APPENDIX OVERVIEW

Table of contents:

A CITYINTRUSION-OPENV DATASET

In this subsection, we present additional information and details for the proposed Cityintrusion-OpenV dataset, including data statistics, visualization results, intrusion Dataset Comparisons, and the correspondence between full name, text prompt, abbreviation, and framework design motivation.

A.1 METHOD AND STATISTICS FOR PROPOSED DATASET

Automatic Generation Method. Inspired by promising intrusion detection works (Sun et al., 2020; Shi et al., 2022; Han et al., 2024b), our Cityintrusion-OpenV dataset is built based on the Cityscape dataset (Cordts et al., 2016a). The main reason is that the Cityscape datasets have segmentation and detection labels for the same original image, which provides a prerequisite for our multiple intrusion detection tasks. Additionally, following the relevant works (Han et al., 2024c), we also design an automatic labeling program to generate Intrusion ('Y') and No-intrusion ('N') labels. Note that the final intrusion detection overlapping pixel points are also set to **20** (Sun et al., 2020). The specific processes are shown as follows.

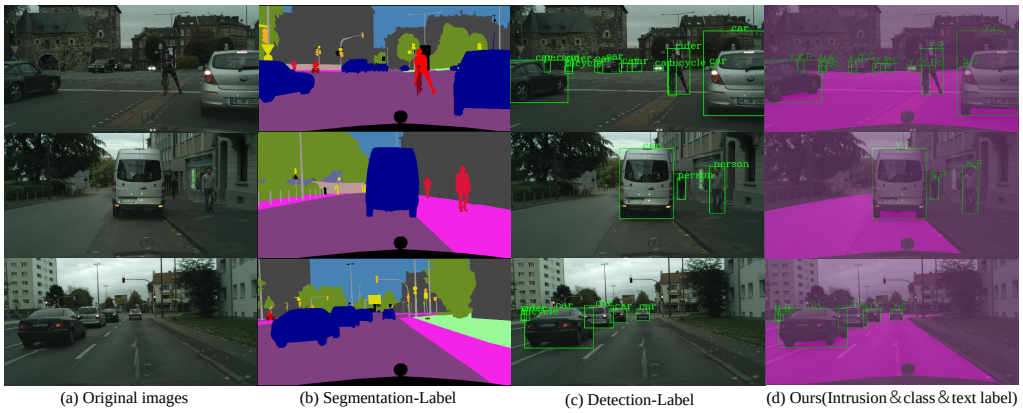
· **Step 1:** We first clean the original Cityscape (Cordts et al., 2016b)/Foggy-Cityscape (Sakaridis et al., 2018). After cleaning, we conduct frame alignment for these datasets. Note that a small number of objects that we don't care about or are incorrectly labeled will be removed in this process.

· **Step 2:** Based on the results in step 1, we can read the bounding box coordinates of the interested intrusion objects from the Cityscape/Foggy-Cityscape datasets. Additionally, we also read the area-of-interest (AoI) in the Cityscapes segmentation dataset (Cordts et al., 2016b).

· **Step 3:** For the obtained area-of-interest (AoI) in step 2, we binarize them with **0** and **1**.

· **Step 4:** After step 3, the bounding box coordinates from step 2 are projected into the binarized area-of-interest (AoI).

· **Step 5:** We calculate the overlapping pixel values between AoI and bounding box in step 4.



(a) Original images (b) Segmentation-Label (c) Detection-Label (d) Ours(Intrusion &class &text label)

Figure 8: More visualization results of our Cityintrusion-OpenV. Unlike previous intrusion detection datasets (Sun et al., 2020; Han et al., 2024b), our datasets encompass all common/possible intrusion categories in Cityscape datasets, providing richer and varied labels that meet the requirements of the proposed OVID task.

· **Step 6:** To get the final intrusion/no-intrusion labels: 'N/Y, Class', we compare overlapping pixel values in step 5 with a setting threshold, where 'N' denotes Non-Intrusion, 'Y' denotes Intrusion, and 'Class' denotes names of intrusion objects. Note that, following previous work (Sun et al., 2020), the threshold is set to 20.

· **Step 7:** To obtain and present our final intrusion detection dataset better, we blended the segmented images containing the intrusion labels in step 6 with the original images in step 1.

· **Step 8:** Finally, to ensure the quality and accuracy of the proposed datasets, a team of three students are organized to manually inspect and verify the annotations.

Statistical Analysis. Then, we conduct a detailed statistical analysis, as shown in Table 8. We provide details of the number of intrusion and non-intrusion cases for each category in the training and validation sets of the dataset, along with the total average. The total average of the whole dataset can reach **18.03**, surpassing previous promising intrusion detection datasets greatly. Rich labels can meet the requirements and provide a data foundation for the proposed OVID task.

Table 8: The detailed statistics of proposed datasets. T and V denote the training and validation datasets, respectively. ‡ denotes the total average in the whole dataset.

Categories	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle
	T|V	T|V	T|V	T|V	T|V	T|V	T|V	T|V
Intrusion cases ('Y')	3567|716	698|226	14545|2493	246|52	219|67	88|9	270|55	1138|361
Non-Intrusion cases ('N')	14427|2703	1109|330	12610|2174	243|41	166|31	83|14	469|94	2591|814
Total cases	17994|3419	1807|556	27155|4667	489|93	385|98	171|23	739|149	3729|1175
Total average‡ (Per very image)	**18.03**							

A.2 More Visualization Results

In order to better present our proposed dataset, we also provide more visualization results, as shown in Figure 8. Different from the previous single category (President) (Sun et al., 2020; Shi et al., 2022) and four categories (President, Motorcycle, Rider, Bicycle) (Han et al., 2024b;c), we can find that our Cityintrusion-OpenV dataset contains multiple different intrusion categories, not only single or four categories. All possible intruder categories can be considered in our dataset, *e.g.*, Person, Rider, Car, Truck, Bus, Train, Motorcycle, Bicycle. Our new dataset can provide the prerequisite for the OVID task. Here, because of the labels, we utilize abbreviations instead of labels in order to easily show our results, *e.g.*, 'N, P' denotes the 'No-Intrusion, Person', Text prompt: Person. 'Y, C' denotes the 'Intrusion, Car', Text prompt: Car. 'Y, Bu' denotes the 'Intrusion, Bus', Text

Table 9: The correspondence between the full name, text prompt, and abbreviation. *Italic* denotes *thing classes* (AoI). All categories are customizable in different scenarios.

# No.	# Full name	# Text prompt	# Abbreviation
# 1	'Person'	'Person'	'P'
# 2	'Rider'	'Rider'	'R'
# 3	'Car'	'Car'	'C'
# 4	'Truck'	'Truck'	'Tk'
# 5	'Bus'	'Bus'	'Bu'
# 6	'Train'	'Train'	'Tn'
# 7	'Motorcycle'	'Motorcycle'	'M'
# 8	'Bicycle'	'Bicycle'	'Bc'
# 9	'*Road*'	'*Road*'	'*Ro*'
# 10	⋮	⋮	⋮

prompt: Bus. 'Y, M' denotes the 'Intrusion, Motorcycle', Text prompt: Motorcycle. The detailed correspondence between full name, text prompt, and abbreviation and can be found in Table 9.

A.3 More Intrusion Dataset Comparisons

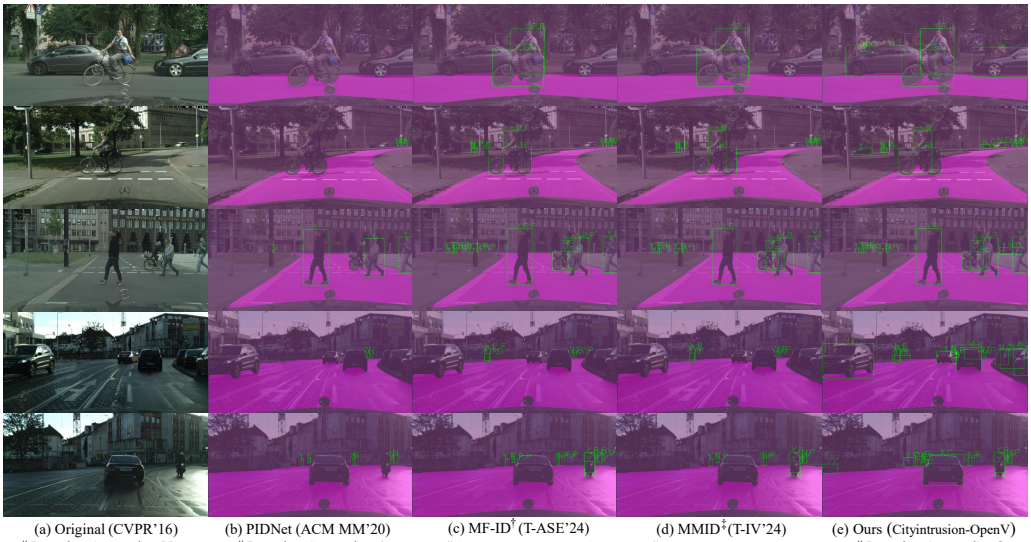
In this subsection, we further compare our proposed datasets with other promising intrusion detection datasets and provide more comparison results to verify the superiority of our dataset, as shown in Figure 9. Compared to previous promising intrusion detection datasets (Sun et al., 2020; Han et al., 2024b;c), our dataset exhibits much superior and richer labels. Besides, the proposed datasets contain **8** intrusion categories, surpassing the previous works **1** or **4** categories.More importantly, our Cityintrusion-OpenV dataset contains text labels, which compensate for the lack of relevant datasets and meet the needs of the proposed OVID task.

A.4 The Correspondence Between Full Name, Text Prompt, and Abbreviation

To better help understand the different intrusion categories and the abbreviations in our paper, we provide the detailed correspondence between the full name, the text prompt, and the abbreviation. The detailed correspondence is shown in the Table 9, *e.g.*, 'Person' (# Full name) → 'Person' (# Text prompt) → 'P' (# Abbreviation), 'Rider' (# Full name) → 'Rider' (# Text prompt) → 'R' (# Abbreviation), 'Car' (# Full name) → 'Car' (# Text prompt) → 'C' (# Abbreviation), 'Truck' (# Full name) → 'Truck' (# Text prompt) → 'Tk' (# Abbreviation), 'Bus' (# Full name) → 'Bus' (# Text prompt) → 'Bu' (# Abbreviation).

A.5 Framework Design Motivation

In this subsection, We first explore two basic yet important questions as motivations for our approach. (1) **Why** do we conduct open vocabulary intrusion detection research? Our goal is to break through the dependencies and limitations of pre-defined categories. Truly enable intrusion detection in the open world. (2) **How** to achieve the specific OVID task? A simple idea is that we can leverage a collaborative model with Open-vocabulary segmentation (OVS), *e.g.*, SAM (Kirillov et al., 2023), FastSAM (Zhao et al., 2023), EfficientSAM (Xiong et al., 2024), and Open-vocabulary detection (OVD), *e.g.*, DetClip (Yao et al., 2023), Grounding DINO (Liu et al., 2024), YOLO-world (Cheng et al., 2024) to train/infer and get final Intrusion/No-intrusion labels. As shown in Table 10, we list and compare some feasible schemes. Unfortunately, although the model of 'OVD+OVS' is a feasible solution, it is not suitable for intrusion detection. The main reason is that the training cost of the End-to-End strategy combined with two **LLVMs** (Large Language Vision Models) is very expensive. To alleviate this problem, we design a new efficient framework for the proposed OVID task, namely OVIDNet. Our framework is established based on OpenSeeD (Zhang et al., 2023). Finally, the OVIDNet is leveraged to collaborate to give the bounding box and mask image for the OVID



(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 9: The compression between our datasets with other promising intrusion detection datasets. Unlike previous intrusion datasets (Sun et al., 2020; Han et al., 2024b), our datasets encompass a broader range of potential intruders. Our datasets can be used to train/evaluate the performance of the OVID task and validate the effectiveness of the proposed strategies.

task, and the final intrusion labels ('N/Y') are given by the intrusion post-processing judgments. The overall framework and pipeline of OVIDNet are illustrated below.

Table 10: The comparison of some feasible schemes for the proposed OVID task. OVD and OVS denote the Open-vocabulary detection and segmentation models. Pre-trained TSM denotes the pre-trained traditional segmentation models, *e.g.*, DeepLabv3+ (Chen et al., 2018), PspNet (Zhao et al., 2017). Retrain denotes whether the model needs to be retrained under different scenarios. We can find that our scheme is End-to-End and has a low training cost.

Scheme	OVD		OVS		Pre-trained TSM	End-to-End	Open-Vocabulary?	Retrain	Training Cost
	Train	Infer	Train	Infer					
S1	✓	✓	✓	✓	✗	✗	✓	✗	Very Large
S2	✓	✓	✗	✗	✓	✓	✗	✓	Large
S3	✓	✓	✗	✓	✗	✗	✓	✗	Low
Ours	✓	✓	✓	✓	✗	✓	✓	✗	Low

B Mechanism proof for Multi-Distributed Noise Mixing strategy

To clarify why the proposed noise mixture improves generalization toward unknown categories, we provide a theoretical proof based on Vicinal Risk Minimization (VRM) (Chapelle et al., 2000). In Equation 3, our perturbation can be written as

$$\mathbf{B}_f = \mathcal{C}\left\{\mathbf{B}_e + (\alpha \cdot \mathbf{N}_u + \beta \cdot \mathbf{N}_g + \gamma \cdot \mathbf{N}_t) \odot \Delta \odot \Theta,\ \mathbf{0},\ \mathbf{1}\right\}. \tag{6}$$

To facilitate subsequent derivation, we set $Z = \alpha \cdot \mathbf{N}_u + \beta \cdot \mathbf{N}_g + \gamma \cdot \mathbf{N}_t$. And $\mathbf{N}_u, \mathbf{N}_g, \mathbf{N}_t$ denote noise sampled from Uniform, Gaussian, and Laplace distributions, respectively. Since Z is a combination of three independent noise sources, its distribution can be expressed as

$$p_Z(z) = \alpha p_u(z) + \beta p_g(z) + \gamma p_t(z), \tag{7}$$

which is a mixed distribution containing (1) fine-grained bounded perturbations (U), (2) moderate Gaussian variations (Gaussian), (3) heavy-tailed structural deviations (Laplace). This directly yields a mixed vicinal distribution in the Vicinal Risk Minimization (VRM) framework. Following previous work (Chapelle et al., 2000), the vicinal risk can be written as

$$\hat{R}_{\text{VRM}}(f) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \mathbb{E}_{z\sim p_Z} \ell(f(x+z), y). \tag{8}$$

Because p_Z is a mixture, VRM can be expressed as

$$\hat{R}_{\text{M-VRM}}(f) = \alpha\hat{R}_u(f) + \beta\hat{R}_g(f) + \gamma\hat{R}_t(f), \tag{9}$$

where $\hat{R}_u(f) = \mathbb{E}_{z\sim p_u}\ell(f(x+z), y), \hat{R}_g(f) = \mathbb{E}_{z\sim p_g}\ell(f(x+z), y), \hat{R}_t(f) = \mathbb{E}_{z\sim p_t}\ell(f(x+z), y)$. Thus, instead of training against a single perturbation model, our model effectively optimizes risk under three complementary vicinal neighborhoods simultaneously.

Let the real open-world perturbation distribution be $q(z)$. If q and p_Z share support and $q \ll p_Z$, then the Radon-Nikodym ratio can be expressed as

$$K := \sup_z \frac{q(z)}{p_Z(z)}, \tag{10}$$

and the K is finite, giving

$$q(z) \leq Kp_Z(z). \tag{11}$$

Because $p_Z(z) = \alpha p_u(z) + \beta p_g(z) + \gamma p_t(z)$ covers bounded signals (U), smooth variations (Gaussian), and heavy-tailed structural changes (Laplace). It has wider support and a smaller worst-case ratio K than any single distribution. Therefore, the true risk is upper-bounded by

$$R_q(f) = \mathbb{E}_{z\sim q}\ell(f(x+z), y) \leq K\hat{R}_{\text{M-VRM}}(f). \tag{12}$$

A smaller constant K implies a tighter bound. Hence, our model can better generalize to unknown object shapes, sizes, and distributions, and improve localization for novel or rare categories. Because their perturbations are more likely to fall inside the large support of the mixture p_Z.

C More Experiment Settings

In this section, we will introduce more implementation details and settings in the experiments. We present more setting details for the experiment, as shown in Table 11. Due to the limitation of our GPUs, we have to set the Image_size to 800 and reduce the iterations to 15000, which inevitably makes some of our results lower than those in the original model (image_size: 1200×1200, max_iter: 368750) (Zhang et al., 2023). To ensure fairness and verify the correctness of our method, we also set CHECKPOINT_PERIOD and EVAL_PERIOD to 15000, respectively, for all experiments. Additionally, we retrain the baseline and verify the method's validity. Note that our OVIDNet framework is built based on the OpenSeeD (Zhang et al., 2023). The OpenSeeD is a simple but efficient framework for open-vocabulary segmentation and detection. Differently, we modify the original framework to meet the requirements of the OVID task. We first add an effective intrusion detection judgment module to obtain the capability for intrusion detection. Then, we propose two strategies for improving generalization and intrusion performance in the open world and verify the effectiveness on multiple dominant datasets and tasks. Therefore, to be fair, our setting mainly refers to Openseed and is adapted to our own tasks. Our OVIDNet framework consists of a text encoder (Clip (Radford et al., 2021)), image encoder (Tiny-swin-transformer (Liu et al., 2021)), decoder, and intrusion detection post-processing module. Note that the final overlapping pixel threshold is set to 20 (Sun et al., 2020).

D The comparisons with other strong open-vocabulary detection systems

To further validate the effectiveness of our model, we conduct a thorough investigation for other open-vocabulary detection models, *e.g.*, YOLO-World (Cheng et al., 2024) and Grounding DINO (Liu et al., 2024), and find that they primarily focus on the open-vocabulary detection domain. However, our task requires not only detection sub-tasks but also segmentation sub-tasks and intrusion detection sub-tasks. Therefore, open-vocabulary detection models alone cannot fulfill our intrusion detection requirements. Besides, we also find that some models support the open-vocabulary instance segmentation. However, these models also cannot meet the needs of the OVID task, as shown in Table 12. Nevertheless, to validate the effectiveness of our proposed model, we conducted some experiments on the YOLO-world and GroundingDINO models. Specifically, we follow previous intrusion detection work (Sun et al., 2020; Han et al., 2024b) and adopt a combined

Table 11: The detailed illustration of the experiment setting.

# Name 1	Setting Category 1	Value	# Name 2	Setting Category 2	Value
TOKENIZER	TEXT	CLIP	WINDOW_SIZE	BACKBONE	7
CONTEXT_LENGTH		18	PATCH_SIZE		4
WIDTH		512	EMBED_DIM		96
HEADS		8	DEPTHS		[2, 2, 6, 2]
LAYERS		12	NUM_HEADS		[3, 6, 12, 24]
# Name 3	Setting Category 3	Value	# Name 4	Setting Category 4	Value
IGNORE_VALUE	ENCODER	255	NHEADS	DECODER	8
LOSS_WEIGHT		1.0	CLASS_WEIGHT		4.0
CONVS_DIM		256	MASK_WEIGHT		5.0
MASK_DIM		256	DICE_WEIGHT		5.0
COMMON_STRIDE		4	BOX_WEIGHT		5.0
TRANSFORMER_ENC_LAYERS		6	GIOU_WEIGHT		2.0
TOTAL_NUM_FEATURE_LEVELS		4	HIDDEN_DIM		256
NUM_FEATURE_LEVELS		3	NUM_OBJECT_QUERIES		300

model approach for testing, *i.e.*, open-vocabulary detection (OVD) + open-vocabulary segmentation (OVS). Here, open-vocabulary detection models contain YOLO-world (Cheng et al., 2024). Besides, the open-vocabulary segmentation model adopts the Clipseg (Lüddecke & Ecker, 2022) to complete the experiments. Note that for the latter (open-vocabulary segmentation model), we initially intended to adopt the SAM model (Kirillov et al., 2023). However, we find that the original SAM requires point or bounding box information for object segmentation. Therefore, in these experiments, we employed the Clipseg model for the open-vocabulary segmentation subtask. The specific experiment results are as shown in Table 13.

Table 12: The proposed OVID task analyses in different open-vocabulary works.

Name	Instance segmentation work	Panoptic segmentation work
Does it segment stuff (road/other)	✗	✓
Are output pixels fully covered?	✗	✓
Can it meet the requirements of the OVID task?	✗	✓

Table 13: The comparison results between different combined models and our OVIDNet model.

Model	Type	Domain	Acc	Parameter
YOLO-World-S (Cheng et al., 2024)+Clipseg (Lüddecke & Ecker, 2022)	OVD+OVS	Normal Foggy	22.64 17.75	163.75M
YOLO-World-L (Cheng et al., 2024)+Clipseg (Lüddecke & Ecker, 2022)	OVD+OVS	Normal Foggy	30.08 24.34	198.75M
OVIDNet(Ours)	End-to-End	Normal Foggy	**32.79** **27.83**	**120.32M**

We can find that, compared with the combined model (YOLO-World-L+Clipseg), our model can surpass it by **2.71%** (normal domain) and **3.49%** (foggy domain). These performance gains demonstrate the effectiveness of our model.

E More Results for OVID Task

In this section, we will provide additional results to test the effectiveness of our framework and strategies. We first report quantitative results of different categories in normal and cross-domain conditions. Then, we present more visualization results. The specific results are shown below.

E.1 Quantitative Results of Different Categories

We first present additional results from various categories using the proposed strategies. Note that we give two types of metrics, *i.e.*, segmentation/detection metrics (IOU, AP, AP@.5) and intrusion

detection metrics (AccY, AccN, Acc), respectively. The former is obtained via a zero-shot manner, the latter via a task-specific transfer manner. We conduct experiments in COCO, Cityscape, and Cityintrusion-OpenV, as shown in Table 14. From Table 14, we can find that when the proposed methodology is added, multiple metrics in multiple categories are improved to a certain extent. Compared with the baseline model, our strategies can surpass it by 3.97% (IOU) and 0.93% (AP), respectively, which verifies the effectiveness of the proposed strategies.

Table 14: The more quantitative results of different intrusion categories. Task: **COCO→Cityscape, Cityintrusion-OpenV**. We provide quantitative results for all possible intrusion categories to test the effectiveness of the proposed strategies. Besides, to comprehensively measure the results across different categories, we report two distinct metrics, *i.e.*, segmentation/detection metrics (IOU, AP, AP@.5) and intrusion detection metrics (AccY, AccN, Acc), respectively. The **bold** is the best result.

-			**Intrusion Categories, Task: COCO→Cityscape, Cityintrusion-OpenV**								
Strategies			**Segmentation and Detection Metrics**								
Baseline	DMG	MDNM	Person(%)	Rider(%)	Car(%)	Truck(%)	Bus(%)	Train(%)	Motorcycle(%)	Bicycle(%)	**Avg**(%)
			IOU|AP|AP@.5	IOU|AP|AP@.5	IOU|AP|AP@.5	IOU|AP|AP@.5	IOU|AP|AP@.5	IOU|AP|AP@.5	IOU|AP|AP@.5	IOU|AP|AP@.5	IOU|AP|AP@.5
✓	✗	✗	64.7|9.1|23.8	0.0|0.0|0.0	80.8|17.2|36.7	24.0|18.6|25.3	62.8|36.1|53.9	2.2|13.7|25.4	45.8|10.3|25.1	69.6|10.4|30.4	43.74|14.43|27.58
✓	✗	✓	67.2|10.6|26.1	0.0|0.0|0.0	82.5|17.1|38.7	36.0|19.1|29.4	47.8|32.4|52.8	4.1|17.7|31.8	50.9|8.2|21.3	68.5|8.8|28.4	44.63|14.24|28.56
✓	✓	✗	69.9|12.3|32.1	0.0|0.0|0.0	80.1|21.4|45.5	24.2|14.5|20.5	51.0|36.2|53.1	0.6|12.5|20.0	56.1|10.7|25.7	72.2| 11.5|36.4	44.26|14.89|**29.16**
✓	✓	✓	66.5|11.0|28.6	0.0|0.0|0.0	86.5|22.8|47.0	37.4|22.1|32.9	60.2|33.3|45.4	0.0|9.6|13.6	58.6|11.9|27.8	72.5|12.2|36.5	**47.71**|**15.36**|28.98
Strategies			**Intrusion Detection Metrics**								
Baseline	DMG	MDNM	Person(%)	Rider(%)	Car(%)	Truck(%)	Bus(%)	Train(%)	Motorcycle(%)	Bicycle(%)	All Categories
			AccY |AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc
✓	✗	✗	12.43|46.39|39.28	0.00|0.00|0.00	22.74|29.48|25.88	30.77|7.32|20.43	31.34|35.48|32.65	0.00|0.00|0.00	14.55|21.28|18.79	12.19|38.70|30.55	18.72|36.19|29.36
✓	✓	✗	18.02|42.47|37.35	0.00|0.00|0.00	22.50|36.38|28.97	28.85|4.88|18.28	25.37|38.71|29.59	0.00|0.00|0.00	20.00|26.60|24.16	18.01|43.12|35.40	20.06|37.56|**30.72**
✓	✗	✓	21.51|39.99|36.12	0.00|0.00|0.00	22.70|42.87|32.10	21.15|14.63|18.28	35.82|32.26|34.69	11.11|7.14|8.70	12.73|23.40|19.46	20.22|42.26|35.49	21.01|38.64|**31.75**
✓	✓	✓	19.55|41.10|36.59	0.00|0.00|0.00	28.12|40.16|33.73	32.69|7.32|21.51	32.84|29.03|31.63	0.00|0.00|0.00	25.45|15.96|19.46	21.61|43.61|36.85	24.43|38.16|**32.79**

Table 15: The more quantitative results of different intrusion categories in the cross-domain task. We further test the effectiveness of the proposed strategies with a task-specific transfer manner. Task: **COCO→Foggy-Cityscape, Cityintrusion-OpenV**. Three foggy conditions are used to conduct comprehensive experiments, *i.e.*, α=0.005,α=0.01,α=0.02. The **bold** is the best result.

-			**Intrusion Categories, Task: COCO→Foggy-Cityscape, Cityintrusion-OpenV**								
Strategies			α = **0.005**								
Baseline	DMG	MDNM	Person(%)	Rider(%)	Car(%)	Truck(%)	Bus(%)	Train(%)	Motorcycle(%)	Bicycle(%)	All Categories
			AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc
✓	✗	✗	14.11|45.14|38.64	0.00|0.00|0.00	25.43|24.38|24.94	28.85|4.88|18.28	31.34|29.03|30.61	0.00|0.00|0.00	9.09|20.21|16.11	10.25|37.10|28.85	20.43|33.58|28.44
✓	✓	✗	19.13|41.21|36.59	0.00|0.00|0.00	23.67|30.96|27.06	38.46|4.88|23.66	29.85|32.26|30.61	0.00|0.00|0.00	23.64|22.34|22.82	18.56|41.15|34.21	21.29|34.75|**29.49**
✓	✗	✓	20.39|38.18|34.45	0.00|0.00|0.00	24.07|38.68|30.88	26.92|12.20|20.43	31.34|29.03|30.61	0.00|7.14|4.35	14.55|27.66|22.82	20.22|40.66|34.38	21.66|36.20|**30.52**
✓	✓	✓	20.81|39.55|35.62	0.00|0.00|0.00	28.28|35.05|31.43	32.69|7.32|21.51	38.81|25.81|34.69	0.00|0.00|0.00	27.27|19.15|22.15	22.44|42.26|36.17	24.96|35.54|**31.40**
Strategies			α = **0.01**								
Baseline	DMG	MDNM	Person(%)	Rider(%)	Car(%)	Truck(%)	Bus(%)	Train(%)	Motorcycle(%)	Bicycle(%)	All Categories
			AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc
✓	✗	✗	14.53|41.66|35.98	0.00|0.00|0.00	26.11|20.84|23.66	30.77|4.88|19.35	29.85|19.35|26.53	0.00|0.00|0.00	14.55|20.21|18.12	13.30|36.12|29.11	21.29|30.64|26.98
✓	✓	✗	19.55|38.81|34.78	0.00|0.00|0.00	24.79|27.51|26.06	36.54|4.88|22.58	29.85|22.58|27.55	0.00|0.00|0.00	20.00|22.34|21.48	20.22|38.94|33.19	22.14|32.16|**28.24**
✓	✗	✓	21.09|36.81|33.52	0.00|0.00|0.00	25.31|33.53|29.14	28.85|7.32|19.35	35.82|29.03|33.67	11.11|7.14|8.70	18.18|27.66|24.16	22.99|39.07|34.13	23.00|33.56|**29.43**
✓	✓	✓	20.95|37.81|34.28	0.00|0.00|0.00	30.00|30.50|30.23	28.85|7.32|19.35	29.85|25.81|28.57	0.00|0.00|0.00	25.45|20.21|22.15	23.55|40.17|35.06	25.94|32.93|**30.20**
Strategies			α = **0.02**								
Baseline	DMG	MDNM	Person(%)	Rider(%)	Car(%)	Truck(%)	Bus(%)	Train(%)	Motorcycle(%)	Bicycle(%)	All Categories
			AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc	AccY|AccN|Acc
✓	✗	✗	14.53|37.22|32.47	0.00|0.00|0.00	26.96|14.86|21.32	30.77|2.44|18.28	28.36|12.90|23.47	0.00|0.00|0.00	20.00|20.21|20.13	15.24|30.96|26.13	22.04|25.88|24.38
✓	✓	✗	21.23|34.48|31.71	0.00|0.00|0.00	26.75|20.10|23.66	32.69|4.88|20.43	29.85|16.13|25.51	0.00|0.00|0.00	23.64|24.47|24.16	22.44|33.05|29.79	23.88|26.90|**25.72**
✓	✗	✓	23.60|33.07|31.09	0.00|0.00|0.00	27.96|26.36|27.21	30.77|2.44|18.28	35.82|19.35|30.61	11.11|7.14|8.70	16.36|20.21|18.79	27.42|33.54|31.66	25.51|28.50|**27.33**
✓	✓	✓	21.79|33.59|31.12	0.00|0.00|0.00	32.25|23.83|28.33	28.85|4.88|18.28	32.84|16.13|27.55	0.00|0.00|0.00	25.45|17.02|20.13	25.48|34.52|31.74	27.72|27.90|**27.83**

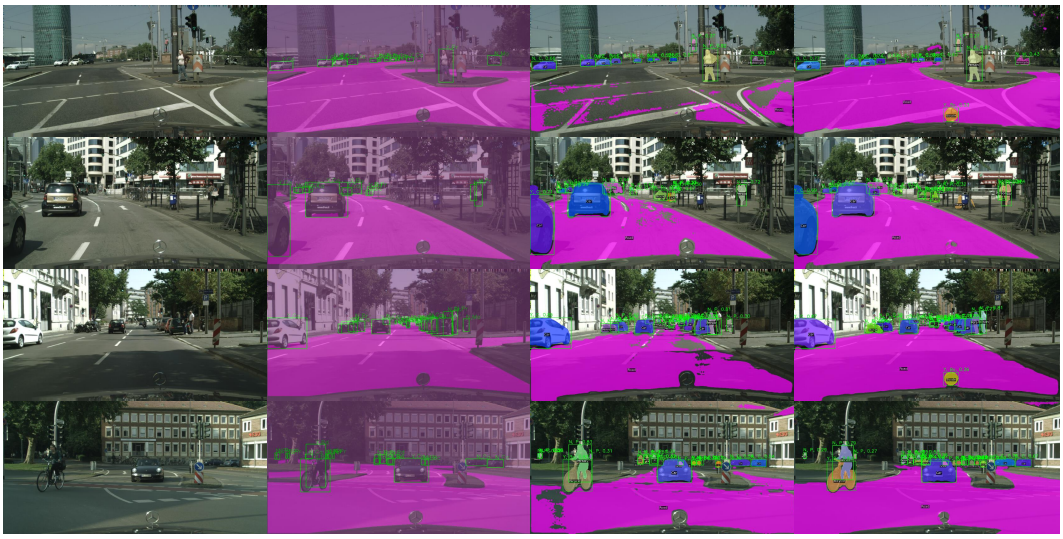
E.2 More Results of Cross-domain Task

Furthermore, we present additional intrusion detection results for various intrusion categories across different cross-domain tasks. In this experiment, we adopt three different foggy coefficients, *i.e.*, α=0.005, α=0.01, and α=0.02. In these experiments, We conduct experiments in COCO, Foggy-Cityscape, and Cityintrusion-OpenV, as shown in Table 15. We can observe that, in various cross-domain tasks, our strategies enhance intrusion detection performance. In different tasks, compared

with the original baseline model, our framework can improve them by 2.96%, 3.22%, and 3.45%, respectively. Furthermore, our proposed approach can effectively improve the performance of intrusion detection for various categories. These performance improvements demonstrate the effectiveness of our approach, as well as the ability of our framework to generalize.

E.3 More Visualization Comparison Results

Finally, we also present more visualization comparison results to verify the effectiveness of the proposed framework and strategies. We set the text prompt of stuff_classes as Road and set the text prompt of thing_classes as'Person', 'Rider', 'Car', 'Truck', 'Bus', 'Train', 'Motorcycle', 'Bicycle', as shown in Figure 10. From Figure 10, we can find that our framework can present promising visualization detection results, not only detecting all intruders correctly but also giving correct Intrusion ('Y')/No-intrusion ('N') labels, which proves the effectiveness of our approach.



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

Figure 10: The visualization comprising results. Here, (a), (b), (c), and (d) denote Original images, Ground truth, Baseline results, and Ours, respectively. Text prompt: Road (stuff_classes), 'Person', 'Rider', 'Car', 'Truck', 'Bus', 'Train', 'Motorcycle', 'Bicycle', (thing_classes). For thing_classes, the abbreviations are used instead of complete labels to easily show our results. The correspondence between the abbreviation and full name can be referred to in the Table 9.

F The Principles of Geometric Constraint Reclassification Strategy

In subsection 5.4, we analyze why the performance of the category 'Rider' is '0.0' in depth. To compensate for this gap and enhance the practicality of our model, we design a simple yet effective reasoning enhancement strategy, *i.e.*, Geometric Constraint Reclassification Strategy. Our GCRS strategy is implemented within the post-processing reasoning phase, specifically designed to mitigate local class ambiguity, *e.g.*, category similarity. The ambiguity occurs when our model predicts the person in the vehicle as a standalone Subject Box B_P (Person), rather than the intended composite Rider class. Therefore, the principle of our GCRS strategy relies on utilizing the spatial topological constraints between predicted bounding boxes. Specifically, by calculating the IoU between the detected B_P and any potential Accessory Box B_V (Bicycle/Motorcycle), we quantify their degree of spatial coupling. If this coupling exceeds a predefined High-Coupling Threshold τ, it constitutes a strong geometric constraint. Then, a mandatory class is reassigned to enforce semantic consistency. The core correction criterion is formally expressed as a conditional reclassification operation, and we can express it as

$$\text{If Class}(B_P) = \text{'Person' and } (\exists B_V \in \{(\text{'Bicycle', 'Motorcycle'})\} \text{ s.t. IoU}(B_P, B_V) > \tau), \quad \text{Then Class}(B_P) \leftarrow \text{'Rider'}, \tag{13}$$

where τ is a threshold. The IoU metric used to quantify the overlap is defined as

$$\text{IoU}(B_P, B_V) = \frac{\text{Area}(B_P \cap B_V)}{\text{Area}(B_P \cup B_V)}. \tag{14}$$

Our GCRS strategy acts as a domain-knowledge-driven geometric filter. And GCRS can significantly enhance our model's accuracy on some challenging composite instances (***e.g.***, Rider category) by prioritizing geometric evidence over primary model scores.

G THE DETAILED INFORMATION OF CITYINTRUSION-OPENV-BDD DATASET

To consider more distribution shift types and enhance the diversity of intrusion scene 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). We clean the original dataset based on the proposed OVID task features. The detailed method can refer to Appendix A.1 in our paper. Our new datasets contain rich intrusion scene types, e.g., multiple different weather (Clear, Cloudy, Rainy, Foggy, Night), different geographic environments (City, Highway, Suburban/Rural), different period of time (Daytime, Dusk, Night), and Different transportation environments (Heavy Traffic, Empty Road). These new domains can meet the experiment's requirements in different distribution shifts. Finally, our datasets contain 1482 training data and 449 evaluation data. We evaluate the performance of our model on these datasets, as shown in the Table 7. We can find that, in different domain shifts, our strategies still present performance improvements. Compared with the baseline model, our model can surpass it by **4.58%**, which verifies the strong robustness of our model and the effectiveness of the proposed strategies.

H LIMITATIONS AND FURTHER WORKS

In this paper, we introduce the Open-Vocabulary Intrusion Detection (OVID) project for the first time, including a new task, an efficient framework, and a strong benchmark for vision-based intrusion detection. Additionally, we design corresponding strategies to enhance intrusion detection performance in real-world scenarios and increase the practicality of the model. However, there are still some limitations that need to be addressed in the future: (1) Enhance the ability to recognize fine-grained categories and improve generalization performance in the real world. (2) Inspired by methods for parameter-efficient fine-tuning (Hu et al., 2022; Ye et al., 2026), we will explore efficient ways to enhance the intrusion detection performance of models at a lower fine-tuning cost.