MUSES: The Multi-Sensor Semantic Perception Dataset for Driving under Uncertainty

Supplementary Material

A Annotation Time

The total annotation time over the 2500 images of the MUSES dataset was 11 827 hours, which translates to more than 15 months of 24-hour/day labeling for one person. During annotation, we observed a large diversity in annotation difficulty, leading to varying annotation times from 3 hours to 8 hours and 40 minutes per image. We provide a detailed breakdown of the annotation times in Tab. 7, where we list the average annotation times for each of our two stages. We thereby separate the initial drawing of the annotation and the subsequent quality control step, where a different annotator refines the annotations by trying to find mistakes in the initially drawn annotations. Quality control contributes significantly to the overall annotation time, emphasizing the significant effort needed to ensure high-quality annotations.

Table 7: Annotation time breakdown by annotation difficulty in HH:MM format. "Draw": Annotation drawing. "QC": Quality control.

Difficulty	Easy	Medium	Hard	Very hard
# Scenes	728	1345	265	9
Draw stage 1	00:45	01:45	02:30	03:00
QC stage 1	00:30	01:00	01:20	01:40
Draw stage 2	00:45	01:15	01:30	01:50
QC stage 2	01:00	01:40	02:00	02:10
Total average time	03:00	05:40	07:20	08:40

B Recording Platform

Fig. 9 shows a picture of the recording car with the mounted waterproof sensor rig.

C Calibration

To fuse multi-sensor information accurately, we need to calibrate the sensors both geometrically and temporally.

Fig. 9: Recording car with waterproof sensor rig.

Geometric calibration. The intrinsic parameters for the lidar and radar are vendor-calibrated. For the event camera, we reconstruct frames following [12] and use the same intrinsic calibration procedure as for the frame camera, using a metrology-grade checkerboard and OpenCV [2]. For the extrinsic calibration, we create a consistent transform graph between frame camera, event camera, and lidar: First, we perform pairwise calibrations: Stereo-calibration between the frame and event camera according to [12], mutual information maximization between lidar and frame camera according to [13], and mutual information maximization between lidar and event camera according to [15]. Next, we formulate a triangular pose graph and optimize it for loop closure (Powell's dog leg method) to get a consistent set of extrinsics. For radar-lidar calibration we follow [4]: the rotation is estimated via correlative scan matching using a Fourier Mellin transform [5] and the translation is simply measured. The IMU/GNSS is calibrated with u-center [1] and subsequent point cloud consistency optimization.

Synchronization. As all sensors record at different frequencies (see Tab. 2), we synchronize their internal clocks and record asynchronously. In post-processing, we match the camera frame with the lidar and radar using their mid-exposure timestamps and choose samples that minimize this delta. For the event camera, we consider the 3 seconds up until the frame camera's exposure end time, and the GNSS is interpolated to match the frame camera's time. For all sensors, timestamps in μs are provided.

Our data-recording computer is synced to a web-based GPS-time server via the network time protocol (NTP) and functions as a master clock. The radar is synced via the NTP. The lidar and frame camera are synced via the precision time protocol (PTP) with software timestamping. The event camera receives synthetic events from the frame camera at exposure start, used for temporal alignment in post-processing. The GNSS clock naturally syncs with GPS satel-

lite atomic clocks. We will release an SDK for lidar and radar ego-motion compensation and projection of all modalities to the frame camera.

D Anonymization

To protect the privacy of all individuals in our dataset, we use a semi-automatic anonymization pipeline to blur faces and license plates. Specifically, we manually draw bounding boxes over all recognizable faces and license plates in the images and use an off-the-shelf object tracker and segmenter [16] to refine the blur mask. Finally, we applied Gaussian blurring to all masks and checked the images individually.

E Uncertainty Type

Assuming human annotation as the gold standard—given our intensive quality control—any human uncertainty is attributed to aleatoric uncertainty (data-inherent and irreducible). "Difficult" pixels are those explained only by additional data available at annotation stage 2 (e.g., videos, lidar). Hence, a model that only has access to stage-1 (camera) data should not be confident in predicting the labels of "difficult" pixels, as their semantics cannot be explained by this data alone. UPQ allows the model to acknowledge uncertainty alongside making a correct prediction, thus encouraging an uncertainty-aware prediction. This rationale only holds for UPQ evaluation of camera-based models, and not multimodal ones.

F Training Details

F.1 Experiment Implementation Details

For our panoptic segmentation experiments, we train on 2 NVIDIA A100 GPUs with a batch size of 8 with all input channels normalized over the entire dataset. We train a Mask2Former [7] with an ImageNet-1K pre-trained Swin-T [11] backbone and following the mmdetection [6] configuration for the frame camera networks.

For the multimodal networks, we use a learning rate of 0.0002 and follow [3] in projecting all secondary modalities onto the 1920×1080 image plane of the frame camera. The lidar points are ego-motion-compensated and projected onto the image plane with 3 channels: range, intensity, and height. As the radar provides the full azimuth-range spectrum, we project every ego-motion-compensated intensity reading up to 150m range as its own individual point, assuming the ground level as height, into the camera plane with 2 channels: range and intensity. For the event camera, we accumulate positive and negative events in individual channels over 30ms, resulting in a 2-channel image. To avoid overly sparse input images, we dilate the projected points. Exemplary inputs for the

quadrimodal Mask2Former are visualized in Fig. 10. To have a fair comparison with the frame-camera-only network, we likewise use ImageNet-1K pre-training for the Swin-T backbones. As the pre-trained Swin backbones expect a 3-channel input, we add empty channels where necessary. To preserve the pixel values, we apply nearest neighbor interpolation in the random resizing operation during training. All inputs of one modality are randomly set to zero during training with a chance of 20%, to discourage an overreliance on individual modalities [3].

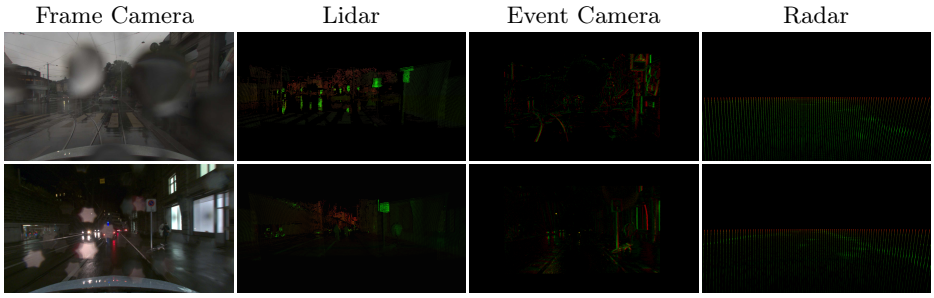

Fig. 10: Example inputs to the quadrimodal Mask2Former. From left to right: Frame camera, projected and ego-motion corrected lidar points, projected event camera, projected and ego-motion corrected radar points. The projections are highlighted for better visualization. Best viewed on a screen at full zoom.

For the semantic segmentation experiments, we use the mmsegmentation [9] framework and train Mask2Former [7] on 8 A100 GPUs with a batch size of 16. We use ImageNet-22K pre-trained weights for the Swin-L [11] backbone and train the network for 70000 iterations, following the hyperparameters used in mmsegmentation.

F.2 Multimodal Architecture Details

For our multimodal experiments, we use a Mask2Former model with different input combinations, utilizing separate pre-trained Swin-T backbones for each modality. For the bimodal networks, we therefore have 2 parallel running backbones and for the quadrimodal network, we have 4 parallel backbones. Each backbone gets a 3-channel input image from a single modality. We fuse each of the 4 outputs (feature pyramid) of the backbones individually with a parallel cross-attention block [3] before passing the fused features to the pixel decoder of Mask2Former. This fusion block allows for parallel fusion of an arbitrary amount of different input modalities. Whereby one backbone has to be picked as the *primary modality* where all other features are fused in parallel by performing standard cross attention between the *primary modality* and each *secondary modality* individually, including a skip connection. The frame camera features thereby serve as the *primary modality* and all other modalities are treated as *secondary modalities*.

F.3 Mask-Classification Baseline for Uncertainty-Aware Panoptic Segmentation

We construct a simple baseline for predicting pixel-level class and instance uncertainty scores with trained mask classification networks, such as Mask2Former [7]. During inference, mask-classification approaches predict pairs $\{(p_i, m_i)\}_{i=1}^{N}$ for each of the N masks (see [8]). $p_i \in \Delta^{K+1}$ denotes the class probability distribution over $K+1$ classes for mask i (the K object classes plus one no-object class $\varnothing$), $m_i \in [0,1]^{H \times W}$ the soft mask prediction over the image with dimensions $H \times W$. Each pixel is assigned to a probability-mask pair i^* according to

$$i^* = \arg\max_{i:c_i \neq \varnothing} p_i(c_i) \cdot m_i[h,w] \tag{2}$$

where c_i is the most likely class for each probability-mask pair. To obtain a class confidence score, we first normalize the mask predictions m_i to sum up to one, overall N masks. We then marginalize overall probability-mask pairs to find a class score

$$s_{class}[h,w] = \sum_{i=1}^{N} p_i(c_{i^*}) \cdot \bar{m}_i[h,w] \tag{3}$$

where $\bar{m}_i$ denotes the normalized mask predictions and c_{i^*} is the predicted class at that pixel. For the instance confidence, we tease out the class influence as follows:

$$\begin{aligned} s_{inst}[h,w] &= \frac{p_{i^*}(c_{i^*}) \cdot \bar{m}_{i^*}[h,w]}{\sum_{i=1}^{N} p_i(c_{i^*}) \cdot \bar{m}_i[h,w]} \\ &= \frac{p_{i^*}(c_{i^*}) \cdot \bar{m}_{i^*}[h,w]}{s_{class}[h,w]} \end{aligned} \tag{4}$$

The denominator corresponds exactly to the class confidence score, whereas the nominator corresponds to the assigned probability-mask pair score during panoptic inference in Eq. (2). This reveals an interesting interpretation: the probability-mask pair score used for panoptic inference can be decomposed into a product of class and instance confidence scores.

G Stage 1 vs. Stage 2: Detailed Results

As mentioned in Sec. 5.1 of the main paper, we try to quantify the added difficulty—and implied subsequent quality—of our second labeling stage onto annotations. We train frame-camera-only Mask2Former with a Swin-T backbone on our final panoptic ground truth (equivalent to H2 labels) and evaluate it on stage 1 (H1) and stage 2 (H2) labels. The results by condition presented in Tab. 8 show a 11.1% drop from H2 to H1, indicating substantially more difficult ground truth after the second stage of the annotation.

To further investigate the quality of the additional annotations, we train a semantic segmentation Mask2Former (Swin-L) exclusively on H1 annotations.

The performance of this model is shown in Tab. 9. The model trained on H1 annotations shows a significant performance drop on ACDC and MUSES datasets. Specifically, there is a drop of 3.3 mIoU on MUSES and 2.0 mIoU on ACDC compared to the model trained on H2 annotations. On the other hand, the scores on Cityscapes are similar for both H1 and H2-trained models. This result is expected because the additional H2 labels primarily address indistinct areas in adverse scenes, which are not present in Cityscapes. Cityscapes labels do not include adverse conditions or auxiliary data, making them more aligned with the H1 labels. As a result, the performance of the H1 and H2 models on Cityscapes is similar. In contrast, the ACDC dataset, which includes some auxiliary data, benefits significantly from the higher-quality H2 labels.

These findings collectively underscore the high quality of the additional H2 labels, as they enhance model performance in challenging conditions where auxiliary data is crucial. Together with the good generalization results in Sec. 5.3, these results indicate that the additional labeled portion of the images (see Fig. 7a) is accurately labeled. As these are also more difficult-to-predict areas, these labels guide models effectively during training, leading to improved generalization (see Sec. 5.3).

Table 8: Annotation stage wise PQ of Mask2Former [7] (Swin-T [11] backbone, frame camera input only) evaluated on stage 1 labels (H1) and stage 2 labels (H2).

	Clear	Fog	Rain	Snow	Day	Night	All
H1	57.9	57.2	57.0	52.4	58.3	51.7	58.0
H2	48.8	46.5	45.4	42.2	49.4	39.4	46.9

Table 9: Mask2Former [7] (Swin-L [11]) performance trained on H2 versus H1 annotations.

mIoU ↑	Cityscapes	ACDC	MUSES
Train on MUSES–H2	73.1	72.0	77.1
Train on MUSES–H1	73.3	70.0	73.8

H Detailed Class-Level Results

For the models presented in Sec. 5.2, we present detailed class-level PQ results in Tabs. 10 and 17 to 22. Adding an event camera to the frame-based camera performs well across the board, with the event camera showing large improvement on small dynamic classes like “motorcycle”, “bicycle” and “person”. Radar

Table 10: Test set class-wise PQ of Mask2Former [7] for different variations of input sensors. A Swin-T [11] backbone is used in all cases.

Frame camera	Event camera	Lidar	Radar	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle
✓	×	×	×	95.1	72.1	76.6	38.3	29.0	39.9	37.5	48.7	69.7	49.6	**84.0**	34.0	19.3	55.7	28.6	42.6	38.2	14.9	17.1
✓	✓	×	×	95.3	73.8	79.4	39.6	31.6	41.9	39.9	52.8	71.2	51.5	82.3	39.2	24.4	59.2	33.4	42.2	35.9	23.3	24.3
✓	×	✓	×	95.7	75.0	79.4	41.4	32.8	45.0	39.2	52.6	71.8	52.4	83.9	40.2	25.9	61.2	35.7	**52.2**	43.2	24.4	22.3
✓	×	×	✓	95.3	76.0	**80.6**	47.1	34.8	**45.9**	**43.4**	60.1	**75.5**	53.1	83.6	42.1	29.7	63.5	**40.9**	45.5	**44.4**	26.7	**24.8**
✓	✓	✓	✓	**95.9**	**76.4**	80.4	**47.4**	**37.6**	45.6	42.8	**60.2**	75.0	**53.8**	83.7	**42.2**	**33.5**	**63.7**	40.0	45.1	42.6	**28.5**	23.9

Table 11: Uncertainty-aware panoptic segmentation baselines and oracles by class in UPQ. Mask2Former [7] with Swin-T [11] is used.

Multimodal	Confidence	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle
×	Constant 100%	95.1	72.1	76.6	38.3	29.0	39.9	37.5	48.7	69.7	49.6	84.0	34.0	19.3	55.7	28.6	42.6	38.2	14.9	17.1
×	Marginalization	89.3	68.6	72.0	36.7	28.9	35.2	27.1	40.6	67.5	50.0	81.4	29.7	21.8	52.7	30.1	37.1	37.2	16.0	20.7
×	Oracle	96.6	84.0	85.9	67.7	59.5	58.5	73.0	69.2	88.4	78.2	97.1	71.8	71.5	76.7	66.1	68.7	80.0	68.9	66.3

Table 12: Class-level results for semantic segmentation with Mask2Former [7] (Swin-L [11], RGB input only). All models are evaluated on the test set of MUSES.

Training Dataset	MUSES IoU↑																		
	road	sidew.	build.	wall	fence	pole	light	sign	veget.	terrain	sky	person	rider	car	truck	bus	train	motorc.	bicycle
Cityscapes [10]	84.1	55.8	81.2	40.7	40.7	53.5	56.1	55.0	77.2	43.7	84.0	47.3	55.9	67.3	53.7	66.6	69.8	32.4	55.0
ACDC [14]	91.7	73.7	85.8	53.6	42.6	61.4	69.1	68.6	79.0	53.3	88.6	59.4	46.8	87.6	66.4	78.5	85.0	27.2	52.6
MUSES	96.5	84.9	91.8	73.3	59.5	68.1	76.7	74.2	87.5	74.5	96.3	72.8	56.5	93.2	67.6	90.2	86.8	48.7	66.2

seems to be specifically good for larger metallic objects like "bus", "train" and "car". Their metal parts make them easier to detect with radar, due to their large radar cross-section. The lidar is generally very helpful in identifying large continuous objects like "building" and "truck". These larger objects usually have good lidar returns, even in more challenging weather conditions. Further, we can also observe an especially large gap of 7.3% PQ from lidar to the other modalities for "sign". This is because traffic signs, which have reflective coatings, give off high-intensity readings for lidar.

For the models presented in Sec. 5.4, we present detailed class-level UPQ results in Tab. 11. The largest performance gaps between the baselines and the oracle exist for small *things* classes, such as "person", "rider", and "motorcycle".

Class-level results for the semantic segmentation experiments are shown in Tab. 12. The models are trained on Cityscapes [10], ACDC [14], or MUSES, and evaluated on the MUSES test set. The class-level results suggest that the "rider" and "motorcycle" classes are the most difficult, potentially due to their rarity.

I Further Dataset Statistics

I.1 Things-Classes Statistics

By projecting the lidar points onto the ground truth, we calculate distance statistics on the *things* classes. For each instance, we filter the points for outliers with a z-score of 1 and average the lidar distance. The resulting average distance per condition is presented in Tab. 13. As expected by its degrading nature onto the lidar, the fog has the lowest average *things* instance distance. We also observe a shift along the day-night axis, attributed to worse visual conditions at night making distant objects harder to identify and label. The average number of instances per image also varies largely between the different conditions. Fog does only have 2.06 instances per image, caused by two reasons. Firstly, it is harder to identify and annotate individual instances doubt-free in this condition, and secondly, heavy fog is mostly present in rural areas with less densely populated scenes.

Table 13: Average *things* class statistics by condition.

	Clear	Fog	Rain	Snow	Day	Night	All
Distance [m]	46.64	38.23	39.25	40.40	44.22	38.44	42.05
# Instances in image	9.56	2.06	12.13	7.92	8.49	7.34	8.03

I.2 Lidar Point Cloud Statistics

We present statistics on the lidar point clouds in Tab. 14 and Fig. 11. Notably, fog significantly impacts the point cloud, reducing the average points in a single lidar scan by one-third. In foggy conditions, only 3.28% of points are farther away than 40m, in contrast to 12.66% in clear weather. This is expected due to the squared attenuation effect of fog particles in the air. It underscores the limitations of lidar in highly adverse conditions, where fewer points are returned, and distant objects become imperceptible in the lidar point cloud.

Table 14: MUSES lidar point cloud statistics by weather condition.

Condition	0-20m	20-40m	>40m	Average point distance [m]	Average # of points	Average # of points in image
Clear	64.31%	23.02%	12.66%	21.95	62 862	42 331
Fog	80.69%	16.03%	3.28%	14.60	41 377	27 346
Rain	61.45%	24.30%	14.25%	22.91	53 563	35 055
Snow	70.97%	20.44%	8.59%	18.63	63 713	42 679

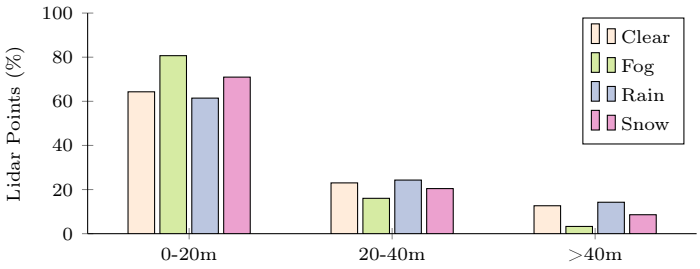

Fig. 11: MUSES share of lidar points in specific distance bins by weather condition.

I.3 Difficulty Map Distribution

We present the distribution by condition of the difficulty map in Tab. 15. 23.49% of off all pixels have a *difficult_ class* label and 6.57% of all *things* pixels have an implicit *difficult_ instance* label.

We present the distribution by condition of the difficulty map in Tab. 16. Terrain and fences have the largest share of *difficult_ class* labels. From the *things* classes, bicycle was specifically difficult to label with 21.6% of the class having a *difficult_ class* and an additional 30.6% having an explicit *difficult_ class* label.

Table 15: Difficulty map distribution by condition in %. *Things difficult_ class* regions are implicitly also *difficult_ instance*. "*difficult_ class*": share of pixels with *difficult_ class* entries in difficulty map. "*difficult_ class* excl. unlabeled": share of pixels with *difficult_ class* entries in difficulty map, excluding unlabeled pixels. "*difficult_ instance*": share of all pixels with explicit *difficult_ instance* entries in the difficulty map. "*things* w/ exp. *difficult_ instance*": share of all *things* pixels with explicit *difficult_ instance* entries in the difficulty map. "*things* w/ imp. *difficult_ instance*": share of all *things* pixels with *difficult_ instance* label, including *difficult_ class* labels that are implicitly also *difficult_ instance*.

Condition	*difficult_ class*	*difficult_ class* excl. unlabeled	*difficult_ instance*	*things* w/ exp. *difficult_ instance*	*things* w/ imp. *difficult_ instance*
Clear	13.97	5.04	0.03	1.08	4.71
Rain	22.71	9.65	0.12	2.64	9.58
Snow	19.35	9.69	0.03	0.80	3.51
Fog	40.68	18.59	0.01	0.62	11.46
Day	10.12	5.86	0.04	1.23	4.57
Night	43.56	17.11	0.06	1.95	9.63
Total	23.49	10.36	0.05	1.51	6.57

Table 16: Difficulty map distribution by class in %. Share of explicit difficulty labels compared to the total labeled pixels of a given class.

	road	sidew.	build.	wall	fence	pole	light	sign	veget.	terrain	sky	person	rider	car	truck	bus	train	motorc.	bicycle
difficult_class	3.4	13.0	9.1	17.4	23.8	12.9	15.8	10.3	20.7	37.2	21.3	16.6	15.9	3.4	5.0	4.6	3.7	15.3	21.6
difficult_instance	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	2.0	0.1	0.6	0.0	0.0	1.5	4.9	30.6

Table 17: *Clear*-test split class-wise PQ of Mask2Former [7] for different variations of input sensors. A Swin-T [11] backbone is used in all cases.

Frame camera	Event camera	Radar	Lidar	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle
✓	×	×	×	97.2	82.6	82.5	40.6	28.0	40.4	38.1	46.1	70.0	51.4	92.5	36.2	23.2	55.3	27.6	35.8	37.7	20.0	22.8
✓	✓	×	×	97.6	83.7	83.8	43.9	28.7	43.2	38.4	49.5	70.7	53.8	92.2	41.6	27.1	59.6	38.2	44.7	34.3	31.4	27.1
✓	×	✓	×	97.8	85.0	84.6	44.9	30.0	46.2	36.7	48.7	71.8	54.6	93.1	43.3	28.7	61.0	35.6	49.1	39.4	30.4	24.5
✓	×	×	✓	97.7	85.2	84.8	52.7	32.7	45.9	41.2	57.3	75.7	51.9	92.6	44.7	31.3	63.2	44.5	36.2	34.0	31.6	28.5
✓	✓	✓	✓	97.6	85.2	86.0	52.9	32.3	47.2	40.8	58.5	75.8	53.8	92.9	45.7	34.0	62.4	42.3	47.8	38.2	32.4	24.4

J European Domain Bias

The MUSES dataset consists exclusively of driving scenes from Switzerland, which may introduce a geographical bias towards Western European environments. This limitation is inherent in the dataset's design and location. While this regional focus ensures consistency and depth within a specific context, it may affect the generalizability of the results to other regions with different driving conditions, infrastructure, and weather patterns.

K Visualization of MUSES Samples

We show further visualizations of MUSES samples in Figs. 12 and 13. The lidar and event camera are thereby projected onto the frame camera for easier inspection. In many of the scenes, the lidar and event camera help identify and clarify unclear areas. This is especially noticeable in rainy conditions, where the frame camera is often blurred by droplets.

L Qualitative Results

We visualize some exemplary panoptic segmentation predictions results of the uni- and quadrimodal Mask2Former with their respective class and instance uncertainty maps in Figs. 14 and 15. The class and instance uncertainty scores are calculated according to Appendix F.3.

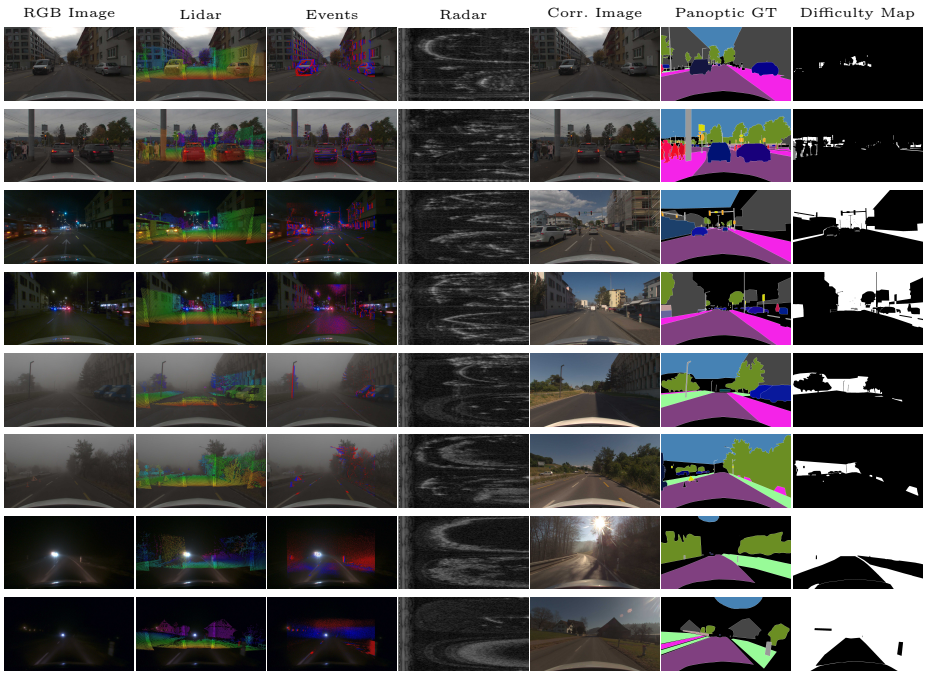

Fig. 12: Visualization of MUSES samples. From left to right: RGB image; motion-compensated lidar points projected and overlaid with the image; events projected onto the image (assuming infinite distance); azimuth-range radar scan (with ranges above a threshold cropped out); corresponding normal-condition image; panoptic ground truth; difficulty map. Best viewed zoomed in.

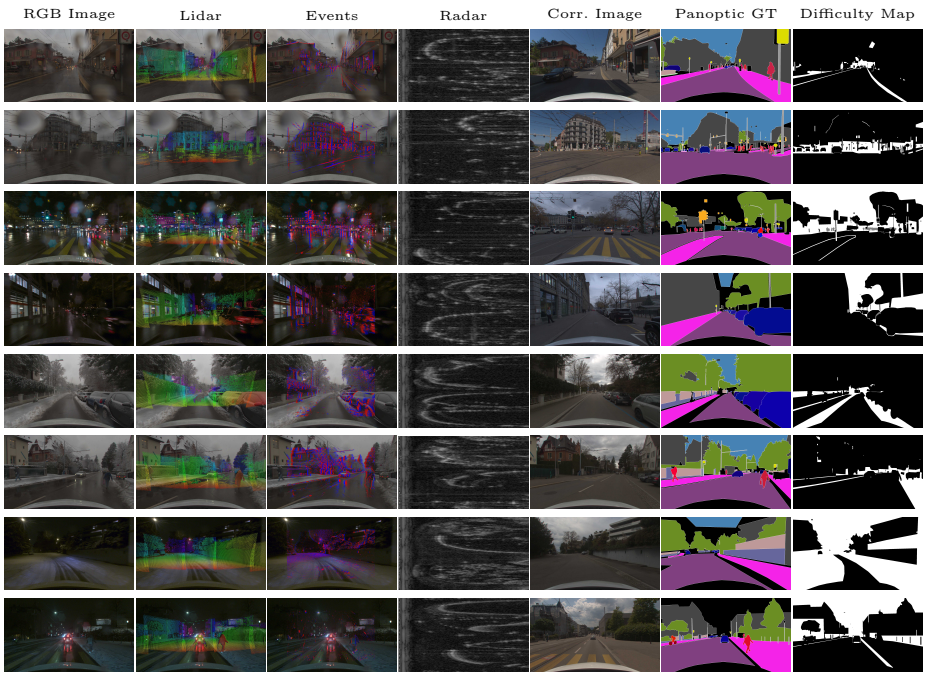

Fig. 13: Visualization of MUSES samples (continued). From left to right: RGB image; motion-compensated lidar points projected and overlaid with the image; events projected onto the image (assuming infinite distance); azimuth-range radar scan (with ranges above a threshold cropped out); corresponding normal-condition image; panoptic ground truth; difficulty map. Best viewed zoomed in.

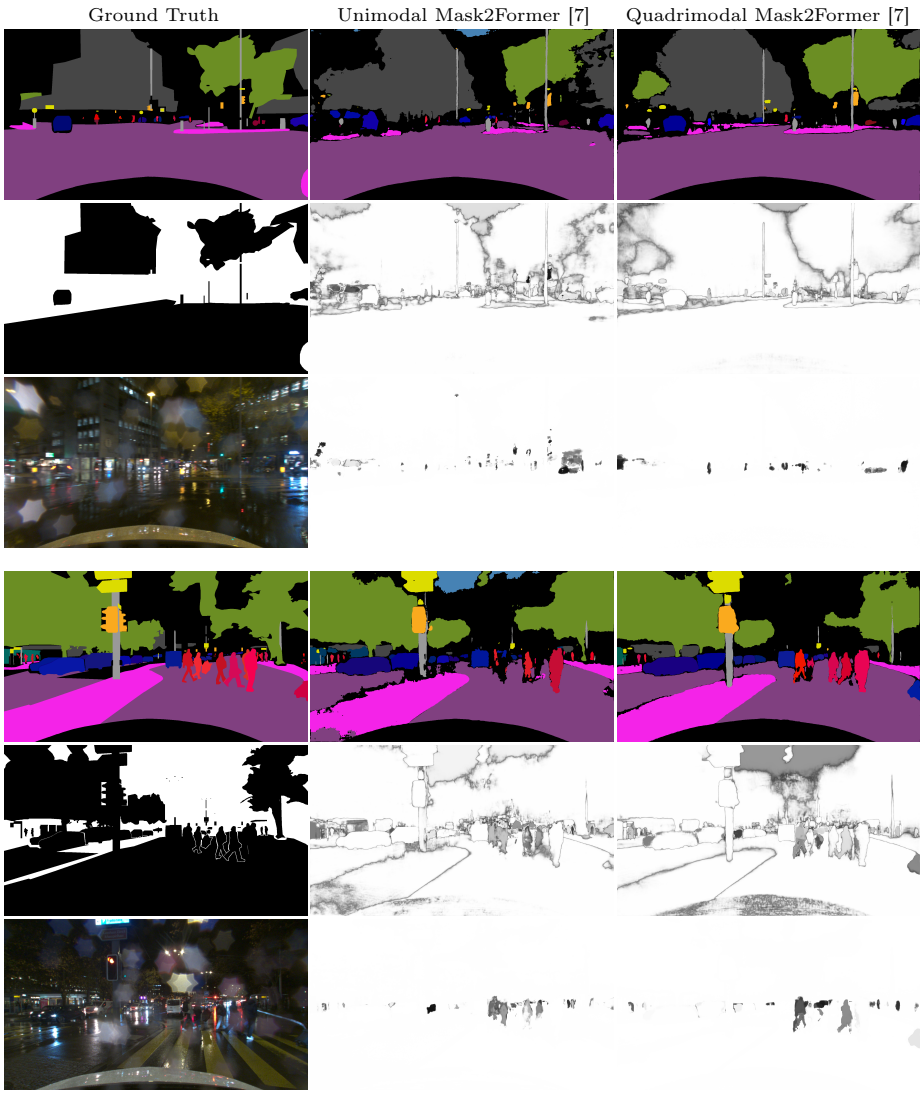

Fig. 14: Qualitative panoptic results. First column from top to bottom: panoptic segmentation ground-truth annotation, difficulty map, input frame camera image. Second and third columns from top to bottom: the panoptic predictions, class uncertainty scores, and instance uncertainty scores. Best viewed on a screen at full zoom.

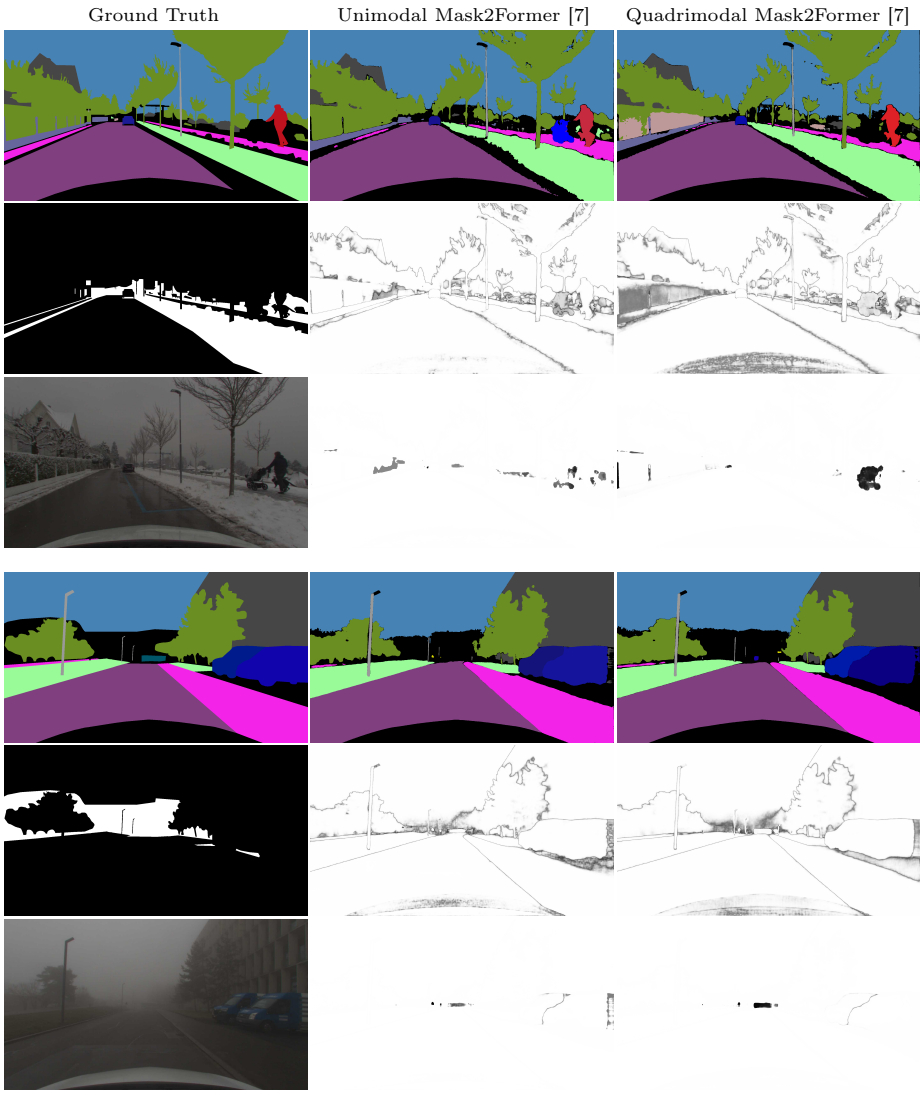

Fig. 15: Qualitative panoptic results (continued). First column from top to bottom: panoptic segmentation ground-truth annotation, difficulty map, input frame camera image. Second and third columns from top to bottom: the panoptic predictions, class uncertainty scores, and instance uncertainty scores. Best viewed on a screen at full zoom.

Table 18: *Fog*-test split class-wise PQ of Mask2Former [7] for different variations of input sensors. A Swin-T [11] backbone is used in all cases.

Frame camera	Event camera	Radar	Lidar	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle
✓	×	×	×	94.8	57.8	69.9	25.1	32.2	45.7	71.9	39.1	67.7	63.8	79.5	44.6	0.0	61.1	43.6	35.5	n/a	0.0	4.4
✓	✓	×	×	94.2	58.1	76.7	12.8	31.2	50.9	72.2	42.9	70.7	65.5	74.2	48.3	0.0	61.6	39.5	32.5	n/a	46.5	10.8
✓	×	✓	×	95.5	62.2	74.8	23.0	30.4	51.6	65.5	41.0	73.5	68.2	77.8	48.8	0.0	66.5	45.2	46.6	n/a	0.0	20.7
✓	×	×	✓	94.0	57.3	76.0	29.1	38.0	50.5	71.0	53.6	77.4	67.4	76.9	43.4	0.0	64.8	48.3	41.2	n/a	0.0	15.3
✓	✓	✓	✓	95.4	60.0	71.1	26.3	43.6	50.9	71.3	51.6	74.7	68.2	76.6	47.3	0.0	68.0	49.3	40.2	n/a	0.0	11.7

Table 19: *Rain*-test split class-wise PQ of Mask2Former [7] for different variations of input sensors. A Swin-T [11] backbone is used in all cases.

Frame camera	Event camera	Radar	Lidar	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle
✓	×	×	×	93.4	71.1	73.3	30.2	26.1	43.0	35.8	55.4	69.1	36.5	81.2	32.4	16.9	54.8	30.0	41.7	39.6	15.7	16.2
✓	✓	×	×	93.7	73.6	76.9	33.4	32.6	42.1	39.8	61.3	70.4	43.1	81.2	38.1	23.1	58.9	28.0	40.5	36.3	16.4	26.1
✓	×	✓	×	94.1	76.1	76.8	34.3	32.9	47.8	41.1	61.0	69.7	34.5	81.9	38.5	24.4	60.8	37.4	49.9	41.3	22.9	22.2
✓	×	×	✓	94.6	79.0	78.3	39.6	37.4	50.4	45.6	67.8	73.6	44.4	83.3	42.7	32.2	63.1	38.9	50.7	50.4	28.3	25.1
✓	✓	✓	✓	94.9	78.8	79.9	42.8	44.5	49.0	44.2	66.9	74.0	40.0	84.0	42.1	35.3	64.4	40.7	40.2	45.8	27.3	26.6

Table 20: *Snow*-test split class-wise PQ of Mask2Former [7] for different variations of input sensors. A Swin-T [11] backbone is used in all cases.

Frame camera	Event camera	Radar	Lidar	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle
✓	×	×	×	94.5	68.4	76.4	43.6	30.8	30.3	35.3	50.9	71.7	34.8	81.1	31.8	18.1	56.7	15.0	64.3	32.4	6.2	14.7
✓	✓	×	×	94.8	70.9	78.1	44.9	33.8	31.1	38.9	54.0	73.0	33.8	80.1	35.8	27.1	58.6	23.8	51.2	37.5	16.4	21.6
✓	×	✓	×	94.9	69.0	78.2	46.7	36.6	34.0	36.6	56.3	72.4	37.7	81.2	37.2	30.0	61.2	24.0	68.0	57.5	16.6	19.5
✓	×	×	✓	94.3	72.4	80.4	50.6	33.0	36.6	38.7	59.3	75.1	39.6	80.5	36.7	19.1	64.2	28.9	60.1	39.9	17.7	22.0
✓	✓	✓	✓	95.0	72.7	79.3	50.3	33.6	34.8	40.2	60.7	75.2	41.3	79.9	36.2	35.7	63.9	23.4	52.8	38.7	24.5	21.7

Table 21: *Day*-test split class-wise PQ of Mask2Former [7] for different variations of input sensors. A Swin-T [11] backbone is used in all cases.

Frame camera	Event camera	Radar	Lidar	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle
✓	×	×	×	95.7	72.9	79.6	38.8	32.1	39.7	43.9	46.2	84.0	53.4	95.7	34.8	19.2	56.2	30.0	46.0	34.9	17.6	16.9
✓	✓	×	×	96.1	74.7	81.6	40.1	33.9	42.3	43.5	49.4	84.8	55.1	96.1	40.2	23.9	59.6	35.3	45.0	28.8	27.8	24.2
✓	×	✓	×	96.2	74.9	81.4	40.3	35.8	45.4	44.4	49.8	85.2	54.6	96.0	41.0	25.0	61.4	38.5	54.1	30.6	26.6	24.2
✓	×	×	✓	95.8	75.6	81.6	46.0	36.5	45.8	44.6	55.7	84.8	54.8	95.3	41.5	26.2	63.1	42.6	48.6	38.3	27.1	24.7
✓	✓	✓	✓	96.0	75.4	81.1	44.7	39.5	44.7	44.6	55.6	84.6	55.8	95.8	42.1	28.6	63.0	41.7	45.0	32.4	28.9	27.5

Table 22: *Night*-test split class-wise PQ of Mask2Former [7] for different variations of input sensors. A Swin-T [11] backbone is used in all cases.

Frame camera	Event camera	Radar	Lidar	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle
✓	×	×	×	94.2	70.6	70.9	37.5	22.3	40.1	29.1	52.9	43.4	39.6	49.4	32.6	19.4	55.0	0.0	22.5	40.4	11.0	17.3
✓	✓	×	×	93.9	72.3	75.2	38.8	26.8	41.4	35.0	58.5	46.5	41.8	40.4	37.2	25.0	58.6	0.0	29.7	40.3	16.7	24.5
✓	×	✓	×	95.1	75.2	75.6	43.3	26.5	44.3	32.4	57.3	47.3	46.6	48.6	38.7	26.8	60.9	0.0	41.5	50.3	20.9	19.8
✓	×	×	✓	94.6	76.7	78.7	49.0	31.2	46.1	41.7	67.3	58.7	48.7	49.0	43.2	33.0	64.1	17.5	27.6	48.2	26.2	24.9
✓	✓	✓	✓	95.6	78.2	79.0	51.8	33.3	46.9	40.5	67.9	57.8	48.6	46.4	42.3	37.9	64.6	12.3	45.7	49.2	28.0	18.9

References

1. u-blox AG: u-center: GNSS evaluation software for Windows. `https://www.u-blox.com/en/product/u-center` (2023)
2. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
3. Broedermann, T., Sakaridis, C., Dai, D., Van Gool, L.: HRFuser: A multi-resolution sensor fusion architecture for 2D object detection. In: ITSC (2023)
4. Burnett, K., Yoon, D.J., Wu, Y., Li, A.Z., Zhang, H., Lu, S., Qian, J., Tseng, W.K., Lambert, A., Leung, K.Y., et al.: Boreas: A multi-season autonomous driving dataset. The International Journal of Robotics Research **42**(1-2), 33–42 (2023)
5. Checchin, P., Gérossier, F., Blanc, C., Chapuis, R., Trassoudaine, L.: Radar scan matching slam using the Fourier-Mellin transform. In: Field and Service Robotics: Results of the 7th International Conference (2010)
6. Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C.C., Lin, D.: MMDetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155 (2019)
7. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: CVPR (2022)
8. Cheng, B., Schwing, A., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. In: NeurIPS (2021)
9. Contributors, M.: MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation` (2020)
10. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
11. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV (2021)
12. Muglikar, M., Gehrig, M., Gehrig, D., Scaramuzza, D.: How to calibrate your event camera. In: CVPR (2021)
13. Pandey, G., McBride, J.R., Savarese, S., Eustice, R.M.: Automatic extrinsic calibration of vision and lidar by maximizing mutual information. Journal of Field Robotics **32**(5), 696–722 (2015)
14. Sakaridis, C., Dai, D., Van Gool, L.: ACDC: The Adverse Conditions Dataset with Correspondences for semantic driving scene understanding. In: ICCV (2021)
15. Ta, K., Bruggemann, D., Brödermann, T., Sakaridis, C., Van Gool, L.: L2E: Lasers to events for 6-dof extrinsic calibration of lidars and event cameras. In: ICRA (2023)
16. Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H.: Fast online object tracking and segmentation: A unifying approach. In: CVPR (2019)