# Appendix

We provide additional details on our dataset in Sec. A. In particular, we report the sensor layout (Sec. A.1), annotation details (Sec. A.2), extensive information on dataset generation (Sec. A.3) and dataset statistics (Sec. A.4).

Moreover, we conduct additional experiments in Sec. B. We provide baselines on multitask learning under continual domain shift (Sec. B.1, and conduct ablation studies on joint training with real-world data (Sec. B.2), the optimal dataset size for each task (Sec. B.3), the comparison between SHIFT and the VIPER dataset (Sec. B.4), and the error analysis for the rainy and foggy domains (Sec. B.5).

Implementation details for each experiment conducted in this work are reported in Sec. C for full reproducibility.

## A. Dataset Details

The detailed user guide and additional information can be found at https://www.vis.xyz/shift.

### A.1. Reference systems and sensor layout

The dataset has three levels of reference systems: *world*, *vehicle*, and *camera*. The world system represents the absolute position of objects. The vehicle system is used for storing all 3D annotations. The camera systems are the reference systems used for each individual camera.

Tab. 5 summarizes the supported sensors. We set up the vehicle system following KITTI's convention and the right-hand rule. Specifically, the origin is located at the center of the ego-vehicle (marked as the blue dot in Fig. 9). Its $x$, $y$, and $z$ axes point in the right, down, and front directions, respectively (Fig. 9, bottom left). All the sensors are located on a circle centered at the vehicle reference system's origin, except for the stereo cameras that are placed on the left to the front camera, with a horizontal displacement of $20\text{cm}$. All the cameras have a field-of-view (FoV) of $90°$. The 128-channel LiDAR sensor has a vertical FoV range of $[-10°, +10°]$ and a scan rate of 1.12M points per second.

| Sensor | Data type | Position |
|---|---|---|
| RGB camera | 24-bit RGB | $5 \times$ RGB cameras (front, left / right $45°$, left / right $90°$). |
| Stereo camera | 24-bit RGB | Additional RGB camera offsetting $20\text{cm}$ toward left from the center. |
| Depth camera | 24-bit Gray | Same as front view RGB camera. |
| Optical flow | 32-bit UV | Same as front view RGB camera. |
| GNSS / IMU | Vector | Center of vehicle. |

Table 5. The data type and position settings of the sensors.

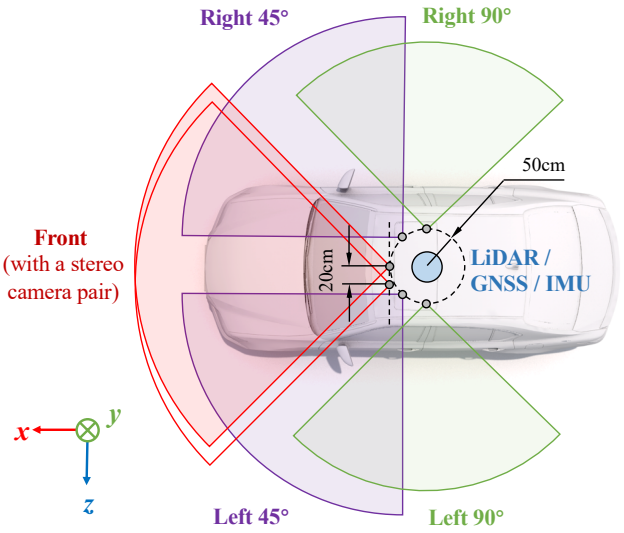


Figure 9. The vehicle system and the sensor layout. Except for stereo cameras, all the cameras are located on a circle centered at the vehicle reference system's origin (blue dot). LiDAR and motion sensors are located at the origin. Axes directions of the vehicle system are shown at the bottom left corner. *Best viewed in color.*

Annotations stored in the vehicle system can be easily converted into the camera systems. Here, the front cameras and LiDAR sensor have camera systems identical to the vehicle system, so no conversion is needed for them. For other cameras, a vehicle-to-camera matrix (*i.e.* intrinsic and extrinsic parameters) is provided to transform the annotations so that they fit each camera.

### A.2. Annotation details

We present detailed specifications for the annotation set provided in SHIFT.

**Object detection** is a fundamental localization task for scene understanding and a basis for numerous downstream driving tasks, including multiple object tracking (MOT) and object re-identification (ReID). We provide 2D/3D bounding box annotations and object identities for six categories of traffic participants, *i.e.* car, truck, bus, bicycle, motorcycle, and pedestrian, together with the visibility attributes 'occluded' and 'truncated'. Moreover, for each box, we provide fine-grained object classes (*e.g.* vehicle model type).

While previous datasets only provide 7 DoF (*i.e.* only yaw angle) 3D boxes [5, 17, 76], we provide 9 DoF annotations and use the Euler angle system (*i.e.*, yaw, roll, pitch) to represent the orientation for bounding boxes in 3D space.

**Image segmentation** is a fundamental pixel-level perception task. For each frame, we provide panoptic (*i.e.* instance and semantic) segmentation labels on the 23 classes of the Cityscapes [10] annotation scheme. Together with 2D bounding boxes, segmentation labels can be used in multi-object tracking and segmentation (MOTS) and multi-object panoptic tracking (MOTP) tasks.

| Category $\mathcal{H}_i$ | Candidate dom. $h_i^{(j)}$ | BDD100K eq. | Environmental parameters | Degrees of shift |
|---|---|---|---|---|
| Time of day | noon | daytime | Sun altitude angle = {90, 75, 60, 45, 30} | altitude angle $\in [-5, 90]$ |
| | morning / afternoon | daytime | Sun altitude angle = {15, 10, 5} | |
| | dawn / dusk | dawn / dusk | Sun altitude angle = {4, 3, 2} | |
| | sunrise / sunset | dawn / dusk | Sun altitude angle = {1, 0, -1} | |
| | night | night | Sun altitude angle = {-2, -3} | |
| | dark night | night | Sun altitude angle = {-4, -5} | |
| Weather | clear | clear | cloudiness = {0, 5} | cloudiness $\in [0, 100]$ |
| | slight cloudy | clear | cloudiness = {10, 15} | |
| | partly cloudy | partly cloudy | cloudiness = {25, 50, 70} | |
| | overcast | overcast | cloudiness = 100 | |
| | small rain | rainy | cloudiness = 70; precipitation = 20; deposit = 60; fog den. = 3 | precipitation $\in [0, 100]$ |
| | mid rain | rainy | cloudiness = 80; precipitation = 50; deposit = 80; fog den. = 3 | |
| | heavy rain | rainy | cloudiness = 100; precipitation = 100; deposit = 100; fog den. = 7 | |
| | small fog | foggy | cloudiness = 60; fog density = 30; fog distance = 15 | fog density $\in [0, 100]$ |
| | heavy fog | foggy | cloudiness = 80; fog density = 90; fog distance = 20 | |
| Vehicle density | sparse | - | num of vehicle = 50 | vehicle per map, vehicle per frame |
| | moderate | - | num of vehicle = 100 | |
| | crowded | - | num of vehicle = 250 | |
| Pedestrian density | sparse | - | num of pedestrians = 100 | pedestrian per map, pedestrian per frame |
| | moderate | - | num of pedestrians = 200 | |
| | crowded | - | num of pedestrians = 400 | |

Table 6. Definitions of the domain category and candidate domains, used for discrete domain shifts. Each category has a group of candidate domains. For each candidate domain, we show its equivalent domain label in BDD100K and the environmental parameters for simulation.

**Depth estimation** is an essential step to extend the 2D perception tasks into the 3D setting. We provide the depth labels aligned with the front-view RGB camera to enable image- and video-based monocular and stereo depth estimation. Depth resolution is $1\text{mm}$.

**Optical flow estimation** is an essential task for driving algorithms involving motion. However, existing large-scale datasets typically do not provide optical flow annotations due to the high labeling cost. Representing the relative motion between each pixel in a pair of images, optical flow can be instrumental in object tracking and ego-motion tasks. We provide the optical flow labels in the UV map format, also used in KITTI [17].

### A.3. Data generation pipeline

We introduce the pipeline that used to generate the discrete and continuous domain shifts.

**Disrete shift.** As discussed in Sec. 3.3, we set up an efficient sampling pipeline that can cover a diverse combination of conditions. To determine the environmental parameters of each sequence, we use a technique similar to random search. In Tab. 6, we define 4 categories of domain shifts, *e.g.*, time of day, weather, vehicle density, and pedestrian density. For the $i$-th category ($1 \leq i \leq 4$), we define a set of *candidate* domains, $\mathcal{H}_i = \{h_i^{(1)}, \cdots, h_i^{(n_i)}\}$, where each candidate $h_i^{(j)}$ corresponds to a certain group of environmental parameters, defined in the Tab. 6. Note that the parameter can be a fixed value or a set of values. For the set of values, we again uniformly sample one value out of the set.

Our sampling method for the discrete domain shifts can be summarized as following. A sequence is generated with a fixed parameter vector $\boldsymbol{\theta} = \theta_1 \cup \cdots \cup \theta_m$, where each $\theta_i$ is sampled uniformly across all candidates in the $i$-th category, *i.e.*,

$$h_i^{(j)} \sim \text{Uniform}(\mathcal{H}_i), \quad \forall i = \{1, 2, 3, 4\} \tag{1}$$

$$\theta_i \sim h_i^{(j)} \tag{2}$$

This pipeline guarantees the uniform marginal distribution of candidates conditioned on any category. Using this pipeline, we can easily add data without breaking the distribution of domains. Moreover, any subset of sequences of SHIFT has the same distribution, allowing a fair experiment on the impact of data amount.

**Continuous shifts.** For sequences with continuous domain shift, the change of parameters happens on one specific domain category $c$, while others are kept unchanged, *i.e.* the frame at time $t$ is generated with the parameter vector

$$\boldsymbol{\theta}(t) = \theta_1 \cup \cdots \cup f_c(t) \cup \cdots \cup \theta_4 \ , \tag{3}$$

where $f_c(t)$ is obtained by linear interpolation of the states listed in the 'Environmental parameters' column in Tab. 6.

| Continuous shift type | Environmental parameters | | |
|---|---|---|---|
| | **Beginning state** ($t = 0$) | **Intermediate state** ($t = 0.2$) | **End state** ($t = 1$) |
| Time of day | Sun altitude angle = 90 | - | Sun altitude angle = -5 |
| Cloudiness | cloudiness = 0 | - | cloudiness = 100 |
| Raininess | cloudiness = 0, precipitation = 0, deposit = 0, fog density = 0 | cloudiness = 80, precipitation = 50, deposit = 80, fog density = 3 | cloudiness = 100, precipitation = 100, deposit = 100, fog density = 7 |
| Fogginess | cloudiness = 0, fog density = 0, fog distance = 0 | cloudiness = 60, fog density = 30, fog distance = 15 | cloudiness = 80, fog density = 90, fog distance = 20 |

Table 7. Definitions of parameters used for continuous domain shifts. The parameters are updated for every frames during driving. The value of parameters are determined by linear interpolation between the state of beginning, intermediate (if applicable) and end.

Specifically, $f_c(t)$ is obtained by interpolating the points

$$(t, \theta) = \left[(0, \theta_{c,\text{begin}}), (0.2, \theta_{c,\text{intermediate}}), (1, \theta_{c,\text{end}})\right], \tag{4}$$

where $t \in [0, 1]$ represents the degree of continuous shift from the minimum to the maximum parameter allowed for a given domain category.

Furthermore, we provide an additional set of sequences presenting domain shifts simultaneously happening along multiple domain shift directions within the same sequence.

**Domain labeling details.** The degree of shift for each domain category is quantified by a numerical value called *severity*. For weather conditions, we use percentage values to indicate the degree of severity, where $0\%$ corresponds to clear weather conditions and $100\%$ represents the most extreme condition allowed by the CARLA simulator for a given weather direction, *e.g.* cloudiness, precipitation, fog density, or fog distance. We describe the time of day using the Sun's altitude angle to disentangle the lighting condition with the sunrise/sunset time. For the object densities, we use the number of objects per frame as the severity (Tab. 6, rightmost column).

### A.4. Dataset statistics

SHIFT is diverse in bounding box scale. Fig. 10 (left) plots the object density measured by boxes per frame and shows coverage from 0 to 30 boxes/frame for SHIFT. We compare the distribution with the BDD100K's MOT set. Due to the sparsity of the vehicle/pedestrians density domains, our dataset has on average a higher density of frames counting less bounding boxes than BDD100K, but the crowded frames ($\geq$ 20 boxes/frame) show similar trends. Moreover, Fig. 10 (right) shows the distribution of bounding box sizes, defined as $\sqrt{wh}$ where $w$ and $h$ are the width and height of a box. SHIFT covers diverse box sizes ranging from 10 to 650 pixels. We also observe that our dataset has 41.2% bounding boxes smaller than 15 pixels while BDD100K has 30.9%, showing that our dataset provides challenging conditions for small object detection and tracking.

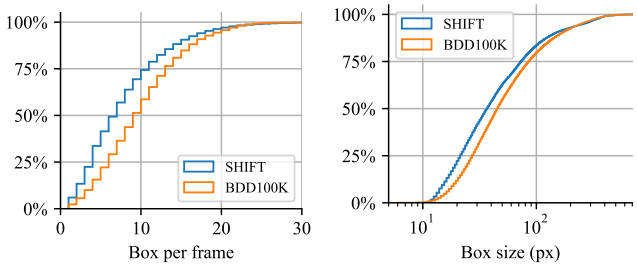


Figure 10. Cumulative distributions of the bounding box per frame (left) and bounding box size measured in $\sqrt{wh}$ (right). We only count the objects in the front camera view. SHIFT covers various object densities and a wide range of object scale.

## B. Additional Experiments

To further highlight the usefulness of SHIFT, we conduct experiments on multitask learning (Sec. B.1) and joint training with real-world data (Sec. B.2). We also investigate the optimal dataset size and sampling rate (Sec. B.3).

### B.1. Multitask learning

In this experiment, we study whether different perception tasks mutually benefit or interfere with each other when jointly learned with a shared feature extractor. The wide variety of tasks supported in SHIFT unlocks new opportunities to investigate different combinations of perception tasks. Special attention is also paid to the robustness of multitask models under incrementally shifted domains.

Specifically, we consider four different perception tasks: semantic segmentation, instance segmentation, monocular depth estimation, and optical flow estimation. Each task requires the model to learn a distinct encoding function: semantic segmentation requires intermediate activations to encode pixel-level information, instance segmentation requires instance-level information, depth estimation requires contextual information and object priors that allow to convert 2D images to 3D cues, and optical flow requires to encode a function of two images that embodies information on motion perception.

| Task | Train | Metric | Source | OOD | | | | | | OOD Avg. | $\Delta_{\text{Source}}$ | $\Delta_{\text{OOD}}$ | $\Delta_{\text{S}\to\text{O}}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | clear-daytime | cloudy | overcast | foggy | rain | dawn/dusk | night | | | | |
| Semantic segmentation (S) | S | mIoU (%) ↑ | 69.1 | 40.6 | 40.6 | 21.5 | 19.6 | 18.1 | 8.9 | 24.9 | - | - | -64.0% |
| | S + D | | **75.2** | 53.8 | 52.6 | 24.3 | 26.6 | 24.0 | 9.9 | 31.9 | **8.9%** | 28.1% | -57.6% |
| | S + F | | 69.4 | 51.8 | 54.7 | 26.4 | 22.4 | 22.7 | 9.8 | 31.3 | 0.4% | 25.8% | -54.9% |
| | S + D + F | | 71.8 | 50.0 | 51.9 | 23.5 | 24.0 | 22.1 | 9.5 | 30.2 | 3.8% | 21.2% | -58.0% |
| | S + I | | 74.8 | **63.9** | **68.1** | **41.0** | **36.8** | **37.3** | **23.6** | **45.1** | 8.2% | **81.3%** | **-39.7%** |
| | S + D + I | | 75.0 | 62.4 | 65.1 | 37.4 | 35.4 | 35.3 | 20.5 | 42.7 | 8.6% | 71.6% | -43.1% |
| | S + F + I | | 72.5 | 58.3 | 59.6 | 35.8 | 27.2 | 28.8 | 14.4 | 37.3 | 4.9% | 50.1% | -48.5% |
| | S + D + F + I | | 74.7 | 60.6 | 59.6 | 37.1 | 32.7 | 33.1 | 19.3 | 40.4 | 8.1% | 62.3% | -45.9% |
| Depth estimation (D) | D | SILog ↓ | 17.8 | 28.3 | 23.1 | 81.9 | 46.3 | 54.6 | 63.2 | 49.6 | - | - | -64.1% |
| | D + S | | 16.9 | 25.2 | 22.4 | 65.7 | 43.0 | 49.4 | 57.6 | 43.9 | 5.6% | 13.0% | -61.6% |
| | D + F | | 19.3 | 25.3 | 20.4 | 66.6 | 45.3 | 50.3 | 54.4 | 43.7 | -7.8% | 13.4% | -55.8% |
| | D + S + F | | 19.6 | 26.9 | 24.7 | 67.8 | 45.1 | 52.1 | 56.6 | 45.5 | -9.2% | 8.9% | -56.9% |
| | D + I | | 17.3 | 21.0 | 16.8 | 66.4 | 35.1 | 42.4 | 48.4 | 38.3 | 2.9% | 29.3% | -54.8% |
| | D + S + I | | **16.0** | **19.5** | **15.4** | 61.1 | **31.2** | **38.5** | **42.7** | **34.7** | **11.0%** | **42.8%** | -53.8% |
| | D + F + I | | 17.8 | 21.4 | 17.9 | **47.9** | 36.4 | 39.9 | 46.3 | 35.0 | 0.1% | 41.8% | **-49.1%** |
| | D + S + F + I | | 17.6 | 21.9 | 18.3 | 53.2 | 37.1 | 42.7 | 50.5 | 37.3 | 1.0% | 33.0% | -52.7% |
| Optical flow estimation (F) | F | EPE (px) ↓ | **6.0** | **6.7** | **6.4** | **9.0** | **9.7** | **9.1** | **11.0** | **8.6** | **-** | **-** | -30.8% |
| | F + S | | 7.8 | 8.3 | 8.5 | 10.4 | 12.0 | 10.9 | 12.6 | 10.4 | -23.1% | -17.4% | -25.7% |
| | F + D | | 6.0 | 6.9 | 6.4 | 9.4 | 10.4 | 9.6 | 11.8 | 9.1 | -0.2% | -5.0% | -34.2% |
| | F + D + S | | 6.1 | 8.5 | 8.3 | 10.6 | 12.1 | 11.0 | 13.0 | 10.6 | -2.1% | -18.5% | -42.4% |
| | F + I | | 9.8 | 9.6 | 9.6 | 10.4 | 11.3 | 10.6 | 12.1 | 10.6 | -38.9% | -18.5% | **-7.7%** |
| | F + S + I | | 7.7 | 8.2 | 7.9 | 9.7 | 10.6 | 9.8 | 11.8 | 9.7 | -22.7% | -10.8% | -20.2% |
| | F + D + I | | 8.0 | 8.3 | 8.2 | 9.9 | 10.6 | 10.0 | 11.9 | 9.8 | -25.5% | -12.1% | -18.4% |
| | F + D + S + I | | 8.1 | 8.4 | 8.4 | 10.1 | 11.0 | 10.2 | 12.1 | 10.0 | -26.4% | -14.0% | -19.2% |
| Instance segmentation (I) | I | mAP (%) ↑, vehicles | 63.9 | 57.4 | 65.7 | 21.9 | 31.2 | 22.7 | 6.6 | 34.2 | - | - | 46.4% |
| | I + S | | 64.9 | 59.1 | 66.2 | 26.4 | **34.4** | 27.1 | **14.3** | **37.9** | 1.5% | **10.7%** | **41.6%** |
| | I + S + D | | 65.0 | 57.9 | 64.9 | 25.9 | 32.6 | 26.1 | 10.9 | 36.4 | 1.6% | 6.3% | 44.0% |
| | I + S + F | | 62.3 | 57.1 | 64.2 | 21.6 | 31.6 | 23.6 | 7.7 | 34.3 | -2.5% | 0.1% | 45.0% |
| | I + D | | **65.9** | **59.3** | 66.9 | **26.8** | 32.2 | **26.3** | 11.4 | 37.2 | **3.1%** | 8.5% | 43.6% |
| | I + D + F | | 65.8 | 50.2 | **67.0** | 21.5 | 31.0 | 22.9 | 6.7 | 33.2 | 2.9% | -3.0% | 49.5% |
| | I + F | | 63.1 | 56.9 | 65.1 | 20.4 | 28.9 | 21.7 | 5.0 | 33.0 | -1.3% | -3.6% | 47.7% |
| | I + S + D + F | | 64.8 | 57.9 | 65.3 | 22.8 | 31.5 | 23.1 | 8.3 | 34.8 | 1.4% | 1.6% | 46.3% |

Table 8. Multitask learning performances. We evaluate 15 combinations of 4 perception tasks: semantic segmentation (S), monocular depth estimation (D), optical flow estimation (F), and instance segmentation (I). The combinations of S + I, S + D, and S + D + I significantly improve on both tasks' source and OOD performance in their respective tasks. ↑ (↓): the higher (lower) the better.

**Multitask model.** To compose a unified multitask model, we use the segmentation model DRN-D-54 [95] as feature extractor and combine it with the heads required for other tasks. The DRN-D-54 model has 8 sequential residual blocks with dilated convolutions and transposed convolutions at the end to generate segmentation results. Here, all the modules of DRN-D-54 are used for semantic segmentation. For instance segmentation, we rely on the Feature Pyramid Network (FPN) [100], Region Proposal Network (RPN), and ROIAlign modules identical to those introduce in Mask R-CNN [21]. FPN uses the 2nd to 5th blocks' outputs of the DRN-D network. For the optical flow and depth estimation, we adapt the decoders similar to FlowNet [98] and U-Net [66]. The decoder has 5 sequential blocks, where each block has one up-sampling layer, followed by a shortcut connection from the feature extractor's corresponding block, and a series of convolution layers. Together with the feature extractor, we obtain an encoder-decoder structure commonly used in dense prediction tasks.

**Experiment setup.** We traverse all 15 combinations for the 4 tasks mentioned above. Our multitask model is trained with 5,000 frames sampled from the clear-daytime domain in SHIFT and evaluated under different discrete domain shifts. To fit the multitask model into the GPU memory, we reduce the image size to $640 \times 400$ pixels. Please note that the performance will be slightly affected by the size-reduced images and thus, it is not directly comparable to our baseline experiments in 2 of the main paper. All combinations are trained for 100 epochs, when convergence is reached for all tasks.

**Experimental results** are summarized in Tab. 8. Every model is trained on the clear-daytime domain and tested on different types of shifted domains, indicated with OOD in the Table. We report the average performance on the out-of-distribution domains as OOD avg. The columns $\Delta_{\text{Source}}$

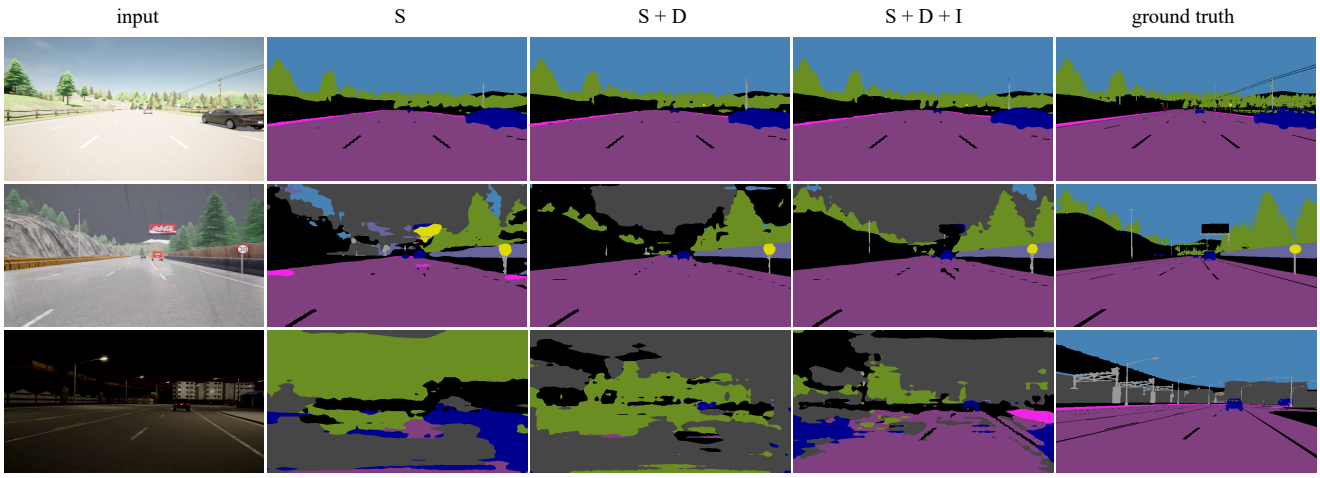


Figure 11. Qualitative results on semantic segmentation. The three rows show the results from the clear-daytime, rainy, and night domain, respectively, of models trained on the clear-daytime domain. The combinations of S + D and S + D + I improve the performance against domain shifts.

and $\Delta_{\text{OOD}}$ report for different multitask models the relative Source / OOD avg. performance change on a given task with respect to the performance of a single-task model trained on that specific task. The column $\Delta_{S \rightarrow O.}$ reports for different multitask models the relative change from Source to OOD avg. performance on a given task. Below are our observations.

***Multitask learning improves robustness.*** We observe that specific combinations of tasks largely improve the single-task model performance on the source domain. For instance, the combination of semantic segmentation (S) + depth (D) + instance segmentation (I) boosts the source domain performance by 8.6% / 11.0% / 1.6% on the respective tasks. Similar improvements are observed for other combinations, including S + I and S + D. We visualize the results of these combinations in Fig. 11. This is possibly due to the intertwined nature of such tasks. In particular, depth and semantics both need to learn contextual features from neighboring pixels, and both instance and semantics segmentation need to segment parts of the image.

Further, multitask learning often significantly increases the generalization of a model to domain shifts. For example, the combination of S + D + I improves the OOD performance in the respective tasks by 71.6% / 42.8% / 6.3%. The improvements are substantially greater than the improvements on the source domain, suggesting that the increase in model's robustness is not attributable to the increase in the overall model's performance as seen on the source domain. We argue that this is potentially due to the model learning more general features that are shareable across tasks and, consequently, also more general under domain shifts. For example, the addition of instance segmentation typically causes the greatest robustness improvements. This might be due to the complex nature of the instance segmentation task, which requires to encode features capable of both detecting and segmenting objects in an image.

***Instance segmentation can only be improved mildly.*** Instance segmentation is only improved at most by 10.7% on OOD performance by other tasks. As previously mentioned, we hypothesize that instance segmentation already learns more general features due to its nature. Thus, the addition of other tasks provides only mild improvements. On the other hand, however, when combined with other tasks, instance segmentation largely boosts their robustness, *e.g.* S + I and D + I.

***Optical flow is heavily affected by other tasks.*** Unlike the previous tasks that benefit from multitask learning, optical flow shows a different behavior. Although optical flow can improve other tasks' robustness (*e.g.* S + F and D + F), the optical flow itself is negatively affected by the addition of other tasks. When jointly trained with other tasks, its performance drops by a large margin, ranging from -0.2% to -38.9%. A possible explanation is that the optical flow task, which takes a pair of frames as input, learns a different encoding function than other non-temporal tasks requiring only one frame. To learn a feature extractor shared across the two different types of inputs, the model shows to sacrifice its effectiveness on the task requiring two images. This suggests that combining different tasks is not trivial; instead, it requires extensive evaluation and comparison. SHIFT provides a playground to develop novel multitask learning techniques and to investigate and solve the multiple challenges presented by such an interesting problem.

***Domain shift is only partially mitigated.*** While the model's robustness can be improved by multitask learning,

the domain shifts provided in SHIFT still pose a tremendous threat to the robustness under domain shift. For all the evaluated tasks, the minimum average OOD performance drop with respect to the corresponding source performance ($\Delta_{S \rightarrow O.}$) amounts to $\sim$ 40%. Under extreme conditions, *e.g.* foggy and night, the performances are degraded even more than 60%, which indicates real-life risks if autonomous vehicles heavily rely on such models.

By introducing SHIFT, which supports multi-domain and multitask studies in a single dataset, we hope to foster future research on multitask domain adaptation algorithms to counteract these domain gaps effectively. Moreover, we hope that the continuous domain shifts provided in our dataset will shed new light on this challenging problem.

## B.2. Joint training with real-world data

We investigate whether the domain variations in our dataset in combination with a specific domain of real-world data can make a model more robust to domain shift compared to a model only trained on the real-world data. Specifically, we jointly train the model with the source domain data (*i.e.*, clear daytime) from BDD100K and all domain variations from ours. The model is then evaluated on other domains of BDD100K. We employ the Faster R-CNN [63] as the model for object detection and DRN-D-54 [95] for semantic segmentation. The models are learned with the same amount of data from BDD100K but with different amounts of data from SHIFT.

**Object detection** results are shown in Tab. 9. We observe that the joint training provides a relative improvement of the source domain and OOD performance amounting to 2.52% and 3.40%, respectively.

**Semantic segmentation** has similar trends. As shown in Tab. 10, source domain mIoU improves from 46.04% to 51.20%, with a relative improvement of 10.34%. Moreover, out-of-domain mIoU rises by a relative 5.30% from 37.37% to 39.76%.

These results suggest that, if a model is trained on a limited real-world domain, jointly training with the variety of domains provided by our dataset will improve the robustness of the model to real-world shifts.

| **Training set** | **source domain** | | **OOD avg.** | |
|---|---|---|---|---|
| | AP | $AP_{75}$ | AP | $AP_{75}$ |
| BDD100K | 0.318 | 0.312 | 0.265 | 0.251 |
| BDD100K + 2k frames | 0.320 | 0.327 | 0.267 | 0.267 |
| BDD100K + 5k frames | **0.326** | **0.334** | **0.274** | **0.271** |
| BDD100K + 10k frames | 0.325 | 0.329 | 0.254 | 0.238 |

Table 9. Joint training for object detection. Generalization ability is improved with a proper amount of data.

| **Training set** | **source domain** | **OOD avg.** |
|---|---|---|
| BDD100K | 46.04 | 37.37 |
| BDD100K + 6k frames | 47.11 | 38.56 |
| BDD100K + 12k frames | **51.20** | **39.76** |
| BDD100K + 24k frames | 51.09 | 39.23 |

Table 10. Joint training for semantic segmentation. We report the mIoU. Generalization ability is improved with a proper amount of data.

| **Frame rate (Hz)** | 0.1 | 0.2 | 0.5 | 1 | 5 | 10 |
|---|---|---|---|---|---|---|
| **# Frames ($\times$1k)** | 7.5 | 15 | 37.5 | 75 | 375 | 750 |
| Seg. (mIoU, %) | 62.6 | 62.9 | **63.1** | 63.0 | 62.9 | - |
| Det. (mAP, %) | 40.6 | 43.1 | 45.8 | 46.8 | **48.4** | - |
| MOT (MOTA, %) | 25.6 | 34.7 | 45.2 | 49.3 | 54.1 | **54.9** |

Table 11. Performance of different tasks at increasing sampling rates. Training and testing on the same 1500 sequences from all domains.

## B.3. Dataset size

To understand the impact of dataset size and optimize the design of the dataset, we conduct ablation studies on: (1) sampling rate and (2) amount of sequences. Every model is trained on clear-daytime sequences.

| **Training sequence** | 350 | 750 | 1500 | 2000 | 3000 |
|---|---|---|---|---|---|
| Seg. (mIoU, %) | 59.4 | 61.4 | 63.0 | 62.6 | **63.1** |
| Det. (mAP, %) | 41.2 | 45.1 | 46.8 | 48.0 | **50.1** |

Table 12. Performance of different tasks at increasing sequences number. Training and testing on the data of 1Hz from all domains.

**Frame rate.** To avoid the model learning from redundant information, we study what is the optimal sampling rate to achieve the best performance on a given task. Here, we test the semantic segmentation, object detection, and multiple obeject tracking performance on a set of images sampled at different frame rates from a fixed set of 2000 sequences. We notice that performance of different tasks starts to saturate at different sampling rates (Tab. 11). For image-based tasks, such as segmentation and detection, we argue that the information provided by adjacent frames can be redundant, and increasing the sampling rate over a certain threshold have insignificant benefits on the resulting model performance. However, for video-based tasks, like multi-object tracking, the inter-frame information is crucial. A lower frame rate leads to lose a considerable amount of information, thus severely reducing the model performance (Tab. 11, third row).

Our dataset is collected at a fixed frame rate of 10Hz, which is necessary to support a wide range of perception tasks. However, according to the experiments on the sampling rate, we also provide a subset sampled at 1Hz for image-based perception tasks.

**Amount of sequences** is another factor affecting the performance. Here, we test semantic segmentation and object detection performance on a varying number of sequences sampled at 1Hz. In Tab. 12, we find that the performance continuously increases up to 3000 sequences. However, the performance gain is diminishing the more sequences we add. This is potentially due to the limited environmental variation in the simulator. To balance between size and learning performance, we set the total number of sequences to 3000 for the discrete set. Together with our sampling pipeline (Sec. A.3), the current size of SHIFT guarantees that for each BDD100K's domain label, we have more than 500 corresponding sequences for training and testing.

### B.4. Comparison with VIPER

As a synthetic dataset, VIPER [64] also presents sequences from discrete domain shifts. Here, we compare the segmentation performance under domain shifts in VIPER, SHIFT and BDD100K (Tab. 13). We find that the adverse conditions presented in VIPER provide a less relevant threat to model generalization, highlighting how SHIFT mimics more closely real-world trends.

| **Dataset** | daytime ($M_0$) | sunset | night | rain | $\frac{\max \Delta M}{M_0}$ |
|---|---|---|---|---|---|
| VIPER | 59.3 | 57.6 | 55.1 | 53.0 | -10.6% |
| SHIFT (*ours*) | 83.6 | 60.4 | 42.8 | 54.6 | -48.8% |
| BDD100K | 47.9 | - | 20.6 | 37.6 | -57.0% |

Table 13. Out-of-distribution performance on different datasets of a segmentation model (DRN-D) trained on the daytime domain. The last column represents the maximal relative performance drop w.r.t. source.

### B.5. Error analysis for foggy and rainy

As noticeable in Fig. 5, detection and segmentation models show a slightly different behavior under different types of domain shift. While it is first worth noticing that segmentation and detection have different label sets, we here analyze the differences in performance on the two domains presenting the largest discrepancy across the two tasks, *i.e.* foggy and rainy. For example, we find that the most drastically affected class for segmentation in the rainy domain is 'sky' (-69% mIoU w.r.t. clear-daytime), with 25% of the corresponding pixels misclassified as 'building', as opposed to only 2% under foggy conditions. For object detection, we find that most of the errors come from missed detections. The shifted domains lower the classification confidence below the pre-selected threshold, with foggy posing a greater challenge (car AP drops by 74% on foggy vs 40% on rainy).

## C. Implementation Details

In this section, we describe the implementation details and metrics for each task in Tab. 2 and Fig. 5.

**Object detection.** We compare Faster R-CNN [63], Cascade R-CNN [6], and YOLO v3 [62]. The backbone network for the first two methods is ResNet-50 [22], while YOLO v3 uses DarkNet [61] as its backbone. We use the mean Average Precision (mAP) as the metric for 2D bounding boxes. We train the models on 50k frames of data, following the "1x" schedule provided in the `mmdetection` library [97].

**Semantic segmentation.** We also compare three models for semantic segmentation, DeepLab v3+ [9], Fully Convolutional Network (FCN) [41], and DRN-D-54 [95]. All three models use the ResNet-50 [22] as the backbone. We train the models with 20k frames of data until they converge (approximately 150 epochs). We use the mean IoU (mIoU) metric for all evaluations on semantic segmentation.

**Instance segmentation.** We use Mask R-CNN [21] with ResNet-50 backbone and follow the same training routine as Faster R-CNN [63]. A segmentation mAP metric is used for evaluation.

**Depth estimation.** We use AdaBins [3] for the depth estimation experiments. It uses a U-Net-like [66] backbone structure and predicts depth with adaptive bins. The model is trained using its official implementation. We follow KITTI's benchmark on depth estimation [17]. Specifically, we use the Scale-invariant Logarithm (SILog) metric evaluated on the central crop of the image (*i.e.* Eigen Crop).

**Optical flow estimation.** We use RAFT [79] for optical flow estimation. The model is fine-tuned from pre-trained weights on the Things Dataset [46], with 10k frames of our data. The End-point Error (EPE) metric is used for evaluation.

## References


[97] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 19

[98] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. 16

[99] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 6, 16, 19

[100] 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*, pages 2117–2125, 2017. 16

[101] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015. 7, 18, 19

[102] 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*, pages 234–241. Springer, 2015. 16, 19

[103] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 472–480, 2017. 6, 7, 16, 18, 19