Towards Robust Multi-Modal Semantic Segmentation with Teacher-Student Framework and Hybrid Prototype Distillation

Supplementary Material

1. Method Details

1.1. Details of the Data Production Pipeline

The stage 1 framework comprises two models: a teacher trained on full multi-modal inputs (*e.g.*, RGB, Depth, Event and LiDAR on DELIVER [5]) and a robust student tailored for missing modalities. Both share the same architecture. For the multi-modal input data $\{x_r, x_d, x_e, x_l, \ldots\}$ where $x_m \in \mathrm{H} \times \mathrm{W} \times 3$, it is processed through a SegFormer-based network [4] to generate feature maps batch $\{f_r^i, f_d^i, f_e^i, f_l^i, \ldots\}$ with $i \in [1, 4]$ means the modality feature in the i^{th} stage. This network architecture treats all modalities equally. Here, each feature has a shape of [bs, dim, H, W], following the Segformer architecture. For example, the first-layer feature of mit-b0 is [bs, 32, 256, 256]. We then split the batch, feeding each feature of shape [32, 256, 256] into the Hybrid Prototype Distillation (HPD) Module. This yields prototype features of size [c, dim], where c is the number of classes (*e.g.*, 25 for the Deliver dataset) and $dim \in [32, 64, 160, 256]$. Finally, we apply random-matching distillation to these prototype features following main text Equation 5 in the main text to compute the loss $\mathcal{L}_{\mathbf{hp}}$.To distinguish the logits outputs of the teacher model and the student model, we also define them as l_t and l_s respectively. We feed the logits into main text Equation 2 to compute $\mathcal{L}_{\mathbf{KL}}$.

The stage 2 framework comprises three models: full-modality nodal, robust modal and the student model. In data processing, we combine the feature maps f_1, f_2 and the logits l_1, l_2 from different teacher models via FSM into a unified feature map f_t and logits l_t while preserving their original dimensions. Then, f_t and l_t guide the student model's learning in the same manner as in Stage 1. As in Fig.5 of the main paper, following the ASM pipeline, we compute the cosine similarity between modality-specific semantic features and the fused features at each teacher-model layer to estimate modality strength. Modalities are ranked by similarity in descending order, and the **lower half**(rounded down) is selected. The corresponding ViT encoders are then added to the parameter set T for updating.

1.2. Details about CPD

The pseudo-code for Cross-modal Prototype Distillation is shown in algorithm 1. First, we extract the intermediate features from each batch. These features are then passed through a prototype extraction module to obtain the corresponding prototype representations. Finally, we compute the distillation loss using a non-negative KL divergence, allowing the teacher model to provide effective guidance to the student model.

Algorithm 1 CPD Loss Computation

Input: Index list $index$, student feature list x_all, teacher feature list x_all_t, label tensor lbl, prototype list $prototype\_all$

Output: Prototype Consistency-based Unsupervised Mutual Distillation loss

Initialize $loss\_pumd \leftarrow 0$
Define $loss\_kl$ as KL divergence loss function
Let $B \leftarrow$ length of x_all
for $i = 0$ to $B - 1$ **do**
  $batch\_label \leftarrow lbl[i]$ with shape expanded to 3D
  for $j = 0$ to 3 **do**
    for $k = 0$ to length of $index - 1$ **do**
      Let $x\_all\_t \leftarrow$ random order of x_all_t
      $x\_all\_feature \leftarrow x\_all[i][j][k]$ with batch dimension added
      $x\_all\_t\_feature \leftarrow x\_all\_t[i][j][k]$ with batch dimension added
      Compute $student\_prototype$
      Compute $teahcer\_t\_prototype$
      Compute $loss\_kl(prototypes)$
    end for
  end for
end for
return $loss\_pumd/B$

1.3. Implementation Logic of Feedback Mechanism

As shown in algorithm 2, in our multi-modal architecture, we maintain a set of N modality-specific Transformer encoders, denoted as $\{E_i\}_{i=1}^{N}$. At each training iteration, all modalities are forwarded through their corresponding encoders, but only a selected subset is allowed to update its parameters. Formally, let $\mathcal{T} \subseteq \{1, \ldots, N\}$ denote the set of encoder indices chosen for training in the current step. Given the input sequence for each modality x_i, the output of encoder E_i is computed as

$$o_i = \begin{cases} E_i(x_i), & \text{if } i \in \mathcal{T}, \\ E_i(x_i) \text{ with gradients disabled}, & \text{if } i \notin \mathcal{T}. \end{cases}$$

For $i \notin \mathcal{T}$, the forward computation is executed under a no-gradient condition, ensuring that no computational graph is

Algorithm 2 Forward Process of Multi-Modal Encoders

Require: Inputs $\{x_i\}_{i=1}^{N}$, training encoder indices train_ids
Ensure: Outputs $\{o_i\}_{i=1}^{N}$

```
Initialize an empty list outputs
for i = 1 to N do
    x ← x_i
    if i ∈ train_ids then
        o_i ← E_i(x)        {encoder participates in training (gradients enabled)}
    else
        Temporarily disable gradient computation
        o_i ← E_i(x)        {inference mode, no gradients}
    end if
    Append o_i to outputs
end for
return outputs
```

constructed and no parameter updates occur for the frozen encoders. Despite this, their outputs are still computed and propagated to downstream modules, allowing all modalities to contribute to the final representation.

This training strategy implements a form of dynamic partial freezing: only the encoders indexed by $\mathcal{T}$ participate in optimization, while the remaining encoders operate in inference mode for that step. Such a mechanism reduces optimization overhead, stabilizes parameter updates across modalities, and serves as a regularization effect during multi-modal learning.

2. Additional Training Details

2.1. Computational Efficiency

In terms of computational efficiency, three components require clarification: (1) the additional computational cost introduced by the self-distillation framework, (2) the efficiency of the proposed HPD module, and (3) the extra overhead brought by the feedback mechanism. The self-distillation framework performs inference with the teacher model at every training iteration, resulting in approximately a 30% increase in computation time compared with standard training. The HPD module is constructed using basic matrix operations and therefore introduces negligible additional overhead. For the feedback mechanism, most parameters are frozen and only the encoder of the disadvantaged modality is updated, leading to roughly a 40% increase in computation cost. Nevertheless, by loading pretrained weights and reducing the number of training epochs from 200 to 120, the overall training time remains comparable to the AnySeg baseline, ensuring a fair and consistent comparison.

As shown in Tab. 1. Our method converges in fewer training epochs than AnySeg, resulting in a shorter overall training time.

Model	Time	Mem.	FLOPs	Model	Time	Mem.	FLOPs
M-SegFormer	12.74h	16.1G	610G	AnySeg	19.24h	22.2G	878G
Ours (HPD)	19.56h	22.7G	884G	Ours (Feedback)	9.36h	16.4G	462G

Table 1. Training efficiency comparison with MiT-B0 on DELIVER (120 epochs, 2×RTX 3090).

2.2. Training Stability

In the Hybrid Prototype Distillation module, the cross-modal distillation component adopts a strategy that randomly matches modality-specific prototypes for cross-modal supervision. Although this approach may appear difficult to optimize, it in fact exhibits strong stability in practice.

As shown in Fig. 1, we visualize the training losses of the Cross-modal Prototype Distillation. The hyperparameters in Eq. 1 and Eq. 8 in the main text are set to $\lambda = 12$ and $\alpha = 100$, respectively, so that the loss values fall within a similar magnitude. The model is trained for 200 epochs. It can be observed that all losses decrease significantly within the first 50 epochs, after which they gradually stabilize and converge. Although the CPD loss exhibits slight fluctuations in later stages, the variation is minor, and the overall training process remains smooth.

Regarding the validation performance (Validation mIoU), the metric increases rapidly during the first 100 epochs with relatively large fluctuations, and then continues to rise steadily with smaller oscillations, eventually approaching convergence.

Poorly designed feedback can introduce noise and lead to suboptimal solutions. As in main text Tab.4, unimodal distillation without parameter freezing degrades performance relative to the HPD baseline. Accordingly, our feedback mechanism aims to enhance the teacher's awareness of vulnerable modality distributions rather than directly improving accuracy. To ensure stable optimization, we adopt a partially frozen training strategy and use pretrained initialization with a 10-epoch warm-up (Sec.4.1 Line 358). Although initialization does not affect convergence, feedback without it is more likely to push both teacher and student toward suboptimal states.

2.3. Anymodal Dropout Settings

In main text Tab. 1(a), we follow the AnySeg setting: all methods adopt **modality direct removal** for Anymodal Dropout during training and evaluation **except CMNeXt**. Due to architectural constraints, CMNeXt does not support modality removal and therefore uses zeroing instead. In other settings, including main text Tabs. 1(b–d) and the ablation studies, we follow the latest protocol of Liao [3] and consistently apply **zeroing**.

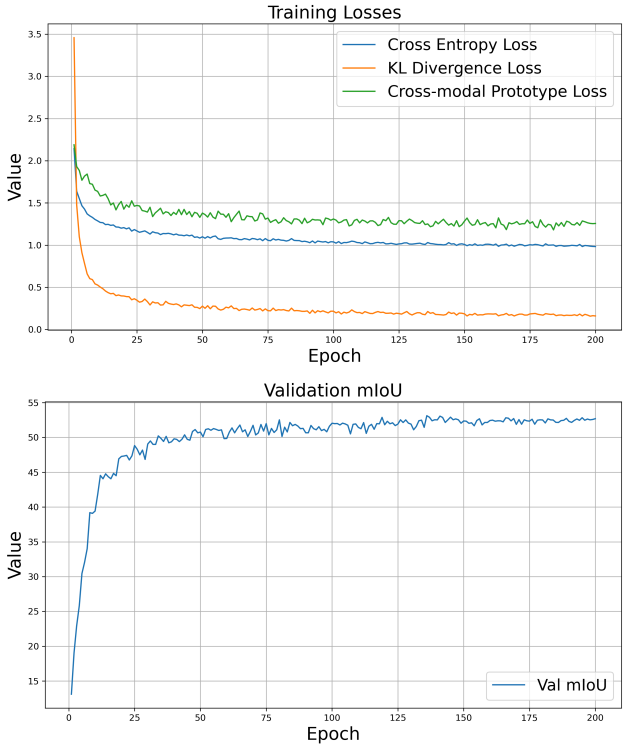

Figure 1. This figure illustrates the training losses and evaluation results obtained after integrating the Cross-modal Prototype Distillation (CPD) module into the training process. The upper part shows the curves of the cross-entropy loss with respect to the labels, the KL divergence loss with respect to the teacher model, and the CPD loss. The lower part presents the validation mIoU curves. The entire teacher–student model is trained at a resolution of 512×512.

2.4. Hyperparameter Complexity

As shown in Fig. 8 of the main paper, model performance remains stable when hyperparameters are set to comparable orders of magnitude. For instance, with fixed λ, varying α around 100 has only a negligible effect (EMM : $49.06 \rightarrow 49.05$). These hyperparameters mainly scale different loss terms to similar magnitudes, ensuring effective contribution from each module, while different choices at comparable orders of magnitude have only a limited impact on overall performance.

3. Supplementary Experiments

3.1. Details About the Metrics

In this work, we employ two types of Metrics. The first is the Entire-Missing Modality Metrics based on AnySeg [7], and the second is the Metrics based on the latest work of Liao *et al*. [3]. Their data processing is similar, with the main difference reflected in the Anymodal dropout method. [7] adopts a method of directly losing the modality, changing $\{x_r, x_d, x_e, x_l\}$ into $\{x_r, x_e\}$, while [3] use a zeroing method $\{x_r, zeros, x_e, zeros\}$.

For the Entire-Missing Modality (EMM) and Random-Missing Modality (RMM) used in the article, we have the following formula definitions:

$$P_p(M_i^{'k}) = p^k \cdot (1-p)^{n-k}, \tag{1}$$

$$\text{EMM} = \text{mIoU}_{\text{EMM}}^{\text{Avg}} = \frac{1}{N}\sum_{i=1}^{N} \text{mIoU}_{M_i'}, \tag{2}$$

$$\text{mIoU}_{\text{EMM}}^{\text{E}}(p) = \sum_{k=0}^{n-1}\sum_{i=1}^{\binom{n}{k}} P_p(M_i^{'k}) \cdot \text{mIoU}_{M_i'}, \tag{3}$$

$$\text{RMM} = \text{mIoU}_{\text{RMM}}^{\text{Avg}} = \frac{1}{N}\sum_{i=1}^{N} \text{mIoU}_{M_i''}, \tag{4}$$

$$\text{mIoU}_{\text{RMM}}^{\text{E}}(p) = \sum_{k=0}^{n-1}\sum_{i=1}^{\binom{n}{k}} P_p(M_i^{''k}) \cdot \text{mIoU}_{M_i''}, \tag{5}$$

here, p is the data missing ratio, N denotes the number of missing modality combinations, Mi' and Mi'' represent the modality combinations after full zeroing/loss and partial zeroing, respectively.

Noisy Modality (NM) is to simulate the real world with noise. It employs Gaussian noise N_G and salt-and-pepper noise N_{SP}. The probability density function of N_G is shown in Eq. 6, which is determined by σ and μ.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \tag{6}$$

With the origin input as X, the noisy input X_N for mIoU_{NM} is defined as Eq 7.

$$X_N = X + N_G(\sigma, \mu) + N_{SP}(D), \tag{7}$$

where, D denotes the noisy density of N_{SP}. These evaluation methods can effectively test the model's robustness under conditions of missing modalities.

3.2. Feedback-only Ablation

The Tab. 2 below reports results for a feedback-only setting without HPD. The baseline is Basic Distillation ($\mathcal{L}_{\text{origin}}$). For comparison, we apply cross-entropy supervision with KL-based distillation on intermediate features while freezing teacher's dominant-modality parameters. The results indicate that the feedback strategy remains effective even without HPD.

Loss Combination	Student		Teacher	
	EMM(Avg)	mIoU	Event + Lidar	mIoU
Basic Distillation	46.42	61.34	1.57	61.92
Basic Distillation with Feedback	47.58(+1.16)	60.80(-0.54)	21.17(+19.60)	60.68(-1.24)

Table 2. Feedback-only experiment without HPD on DELIVER.

3.3. Extreme Scenarios Result

Although our method is designed to enhance robustness under extreme cases of modality absence, it also improves robustness in real-world scenarios. This is intuitively reasonable, as real-life conditions such as nighttime or insufficient exposure can be approximated as forms of modality missingness. To validate this viewpoint, we analyzed the performance of different models under ten extreme scenarios in the DeLiVER dataset. We evaluated the segmentation mIoU performance of different models using MIT-B0 as the backbone under various conditions, and these models have similar numbers of parameters. AnySeg and RobustSeg are both optimized for robustness based on the same baseline model, M-SegFormer, allowing us to compare the effectiveness of different self-distillation frameworks. As shown in Tab. 3, the simple self-distillation framework used by AnySeg leads to an overall performance decline compared with M-SegFormer, indicating that AnySeg only improves robustness under missing-modality conditions. However, it performs poorly in extreme cases where all modalities are present. In contrast, RobustSeg maintains full-modality performance while also demonstrating strong robustness under extreme conditions. For example, in the **sun** scenario, performance remains stable (from 63.43 to 63.66), while in the **night** scenario, as highlighted in red, the model improves from 59.89 to 61.20. RobustSeg also maintains state-of-the-art mean performance and even surpasses CMNeXt, which has more parameters.

For some scenarios, such as lidar-jitter, our method shows performance degradation (62.75 $\rightarrow$ 62.21). We attribute this to the limitation that the noise used to simulate missing modalities cannot fully replicate such complex input degradations. Designing more comprehensive and effective noise simulation strategies will be the focus of our future work.

3.4. More Segmentation Visualization Results

As illustrated in Fig. 2, we deliberately consider two representative categories of challenging scenarios during inference to evaluate the robustness of different multimodal segmentation models in realistic applications: **(1) the R–D–L missing-modality combination**, which simulates partial sensor failure or data corruption, and **(2) night-time low-light conditions with LiDAR jitter**, which reflect extreme physical environments with degraded sensing quality. In the first scenario, existing approaches such as CMNeXt, M-SegFormer, and AnySeg exhibit significant performance degradation due to feature distribution shifts caused by missing modalities, leading to erroneous classifications, incomplete scene structures, and the disappearance of distant or small objects. In contrast, our method preserves coherent scene geometry and yields more accurate semantic boundaries even when only a subset of modalities is available, demonstrating the effectiveness of our modality-robust fusion strategy in mitigating sensor-missing issues.

In the second scenario, characterized by the combination of low-light RGB degradation and unstable LiDAR returns, prior methods are highly sensitive to illumination loss and signal jitter, resulting in blurred boundaries, missing foreground objects, and large misclassified regions. Our approach, however, maintains stable segmentation quality across key semantic regions such as roads, buildings, and foreground targets, producing results that remain closely aligned with the ground truth. These observations collectively indicate that our method exhibits superior robustness and generalization under extreme physical conditions.

4. Detailed T-SNE Visualization

For the main text Figure 5, which presents the t-SNE visualization of the features, we provide a more detailed version in Fig. 3 to help illustrate the process of feature evolution. We visualize the intermediate features and the prototype distributions of the depth and RGB modalities to analyze how CPD influences cross-modal interaction, as shown in Fig. 3. In the first row, the class features become increasingly separable as training progresses, while the mixed regions (purple box) shrink, indicating that CPD improves the discriminability of the learned representations.

In the second and third rows, depth prototypes mainly capture large-scale scene structures, whereas RGB prototypes focus more on fine-grained details [1, 2, 6]. During the first 80 steps, depth features gradually group the 25 categories into two broader clusters—foreground details and coarse background structures (red box). A similar trend appears in the RGB modality, and the expansion of the blue-box regions correlates with their closeness to depth information, showing that depth characteristics are effectively transferred to RGB.

From steps 80 to 120, the inter-class distances of the depth modality begin to increase. This indicates that depth features learn finer details from the RGB modality, leading to more precise structural representation and ultimately enhancing model robustness.

Model (#Params(M))	cloud	fog	night	rain	sun	motion-blur	over-exposure	under-exposure	lidar-jitter	event-lowers	Mean
M-SegFormer (6.10)	64.10	62.06	59.89	60.23	63.43	59.36	61.59	58.38	62.75	60.87	61.27
CMNeXt (10.31)	61.61	60.97	57.65	59.91	61.51	58.26	60.85	54.78	60.72	60.28	59.65
AnySeg (6.10)	61.61	59.18	57.75	56.93	61.23	58.15	58.21	55.95	59.43	57.42	58.59
MAGIC (3.72)	62.77	59.70	59.47	57.50	61.36	59.22	59.72	56.76	60.28	57.83	59.46
RobustSeg (6.10)	64.12	61.59	61.20	59.86	63.66	60.25	61.83	58.98	62.21	60.16	61.39

Table 3. mIoU results of semantic segmentation across ten different degradation cases.

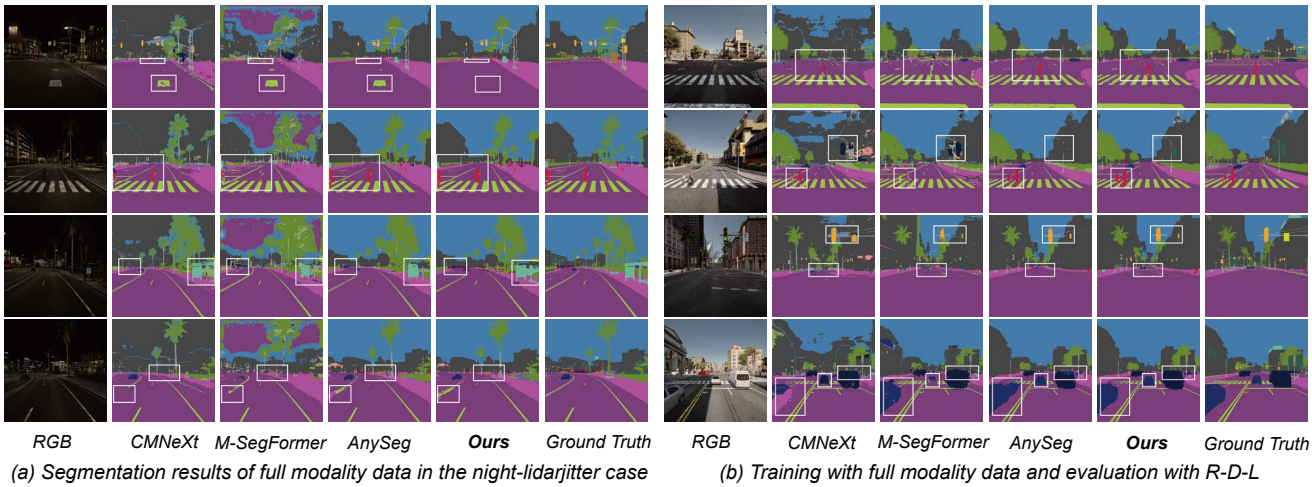

Figure 2. Segmentation results of the model trained with our framework under full-modality and Event missing conditions.

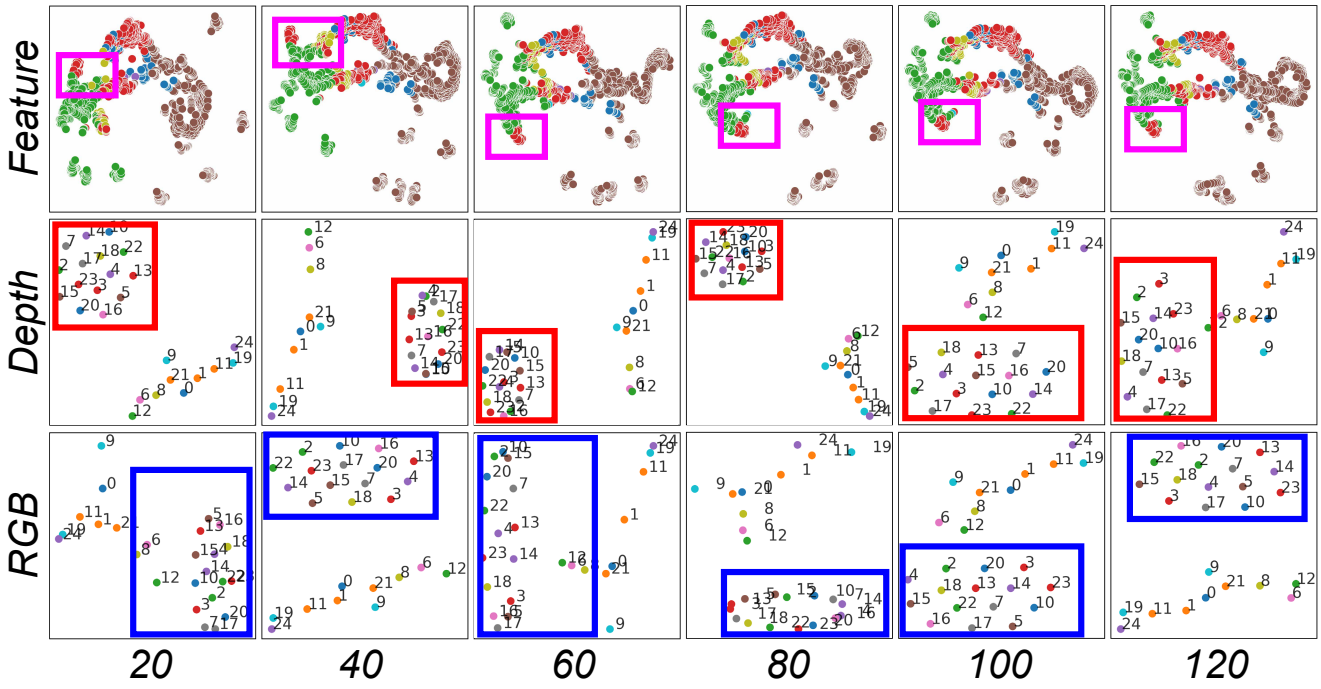

Figure 3. This chart illustrates the visualization of model features across different numbers of epochs. The horizontal axis indicates the number of epochs, while the vertical axis (top to bottom) shows t-SNE visualizations of mixed features, Depth prototypes, and RGB prototypes.

References

[1] Jiaxin Cai, Jingze Su, Qi Li, Wenjie Yang, Shu Wang, Tiesong Zhao, Shengfeng He, and Wenxi Liu. Keep the balance: A parameter-efficient symmetrical framework for rgb+ x semantic segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10587–10598, 2025. 4

[2] Zhiwei Hao, Zhongyu Xiao, Yong Luo, Jianyuan Guo, Jing Wang, Li Shen, and Han Hu. Primkd: Primary modality guided multimodal fusion for rgb-d semantic segmentation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 1943–1951, 2024. 4

[3] Chenfei Liao, Kaiyu Lei, Xu Zheng, Junha Moon, Zhixiong Wang, Yixuan Wang, Danda Pani Paudel, Luc Van Gool, and Xuming Hu. Benchmarking multi-modal semantic segmentation under sensor failures: Missing and noisy modality robustness. *arXiv preprint arXiv:2503.18445*, 2025. 2, 3

[4] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021. 1

[5] Jiaming Zhang, Ruiping Liu, Hao Shi, Kailun Yang, Simon Reiß, Kunyu Peng, Haodong Fu, Kaiwei Wang, and Rainer Stiefelhagen. Delivering arbitrary-modal semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1136–1147, 2023. 1

[6] Xu Zheng, Yuanhuiyi Lyu, Jiazhou Zhou, and Lin Wang. Centering the value of every modality: Towards efficient and resilient modality-agnostic semantic segmentation. In *European Conference on Computer Vision*, pages 192–212. Springer, 2024. 4

[7] Xu Zheng, Haiwei Xue, Jialei Chen, Yibo Yan, Lutao Jiang, Yuanhuiyi Lyu, Kailun Yang, Linfeng Zhang, and Xuming Hu. Learning robust anymodal segmentor with unimodal and cross-modal distillation. *arXiv preprint arXiv:2411.17141*, 2024. 3