上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

ICLR
International Conference On Learning Representations

OVID: Open-Vocabulary Intrusion Detection

Fujun Han[1,2], Jingqi Ye[1,3], Chenglong Zhang[2,3], Peng Ye[1,4]*

[1] Shanghai Artificial Intelligence Laboratory
[2] School of Data Science, The Chinese University of Hong Kong, Shenzhen
[3] University of Science and Technology of China
[4] The Chinese University of Hong Kong
hanfujun@cuhk.edu.cn

Introduction

Various vision intrusion detection models have achieved great success in many scenarios, e.g., autonomous driving, intelligent monitoring and security. However, their reliance on pre-defined classes limits their applicability in open-world intrusion detection scenarios. To remedy these, we introduce the Open-Vocabulary Intrusion Detection (OVID) project for the first time.

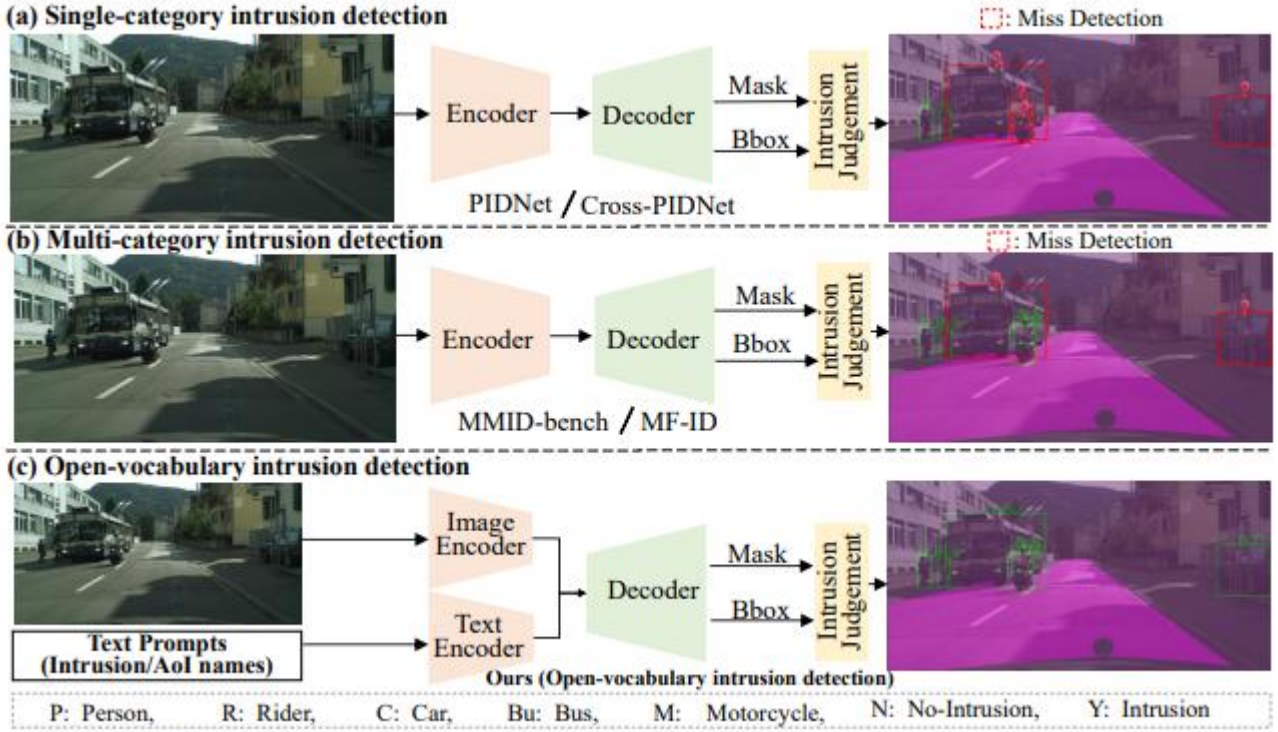
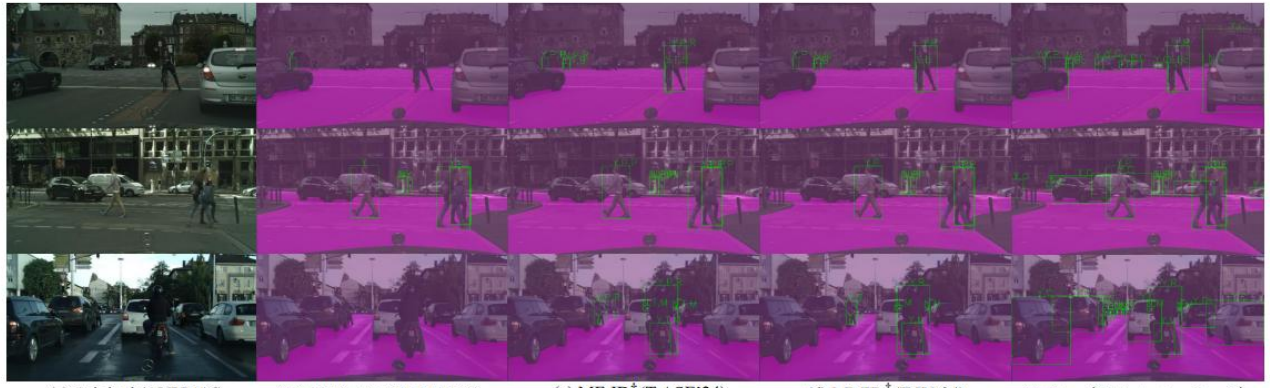


Figure 1: **Workflow comparisons** of different intrusion detection methods. Here, (a), (b), and (c) denote the Single-category, Multi-category, and proposed Open-vocabulary intrusion detection paradigms, respectively. '?' denotes the missed detection (False Negative). We can find that previous works can only detect the pre-defined intrusion category; our framework can detect more categories correctly, which demonstrates the validity of our paradigm.

Cityintrusion-OpenV

Systematic Datasets



(a) Original (CVPR'16) # Intrusion Categories: **None** | (b) PIDNet (ACM MM'20) # Intrusion Categories: **1** | (c) MF-ID†(T-ASE'24) # Intrusion Categories: **4** | (d) MMID‡(T-IV'24) # Intrusion Categories: **4** | (e) Ours (Cityintrusion-OpenV) # Intrusion Categories: **8**

Figure 2: The visualization comparison between our datasets and other promising intrusion detection datasets. † and ‡ denote fine-grained and multiple domains, respectively.

Table 1: The comparison between previous intrusion detection datasets and our datasets. † denotes multiple domains.

Intrusion Detection Dataset Names	Categories	'Y'/'N' Cases	Cases per image
Cityintrusion (Sun et al., 2020; Shi et al., 2022)	1	4599/15084	7.3
Cityintrusion-Multicategory (Han et al., 2024b)	4	5431/22683	9.59
Multi-Domain Multi-Category (Han et al., 2024c;a)	4	5431/22683†	9.59
Ours (Cityintrusion-OpenV)	8	24750/37899	**18.03**

- Our datasets have 8 intrusion categories.
- 18.03 cases per image in the whole dataset/ about 2× up compared to others.

OVIDNet Framework

Proposed Framework and Methods

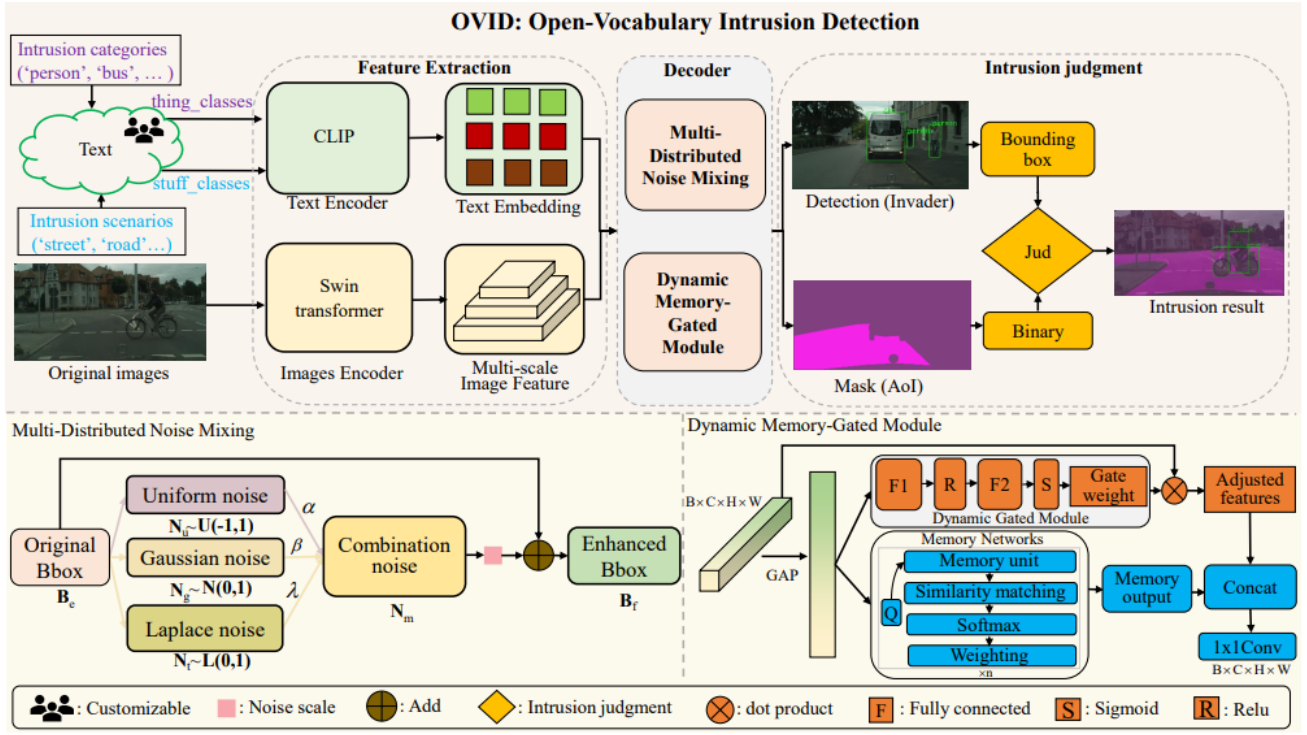


Figure 3: The overall framework and pipeline of our proposed OVIDNet. The input of OVIDNet consists of two different modalities: Text and Images. The text includes some customizable and common intrusion categories and scenarios. The image denotes the corresponding original images. Then, the text and images are sent to different encoders to extract features, *i.e.*, clip and tiny-swin-transformer, respectively. These features will be sent to the decoder for decoding and prediction. In the encoder, we design a multi-distributed noise mixing strategy and a dynamic memory-gated module to enhance generalization in open scenarios. Finally, we extract the predicted bounding box and predicted AoI mask to calculate the overlapping pixels and give the final intrusion results. Once the overlapping pixels are greater than the threshold (t), it will be judged as an intrusion. Otherwise, it will be judged as no-intrusion. We use abbreviations to represent the full name to better express the intrusion results. The detailed correspondence can be found in **Appendix A.4**.

Multi-Distributed Noise Mixing Strategy

Multi-Distributed Noise Mixing Strategy

4.3 Multi-Distributed Noise Mixing Strategy

In the original OpenSeeD model, noise generation methods usually use a uniform noise distribution and a fixed percentage of noise dynamics. Therefore, we can express it as

$$\mathbf{B}_f = \mathcal{C}\left\{\mathbf{B}_e + \mathbf{N}_r \odot \Delta \odot \Upsilon,\ \mathbf{0},\ \mathbf{1}\right\}, \tag{2}$$

where $\mathbf{B}_e$ denotes the set of the bounding box, $\mathbf{B}_e$ can be expressed by center point ($\mathbf{x}$, $\mathbf{y}$) and width, height ($\mathbf{w}$, $\mathbf{h}$). Δ denotes the range of the disturbance and $\Delta = \left\{\frac{\mathbf{w}}{2}, \frac{\mathbf{h}}{2}, \mathbf{w}, \mathbf{h}\right\}$. Υ is a constant noise scaling factor. $\mathbf{N}_r$ denotes the random distribution and $\mathbf{N}_r \sim \mathcal{U}(-1, 1)$. $\odot$ denotes the element-wise product. $\mathbf{C}$ denotes that all value is clamped between $\mathbf{0}$ and $\mathbf{1}$. However, in the real world, this method can not adapt to dynamic environments and scenarios, *e.g.*, different sizes and changing objects, and challenging intrusion scenarios. Therefore, to address this issue, we propose a new **M**ulti-**D**istributed **N**oise **M**ixing Strategy, as shown in Equation 3. The idea of the proposed Multi-Distributed Noise Mixing Strategy is very simple yet effective. When confronted with complex dynamic environments, models need to cope with the variations of different targets and scenarios. Specifically, for tiny objects, fine-grained perturbations are used to preserve their detailed features. Meanwhile, large-scale perturbations to strengthen the global features for large objects.

$$\mathbf{B}_f = \mathcal{C}\left\{\mathbf{B}_e + (\alpha \cdot \mathbf{N}_u + \beta \cdot \mathbf{N}_g + \gamma \cdot \mathbf{N}_t) \odot \Delta \odot \Theta,\ \mathbf{0},\ \mathbf{1}\right\}, \tag{3}$$

where $\mathbf{N}_u \sim \mathcal{U}(-1, 1)$, $\mathbf{N}_g \sim \mathcal{N}(0, 1)$ and $\mathbf{N}_t \sim \mathcal{L}(0, 1)$. $\mathcal{L}$ denotes the Laplace distribution. α, β and γ is the coefficient of $\mathcal{U}$, $\mathcal{N}$, and $\mathcal{L}$ distributions, respectively. Note that $\alpha + \beta + \gamma = 1$. Θ denotes the proposed dynamic varying noise ratio based on the detection area of the bounding box, and $\Theta = \tau \cdot (1 + \log(1 + \mathbf{A}))$. $\mathbf{A}$ denotes the area of the bounding box and $\mathbf{a} = \mathbf{w} \cdot \mathbf{h}$. Besides, $\mathbf{C}$ and Δ are defined as the same as the Equation 2. Our detailed algorithm is shown in Algorithm 1, and the mechanism proof of the Multi-Distributed Noise Mixing strategy is shown in **Appendix B**.

Algorithm 1 **M**ulti-**D**istributed **N**oise **M**ixing Strategy

Require: Bounding box parameters $\mathbf{B}_e$, Noise scale τ; Uniform noise weight α, Gaussian noise weight β, Laplace noise weight γ
Ensure: Augmented bounding box parameters $\mathbf{B}_f$.

▷ $\mathbf{D}$ is a tensor of the same shape as $\mathbf{B}_e$
Initialize $\mathbf{D} \leftarrow \mathbf{0}$
▷ Compute the area of each bounding box
$\mathbf{A} \leftarrow \mathbf{B}_e[:, 2] \cdot \mathbf{B}_e[:, 3]$.
▷ Compute dynamic noise scale for each box
$\mathbf{\Theta} \leftarrow \tau \cdot (1 + \log(1 + \mathbf{A}))$
▷Define perturbation directions for center and size
$\mathbf{\Delta}[:, :2] \leftarrow \mathbf{B}_e[:, 2:]/2$ # Perturb center
$\mathbf{\Delta}[:, 2:] \leftarrow \mathbf{B}_e[:, 2:]$ # Perturb width and height
▷ Generate noise from multiple distributions
$\mathbf{N}_u \sim \mathcal{U}(-1, 1)$, $\mathbf{N}_g \sim \mathcal{N}(0, 1)$ and $\mathbf{N}_t \sim \mathcal{L}(0, 1)$
▷ Compute the weighted combination of noise
$\mathbf{N}_\mathrm{m} \leftarrow \alpha \cdot \mathbf{N}_u + \beta \cdot \mathbf{N}_g + \gamma \cdot \mathbf{N}_l$
▷ Add scaled noise to bounding box parameters
$\mathbf{B}_f \leftarrow \mathbf{B}_e + (\mathbf{N}_\mathrm{m} \odot \mathbf{\Delta} \odot \mathbf{\Theta})$
▷ Clamp augmented bounding boxes to the valid range
$\mathbf{B}_f \leftarrow \mathrm{Clamp}(\mathbf{B}_f, \mathbf{0}, \mathbf{1})$
return $\mathbf{B}_f$

Dynamic Memory-Gated Module

4.4 Dynamic Memory-Gated Module

To address the challenges of insufficient long-term dependency modeling and poor dynamic scene adaptation in our OVID task, we propose a Dynamic Memory-Gated Module. Given a input feature $\mathbf{X} \in \mathbb{R}^{\mathcal{B}\times\mathcal{C}\times\mathcal{H}\times\mathcal{W}}$, we first use global average pooling (GAP) to extract a global context query vector ($\mathbf{Q} \in \mathbb{R}^{\mathcal{B}\times\mathcal{C}}$), express it as $\mathbf{Q} = \mathrm{GAP}(\mathbf{X})$, where $\mathbf{B}$, $\mathbf{C}$, $\mathbf{H}$ and $\mathbf{W}$ denotes the batch_size, channel, height and width. Then, we introduce a dynamic memory retrieval module and express it as

$$\mathbf{O}_m = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{M}_K^{\mathbf{T}}}{\sqrt{d}}\right)\mathbf{M}_V, \tag{4}$$

where $\mathbf{Q} \in \mathbb{R}^{\mathcal{B}\times\mathcal{C}}$ denotes query vector. $\mathbf{M}_K \in \mathbb{R}^{\mathcal{B}\times\mathcal{C}}$ denotes the memory key, and $\mathbf{M}$ is the number of memory units. $\mathbf{M}_V \in \mathbb{R}^{\mathcal{M}\times\mathcal{C}}$ denotes the memory value, and $\mathbf{M}_V$ is used to store the feature information corresponding to the key. $\mathbf{O}_m \in \mathbb{R}^{\mathcal{B}\times\mathcal{C}}$ denotes the memory output by retrieving. Finally, retrieved memory output ($\mathbf{O}_m$) and input features ($\mathbf{X}$) are fused by concatenation and 1×1 Conv. Therefore, we can express this principle as

$$\mathbf{X}_f = \mathrm{Conv1x1}(\mathrm{Concat}(\mathbf{X} \odot \mathbf{W}, \mathbf{O}_m)), \tag{5}$$

where $\mathbf{X}_f$ denotes the fusion feature. $\mathbf{W}$ denotes the generate dynamic weights and $\mathbf{W} = \sigma(\mathbf{W}_2\mathrm{ReLU}(\mathbf{W}_1\mathbf{Q}))$. $\mathbf{W}_1, \mathbf{W}_2$ denotes the weight of fully connected networks. $\mathbf{W}_1 \in \mathbb{R}^{\mathcal{C}\times d}$, $\mathbf{W}_2 \in \mathbb{R}^{d\times\mathcal{C}}$. σ denotes the sigmoid function.

Experiment and Results

Experiment settings

Implementation Details. We conduct all experiments on a computer with 8 NVIDIA GeForce RTX 2080Ti GPUs. Unless specified, the Max_Iter, Batch_size_total, CHECKPOINT_PERIOD, EVAL_PERIOD of all experiments are set to 15000, 8, 15000, and 15000, respectively. The image encoder and text encoder adopt tiny-swin-transformer (Liu et al., 2021) and Clip (Radford et al., 2021), respectively. More hyperparameter details can be found in **Appendix C**.

Datasets. Our experiments are conducted in some publiy datasets, *e.g.*, COCO (Lin et al., 2014), Citytscape (Cordts et al., 2016a), Foggy-Cityscape (Sakaridis et al., 2018) and Cityintrusion-OpenV. In addition, to provide more visualization results, we also test and report visualization demo results on other datasets, *e.g.*, the ShanghaiTech Campus dataset (Luo et al., 2017), and the UA-DETRAC (Wen et al., 2020). Note that in our experiment, we adopt two manners, *i.e.*, zero-shot and task-specific transfer, to evaluate the performance of the model.

Metrics. In order to report the quantitative results of our experiments more comprehensively, inspired by some previous promising work (Han et al., 2024c;b), the mIOU(%) and mAP(%) are utilized to evaluate the sub-task performance of segmentation and object detection. For the intrusion detection performance, we also use three different intrusion detection metrics: AccY(%), AccN(%), and Acc(%) to quantify. Besides, some additional metrics, *e.g.*, panoptic segmentation metrics, PQ(%), SQ(%), RQ(%), and every AP(%), AP@.5(%) of intrusion categories, are reported to evaluate the zero-shot performance of the model.

Comparison Models. We compare with the OpenSeeD (Zhang et al., 2023) because of its promising multi-task capability and performance in open-vocabulary tasks. The multi-task feature is consistent with our task. Besides, we also compare the latest intrusion works, *e.g.*, PIDNet (Sun et al., 2020), Cross-PIDNet (Shi et al., 2022), MMID-bench (Han et al., 2024c), MF-ID (Han et al., 2024b).

Experiment and Results

Main Results

Table 2: The zero-shot and task-specific transfer comparison results between promising multi-task open-vocabulary work and OVIDNet in different datasets. More results are shown in **Appendix D.**

-	Zero-shot Detection (Panoptic segmentation)				Task-specific Transfer (Intrusion detection)			
Model	Test data 1	RQ(%)	SQ(%)	PQ(%)	Test data 2	AccY(%)	AccN(%)	Acc(%)
OpenSeeD	Cityscape	18.22	43.68	14.03	Ours (Normal)	18.72	36.19	29.36
	Foggy-Cityscape	18.07	36.71	14.28	Ours (Foggy)	22.04	25.88	24.38
OVIDNet (Ours)	Cityscape	20.36	36.17	**16.22**	Ours (Normal)	24.43	38.16	**32.79**
	Foggy-Cityscape	19.05	33.71	**15.40**	Ours (Foggy)	27.72	27.90	**27.83**

Compared with promising open-vocabulary works. We first compare the multiple performances with the promising OpenSeeD model and report three zero-shot detection performances, *i.e.*, PQ(%), SQ(%), RQ(%), and three task-specific transfer intrusion performances, *i.e.*, AccY(%), AccN(%), Acc(%), as shown in Table 2. We can find that in different tasks, for the panoptic segmentation performance (PQ), compared with OpenSeeD, our methods can improve it by 2.19% and 1.12%, respectively. Besides, for intrusion detection performance (Acc), our model can surpass it by 3.43% and 3.45%, respectively, which verifies the effectiveness of the proposed model and strategies.

Experiment and Results

Main Results

Table 3: The comparison between our work and promising intrusion detection works. 'close' and 'open' denote the different detection structures. 'ZSD' denotes Zero-shot detection. ✓ and ✗ denote the intrusion category as assessable or not assessable, respectively. † denotes that the backbone is BNet.

Method	Venue	Structure	ZSD	P(%)	R(%)	M(%)	Bc(%)	Tk(%)	Bu(%)	Tn(%)	C(%)
PIDNet (Sun et al., 2020)	ACM MM'20	close	✗	67.1	✗	✗	✗	✗	✗	✗	✗
			✗	63.3†	✗	✗	✗	✗	✗	✗	✗
Cross-PIDNet (Shi et al., 2022)	T-IV'21	close	✗	74.7	✗	✗	✗	✗	✗	✗	✗
			✗	72.1†	✗	✗	✗	✗	✗	✗	✗
MF-ID (Han et al., 2024b)	T-ASE'24	close	✗	45.8	39.8	34.5	38.2	✗	✗	✗	✗
MMID-bench (Han et al., 2024c)	T-IV'24	close	✗	37.4	34.6	20.7	33.1	✗	✗	✗	✗
OVIDNet (Ours)	-	open	✓	✓	✓	✓	✓	✓	✓	✓	✓

Compared with promising intrusion detection works. We also compare some intrusion detection works, as shown in Table 3. Note that the detailed results of our model can be seen in Tabel 14 of **Appendix E.1**. We can see that, compared with previous intrusion works, our model not only has an open structure but also detects more intrusion categories. More importantly, our model has strong generalization capability and achieves zero-shot detection, which is not only pre-trained/pre-undefined categories. In addition, we can observe that as the task difficulty increases, *i.e.*, common intrusion detection task (PIDNet, Cross-PIDNet, MF-ID)→domain adaptation intrusion detection task (MMID-bench)→Open-vocabulary intrusion detection task (OVID), the performance of each category continuously decreases. The main reason is that, as the difficulty of different intrusion detection tasks increases, the requirements of different intrusion detection frameworks are also raised in the open world, especially their generalization and zero-shot capabilities.

Experiment and Results

Main Results

Zero-shot and Task-specific transfer evaluation results on proposed strategies. We then test the zero-shot/task-specific transfer performance of the proposed strategies. Specifically, we train our model on the COCO dataset and validate it on the Cityscape datasets to obtain the segmentation and detection performance in a zero-shot manner. Besides, we also test the intrusion detection performance on the proposed Cityintrusion-OpenV datasets by a task-specific transfer manner, as shown in Table 4. **B** denotes the baseline. We can observe that as different strategies are added, multiple performances are improved, not only intrusion detection but also zero-shot performance, *e.g.*, PQ(%) and mIOU(%). Compared with the baseline, the intrusion performance (Acc) can surpass it by 3.43%. In addition, the zero-shot performance can surpass it by 2.19% (PQ) and 1.03% (mIOU), respectively. More detailed results can be found in **Appendix E.1**.

Table 4: Zero-shot and Task-specific transfer quantitative results of the proposed different strategies.

B	DMG	MDNM	PQ(%)	mIOU(%)	mAP@.5(%)	AccY(%)	AccN(%)	Acc(%)
✓	✗	✗	14.03	28.34	27.58	18.72	36.19	29.36
✓	✓	✗	15.80	28.78	29.16	20.06	37.56	30.72
✓	✗	✓	15.33	29.40	28.56	21.01	38.64	31.75
✓	✓	✓	16.22	29.37	28.98	24.43	38.16	**32.79**

Experiment and Results

Main Results

Generalization Verification in cross-domain tasks. We further test the performance of our OVIDNet framework and strategies in cross-adverse weather tasks, *e.g.*, Normal→Foggy, to verify generalization capabilities. Note that all performance results are given by the pre-trained (in Normal weather) and inference (in adverse weather) manners, as shown in Figure 4. We can find that our OVIDNet is effective even in adverse weather and exhibits promising intrusion performance. Under three different foggy coefficient setting, *i.e.*, $\alpha = 0.005$, $\alpha = 0.01$, $\alpha = 0.02$, our OVIDNet can surpass the baseline model by 2.96%, 3.22%, and 3.45%, respectively. Besides, our strategies can also improve the zero-shot performance under cross-domain tasks, *e.g.*, compared with the baseline of three different foggy coefficient settings, the PQ(%) in the Normal→Foggy tasks can surpass them by 1.29%, 1.21%, and 1.12%, respectively. More details results can be found in **Appendix E.2**.

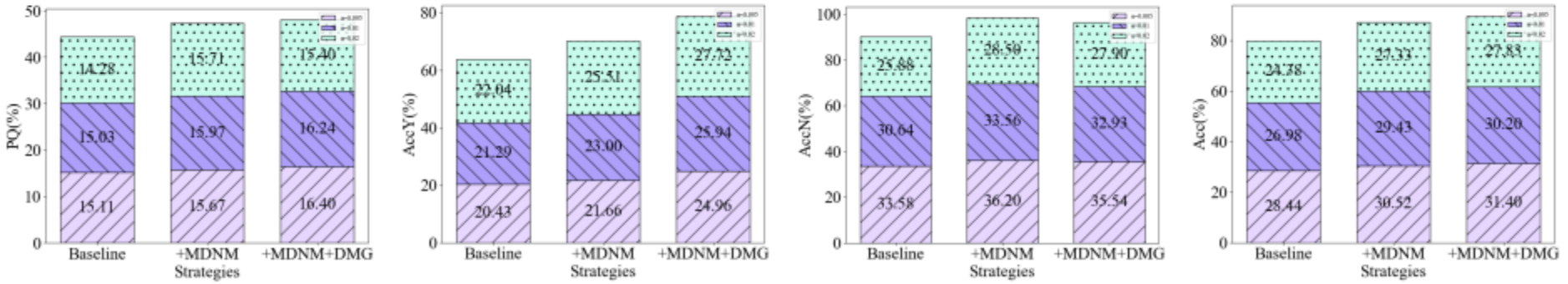
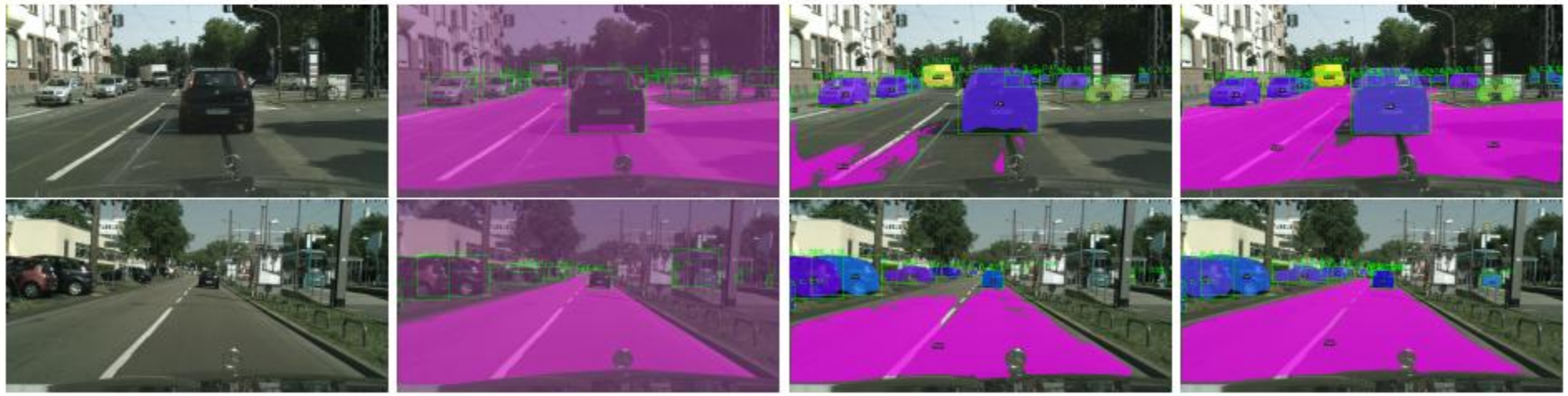


Figure 4: Generalization Verification in cross-domain tasks.

Experiment and Results

Main Results

Visualization Comparisons. We also present some visualization comparison results to verify the zero-shot performance and effectiveness of the proposed framework and methods, as shown in Figure 5. We can find that our framework can present promising visualization detection results, not only detecting intrusion behaviors correctly but also giving correct Intrusion ('Y')/No-intrusion ('N') labels, which proves the effectiveness of our framework and approach. Note that our OVIDNet can improve the zero-shot segmentation performance of AoI; in this case, the AoI is the road. More visualization comparison results are presented in **Appendix E.3**.



(a) Original Images (b) Ground Truth (c) Baseline (d) Ours

Figure 5: The visualization comparison results.

Ablation Experiments

Ablation Experiment Results

Multi-Distributed Noise Mixing Strategy. We analyze the proposed multi-distributed noise mixing strategy and conduct extra ablation experiments to verify its effectiveness. The detailed results are shown in Table 5. We can find that when the α=**0.5**, β=**0.1**, γ=**0.4**, the intrusion detection performance can reach the best, with a 31.75% intrusion accuracy. The main reason is that, in task-specific transfer, the model focuses more on texture features and spatial perturbations. Besides, the transfer task is performed during normal weather. Thus, the need for weather changes and light perturbations is low. In this paper, we set the α, β, γ to **0.5**, **0.1** and **0.4**, respectively.

Dynamic Memory-Gated Module. We also explore the effect of different memory unit sizes on intrusion detection performance, as shown in Table 5. IOU^r denotes the zero-shot segmentation results of the road. We can see that the best intrusion performance can be reached when **M=40**. The main reason is that the larger memory can help capture richer history and global features, especially in open-world intrusion detection. However, larger memory units also introduce more irrelevant information, making it difficult to focus on key memory features. Conversely, fewer memory units can help the model focus on features relevant to intrusion detection, but if the memory units are too low, it will lead to a loss of diversity and complexity required for the intrusion task, affecting the understanding of complex intrusion scenarios. In this paper, we set the memory units to **40**.

Table 5: The ablation experiments of the proposed strategies. **B** denotes the baseline.

Ablation 1: Multi-Distributed Noise Mixing Strategy							
Methods				Metrics			
B	# $\mathcal{U}(\alpha)$	# $\mathcal{N}(\beta)$	# $\mathcal{L}(\gamma)$	PQ	IOU^r(%)	mAP(%)	Acc(%)
✓	✓(1)	✗	✗	14.03	74.1	27.6	29.36
✓	✓(0.5)	✓(0.4)	✓(0.1)	14.37	70.1	32.1	30.28
✓	✓(0.5)	✓(0.2)	✓(0.3)	14.78	68.8	25.8	30.48
✓	✓(0.5)	✓(0.3)	✓(0.2)	14.53	78.5	25.9	31.03
✓	✓(0.5)	✓(0.1)	✓(0.4)	15.33	74.6	28.6	**31.75**
Ablation 2: Dynamic Memory-Gated Module							
Methods				Metrics			
B	Memeroy units			PQ	IOU^r(%)	mAP(%)	Acc(%)
✓	✗			14.03	74.1	27.6	29.36
✓	✓(M=30)			14.84	76.0	27.5	30.31
✓	✓(M=40)			15.80	76.5	29.5	**30.72**
✓	✓(M=50)			15.27	80.1	27.9	30.34

More Insightful and Interesting Experiments

Why is the performance result of category 'Rider' is '0.0?'

In some table, we find that the performance of category 'Rider' is '**0.0**', *e.g.*, Table 14 and Table 15. To answer this question, 1) we first investigate some of the latest open-vocabulary works (Bianchi et al., 2024; Ma et al., 2025). Some works denote that the understanding of fine-grained properties of objects and their parts is important. From this view, we conduct some experiments and provide the visualization comparison, as shown in Figure 6. We can find that our model recognizes the category 'Rider' as the category 'Person'. The main reason is that these two categories have similar features. 2) Besides, in the training dataset, the number of category 'Person' is much larger than the category 'Rider,' which leads to category imbalance. Therefore, these two factors will make it difficult to recognize the fine-grained category 'Rider'. To compensate for this gap, we design a simple yet effective reasoning enhancement strategy, *i.e.*, Geometric Constraint Reclassification Strategy (GCRS). The detaled princlples of GCRS is shown in **Appendix F**. Then, we conduct several experiments to verify the effectiveness of the GCRS strategy, as shown in Table 6. We can find that when applying the proposed strategy, the performance of the rider category improved from 0 to 23.56. Additionally, intrusion detection performance has improved, *i.e.*, 32.79→33.55.

Table 6: The performance of GCRS.

Performance	Rider_Y	Rider_N	Rider	Acc
OVIDNet (Ours)	0	0	0	32.79
+ GCRS	11.95	31.52	23.56	**33.55**

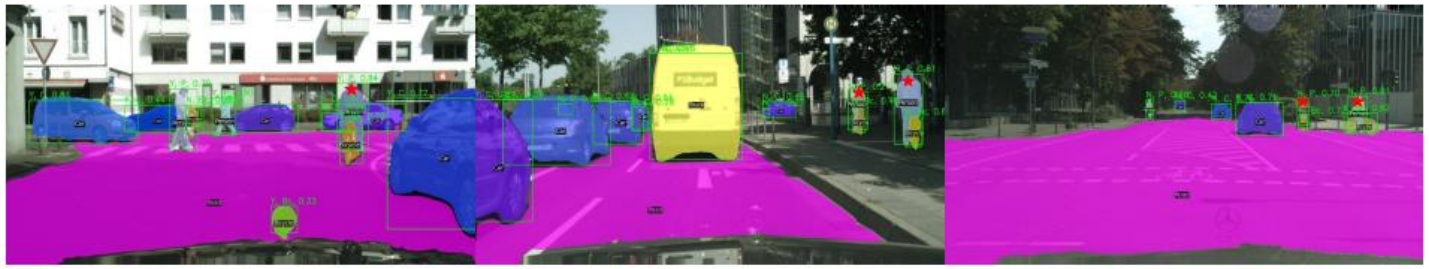


Figure 6: Some cases of recognizing 'Rider (R)' as 'Person (P)'. ★ denotes the case locations.

More Insightful and Interesting Experiments

Real-scenario application exploration

Figure 7: The visualization demo results in real scenarios. We directly utilize our framework to infer public static scenario datasets without any retraining process. We give three different text prompts customizable results, *i.e.*, 2 text prompts, 3 text prompts, and 9 text prompts, respectively.

We can observe that our framework can detect and judge intrusion behavior, demonstrating the practicality and effectiveness of the proposed framework.

Discussions

To consider more distribution shift types and enhance the diversity of intrusion scenes in open-world deployment, we created a new intrusion detection dataset for the OVID task, namely Cityintrusion-OpenV-BDD. The new dataset is built based on the BDD-100K datasets (Yu et al., 2020). The detailed information can be found in **Appendix G**. Our new datasets contain rich intrusion scene types.We evaluate the performance of our model on the datasets, as shown in the Table 7. We can find that, in different domain shifts, our strategies still present promising performance improvements. Compared with the baseline, our model can surpass it by **4.58%**, which verifies the strong robustness of our model and the effectiveness of the proposed strategies.

Table 7: The detailed experimental results on Cityintrusion-OpenV-BDD dataset.

Baseline	DMG	MDNM	AccY(%)	AccN(%)	Acc(%)	Gain
✓	✗	✗	20.99	17.22	18.69	-
✓	✓	✗	20.16	19.27	19.62	+0.93
✓	✗	✓	20.26	20.38	20.33	+1.64
✓	✓	✓	25.79	21.66	**23.27**	**+4.58**

Conclusions

In this paper

- We propose a new and vital intrusion detection task, Open-Vocabulary Intrusion Detection (OVID). This is the first multi-modal attempt in the vision-based intrusion detection task. A new benchmark, including a relative dataset, an efficient multi-modal framework, and some strong baselines, is given for the specific task.

- Two effective strategies are proposed to improve the generalization and enhance the performance of the intrusion detection task in open scenarios, i.e., the Multi-Distributed Noise Mixing and the Dynamic Memory-Gated module.

- Rich experiments and comparisons are done to demonstrate the effectiveness of the proposed framework and strategies. In the future, we will further explore more useful methods to improve performance.