting coherent and temporally smooth scene representations.
- **Sec. 3:** Exact definitions of the neural network architectures used in the paper.
- **Sec. 4:** An ablation study of various hyper-parameters of the proposed model and several baselines.
- **Sec. 5:** Discussion about the domain gap between synthetic data and real data and some visual results on pixel-level domain adaptation.

1. Details of our Scene Model

In this section we provide a comprehensive list of all attributes (parameters) contained in our scene model, explain the annotation process on real images and analyze the resulting data set statistics.

1.1. Scene attributes

We first provide a comprehensive list of scene attributes (parameters) that our model considers in Tab. 1. The table describes each attribute, assigns a unique ID that we later use as reference and tells whether or not the attribute was manually annotated for real images.

A directed acyclic graph relates the scene attributes and can be used for sampling synthetic road scenes. Since the graph is directed and acyclic, ancestral sampling can be employed for efficient generation of data [1]. We refer to Sec. 2 and Fig. 5 for the actual relations between the scene attributes. The (conditional) probability distributions for each node in the graph are hand-defined such that diverse scenes are encouraged, *i.e.*, more uniform distributions for many attributes are used. Hard constraints for infeasible outcomes are enforced by setting some probabilities to 0. Finally, we also used slightly different settings for the two data sets KITTI [3] and NuScenes [6], like the default/mean lane width, the number of maximum lanes, *etc*., to reflect the differences of road layouts in different geographical locations.

1.2. Annotation of scene attributes

As described in the main paper, we collected scene attribute annotation for real images on both KITTI [3] and NuScenes [6] data sets. The list of attributes we are collecting is highlighted with green IDs in Tab. 1. It can be easily imagined that some of these attributes are easier to annotate than others. Asking a human annotator whether the road the car is driving on is a curve or not is relatively easy compared to asking for a centimeter-accurate estimate of the distance to a side road. To annotate attributes related to distances or widths more information is required to be shown to the annotator than the plain RGB image. In our case, we have access to Lidar point clouds for both data sets, which aids the annotator by providing distances for some reference points in the image. Fig. 1 shows our annotation interface, where available pixels with depth information are highlighted in red. In addition to RGB and depth, we also provide a semantic segmentation and a top-view map from OpenStreetMap [7] if GPS was available and accurate.

We found that the depth information was crucial for annotators to estimate distances to intersections. Nonetheless, we also found that annotations for distances perpendicular to the camera axis, *e.g.*, the width of lanes or sidewalks, are difficult, which is why we dropped these attributes for annotation. Note, however, that these attributes are very well included in our model and simulated data!

1.3. Data distribution of scene attributes

Finally, we show the distribution of binary attributes Θ_b on both the KITTI [3] and the simulated data sets. Fig. 2 shows the distribution on training and testing set, where we can observe a few categories with a strong class imbalance. Attributes B4, B8 and B11 are very rare. Fig. 3 shows the same statistics for the NuScenes [6] data set and again the

[*] Work done during an internship at NEC Laboratories America.

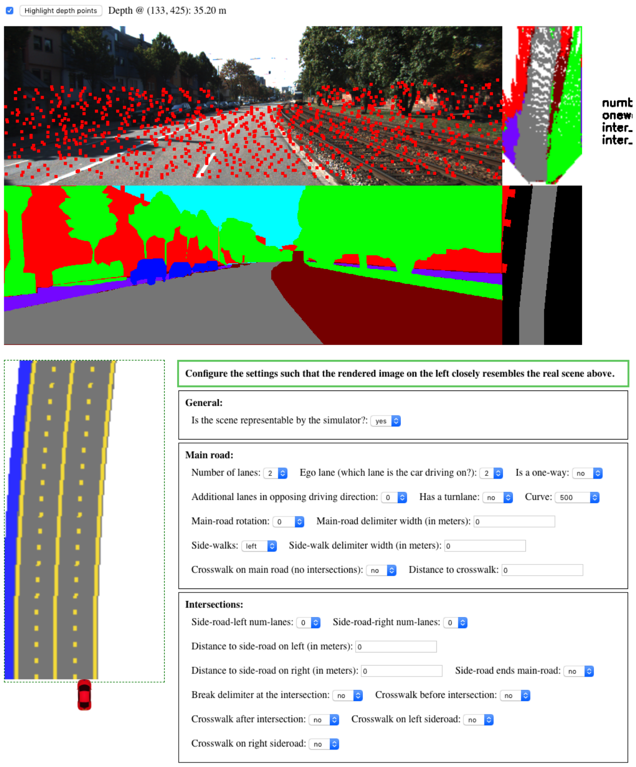
ID	Description
B1	Is the main road curved?
B2	Is the main road a one-way?
B3	Does the main road have a delimiter?
B4	Is there a delimiter between road and side walks?
B5	Does a sidewalk exist on the left of the main road?
B6	Does a sidewalk exist on the right of the main road?
B7	Does a crosswalk exist before the intersection?
B8	Does a crosswalk exist after the intersection?
B9	Does a crosswalk exist on the left side road of the intersection?
B10	Does a crosswalk exist on right side road of the intersection?
B11	Does a crosswalk exist on the main road w/o intersection?
B12	Does a left side road exist?
B13	Does a right side road exist?
B14	Does the main road end after the side roads?
M1	Number of lanes on the left of the ego-lane (maximum 6)
M2	Number of lanes on the right of the ego-lane (maximum 6)
C1	Rotation angle of the main road (*e.g.*, when car makes a turn)
C2	Width of the right side road
C3	Width of the left side road
C4	Width of a delimiter on the main road
C5	Distance to right side street
C6	Distance to left side street
C7	Distance to crosswalk on the main road without intersections
C8	Width of delimiter between main road and sidewalk
C9	Curve radius of the main road
C10-22	Lane widths ($6 \times 2 + 1$)

Table 1: The list of all our scene attributes Θ is divided into groups as in the main paper: binary Θ_b, multi-class Θ_m and continuous Θ_c. Each attribute is assigned an ID preceded by its group ID (B, M or C). The color of the ID indicates if manual annotation on real data exists (green). Attributes only available in simulation are marked red.

distributions of binary classes are extremely biased in both training and validation. Fig. 4 shows the distribution for simulated data where we can observe a much more balanced distribution. Simulated data is able to generate road layouts that rarely occur in real data. Note that the simulated distribution is hand-defined and can be adjusted to ones needs.

1.4. Evaluation metrics

The evaluation metric for binary and multi-class variables chosen in the main paper is plain accuracy (Accu.-Bi and Accu.-Mc). For most of our scene attributes the label distribution is balanced enough and computing accuracy is a valid evaluation metric. However, some attributes are highly imbalanced. We initially used precision and recall to compute an F1-score, but found it actually more difficult to judge the relative performance between different approaches due to some extremely imbalanced attributes. This makes the F1-score highly sensitive to correctly or incorrectly predicting just a single test example (out of more than 2000 for KITTI [3]). To remedy this issue in the future and to use the F1-score, we plan to either exclude these few attributes or, preferably, to increase the quantity of our test set.



☑ Depth @ (133, 425): 35.20 m

Configure the settings such that the rendered image on the left closely resembles the real scene above.

General:

Is the scene representable by the simulator?: yes

Main road:

Number of lanes: 2 Ego lane (which lane is the car driving on?): 2 Is a one-way: no

Additional lanes in opposing driving direction: 0 Has a turnlane: no Curve: 500

Main-road rotation: 0 Main-road delimiter width (in meters): 0

Side-walks: left Side-walk delimiter width (in meters): 0

Crosswalk on main road (no intersections): no Distance to crosswalk: 0

Intersections:

Side-road-left num-lanes: 0 Side-road-right num-lanes: 0

Distance to side-road on left (in meters): 0

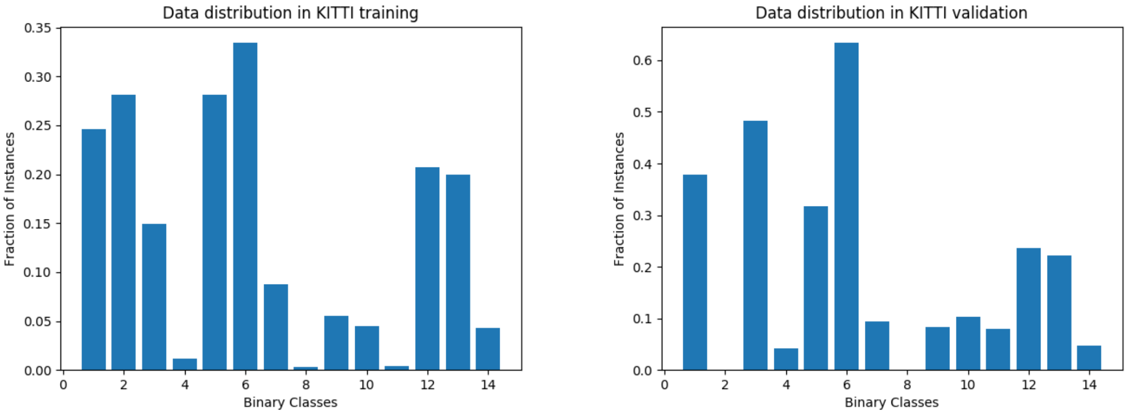
Distance to side-road on right (in meters): 0 Side-road ends main-road: no

Break delimiter at the intersection: no Crosswalk before intersection: no

Crosswalk after intersection: no Crosswalk on left sideroad: no

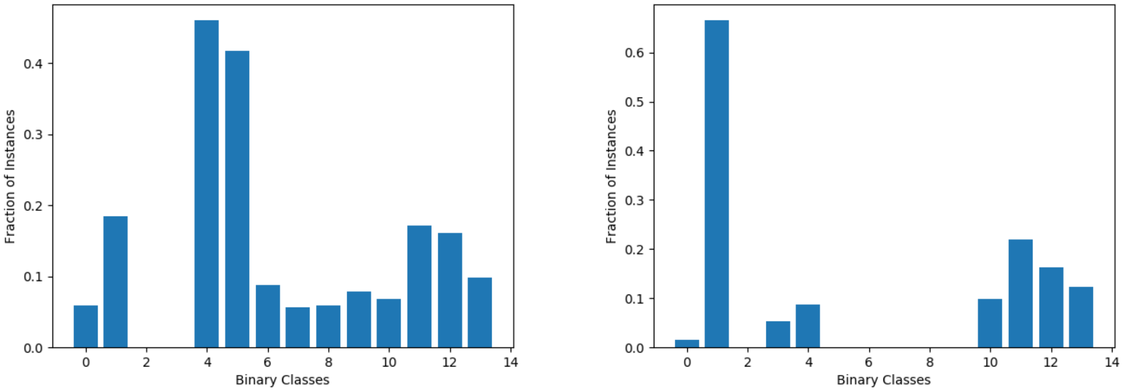
Crosswalk on right sideroad: no

Figure 1: The interface for annotating scene attributes. The user sees the perspective RGB image on the top as well as distances for sparse key points overlaid (the user can choose to overlay or not and distances are displayed when hovering over the image). The user also sees semantic segmentation output, the semantic points mapped into the top-view via a densely predicted depthmap (monocular depth estimation trained on the sparse ground truth points) and the OpenStreetMap map if GPS was available. The scene attributes are put into a web form at the bottom part of the interface. The user adjusts the web form which will re-render the scene after every change and display the current scene rendering at the bottom right.



(a) (b)

Figure 2: Data distribution in KITTI [3] training (a) and test (b) set for binary classes.



(a) (b)

Figure 3: Data distribution in NuScenes [6] training (a) and test (b) set for binary classes.

2. Graphical model details

In this section, we provide more details for our single-image and temporal CRFs, including detailed mathematical forms of potential functions that we omit in the main paper due to space limitations as well as the constraints/dependencies we enforce between scene attributes. An illustration of our graphical model can be viewed in Fig. 5.

2.1. Potential functions in single image CRF

Here we provide detailed information about how we define $\mathcal{S}$, $\mathcal{Q}$ and f_c in our single image CRF. Details can be viewed in Tab. 2.

2.2. Potential functions in temporal CRF

Denoting a video sequence as $\mathcal{V} = \{\mathbf{x}^t\}$, where $t \in \mathcal{T} = \{1, \dots, T\}$ is the frame index, we enforce temporal smoothness by introducing pairwise terms among frames as

$$\begin{aligned} E_v(\Theta^{\mathcal{T}}|\mathcal{V}) = \sum_{t\in\mathcal{T}} E(\Theta^t|\mathbf{x}^t) + \\ \sum_{t\in\mathcal{T}^{-1},i,p} E_{fp}(\Theta_{\mathrm{b}}^t[i], \Theta_{\mathrm{b}}^{t+1}[i], \Theta_{\mathrm{m}}^t[p], \Theta_{\mathrm{m}}^{t+1}[p]) + \\ \sum_{t\in\mathcal{T}^{-1},m} E_{vp}(\Theta_{\mathrm{c}}^t[m], \Theta_{\mathrm{c}}^{t+1}[m]) \ , \end{aligned} \quad (1)$$

where $\Theta^{\mathcal{T}}$ and Θ^t are the overall attribute predictions in the entire video sequence and that of the t-th frame, respectively, and $\mathcal{T}^{-1} = \{1, \dots, T-1\}$.

The newly introduced potential E_{fp} enforces temporal consistency between consecutive frames and is defined as

$$w_{cls} \times \left([\Theta_{\mathrm{b}}^t[i] \neq \Theta_{\mathrm{b}}^{t+1}[i]] + [\Theta_{\mathrm{m}}^t[p] \neq \Theta_{\mathrm{m}}^{t+1}[p]]\right) , \quad (2)$$

where w_{cls} is a hyper-parameter controlling the penalty of

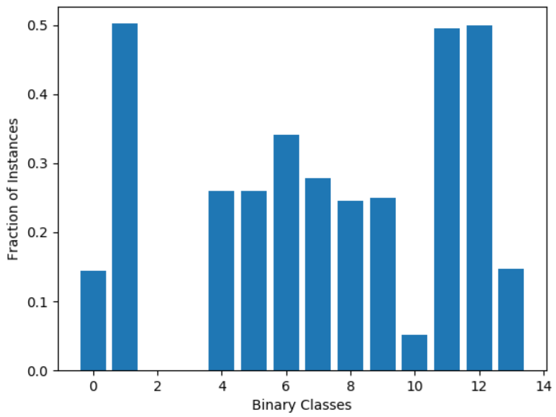


Figure 4: Data distribution in simulated data set for binary classes.

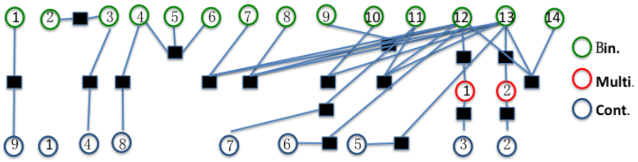


Figure 5: The illustration of our single image CRF. The green, red and blue circles represent the binary, multi-class and continuous group of variables, respectively. The number inside the circle denotes the i-th, p-th and m-th variable of the corresponding variable group.

assigning different labels to the same scene attribute in consecutive frames. In practice, we set $w_{cls} = 1000$ so that we will receive high penalties when our pre-defined rules are violated. The second new potential, E_{vp}, allows smooth changes for continuous variables and is defined as

$$E_{vp}(\Theta_{\mathrm{c}}^{t}[m], \Theta_{\mathrm{c}}^{t+1}[m]) = \|\Theta_{\mathrm{c}}^{t}[m] - \Theta_{\mathrm{c}}^{t+1}[m]\|^2 \ . \qquad (3)$$

Finally, note that due to the fact that ground truth is not available for all frames, we do not introduce per-potential weights. However, our CRF is amenable to piece-wise [8] or joint learning [2, 9] if ground-truth is provided.

$\mathcal{S}$	
i	p
12	1
13	2

$\mathcal{Q}$	
i	m
1	9
3	4
4	8
1	3
2	2
11	7

f_c		
$\Theta_{\mathrm{c}}[2] = 0$	$\Theta_{\mathrm{m}}[2] \neq 0$	
$\Theta_{\mathrm{c}}[3] = 0$	$\Theta_{\mathrm{m}}[1] \neq 0$	
$\Theta_{\mathrm{b}}[2] = 1$	$\Theta_{\mathrm{b}}[3] = 1$	
$\Theta_{\mathrm{b}}[9] = 1$	$\Theta_{\mathrm{b}}[12] = 0$	
$\Theta_{\mathrm{b}}[10] = 1$	$\Theta_{\mathrm{b}}[13] = 0$	
$\Theta_{\mathrm{b}}[12] = 0$	$\Theta_{\mathrm{c}}[6] = 1$	
$\Theta_{\mathrm{b}}[13] = 0$	$\Theta_{\mathrm{c}}[5] = 1$	
$\Theta_{\mathrm{b}}[7] = 1$	$\Theta_{\mathrm{b}}[12] = 0$	$\Theta_{\mathrm{b}}[13] = 0$
$\Theta_{\mathrm{b}}[8] = 1$	$\Theta_{\mathrm{b}}[12] = 0$	$\Theta_{\mathrm{b}}[13] = 0$
$\Theta_{\mathrm{b}}[14] = 1$	$\Theta_{\mathrm{b}}[12] = 0$	$\Theta_{\mathrm{b}}[13] = 0$
$\Theta_{\mathrm{b}}[11] = 1$	$\Theta_{\mathrm{b}}[12] = 0$	$\Theta_{\mathrm{b}}[13] = 1$
$\Theta_{\mathrm{b}}[11] = 1$	$\Theta_{\mathrm{b}}[12] = 1$	$\Theta_{\mathrm{b}}[13] = 0$
$\Theta_{\mathrm{b}}[11] = 1$	$\Theta_{\mathrm{b}}[12] = 1$	$\Theta_{\mathrm{b}}[13] = 1$
$\Theta_{\mathrm{b}}[4] = 1$	$\Theta_{\mathrm{b}}[5] = 0$	$\Theta_{\mathrm{b}}[6] = 0$

Table 2: Definition of $\mathcal{S}, \mathcal{Q}$ and f_c in our graphical model. Note that $\mathcal{S} = \{(i,p)\}$ and $\mathcal{Q} = \{(i,m)\}$. $f_c(\Theta_{\mathrm{b}}[i], \Theta_{\mathrm{c}}[p], \Theta_{\mathrm{c}}[m])$ defines all the conflicts in multi-task attribute predictions.

3. Network architectures

We further provide figures that detail the neural network architectures used for the functions $g(\cdot;\gamma_g)$, $h(\cdot;\gamma_h)$ and $d(\cdot;\gamma_d)$ of our proposed approach. As a reminder, g takes a semantic top-view $\mathbf{x} \in \mathbb{R}^{H\times W\times C}$ as input and computes a one-dimensional feature vector $f_{\mathbf{x}} \in \mathbb{R}^D$. The function h takes $f_{\mathbf{x}}$ as input and predicts the final scene model parameters Θ_{b}, Θ_{m} and Θ_{c} via a multi-task network. The discriminator function d is used for bridging the domain gap and makes a binary decision (“real” or “fake”) for inputs $f_{\mathbf{x}}$. Fig. 6 shows the architecture of these functions. Here we represent each convolutional layer as Conv$\{k\}$: $dim\_in - dim\_out - fs - s$, where dim_in is the number of input feature channels, dim_out is the number of output feature channels, fs is the filter size and s is the stride length. We represent each fully connected layer as FC$\{k\}$: $dim\_in - dim\_out$. After each convolutional layer, batch normalization [4] and a leaky ReLU layer [5] (with $\alpha = 0.2$) are added. Except for the last fully connected layer, every fully connected layer in d and h are followed by batch normalization [4] and ReLU.

Due to space limitations in the main paper, Figure 5 (high-level design of baseline methods and our proposed approach) turned out relatively small. We thus show the same figure again here in Fig. 7 where we split it into two separate rows for a better display.

4. Ablation Study

For our proposed method, **H-BEV+DA**, we report the results of the ablation study for various hyper-parameters. Tables 3 to 7 show the outcomes for the following parameters, respectively: learning rate, dimensionality of neural networks g and h, number of epochs for training, weightings for loss functions of real and simulated data, as well as the adversarial loss. All experiments are done on the KITTI

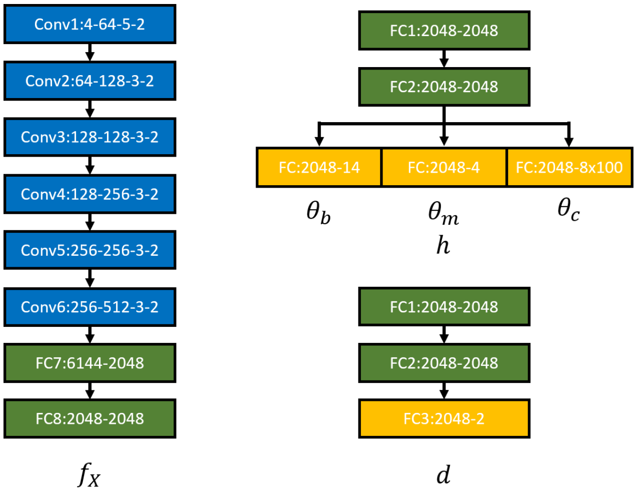


Figure 6: Network Architectures of g (left), h (right top) and d (right bottom).

Learning rate	Accu-Bin↑	Accu-Mc↑	MSE↓	IoU↑
1e-2	0.792	0.734	0.147	0.349
1e-3	0.819	0.760	0.156	0.363
2e-4	0.819	0.724	0.107	0.324
1e-4	0.812	0.717	0.112	0.316
5e-5	0.794	0.707	0.082	0.301
1e-5	0.788	0.676	0.123	0.238

Table 3: Varying the learning rate for **H-BEV+DA** on KITTI.

Dim. g & h	Accu-Bin↑	Accu-Mc↑	MSE↓	IoU↑
32 & 64	0.824	0.732	0.154	0.410
64 & 128	0.824	0.756	0.128	0.399
256 & 1024	0.823	0.773	0.116	0.363
1024 & 2048	0.829	0.759	0.151	0.382
4096 & 4096	0.826	0.735	0.185	0.365

Table 4: Varying the feature dimensionality of neural networks g and h for **H-BEV+DA** on KITTI.

# Epochs	Accu-Bin↑	Accu-Mc↑	MSE↓	IoU↑
1	0.800	0.706	0.096	0.193
5	0.807	0.662	0.097	0.260
10	0.812	0.721	0.135	0.325
50	0.823	0.747	0.142	0.345
100	0.812	0.740	0.148	0.351
200	0.816	0.793	0.136	0.354

Table 5: Varying the number of epochs of training for **H-BEV+DA** on KITTI.

data set [3]. We did a similar hyper-parameter search for all baselines reported in the main paper and chose best models accordingly.

Weights λ^{r} & λ^{s}	Accu-Bin↑	Accu-Mc↑	MSE↓	IoU↑
1.0 & 1.0	0.825	0.727	0.099	0.384
1.0 & 0.1	0.812	0.739	0.112	0.370
0.1 & 1.0	0.822	0.755	0.131	0.368
1.0 & 2.0	0.821	0.753	0.145	0.363
2.0 & 1.0	0.828	0.775	0.161	0.369
1.0 & 5.0	0.821	0.748	0.137	0.348
5.0 & 1.0	0.820	0.745	0.146	0.329

Table 6: Varying the weight of the loss functions for real and simulated data, λ^{r} and λ^{s}, for **H-BEV+DA** on KITTI.

Weight λ^{adv}	Accu-Bin↑	Accu-Mc↑	MSE↓	IoU↑
0.1	0.828	0.777	0.119	0.375
0.5	0.815	0.750	0.135	0.379
1.0	0.821	0.742	0.165	0.380
2.0	0.819	0.793	0.163	0.367
5.0	0.825	0.756	0.150	0.357
10.0	0.827	0.734	0.109	0.372

Table 7: Varying the weight of the adversarial loss λ^{adv} for **H-BEV+DA** on KITTI.

5. Domain gap between synthetic and real data

Due to the noise as well as the limitations of the perception power of sensors, the presence of domain gaps between real and synthetic data is inevitable. However, we claim that the noise pattern that appears in the real top-view map, which is mainly caused by limited field of view (FOV) of the perspective camera along with the sparsity of data points that correspond to the marginal area of a camera's perceptive field, is structured and learnable.

As our initial attempts to bridging this domain gap, we conducted experiments with pixel-level domain adaptation, which is easy to visualize and helps us better understand the process of the domain adaptation. We use a similar method for pixel-level domain adaptation as proposed in [10]. Different from [10], which directly predicts each pixel's RGB value in the transferred domain, we instead manipulate the input image by predicting a noise mask as well as a pixel flow map to mimic the noise in the input domain.

While we ultimately chose feature-level domain adaptation due to its effectiveness and simplicity in implementation, as mentioned in the main paper, we still want to share our initial results with pixel-level domain adaptation here because it provides further insights. To better illustrate that domain adaptation has noticeable positive effects on training the whole system, we thus show both qualitative results on domain adaptation on the pixel-level in Fig. 8 and quantitative results on performance improvements in predicting different attributes in Tab. 8. As can be seen in Fig. 8, the domain adaptation module learns to blend a similar pattern of the noise that occurs in the real data onto the ideal synthetic data, thereby creating real-like data. In Tab. 8, we notice a

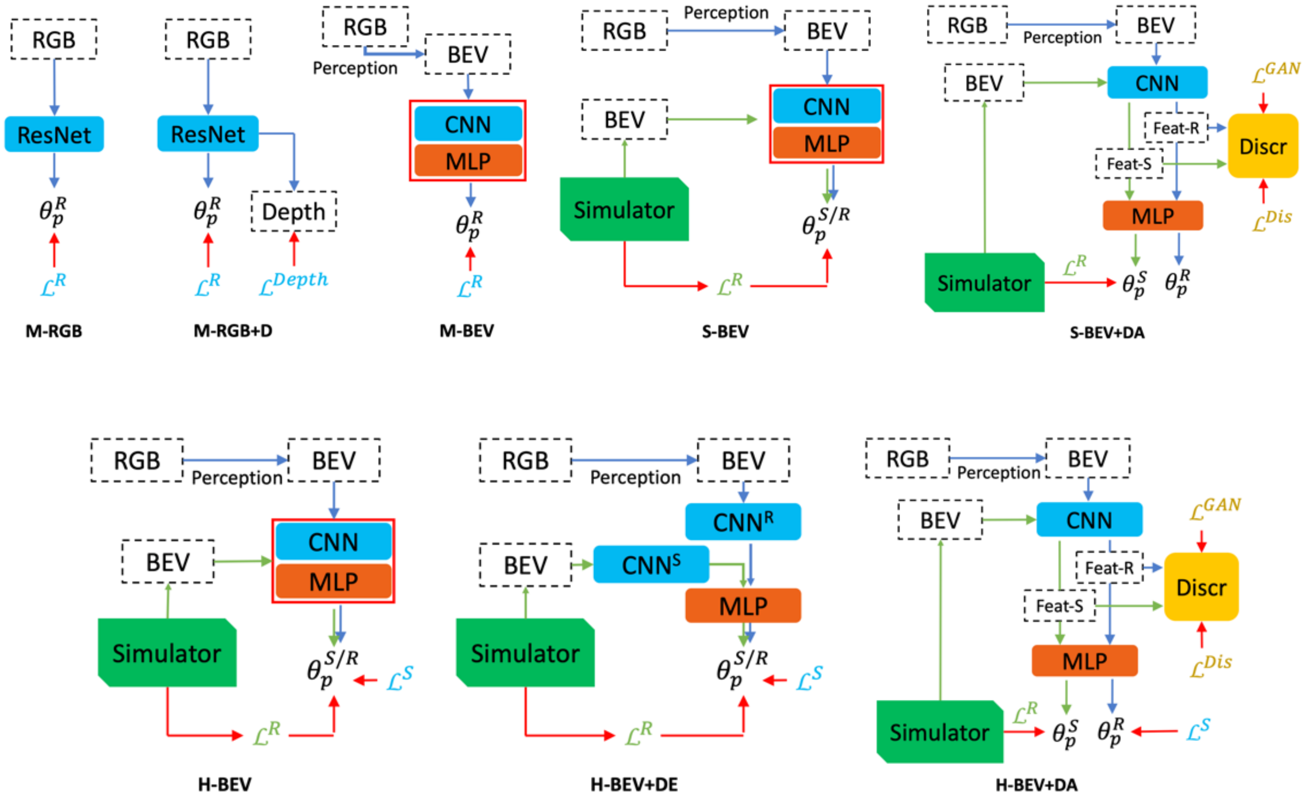


Figure 7: Overview of all models we are comparing in our quantitative evaluation in Table 1 of the main paper. All "M-" and "S-" models (manual annotation and simulation-only) are in the first row, while the proposed hybrid ("H-") models are in the second row. Note that "CNN", "MLP" and "Discr" correspond to the functions g, h and d, respectively.

clear increase in classification accuracy for attributes B2, B3, B7, B8, B12 and B14 after adding domain adaptation (DA). We also see observable drops in L1-distance on continuous attributes like C2, C3 and C7.

References

[1] Christopher M. Bishop. *Pattern Recogntion and Machine Learning*. Springer, 2007.

[2] Justin Domke. Learning graphical model parameters with approximate marginal inference. *PAMI*, 35(10):2454–2467, 2013.

[3] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets Robotics: The KITTI Dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[4] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *ICML*, 2015.

[5] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier Nonlinearities Improve Neural Network Acoustic Models. In *ICML*, 2013.

[6] NuTonomy. The NuScenes data set. `https://www.nuscenes.org`, 2018.

[7] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . `https://www.openstreetmap.org`, 2017.

[8] Charles Sutton and Andrew McCallum. Piecewise training for undirected models. In *UAI*, 2005.

[9] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015.

[10] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.

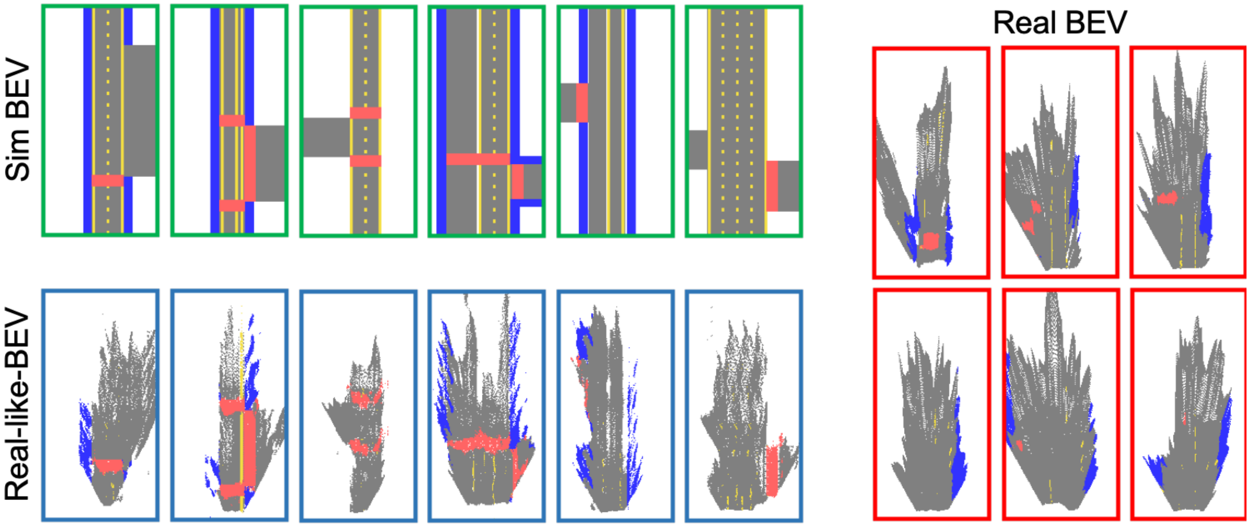


Figure 8: Bridging the domain gap with pixel-level DA.

Method	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14
S-BEV	.622	.013	.518	.959	.599	.768	.704	.733	.917	.853	.920	.556	.768	.786
S-BEV+DA	.622	.726	.822	.959	.589	.722	.905	1.00	.917	.898	.921	.728	.796	.851
M-BEV	.662	.824	.622	.959	.565	.493	.959	1.00	.917	.898	.920	.822	.867	.969
H-BEV+DA	.634	.963	.734	.959	.612	.651	.911	1.00	.917	.898	.921	.796	.865	.964

Method	M1	M2	C1	C2	C3	C4	C5	C6	C7	C8	C9
M-BEV	.667	.886	.126	.181	.192	.146	.176	.208	.637	.053	.094
H-BEV+DA	.688	.896	.146	.105	.086	.128	.219	.233	.570	.051	.093

Table 8: Per-class accuracy (B&M) and L1-distance (C) with and without pixel-level DA on data set KITTI.