OpenESS: Event-based Semantic Scene Understanding with Open Vocabularies

Supplementary Material

Table of Contents

A. Additional Implementation Details

In this section, we provide additional details to assist the implementation and reproduction of the approaches in the proposed OpenESS framework.

A.1. Datasets

In this study, we follow prior works [2, 7, 8, 16] by using the ***DDD17-Seg*** [2] and ***DSEC-Semantic*** [16] datasets for evaluating and validating the baselines, prior methods, and the proposed OpenESS framework. Some specifications related to these two datasets are listed as follows.

- **DDD17-Seg** [2] serves as the first benchmark for ESS. It is a semantic segmentation extension of the DDD17 [3] dataset, which includes hours of driving data, capturing a variety of driving conditions such as different times of day, traffic scenarios, and weather conditions. Alonso and Murillo [2] provide the semantic labels on top of DDD17 to enable event-based semantic segmentation. Specifically, they proposed to use the corresponding gray-scale images along with the event streams to generate an approximated set of semantic labels for training, which was proven effective in training models to segment directly on event-based data. A three-step procedure is applied: *i)* train a semantic segmentation model on the gray-scale images in the *Cityscapes* dataset [5]; *ii)* Use the trained model to label the gray-scale images in DDD17; and *iii)* Conduct a post-processing step on the generated pseudo labels, including class merging and image cropping. The dataset specification is shown in Tab. A. In total, there are 15950 training and 3890 test samples in the DDD17-Seg dataset. Each pixel is labeled across six semantic classes, including `flat`, `background`, `object`, `vegetation`, `human`, and `vehicle`. For each sample, we convert the event streams into a sequence of 20 voxel grids, each consisting of 32000 events and with a spatial resolution of 352 × 200. For additional details of this dataset, kindly refer to http://sensors.ini.uzh.ch/news_page/DDD17.html.
- **DSEC-Semantic** [16] is a semantic segmentation extension of the DSEC (Driving Stereo Event Camera) dataset [6]. DSEC is an extensive dataset designed for advanced driver-assistance systems (ADAS) and autonomous driving research, with a particular focus on event-based vision and stereo vision. Different from DDD17 [3], the DSEC dataset combines data from event-based cameras and traditional RGB cameras. The inclusion of event-based cameras (which capture changes in light intensity) alongside regular cameras provides a rich, complementary data source for perception tasks. The dataset typically features high-resolution images and event data, providing detailed visual information from a wide range of driving conditions, including urban, suburban, and highway environments, various weather conditions, and different times of the day. This diversity is crucial for developing systems that can operate reliably in real-world conditions. Based on such a rich collection, Sun *et al.* [16] adopted a similar pseudo labeling procedure as DDD17-Seg [2] and generated the semantic labels for eleven sequences in DSEC, dubbed as DSEC-Semantic. The dataset specification is shown in Tab. B. In total, there are 8082 training and 2809 test samples in the DSEC-Semantic dataset. Each pixel is labeled across eleven semantic classes, including `background`, `building`, `fence`, `person`, `pole`, `road`, `sidewalk`, `vegetation`, `car`, `wall`, and `traffic-sign`. For each sample, we convert the event streams into a sequence of 20 voxel grids, each consisting of 100000 events and with a spatial resolution of 640 × 440. For additional details of this dataset, kindly refer to

Table A. The specifications of the *DDD17-Seg* dataset [2].

-	Training					Test
Seq	dir0	dir3	dir4	dir6	dir7	dir1
# Frames	11785	20051	41071	28411	58650	71680
# Events	5550	1320	6945	1140	995	3890
Resolution	352 × 200					352 × 200
# Classes	6 Classes					6 Classes

Table B. The specifications of the *DSEC-Semantic* dataset [16].

-	Training								Test		
Seq	00_a	01_a	02_a	04_a	05_a	06_a	07_a	08_a	13_a	14_c	15_a
# Frames	939	681	235	701	1753	1523	1463	787	379	1191	1239
# Events	933	675	229	695	1747	1517	1457	781	373	1185	1233
Resolution	640 × 440								640 × 440		
# Classes	11 Classes								11 Classes		

https://dsec.ifi.uzh.ch/dsec-semantic.

A.2. Text Prompts

To enable the conventional evaluation of our proposed open-vocabulary approach on an event-based semantic segmentation dataset, we need to use the pre-defined class names as text prompts to generate the text embedding. Specifically, we follow the standard templates [13] when generating the embedding. The dataset-specific text prompts defined in our framework are listed as follows.

- **DDD17-Seg.** There is a total of six semantic classes in the DDD17-Seg dataset [2], with static and dynamic components of driving scenes. Our defined text prompts of this dataset are summarized in Tab. C. For each semantic class, we generate for each text prompt the text embedding using the CLIP text encoder and then average the text embedding of all text prompts as the final embedding of this class.
- **DSEC-Semantic.** There is a total of eleven semantic classes in the DSEC-Semantic dataset [16], ranging from static and dynamic components of driving scenes. Our defined text prompts of this dataset are summarized in Tab. D. For each semantic class, we generate for each text prompt the text embedding using the CLIP text encoder and then average the text embedding of all text prompts as the final embedding of this class.

A.3. Superpixels

In image processing and computer vision, superpixels can be defined as a scheme that groups pixels in an image into perceptually meaningful atomic regions, which are used to replace the rigid structure of the pixel grid [1]. Superpixels provide a more natural representation of the image structure, often leading to more efficient and effective image processing. Here are some of their key aspects:

- **Grouping Pixels.** Superpixels are often formed by clustering pixels based on certain criteria like color similarity, brightness, texture, and other low-level patterns [1], or more recently, semantics [10]. This results in contiguous regions in the image that are more meaningful than individual pixels for many applications [4, 11, 12, 17].
- **Reducing Complexity.** By aggregating pixels into superpixels, the complexity of image data is significantly reduced [15]. This reduction helps in speeding up subsequent image processing tasks, as algorithms have fewer elements (superpixels) to process compared to the potentially millions of pixels in an image.
- **Preserving Edges.** One of the primary goals of superpixel segmentation is to preserve important image edges. Superpixels often adhere closely to the boundaries of objects in the image, making them useful for tasks that rely on accurate edge information, like object recognition and scene understanding.

In this work, we propose to first leverage calibrated frames to generate coarse, instance-level superpixels and then distill knowledge from a pre-trained image backbone to the event segmentation network. Specifically, we resort to the following two ways to generate the superpixels.

- **SLIC.** The first way is to leverage the heuristic Simple Linear Iterative Clustering (SLIC) approach [1] to

Table C. The text prompts defined on the *DDD17-Seg* dataset [2] (6 classes) used for generating the CLIP text embedding.

	DDD17 (6 classes)	
#	**class**	**text prompt**
0	`flat`	'road', 'driveable', 'street', 'lane marking', 'bicycle lane', 'roundabout lane', 'parking lane', 'terrain', 'grass', 'soil', 'sand', 'lawn', 'meadow', 'turf'
1	`background`	'sky', 'building'
2	`object`	'pole', 'traffic sign pole', 'traffic light pole', 'traffic light box', 'traffic-sign', 'parking-sign', 'direction-sign'
3	`vegetation`	'vegetation', 'vertical vegetation', 'tree', 'tree trunk', 'hedge', 'woods', 'terrain', 'grass', 'soil', 'sand', 'lawn', 'meadow', 'turf'
4	`human`	'person', 'pedestrian', 'walking people', 'standing people', 'sitting people', 'toddler'
5	`vehicle`	'car', 'jeep', 'SUV', 'van', 'caravan', 'truck', 'box truck', 'pickup truck', 'trailer', 'bus', 'public bus', 'train', 'vehicle-on-rail', 'tram', 'motorbike', 'moped', 'scooter', 'bicycle'

Table D. The ext prompts defined on the *DSEC-Semantic* dataset [16] (11 classes) used for generating the CLIP text embedding.

	DSEC-Semantic (11 classes)	
#	**class**	**text prompt**
0	`background`	'sky'
1	`building`	'building', 'skyscraper', 'house', 'bus stop building', 'garage', 'carport', 'scaffolding'
2	`fence`	'fence', 'fence with hole'
3	`person`	'person', 'pedestrian', 'walking people', 'standing people', 'sitting people', 'toddler'
4	`pole`	'pole', 'electric pole', 'traffic sign pole', 'traffic light pole'
5	`road`	'road', 'driveable', 'street', 'lane marking', 'bicycle lane', 'roundabout lane', 'parking lane'
6	`sidewalk`	'sidewalk', 'delimiting curb', 'traffic island', 'walkable', 'pedestrian zone'
7	`vegetation`	'vegetation', 'vertical vegetation', 'tree', 'tree trunk', 'hedge', 'woods', 'terrain', 'grass', 'soil', 'sand', 'lawn', 'meadow', 'turf'
8	`car`	'car', 'jeep', 'SUV', 'van', 'caravan', 'truck', 'box truck', 'pickup truck', 'trailer', 'bus', 'public bus', 'train', 'vehicle-on-rail', 'tram', 'motorbike', 'moped', 'scooter', 'bicycle'
9	`wall`	'wall', 'standing wall'
10	`traffic-sign`	'traffic-sign', 'parking-sign', 'direction-sign', 'traffic-sign without pole', 'traffic light box'

efficiently group pixels from frame I_i^{img} into a total of M_{slic} segments with good boundary adherence and regularity. The superpixels are defined as $I_i^{sp} = \{\mathcal{I}_i^1, \mathcal{I}_i^2, ..., \mathcal{I}_i^{M_{slic}}\}$, where M_{slic} is a hyperparameter that needs to be adjusted based on the inputs. The generated superpixels satisfy $\mathcal{I}_i^1 \cup \mathcal{I}_i^2 \cup ... \cup \mathcal{I}_i^{M_{slic}} = \{1, 2, ..., H \times W\}$. Several examples of the SLIC-generated superpixels are shown in the second row of Fig. A, where each of the color-coded patches represents one distinct and semantically coherent superpixel.

- **SAM.** For the second option, we use the recent Segment Anything Model (SAM) [10] which takes I_i^{img} as

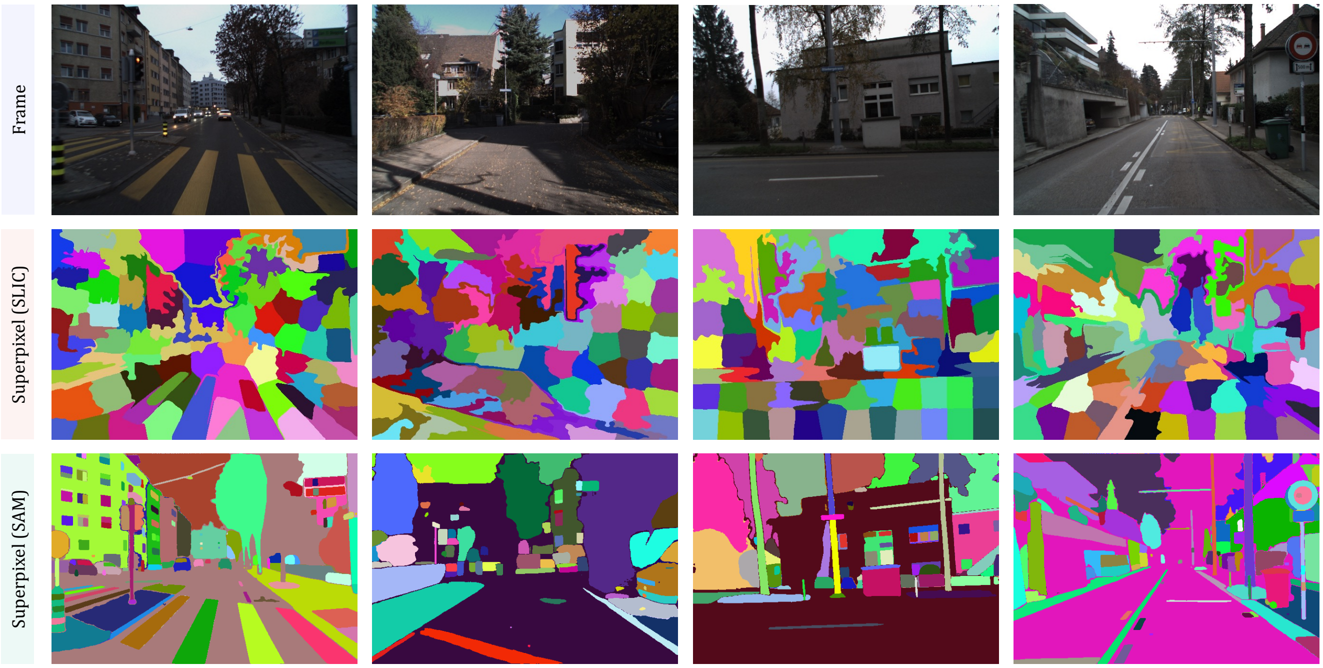


Figure A. **Examples of superpixels** generated by SLIC [1] (the 2nd row) and SAM [10] (the 3rd row). The parameter M_{slic} in the SLIC algorithm is set to 100. Each colored patch represents one distinct and semantically coherent superpixel. Best viewed in colors.

the input and outputs M_{sam} class-agnostic masks. For simplicity, we use M to denote the number of superpixels used during knowledge distillation, *i.e.*, $\{I_i^{sp} = \{\mathcal{I}_i^1, ..., \mathcal{I}_i^k\} | k = 1, ..., M\}$. Several examples of the SAM-generated superpixels are shown in the third row of Fig. A, where each of the color-coded patches represents one distinct and semantically coherent superpixel.

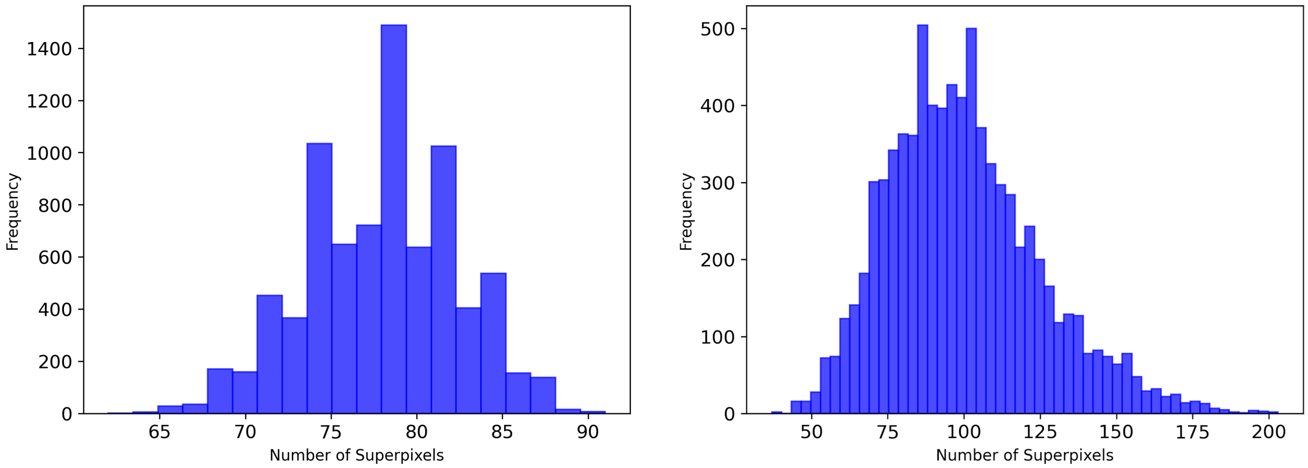
We calculate the SLIC and SAM superpixel distributions on the training set of the DSEC-Semantic dataset [16] and show the corresponding statistics in Fig. B. As can be observed, the SLIC-generated superpixels often contain more low-level visual cues, such as color similarity, brightness, and texture. On the contrary, superpixels generated by SAM exhibit clear semantic coherence and often depict the boundaries of objects and backgrounds. As verified in the main body of this paper, the semantically richer SAM superpixels bring higher performance gains in our Frame-to-Event Contrastive Learning framework.

Meanwhile, we provide more fine-grained examples of the SLIC algorithm using different M_{slic}, *i.e.*, 25, 50, 100, 150, and 200. The results are shown in Fig. C. Specifically, the number of superpixels M_{slic} should reflect the complexity and detail of the image. For images with high detail or complexity (like those with many objects or textures), a larger M_{slic} can capture more of this detail. Conversely, for simpler images, fewer superpixels might be sufficient. Usually, more superpixels mean smaller superpixels. Smaller superpixels can adhere more closely to object boundaries and capture finer details, but they might also capture more noise. Fewer superpixels result in larger, more homogeneous regions but may lead to a loss of detail, especially at the edges of objects. The choice also depends on the specific application. For instance, in object detection or segmentation tasks where boundary adherence is crucial, a higher number of superpixels might be preferable. In contrast, for tasks like image compression or abstraction, fewer superpixels might be more appropriate. Often, the optimal number of superpixels is determined empirically. This involves experimenting with different values and evaluating the results based on the specific criteria of the task or application. In our event-based semantic segmentation task, we choose $M_{slic} = 100$ for our Frame-to-Event Contrastive Learning on the DSEC-Semantic dataset [16], and $M_{slic} = 25$ on the DDD17-Seg dataset [2].

Since I_i^{evt} and I_i^{img} have been aligned and synchronized, we can group events from I_i^{evt} into superevents $\{V_i^{sp} = \{\mathcal{V}_i^1, ..., \mathcal{V}_i^l\} | l = 1, ..., M\}$ by using the known event-pixel correspondences.

A.4. Backbones

As mentioned in the main body of this paper, we establish three open-vocabulary event-based semantic segmentation settings based on the use of three different event representations, *i.e.*, `frame2voxel`, `frame2recon`, and `frame2spike`. It is worth noting that these three event representations tend to have their own advantages.



(a) Histogram of SLIC-Generated Superpixels

(b) Histogram of SAM-Generated Superpixels

Figure B. **The statistical distributions** of superpixels generated by SLIC [1] (subfigure a) and SAM [10] (subfigure b).

Table E. **The per-class segmentation results** of annotation-free event-based semantic segmentation approaches on the test set of *DSEC-Semantic* [16]. Scores reported are IoUs in percentage (%). For each semantic class, the best score in each column is highlighted in **bold**.

Method	mIoU	background	building	fence	person	pole	road	sidewalk	vegetation	car	wall	traffic-sign	Acc
Annotation-Free ESS													
MaskCLIP [19]	21.97	26.45	52.59	0.20	0.04	**4.19**	65.76	2.96	48.02	40.67	0.67	0.08	58.96
FC-CLIP [18]	39.42	87.49	69.68	**14.39**	**17.53**	0.29	71.76	34.56	71.30	63.19	2.98	0.50	79.20
OpenESS (Ours)	**43.31**	**92.53**	**74.22**	11.96	0.00	0.41	**87.32**	**55.09**	**74.23**	**64.25**	**7.98**	**8.47**	**86.18**

We supplement additional implementation details regarding the used event representations as follows.

- **Frame2Voxel.** For the use of *voxel grids* as the event embedding, we follow Sun *et al.* [16] by converting the raw events ε_i into the regular voxel grids $I_i^{vox} \in \mathbb{R}^{C\times H\times W}$ as the input to the event-based semantic segmentation network. This representation is intuitive and aligns well with conventional event camera data processing techniques. It is suitable for convolutional neural networks as it maintains spatial and temporal relationships. Specifically, with a predefined number of events, each voxel grid is built from non-overlapping windows as follows:

$$I_i^{vox} = \sum_{\mathbf{e}_j \in \varepsilon_i} p_j \delta(\mathbf{x}_j - \mathbf{x})\delta(\mathbf{y}_j - \mathbf{y}) \max\{1 - |t_j^* - t|, 0\}, \tag{1}$$

where δ is the Kronecker delta function; $t_j^* = (B-1)\frac{t_j - t_0}{\Delta T}$ is the normalized event timestamp with B as the number of temporal bins in an event stream; ΔT is the time window and t_0 denotes the time of the first event in the window. It is worth noting that *voxel grids* can be memory-intensive, especially for high-resolution sensors or long-time windows. They might also introduce quantization errors due to the discretization of space and time. For additional details on the use of *voxel grids*, kindly refer to https://github.com/uzh-rpg/ess.

- **Frame2Recon.** For the use of *event reconstructions* as the event embedding, we follow Sun *et al.* [16] and Rebecq *et al.* [14] by converting the raw events ε_i into the regular frame-like event reconstructions $I_i^{rec} \in \mathbb{R}^{H\times W}$ as the input to the event-based semantic segmentation network. This can be done by accumulating events over short time intervals or by using algorithms to interpolate or simulate frames. This approach is compatible with standard image processing techniques and algorithms developed for frame-based vision. It is more familiar to practitioners used to working with conventional cameras. In this work, we adopt the E2VID model [14] to generate the *event reconstructions*. This process can be described as follows:

$$\mathbf{z}_k^{rec} = E_{\text{e2vid}}(I_k^{vox}, \mathbf{z}_{k-1}^{rec}), \quad k = 1, ..., N, \tag{2}$$

$$I_i^{rec} = D_{\text{e2vid}}(\mathbf{z}^{rec}), \tag{3}$$

where I_k^{vox} denotes the *voxel grids* as defined in Eq. (1); E_{e2vid} and D_{e2vid} are the encoder of decoder of the E2VID

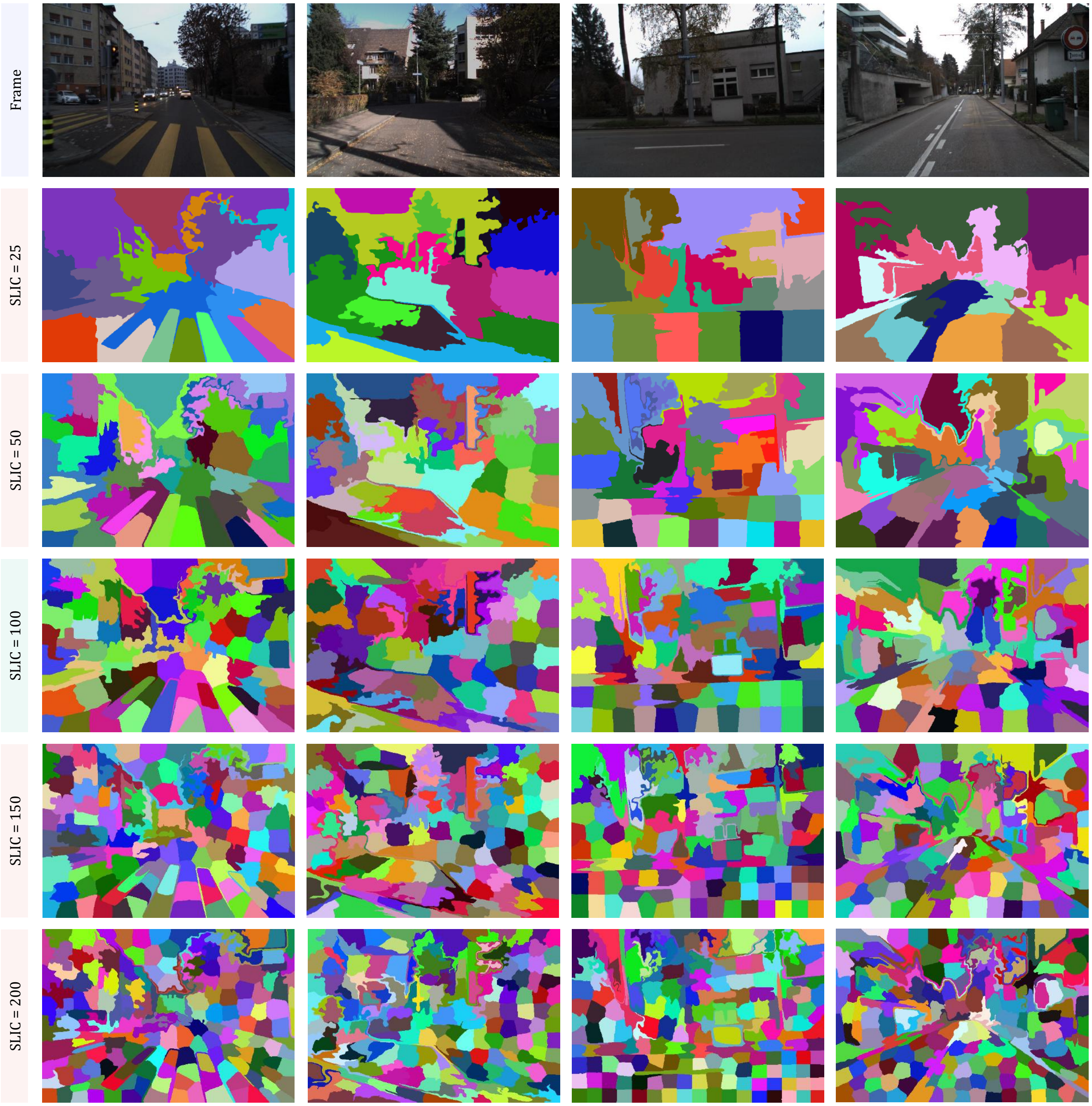


Figure C. **Examples of superpixels** generated by SLIC [1] with different numbers of superpixels M_{slic} (25, 50, 100, 150, and 200). Each colored patch represents one distinct and semantically coherent superpixel. Best viewed in colors.

model [14], respectively. It is worth noting that *event reconstructions* can lose the fine temporal resolution that event cameras provide. They might also introduce artifacts or noise, especially in scenes with fast-moving objects or low event rates. For additional details on the use of *event reconstructions*, kindly refer to https://github.com/uzh-rpg/rpg_e2vid.

- **Frame2Spike.** For the use of *spikes* as the event embedding, we follow Kim *et al.* [9] by converting the raw events ε_i into spikes $I_i^{spk} \in \mathbb{R}^{H\times W}$ as the input to the event-based semantic segmentation network. The spike representation keeps the data in its raw form – as individual spikes or events. This representation preserves the high temporal resolution of the event data and is highly efficient in terms of memory and computation, especially for sparse scenes. The rate coding is used as the spike encoding scheme due to its reliable performance across various tasks. Each pixel value with a random number ranging between $[s_{min}, s_{max}]$ at every time step is recorded, where s_{min} and s_{max} are the minimum and maximum possible pixel intensities, respectively. If the random number is greater than the pixel intensity, the Poisson spike generator outputs a spike with amplitude 1. Otherwise, the Poisson spike generator does not yield any spikes. The spikes in a certain time window are accumulated to generate a frame, where such frames will serve as the input to the event-based semantic segmentation network. It is worth noting that processing raw spike data requires specialized algorithms, often inspired by neuromorphic computing. It might not be suitable for traditional image processing techniques and can be challenging to interpret and visualize. For additional details on the use of *spikes*, kindly refer to https://github.com/Intelligent-Computing-Lab-Yale/SNN-Segmentation.

To sum up, each event representation has its unique characteristics and is suitable for different applications or processing techniques. Our proposed OpenESS framework is capable of leveraging each of the above event representations for efficient and accurate event-based semantic segmentation in an annotation-free and open-vocabulary manner. Such a versatile and flexible way of learning verifies the broader application potential of our proposed framework.

A.5. Evaluation Configuration

Following the convention, we use the Intersection-over-Union (`IoU`) metric to measure the semantic segmentation performance for each semantic class. The IoU score can be calculated via the following equation:

$$\texttt{IoU} = \frac{TP}{TP + FP + FN}~, \tag{4}$$

where TP (True Positive) denotes pixels correctly classified as belonging to the class; FP (False Positive) denotes pixels incorrectly classified as belonging to the class; and FN (False Negative) denotes pixels that belong to the class but are incorrectly classified as something else.

The `IoU` metric measures the overlap between the predicted segmentation and the ground truth for a specific class. It returns a value between 0 (no overlap) and 1 (perfect overlap). It is a way to summarize the `mIoU` values for each class into a single metric that captures the overall performance of the model across all classes, *i.e.*, mean IoU (`mIoU`). The `mIoU` of a given prediction is calculated as:

$$\texttt{mIoU} = \frac{1}{C}\sum_{i=1}^{C}\texttt{IoU}_i~, \tag{5}$$

where C is the number of classes and $\texttt{IoU}_i$ denotes the score of class i. `mIoU` provides a balanced measure since each class contributes equally to the final score, regardless of its size or frequency in the dataset. A higher `mIoU` indicates better semantic segmentation performance. A score of 1 would indicate perfect segmentation for all classes, while a score of 0 would imply an absence of correct predictions. In this work, all the compared approaches adopt the same `mIoU` calculation as in the ESS benchmarks [2, 16]. Additionally, we also report the semantic segmentation accuracy (`Acc`) for the baselines and the proposed framework.

B. Additional Experimental Results

In this section, we provide the class-wise IoU scores for the experiments conducted in the main body of this paper.

B.1. Annotation-Free ESS

The per-class zero-shot event-based semantic segmentation results are shown in Tab. E. For almost every semantic class, we observe that the proposed OpenESS achieves much higher IoU scores than MaskCLIP [19] and FC-CLIP [18]. This validates the effectiveness of OpenESS for conducting efficient and accurate event-based semantic segmentation without using either the event or frame labels.

B.2. Annotation-Efficient ESS

The per-class linear probing event-based semantic segmentation results are shown in the first block of Tab. F and Tab. G. Specifically, compared to the random initialization baseline, a self-supervised pre-trained network always provides better features. The quality of representation learning often determines the linear probing performance. The network pre-trained using our frame-to-event contrastive distillation and text-to-event consistency regularization tends to achieve higher event-based semantic segmentation results than MaskCLIP [19] and FC-CLIP [18]. Notably, such improvements are holistic across almost all eleven semantic classes in the dataset. These results validate the effectiveness of the proposed OpenESS framework in tackling the challenging event-based semantic segmentation task.

The per-class annotation-efficient event-based semantic segmentation results of the `frame2vodel` and `frame2recon` settings under 1%, 5%, 10%, and 20% annotation budgets are shown in Tab. F and Tab. G, respectively. Similar to the findings and conclusions drawn above, we observe clear superiority of the proposed OpenESS framework over the random initialization, MaskCLIP [19], and FC-CLIP [18] approaches. Such consistent performance improvements validate again the effectiveness and superiority of the proposed frame-to-event contrastive distillation and text-to-event consistency regularization. We hope our framework can lay a solid foundation for future

Table F. **The per-class segmentation results** of annotation-efficient event-based semantic segmentation approaches on the test set of *DSEC-Semantic* [16]. All approaches adopted the `frame2voxel` representation. Scores reported are IoUs in percentage (%). For each semantic class under each experimental setting, the best score in each column is highlighted in **bold**.

Method	**mIoU**	background	building	fence	person	pole	road	sidewalk	vegetation	car	wall	traffic-sign	**Acc**
Linear Probing													
Random	6.70	7.85	3.37	0.00	0.00	0.00	38.60	0.00	23.83	0.01	0.00	0.00	37.94
MaskCLIP [19]	33.08	75.04	65.06	4.63	0.00	**6.47**	77.06	17.07	55.89	52.17	0.69	**9.78**	76.39
FC-CLIP [18]	43.00	92.53	72.59	**12.43**	0.02	0.00	88.14	52.84	71.92	64.02	**10.54**	7.95	86.00
OpenESS (Ours)	**44.26**	**93.64**	**75.40**	11.82	**1.16**	0.75	**90.29**	**57.96**	**73.15**	**65.36**	9.69	7.67	**87.55**
Fine-Tuning (1%)													
Random	26.62	81.63	33.13	1.77	0.97	7.58	76.81	17.45	51.05	18.64	0.37	3.40	70.04
MaskCLIP [19]	33.89	87.56	53.24	2.34	0.60	8.92	81.71	25.76	59.37	42.56	2.52	8.24	77.79
FC-CLIP [18]	39.12	91.64	59.78	**8.93**	0.00	7.84	**87.58**	**46.58**	66.87	51.30	**4.74**	5.10	82.12
OpenESS (Ours)	**41.41**	**93.01**	**74.01**	3.21	**10.78**	**14.58**	84.50	34.78	**69.82**	**55.12**	4.47	**11.21**	**84.41**
Fine-Tuning (5%)													
Random	31.22	77.13	50.32	**12.36**	1.26	0.00	86.03	41.22	21.48	50.67	2.96	0.04	71.38
MaskCLIP [19]	37.03	91.09	60.52	4.35	11.90	**11.73**	81.24	23.56	61.77	45.93	2.75	12.45	79.58
FC-CLIP [18]	43.71	92.91	**71.21**	10.84	0.00	5.60	**90.11**	57.54	**71.30**	**61.04**	**11.41**	8.81	**86.38**
OpenESS (Ours)	**44.97**	**93.58**	70.18	8.44	**18.22**	11.01	89.72	**57.76**	67.44	56.06	9.59	**12.70**	85.46
Fine-Tuning (10%)													
Random	33.67	85.79	49.85	6.78	8.00	**15.51**	80.78	25.72	58.18	29.97	0.82	8.93	76.69
MaskCLIP [19]	38.83	92.34	69.96	3.64	5.85	12.98	82.23	23.61	66.39	53.23	3.47	13.46	82.36
FC-CLIP [18]	44.09	93.62	72.86	**10.88**	0.00	8.23	**89.81**	**57.05**	**71.95**	60.64	9.58	10.42	86.66
OpenESS (Ours)	**46.25**	**93.92**	**73.34**	8.13	**18.61**	15.41	89.03	52.56	71.76	**61.71**	**9.99**	**14.26**	**86.72**
Fine-Tuning (20%)													
Random	41.31	91.08	67.90	4.68	17.90	**17.41**	85.11	43.24	66.62	43.95	5.03	11.55	82.99
MaskCLIP [19]	42.40	93.19	72.49	5.52	18.21	16.17	84.29	35.04	69.44	54.47	2.43	15.15	84.09
FC-CLIP [18]	47.77	91.05	70.90	7.04	**21.10**	14.84	**91.13**	**64.28**	71.62	61.73	**13.25**	**18.55**	86.95
OpenESS (Ours)	**48.28**	**94.21**	**74.66**	**10.49**	20.46	16.27	90.15	57.66	**73.71**	**63.95**	11.20	18.29	**87.57**

works in the established annotation-efficient event-based semantic segmentation.

C. Qualitative Assessment

In this section, we provide sufficient qualitative examples to further attest to the effectiveness and superiority of the proposed framework.

C.1. Open-Vocabulary Examples

The key advantage of our proposed OpenESS framework is its capability to leverage open-world vocabularies from the CLIP text embedding space. Unlike prior event-based semantic segmentation, which relies on pre-defined and fixed categories, our open-vocabulary segmentation aims to understand and categorize image regions into a broader, potentially unlimited range of categories. We provide more open-vocabulary examples in Fig. D. As can be observed, given proper text prompts like *"road"*, *"sidewalk"*, and *"building"*, our proposed OpenESS framework is capable of generating semantically meaningful attention maps for depicting the corresponding regions. Such a flexible framework can be further adapted to new or unseen categories without the need for extensive retraining, which is particularly beneficial in dynamic environments where new objects or classes might frequently appear. Additionally, the open-vocabulary segmentation pipeline allows users to work with a more extensive range of objects and concepts, enhancing the user experience and interaction capabilities.

C.2. Visual Comparisons

In this section, we provide more qualitative comparisons of our proposed OpenESS framework over prior works [16, 19] on the DSEC-Semantic dataset. Specifically, the visual comparisons are shown in Fig. E and Fig. F. As can be observed, OpenESS shows superior event-based semantic segmentation performance over prior works across a wide range of event scenes under different lighting and weather conditions. Such consistent segmentation performance improvements provide a solid foundation to validate the ef-

Table G. **The per-class segmentation results** of annotation-efficient event-based semantic segmentation approaches on the test set of *DSEC-Semantic* [16]. All approaches adopted the `frame2recon` representation. Scores reported are IoUs in percentage (%). For each semantic class under each experimental setting, the best score in each column is highlighted in **bold**.

Method	**mIoU**	background	building	fence	person	pole	road	sidewalk	vegetation	car	wall	traffic-sign	**Acc**
Linear Probing													
Random	6.22	7.55	5.48	0.00	0.00	0.00	39.79	0.00	15.64	0.01	0.00	0.00	36.60
MaskCLIP [19]	27.09	59.82	62.14	1.60	0.00	4.54	69.71	5.34	47.85	38.51	0.40	8.12	70.59
FC-CLIP [18]	40.08	**89.22**	**69.08**	**14.62**	**26.90**	0.00	83.14	21.79	**69.56**	57.78	**7.86**	0.92	82.70
OpenESS (Ours)	**44.08**	88.56	61.43	6.05	21.54	**12.36**	**91.43**	**63.04**	64.01	**60.52**	6.18	**9.76**	**84.48**
Fine-Tuning (1%)													
Random	23.95	76.37	29.59	1.73	0.00	5.75	78.12	9.73	48.96	11.56	0.28	1.38	69.20
MaskCLIP [19]	30.73	79.25	47.26	0.13	1.17	5.04	78.78	19.72	56.13	43.74	1.13	5.70	74.25
FC-CLIP [18]	38.99	87.75	61.48	3.47	4.60	8.06	88.96	55.12	64.41	47.16	**3.61**	4.23	82.90
OpenESS (Ours)	**43.17**	**87.85**	**66.15**	**8.82**	**21.52**	**12.41**	**89.36**	**55.35**	**72.45**	**48.76**	3.40	**8.81**	**84.56**
Fine-Tuning (5%)													
Random	30.42	80.25	38.43	5.50	13.45	9.08	83.45	30.88	51.75	19.53	0.16	2.19	73.65
MaskCLIP [19]	36.33	85.80	60.43	2.60	8.70	7.47	83.10	34.04	64.80	39.60	3.07	**10.00**	80.37
FC-CLIP [18]	43.34	88.28	64.90	6.94	**20.96**	**9.58**	**91.18**	**62.35**	68.09	52.39	4.93	7.16	84.93
OpenESS (Ours)	**45.58**	**89.11**	**70.83**	**10.92**	20.21	1.99	91.04	60.76	**72.07**	**67.91**	**12.90**	3.69	**86.93**
Fine-Tuning (10%)													
Random	34.11	81.85	46.28	4.87	11.30	10.20	85.32	43.16	55.34	32.72	1.28	2.90	77.48
MaskCLIP [19]	40.13	87.31	62.54	4.93	5.09	**12.86**	88.30	50.60	64.74	55.21	0.32	9.51	83.52
FC-CLIP [18]	45.35	89.71	69.00	6.64	22.37	8.33	91.20	64.09	69.34	61.73	7.23	9.19	86.29
OpenESS (Ours)	**48.94**	**90.63**	**71.68**	**12.41**	**29.32**	9.42	**92.53**	**66.19**	**73.76**	**69.03**	**10.71**	**12.71**	**87.84**
Fine-Tuning (20%)													
Random	39.25	87.14	61.80	6.77	3.51	13.19	88.53	56.12	61.95	44.65	1.29	6.84	82.51
MaskCLIP [19]	43.37	89.83	69.80	7.07	8.93	10.67	88.88	52.65	70.71	60.03	3.10	15.39	85.69
FC-CLIP [18]	47.18	91.20	71.39	**11.53**	24.92	9.60	91.58	63.88	71.52	63.44	7.55	12.36	87.07
OpenESS (Ours)	**49.74**	**91.28**	**73.43**	10.69	**27.18**	**13.85**	**92.84**	**67.59**	**74.20**	**69.22**	**10.62**	**16.21**	**88.26**

fectiveness and superiority of the proposed frame-to-event contrastive distillation and text-to-event consistency regularization. For additional qualitative comparisons, kindly refer to Appendix C.4.

C.3. Failure Cases

As can be observed from Fig. D, Fig. E, and Fig. F, the existing event-based semantic segmentation approaches still have room for further improvements. Similar to the conventional semantic segmentation task, it is often hard to accurately segment the boundaries between the semantic objects and backgrounds. In the context of event-based semantic segmentation, such a problem tends to be particularly overt. Unlike traditional cameras that capture dense, synchronous frames, event cameras generate sparse, asynchronous events, which brings extra difficulties for accurate boundary segmentation. Meanwhile, the current framework finds it hard to accurately predict the minor classes, such as *fence*, *pole*, *wall*, and *traffic-sign*. We believe these are potential directions that future works can explore to further improve the event-based semantic segmentation performance on top of existing frameworks.

C.4. Video Demos

In addition to the qualitative examples shown in the main body and this supplementary file, we also provide several video clips to further validate the effectiveness and superiority of the proposed approach. Specifically, we provide three video demos in the attachment, named `demo1.mp4`, `demo2.mp4`, and `demo3.mp4`. The first two video demos show open-vocabulary event-based semantic segmentation examples using the class names and open-world vocabularies as the input text prompts, respectively. The third video demo contains qualitative comparisons of the semantic segmentation predictions among our proposed OpenESS and prior works. All the provided video sequences validate again the unique advantage of the proposed open-vocabulary event-based semantic segmentation framework.

Kindly refer to our GitHub repository[1] for additional details on accessing these video demos.

D. Broader Impact

In this section, we elaborate on the positive societal influence and potential limitations of the proposed open-vocabulary event-based semantic segmentation framework.

D.1. Positive Societal influence

Event-based cameras can capture extremely fast motions that traditional cameras might miss, making them ideal for dynamic environments. In robotics, this leads to better object detection and scene understanding, enhancing the capabilities of robots in the manufacturing, healthcare, and service industries. In autonomous driving, event-based semantic segmentation provides high temporal resolution and low latency, which is crucial for detecting sudden changes in the environment. This can lead to faster and more accurate responses, potentially reducing accidents and enhancing road safety. Our proposed OpenESS is designed to reduce the annotation budget and training burden of existing event-based semantic segmentation approaches. We believe such an efficient way of learning helps increase the scalability of event-based semantic segmentation systems and in turn contributes positively to impact society by enhancing safety, efficiency, and performance in various aspects.

D.2. Potential Limitation

Although our proposed framework is capable of conducting annotation-free and open-vocabulary event-based semantic segmentation and achieves promising performance, there tend to exist several potential limitations. Firstly, our current framework requires the existence of synchronized event and RGB cameras, which might not be maintained by some older event camera systems. Secondly, we directly adopt the standard text prompt templates to generate the text embedding, where a more sophisticated design could further improve the open-vocabulary learning ability of the existing framework. Thirdly, there might still be some self-conflict problems in our frame-to-event contrastive distillation and text-to-event consistency regularization. The design of a better representation learning paradigm on the event-based data could further resolve these issues. We believe these are promising directions that future works can explore to further improve the current framework.

E. Public Resources Used

In this section, we acknowledge the use of public resources, during the course of this work.

E.1. Public Datasets Used

We acknowledge the use of the following public datasets, during the course of this work:

• DSEC[2]	CC BY-SA 4.0
• DSEC-Semantic[3]	CC BY-SA 4.0
• DDD17[4]	CC BY-SA 4.0
• DDD17-Seg[5]	Unknown
• E2VID-Driving[6]	GNU General Public License v3.0

E.2. Public Implementations Used

We acknowledge the use of the following public implementations, during the course of this work:

• ESS[7]	GNU General Public License v3.0
• E2VID[8]	GNU General Public License v3.0
• HMNet[9]	BSD 3-Clause License
• EV-SegNet[10]	Unknown
• SNN-Segmentation[11]	Unknown
• CLIP[12]	MIT License
• MaskCLIP[13]	Apache License 2.0
• FC-CLIP[14]	Apache License 2.0
• SLIC-Superpixels[15]	Unknown
• Segment-Anything[16]	Apache License 2.0

[1] https://github.com/ldkong1205/OpenESS

[2] https://dsec.ifi.uzh.ch

[3] https://dsec.ifi.uzh.ch/dsec-semantic

[4] http://sensors.ini.uzh.ch/news_page/DDD17.html

[5] https://github.com/Shathe/Ev-SegNet

[6] https://rpg.ifi.uzh.ch/E2VID.html

[7] https://github.com/uzh-rpg/ess

[8] https://github.com/uzh-rpg/rpg_e2vid

[9] https://github.com/hamarh/HMNet_pth

[10] https://github.com/Shathe/Ev-SegNet

[11] https://github.com/Intelligent-Computing-Lab-Yale/SNN-Segmentation

[12] https://github.com/openai/CLIP

[13] https://github.com/chongzhou96/MaskCLIP

[14] https://github.com/bytedance/fc-clip

[15] https://github.com/PSMM/SLIC-Superpixels

[16] https://github.com/facebookresearch/segment-anything

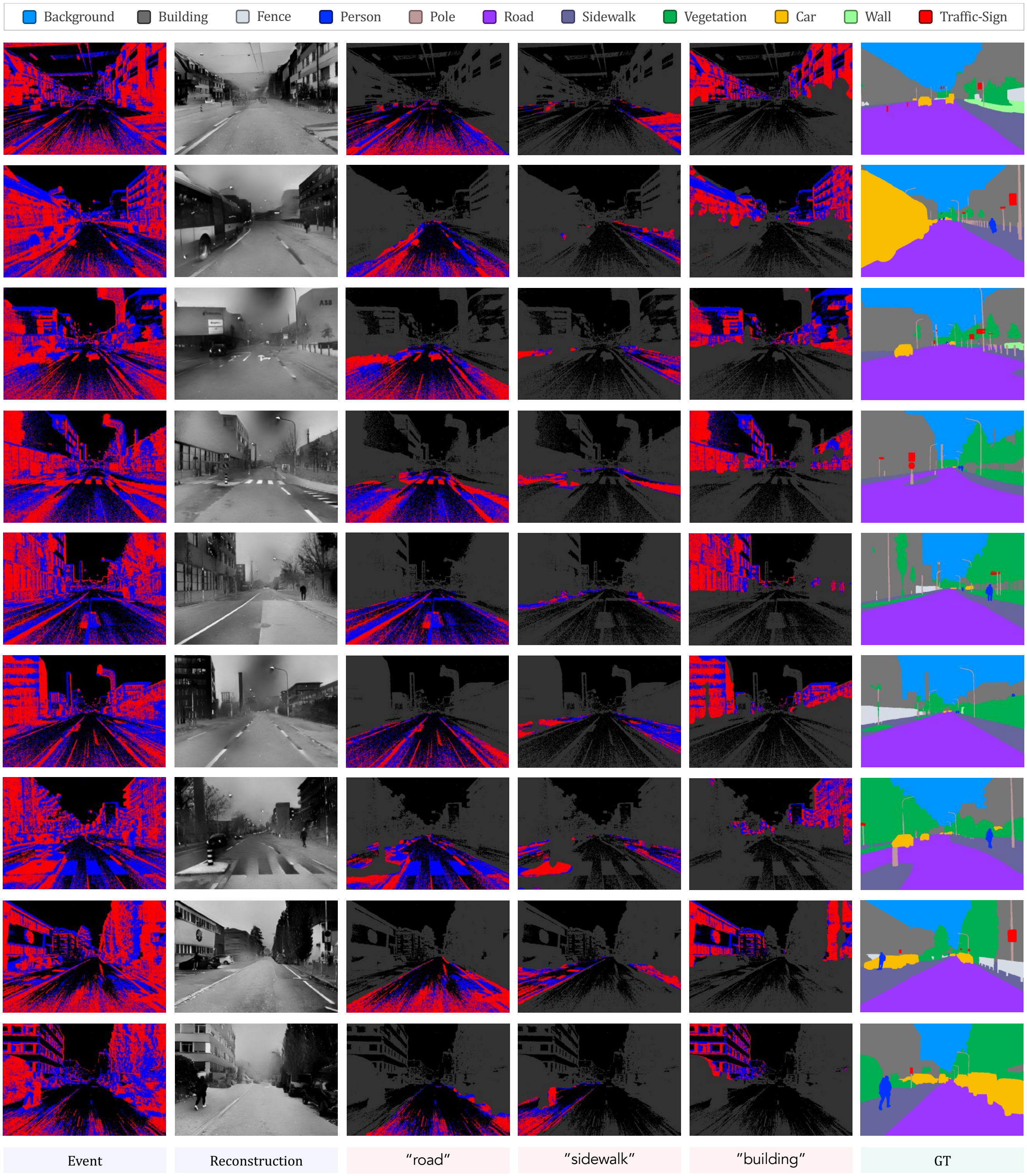


Figure D. **Qualitative examples** of the language-guided attention maps generated by the proposed OpenESS framework. For each sample, the regions with a high similarity score to the text prompts are highlighted. Best viewed in colors and zoomed-in for additional details.

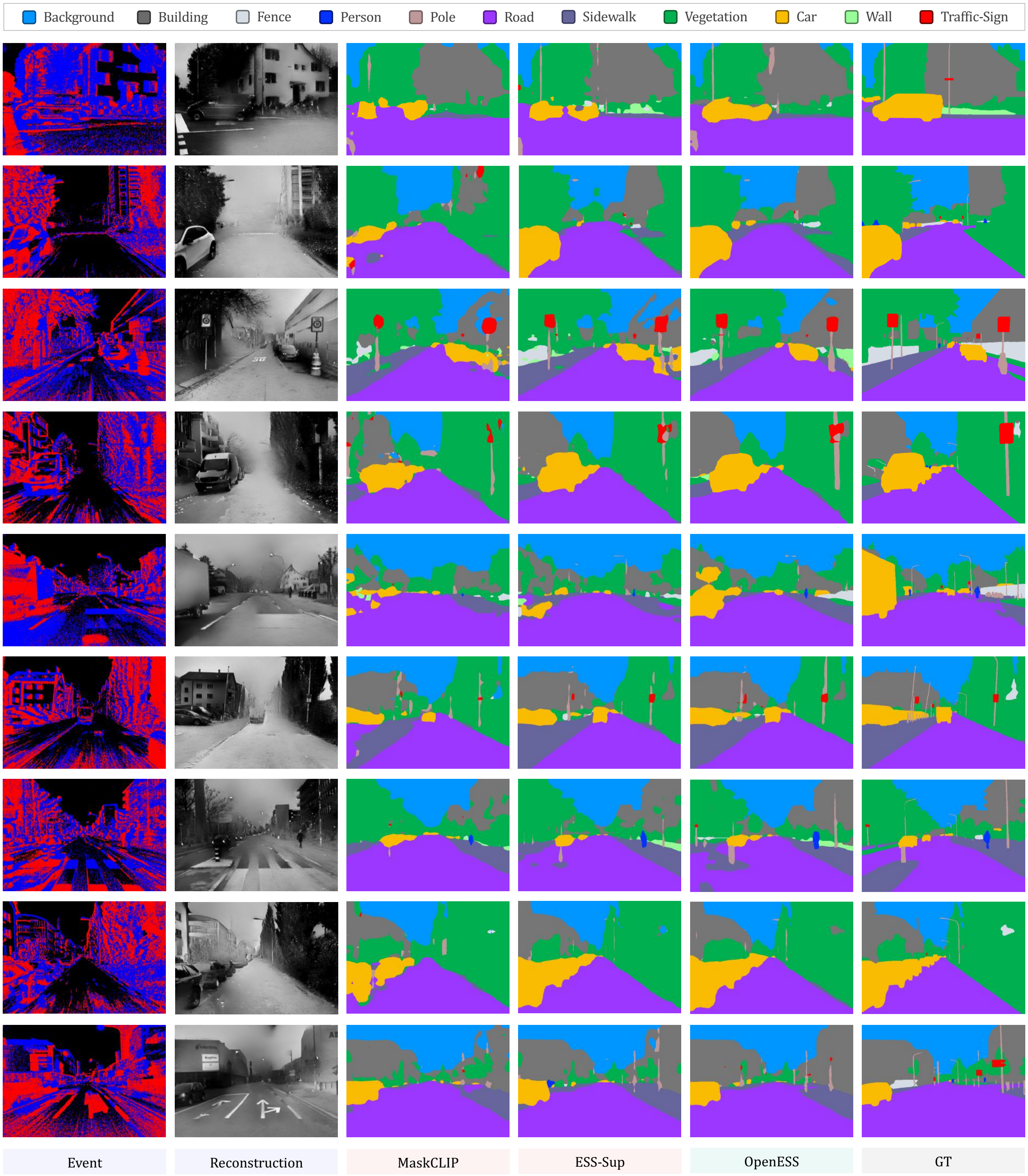


Figure E. **Qualitative comparisons** (1/2) among different ESS approaches on the *test* set of *DSEC-Semantic* [16]. Best viewed in colors.

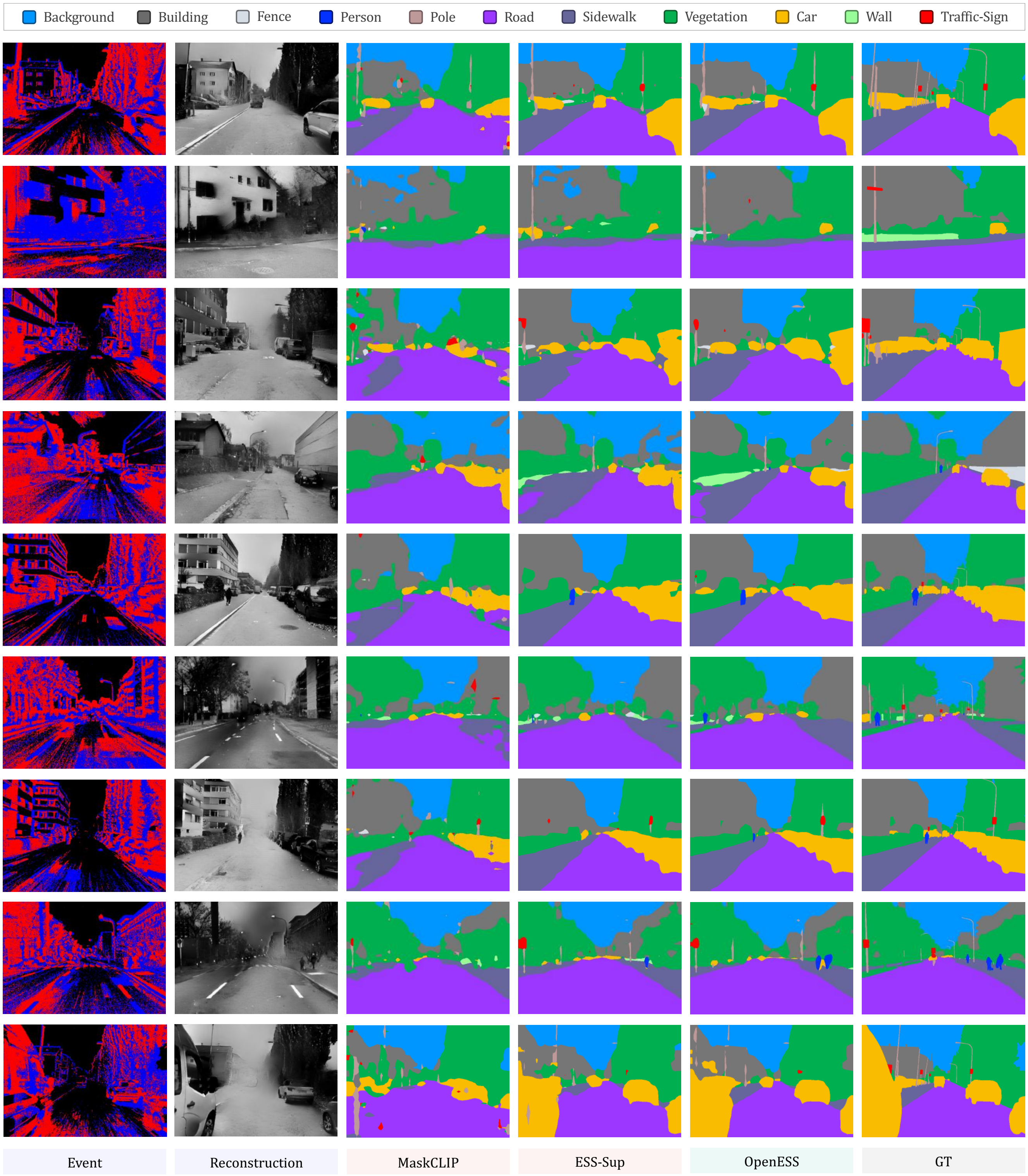


Figure F. **Qualitative comparisons** (2/2) among different ESS approaches on the *test* set of *DSEC-Semantic* [16]. Best viewed in colors.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012. 2, 4, 5, 6

[2] Inigo Alonso and Ana C. Murillo. Ev-segnet: Semantic segmentation for event-based cameras. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–10, 2019. 1, 2, 3, 4, 7

[3] Jonathan Binas, Daniel Neil, Shih-Chii Liu, and Tobi Delbruck. Ddd17: End-to-end davis driving dataset. In *International Conference on Machine Learning Workshops*, pages 1–9, 2017. 1

[4] Runnan Chen, Youquan Liu, Lingdong Kong, Nenglun Chen, Xinge Zhu, Yuexin Ma, Tongliang Liu, and Wenping Wang. Towards label-free scene understanding by vision foundation models. In *Advances in Neural Information Processing Systems*, 2023. 2

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. 1

[6] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 6(3): 4947–4954, 2021. 1

[7] Ryuhei Hamaguchi, Yasutaka Furukawa, Masaki Onishi, and Ken Sakurada. Hierarchical neural memory network for low latency event processing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22867–22876, 2023. 1

[8] Zexi Jia, Kaichao You, Weihua He, Yang Tian, Yongxiang Feng, Yaoyuan Wang, Xu Jia, Yihang Lou, Jingyi Zhang, Guoqi Li, and Ziyang Zhang. Event-based semantic segmentation with posterior attentio. *IEEE Transactions on Image Processing*, 32:1829–1842, 2023. 1

[9] Youngeun Kim, Joshua Chough, and Priyadarshini Panda. Beyond classification: Directly training spiking neural networks for semantic segmentation. *Neuromorphic Computing and Engineering*, 2(4):044015, 2022. 6

[10] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 2, 3, 4, 5

[11] Youquan Liu, Lingdong Kong, Jun Cen, Runnan Chen, Wenwei Zhang, Liang Pan, Kai Chen, and Ziwei Liu. Segment any point cloud sequences by distilling vision foundation models. In *Advances in Neural Information Processing Systems*, 2023. 2

[12] Xidong Peng, Runnan Chen, Feng Qiao, Lingdong Kong, Youquan Liu, Tai Wang, Xinge Zhu, and Yuexin Ma. Learning to adapt sam for segmenting cross-domain point clouds. *arXiv preprint arXiv:2310.08820*, 2023. 2

[13] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763, 2021. 2

[14] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(6):1964–1980, 2019. 5, 6

[15] David Stutz, Alexander Hermans, and Bastian Leibe. Superpixels: An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, 166:1–27, 2018. 2

[16] Zhaoning Sun, Nico Messikommer, Daniel Gehrig, and Davide Scaramuzza. Ess: Learning event-based semantic segmentation from still images. In *European Conference on Computer Vision*, pages 341–357, 2022. 1, 2, 3, 4, 5, 7, 8, 9, 12, 13

[17] Jingyi Xu, Weidong Yang, Lingdong Kong, Youquan Liu, Rui Zhang, Qingyuan Zhou, and Ben Fei. Visual foundation models boost cross-modal unsupervised domain adaptation for 3d semantic segmentation. *arXiv preprint arXiv:2403.10001*, 2024. 2

[18] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. In *Advances in Neural Information Processing Systems*, 2023. 5, 7, 8, 9

[19] Chong Zhou, Chen Change Loy, and Bo Da. Extract free dense labels from clip. In *European Conference on Computer Vision*, pages 696–712, 2022. 5, 7, 8, 9