

# Ada-iD: Active Domain Adaption for Intrusion Detection

Anonymous Authors

## A DETAILED PROCESS OF MEJ STRATEGY

In this section, we will introduce the spectral clustering process in detail. Spectral clustering is an effective graph theory-based clustering method [22], which mainly includes six steps, which are Similarity Matrix, Adjacency matrix (W), Degree Matrix (D), Laplacian Matrix, Feature Matrix, and Clustering, respectively. The detailed workflow is shown in A.

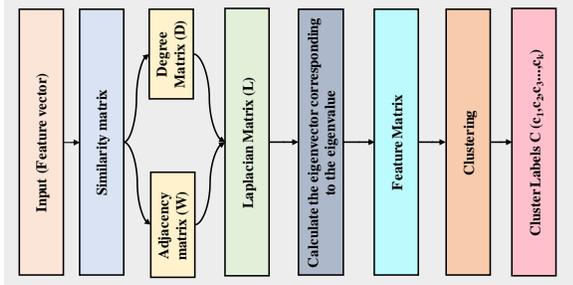


Figure A: The workflow of spectral clustering. Spectral clustering mainly contains six steps, Similarity Matrix, Adjacency matrix (W), Degree Matrix (D), Laplacian Matrix, Feature Matrix, and Clustering, respectively.

Besides, we also give the detailed calculation process. In subsection 3.5, given an input image  $I_i$ , we defined the final results matrix  $\mathcal{X}$  and  $\mathcal{G} = S(\mathbf{N}_k | \mathbf{F}_c, \mathbf{F}_p)$ . For spectral clustering  $\mathcal{S}$ , we first need to calculate the similarity matrix ( $w_{i,j}$ ) by the Gaussian kernel function and express it as

$$w_{i,j} = \begin{cases} \mathbf{K}(\mathbf{F}_i, \mathbf{F}_j) = \exp\left\{-\frac{\|\mathbf{F}_i - \mathbf{F}_j\|_2^2}{2\sigma^2}\right\} & \text{if } (i, j \in \mathbf{E}) \\ 0 & \text{if } (i, j \notin \mathbf{E}), \end{cases} \quad (1)$$

where  $\|\mathbf{F}_i - \mathbf{F}_j\|_2^2$  is the squared Euclidean distance, and  $\sigma$  is the bandwidth parameter of the Gaussian kernel. In spectral clustering, we can define Graph-based as  $\mathbf{G} = \{\mathbf{V}, \mathbf{E}\}$ , where  $\mathbf{V}$  and  $\mathbf{E}$  denotes the sets of vertex and edge and  $\mathbf{V}$  is expressed as  $\mathbf{V} = \{1, 2, 3, \dots, \mathbf{N}\}$ , and  $\mathbf{E}$  is expressed as  $\mathbf{E} : \mathbf{W} = [w_{i,j}]$ ,  $1 \leq i, j \leq \mathbf{N}$ . Note that the calculation way of our Inter-class similarity (IS) is the same as the definition of computing the similarity matrix in spectral clustering [22]! Besides, we can define  $\mathbf{A} \in \mathbf{V}$ ,  $\mathbf{B} \in \mathbf{V}$ , and  $\mathbf{A} \cap \mathbf{B} = \emptyset$ , then we express it as  $\mathcal{W}(\mathbf{A}, \mathbf{B}) = \sum_{i \in \mathbf{A}, j \in \mathbf{B}} w_{i,j}$ . Suppose that the  $\mathbf{K}$  categories existed, then we can know that  $\mathbf{V} = \bigcup_{k=1}^{\mathbf{K}} \mathbf{A}_k$ , and  $\mathbf{A}_i \cap \mathbf{A}_j = \emptyset, \forall i, j \in \{1, 2, \dots, \mathbf{K}\}$ . The object function can be defined as

$$\min_{\{\mathbf{A}_k\}_{k=1}^{\mathbf{K}}} \mathcal{N}_{cut}(\mathbf{A}_1, \mathbf{A}_2 \dots \mathbf{A}_K) = \sum_{k=1}^{\mathbf{K}} \frac{\mathcal{W}(\mathbf{A}_k, \overline{\mathbf{A}_k})}{\sum_{i \in \mathbf{A}_k} \mathbf{D}_i}, \quad (2)$$

where  $\mathbf{D}_i$  denotes the degree matrix and  $\mathbf{D}_i = \sum_{j=1}^{\mathbf{N}} w_{i,j}$ .  $\overline{\mathbf{A}_k}$  denotes the complementary set of  $\mathbf{A}_k$ . We then use the trace of the matrix

to rewrite Equation 2 as

$$\min_{\{\mathbf{A}_k\}_{k=1}^{\mathbf{K}}} \mathcal{N}_{cut}(\mathbf{A}_1, \dots \mathbf{A}_K) = \text{tr} \begin{pmatrix} \frac{\mathcal{W}(\mathbf{A}_1, \overline{\mathbf{A}_1})}{\sum_{i \in \mathbf{A}_1} \mathbf{D}_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{\mathcal{W}(\mathbf{A}_K, \overline{\mathbf{A}_K})}{\sum_{i \in \mathbf{A}_K} \mathbf{D}_K} \end{pmatrix} = \text{tr}(\mathbf{O} \cdot \mathcal{P}^{-1}), \quad (3)$$

where the matrix  $\mathbf{O}$  is expressed as

$$\mathbf{O} = \begin{pmatrix} \mathcal{W}(\mathbf{A}_1, \overline{\mathbf{A}_1}) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathcal{W}(\mathbf{A}_K, \overline{\mathbf{A}_K}) \end{pmatrix}_{K \times K}, \quad (4)$$

and the matrix  $\mathcal{P}$  is expressed as

$$\mathcal{P} = \begin{pmatrix} \sum_{i \in \mathbf{A}_1} \mathbf{D}_i & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sum_{i \in \mathbf{A}_K} \mathbf{D}_i \end{pmatrix}_{K \times K}. \quad (5)$$

Suppose that  $x_i \in \{0, 1\}$  and  $\sum_{j=1}^{\mathbf{K}} x_{ij} = 1, 1 \leq i \leq \mathbf{N}, 1 \leq j \leq \mathbf{K}$ .  $x_{ij}$  denotes the  $i$ -th sample belongs to the  $j$ -th category.  $\mathbf{X} = (x_1, x_2 \dots x_N)_{\mathbf{N} \times \mathbf{K}}^T$ . Then, we can know that

$$\begin{aligned} \mathbf{X}^T \mathbf{X} &= (x_1, x_2 \dots x_N) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{N}_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{N}_K \end{pmatrix}_{K \times K} \\ &= \begin{pmatrix} \sum_{i \in \mathbf{A}_1} \mathbf{1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sum_{i \in \mathbf{A}_K} \mathbf{1} \end{pmatrix}_{K \times K}, \end{aligned} \quad (6)$$

where  $\mathbf{N}_K$  denotes the number of samples belonging to category  $\mathbf{K}$  in  $\mathbf{N}$  samples.  $\sum_{k=1}^{\mathbf{K}} \mathbf{N}_K = \mathbf{N}$ ,  $\mathbf{N}_K = \sum_{i \in \mathbf{A}_K} \mathbf{1}$ . Therefore, we can find that the Equation 5 can be rewritten as

$$\begin{aligned} \mathcal{P} &= \begin{pmatrix} \sum_{i \in \mathbf{A}_1} \mathbf{D}_i & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sum_{i \in \mathbf{A}_K} \mathbf{D}_i \end{pmatrix} \\ &= \sum_{i=1}^{\mathbf{N}} x_i \mathbf{D}_i x_i^T = \mathbf{X}^T \mathbf{D} \mathbf{X}, \end{aligned} \quad (7)$$

where  $\mathbf{D} = \text{diag}(\mathbf{W} \cdot \mathbf{1}_N)$  and the Equation 4 can be written as

$$\mathbf{O} = \mathbf{X}^T \mathbf{D} \mathbf{X} - \mathbf{X}^T \mathbf{W} \mathbf{X}, \quad (8)$$

where  $\mathbf{X}^T \mathbf{W} \mathbf{X}$  can be described as

$$\begin{aligned}
X^T \mathbf{W} X &= (x_1 \cdots x_N) \begin{pmatrix} \mathbf{W}_{11} & \cdots & \mathbf{W}_{1N} \\ \vdots & \ddots & \vdots \\ \mathbf{W}_{N1} & \cdots & \mathbf{W}_{NN} \end{pmatrix} \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{pmatrix} \\
&= \left( \sum_i x_i \mathbf{W}_{i1} \cdots \sum_{i=1}^N x_i \mathbf{W}_{iN} \right) \begin{pmatrix} x_1^T \\ \vdots \\ x_n^T \end{pmatrix} \\
&= \begin{pmatrix} \sum_{i \in A_1} \sum_{j \in A_1} \mathbf{W}_{ij} & \cdots & \sum_{i \in A_1} \sum_{j \in A_K} \mathbf{W}_{ij} \\ \vdots & \ddots & \vdots \\ \sum_{i \in A_K} \sum_{j \in A_1} \mathbf{W}_{ij} & \cdots & \sum_{i \in A_K} \sum_{j \in A_K} \mathbf{W}_{ij} \end{pmatrix} \\
&= \sum_{i=1}^N \sum_{j=1}^N x_i \mathbf{W}_{ij} x_j^T.
\end{aligned} \tag{9}$$

Thus, the Equation 8 can be rewritten as

$$\begin{aligned}
O &= X^T \mathbf{D} X - X^T \mathbf{W} X \\
&= X^T (\mathbf{D} - \mathbf{W}) X \\
&= X^T \mathbf{L} X.
\end{aligned} \tag{10}$$

Besides, the Equation 3 can be rewritten as

$$\min_{\{\mathbf{A}_k\}_{k=1}^K} N_{cut}(\mathbf{A}_1, \cdots, \mathbf{A}_K) = \text{tr} \left\{ X^T (\mathbf{D} - \mathbf{W}) X \cdot \left( X^T \mathbf{D} X \right)^{-1} \right\}, \tag{11}$$

where  $\mathbf{L}$  denotes the Laplacian Matrix and  $\mathbf{L}$  can be expressed as  $\mathbf{L} = \mathbf{D} - \mathbf{W}$ . In spectral clustering, we perform eigendecomposition of  $\mathbf{L}$  to obtain the eigenvalues and corresponding eigenvectors. Then, we select the eigenvectors corresponding to the smallest  $k$  eigenvalues to form the **feature matrix** and apply  $k$ -means clustering to divide the data into  $k$  cluster. Then, to tackle the problem of class imbalance, the resampling is used to the cluster results  $\mathcal{G}$  and the process can be expressed as  $\mathcal{G}' = \mathbf{R}_s(\mathcal{G})$ .  $\mathbf{R}_s$  denotes the resampling. Building on the resampled dataset  $\mathcal{G}'$ , we compute the within-cluster confidence uncertainty, denoted as  $C_k$ , for each cluster  $k$ . This uncertainty is quantified by the standard deviation of the resampled confidence scores

$$C_k = \sqrt{\frac{1}{N_k - 1} \sum_{i=1}^{N_k} (C_k^i - \bar{C}_i)^2}, \tag{12}$$

where  $C_k$  denotes the confidence uncertainty.  $\bar{C}_i$  and  $C_k^i$  denotes the confidence average and  $i$ -th resampled samples confidence scores in  $k$ -th cluster. Besides, the class probability distribution uncertainty can be calculated by

$$\mathcal{P}_k = -\frac{1}{|N_k|} \sum_{i \in N_k} \sum_{j=1}^4 \mathbf{P}_{ij} \log \mathbf{P}_{ij}, \tag{13}$$

where  $N_k$  is the number of re-samples in cluster  $k$ .  $\mathbf{P}_{ij}$  denotes the the probability of the  $j$ -th category of the  $i$ -th samples. Therefore, the overall uncertainty score of input image  $\mathbf{I}_t$  is expressed as

$$\mathcal{M}(\mathbf{U}_d^t | \mathbf{I}_t, \Theta^n) = \frac{1}{N_k} \sum_{k=1}^{N_k} (C_k^t + \mathcal{P}_k^t), \tag{14}$$

where  $C_k^t$  and  $\mathcal{P}_k^t$  denotes uncertainty of confidence, probability distribution, and tiny object of image  $\mathbf{I}_t$ , respectively.

## B MORE IMPLEMENTATION DETAILS

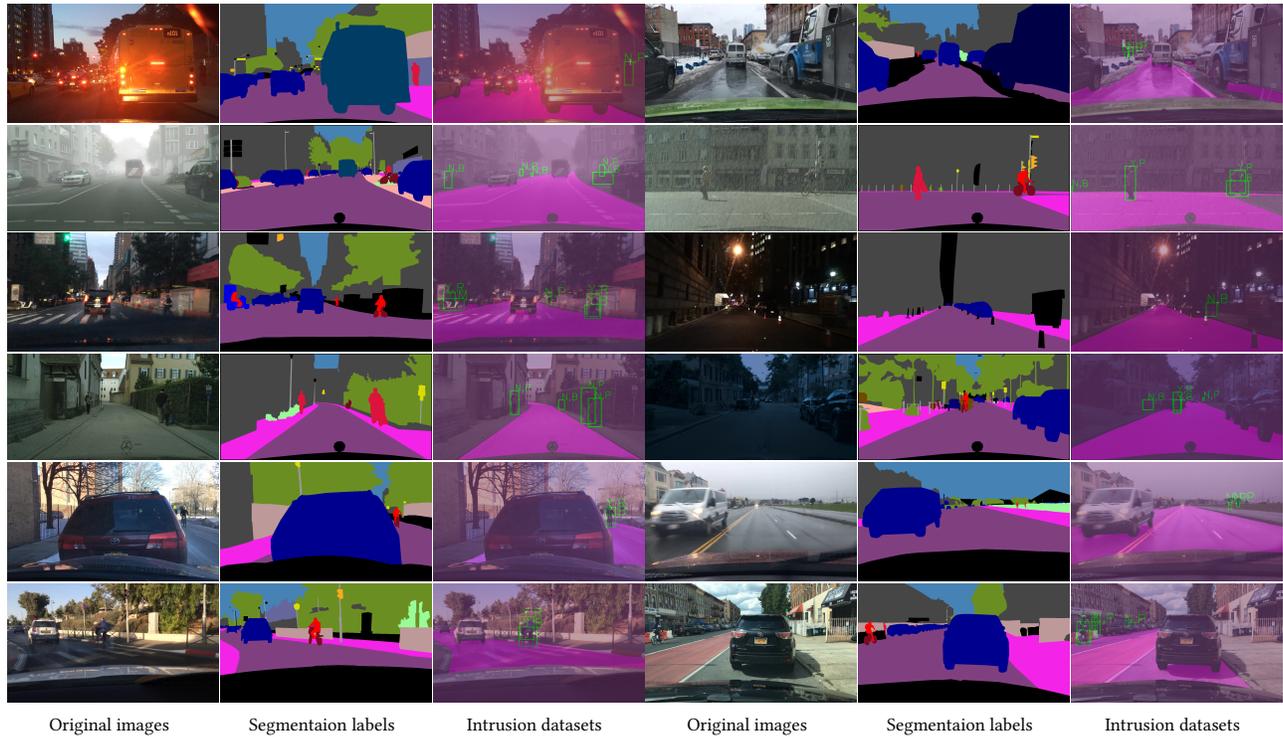
In this section, we present more implementation details. As shown in Table A. Here, Setting 1 is about experimental parameter settings, including multiple hyperparameters, e.g., optimizer weight decay, warmup epochs, warmup initial momentum, box loss gain, cls loss gain, etc. Setting 2 is about the number of sampling times during the experimental process.  $1^{st}$ ,  $2^{nd}$ ,  $\dots$   $20^{th}$  denotes the rounds. In the unlabeled target samples, the number of T1 (**Rainy**), T2 (**Foggy**), and T3 (**Night**) is 2502, and the number of T4 (**BDD-intrusion**) is 839. Therefore, we adopt the method of rounding to the nearest whole number to select samples, i.e., final query samples =  $\lceil \mathbf{N} \cdot \mathbf{R} \rceil$ , where  $\mathbf{N}$  denote the number of samples in the target domain,  $\mathbf{R}$  denotes the rate of sampling,  $\lceil \cdot \rceil$  denotes the rounding function, e.g.,  $1^{st}$ :  $2502 \times 0.1\% \approx 3$  images,  $2^{nd}$ :  $2502 \times 0.2\% \approx 5$  (added 2 images). All selected samples are sent to annotate. Then, the labeled samples and source domain are merged to retrain the proposed framework ADAID-YOLO until meeting the max annotation budget  $\mathbf{B}$ .

Setting 1	Value	Setting 2 (Budget B)	Query samples			
			T1: F	T2: R	T3: Ng	T4: B
batch_size	24	Round 1 ( $1^{st}$ )	3	3	3	1
epoch	150	Round 2 ( $2^{nd}$ )	5	5	5	2
initial learning rate	0.01	Round 3 ( $3^{rd}$ )	8	8	8	3
SGD momentum	0.937	Round 4 ( $4^{th}$ )	10	10	10	3
optimizer weight decay	$5e^{-4}$	Round 5 ( $5^{th}$ )	13	13	13	4
warmup epochs	3.0	Round 6 ( $6^{th}$ )	15	15	15	5
warmup initial momentum	0.8	Round 7 ( $7^{th}$ )	18	18	18	6
warmup initial bias lr	0.1	Round 8 ( $8^{th}$ )	20	20	20	7
box loss gain	0.05	Round 9 ( $9^{th}$ )	23	23	23	8
cls loss gain	0.5	Round 10 ( $10^{th}$ )	25	25	20	8
cls BCELoss positive_weight	1.0	Round 20 ( $20^{th}$ )	50	50	50	17

**Table A: The detailed illustration of experiments setting. ‘T’ denotes the target domain. ‘R’, ‘F’, ‘Ng’, and ‘B’ denote the Normal-Rainy, Normal-Foggy, and Normal-Night and BDD-intrusion datasets, respectively.**

## C DETAILED INFORMATION FOR DATASETS

In this section, we will provide more information for using datasets, including description, statistical analysis and visualization images. The detailed description and statistical analysis are shown in Table B. We conduct comprehensive experiments on five different source/target domains to verify the effectiveness of our sampling strategy. Here, the Normal-CMC, Foggy-CMC, Rainy-CMC, Night-CMC, and BDD-intrusion datasets are built on public Cityscapes, Cityperson, and BDD-100K datasets [10]. In the validation set, the intrusion cases reach 11.82 and 4.74 Intrusion/No-Intrusion labels, respectively. Note that when we conduct experiments on BDD-intrusion datasets, only 839 images in Normal-CMC are chosen to serve as the source domain. The main reason is that 839 images (samples) can meet the characteristics of our intrusion detection task, i.e., containing detection and segmentation labels in the target domain (BDD-intrusion). Besides, we also provide detailed visualization images. As presented in Figure B. From Figure B, we



**Figure B: An illustration of related datasets. Here, we use domain intrusion datasets and provide some visualization images from some intrusion datasets, Normal-CMC, Foggy-CMC, Rainy-CMC, Night-CMC, and BDD-intrusion [10]. These datasets contain multiple different and common domains, e.g., Normal, Foggy, Night, Rainy, and Snow, which can meet the requirements of our ADA-ID task.**

can find that: 1) These intrusion datasets contain multiple different normal/adverse weather (domains), including **Sunny, Rainy, Snow, Night, and Foggy**. These multiple different domains provide a cross-domain foundation for our ADA-ID task. 2) Five datasets can provide correct category/Intrusion labels, which denotes these datasets can meet the requirements of our ADA-ID task.

## D BASELINE MODEL DESCRIPTION

In this section, we provide detailed illustrations for using classic models, including Random Sampling, Least Confident [36], Margin Sampling [27], and Entropy Sampling [36].

- 1) **Random Sampling**: *RS* denotes samples are chosen by Random strategy from the unlabeled target domain. And these samples will be labeled by Expert (Oracle).

- 2) **Least Confident** [36]: *LC* indicates that the samples of lowest confidence are chosen for annotation.

- 3) **Margin Sampling** [27]: *MS* indicates that samples where the difference between the model's most likely and second most likely predicted categories is the smallest will be chosen to annotate.

- 4) **Entropy Sampling** [36]: *ES* is used to measure the uncertainty of a system, the greater the entropy, the greater the uncertainty of the system; the smaller the entropy, the lesser the uncertainty of the system. The samples of the most entropy will be chosen to annotate.

Datasets	Descriptions	Classes	Train set	Val set	Intrusion cases (Val set)
Normal-CMC	Source Domain	4	2502	429	11.82
Rainy-CMC	Target Domain 1	4	2502	429	11.82
Foggy-CMC	Target Domain 2	4	2502	429	11.82
Night-CMC	Target Domain 3	4	2502	429	11.82
BDD-intrusion	Target Domain 4	4	839	429	4.74

**Table B: The detailed description and statistics of using datasets. In our experiments, the Source domain is Normal-CMC, and the source domain denotes Normal weather. Four different target domains are existed, Foggy-CMC, Rainy-CMC, Night-CMC, and BDD-intrusion, respectively. The target domain denotes the Adverse weather. Adequate source and target domain datasets provide a foundation for the proposed ADA-ID task.**

## E MORE EXPERIMENT RESULTS

### E.1 More quantitative experiment.

In this subsection, we will conduct more experiments to illustrate the effectiveness of our strategies. We first test the performance of

Task (Source→Target)	AL Strategy	Framework: ADAID-YOLO						Closed Gap ΔmIOU(%) ↑ / ΔmAP@.5(%) ↑ / ΔAcc(%) ↑
		Annotation budget: 0.1% (1/839), 1 <sup>st</sup>						
		mIOU(%)	mAP@.5(%)	mAP@.5:.95(%)	AccY(%)	AccN(%)	Acc(%)	
N→B	Source Only	76.6	26.9	11.1	27.0	27.2	27.2	-
	Random	77.3	27.7	11.6	21.7	31.5	28.5	+7.9% / +10.4% / +19.7%
	Margin Sampling [27]	74.9	26.5	10.6	22.2	32.9	29.6	-19.1% / -5.2% / +36.4%
	Entropy [36]	75.0	28.2	12.0	20.9	34.3	30.2	-18.0% / +16.9% / +45.5%
	RIPU [37]	76.8	28.8	11.7	26.5	32.6	30.8	+2.2% / +24.7% / +54.5%
	Least Confident [36]	75.6	27.8	11.5	26.9	33.2	31.3	-11.2% / +11.7% / +62.1%
	Ours	<b>78.2</b>	<b>29.9</b>	12.0	26.5	34.1	<b>31.8</b>	+18.0% / +39.0% / +69.7%
	Oracle (Fully-Supervised)	85.5	34.6	13.8	29.1	35.9	33.8	-

**Table C: The quantitative results on different adaptation scenarios under 0.1% (1<sup>st</sup>) annotation budget. Here, ‘N’ and ‘B’ denote using different datasets: Normal-CMC and BDD-intrusion, respectively. Source only denotes the training on the source domain and inference on the target domain. Oracle denotes training and inference on the target domain.**

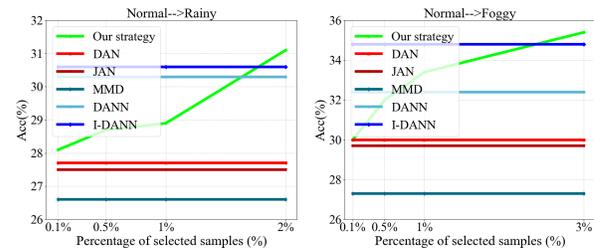
different cross-domain tasks, e.g., Normal-CMC→BDD-intrusion. Like section 4, we also test the performance under the annotation budget 0.1% (1/839) and provide multiple metrics, e.g., mIOU, mAP, AccY, AccN and Acc. The results are shown in Table C. From Table C, we can find that for intrusion detection performance, compared with source only, our strategies can surpass it by 4.6% and outperform classic active learning strategies. Besides, for the performance of segmentation and detection, our strategy also shows the best performance, 78.2% mIOU and 29.9% mAP, which demonstrates the effectiveness of our proposed approach.

## E.2 Compared with more UDA works.

In this subsection, we will conduct more comparison experiments to explore the superiority of our strategy. Specifically, we first test the performance of various target annotation budgets. Then, we compare the performance between our strategy and some common but effective promising UDA methods, e.g., DAN [9], and I-DANN [10], etc. The results are shown in Figure C. From Figure C, we can find that the best performance can be reached when the promising I-DANN is used. Besides, we also can see that 1) In different cross-domain tasks, when using our strategy, 0.5% of data annotations can surpass the performance of multiple UDA methods, e.g., JAN [19], etc. 2) In Normal→Rainy cross-domain tasks, our sampling approaches can effectively improve the performance of intrusion detection, even surpassing the performance of best UDA method with only 2% data annotation, which demonstrates the superiority of our approach.

## E.3 Visualisation Comparisons.

In this subsection, we will present more comparison visualization results to verify the rationality of the proposed sampling strategy. To be fair, we compare the visualization results on multiple different sampling approaches and several cross-domain tasks, e.g., N→F, N→B, N→Ng. All experiments are conducted in round 1. The specific visualization results are presented D. From the D, we can find that, when adopting 0.1% samples (1<sup>st</sup> round), our strategies can

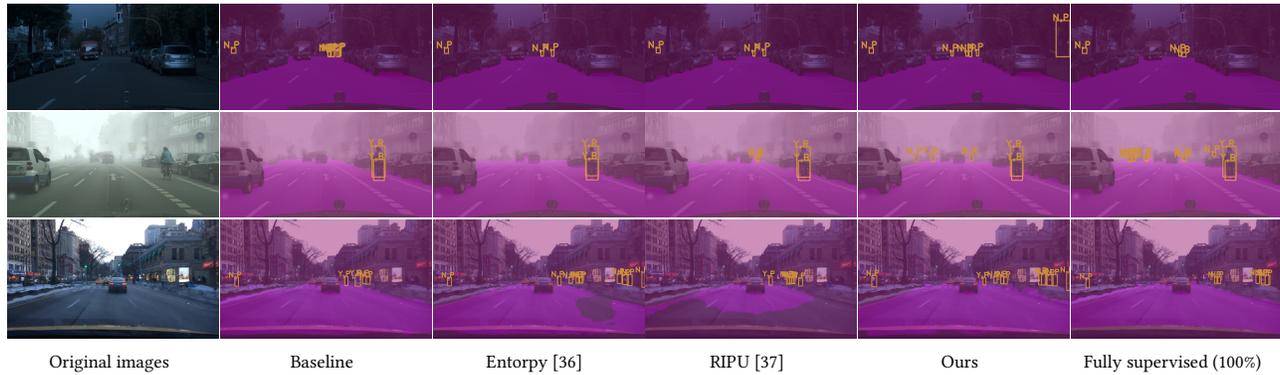


**Figure C: The comparison results of promising UDA works. Task: N→R and N→F. We can find that our sampling approaches can effectively improve the performance of intrusion detection, even surpassing the performance of the best UDA method on N→R with only 2% data annotation.**

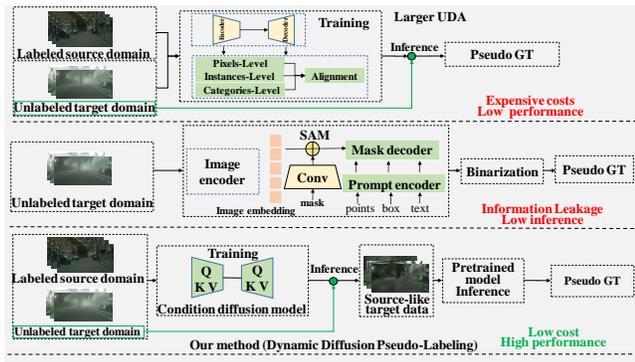
select informative samples for the ADA-ID task, not only effectively detecting the typical Intrusion/No-Intrusion behaviors and intruder category, but also giving correct Intrusion/No-Intrusion labels (‘Y’/‘N’). These advantages indicate that our proposed sampling strategies are effective.

## F COMPARISON OF PSEUDO LABEL STRATEGY

In this section, we introduce and compare three different of obtaining the pseudo GT detailed. As shown in E. To get Pseudo-Labeling ( $\hat{GT}$ ) to help identify areas of uncertainty in segmentation, we need to design an effective Pseudo-Labeling strategy. However, the ground truth of the target domain is not known. How do we get pseudo ground truth? We think that three different but feasible ways can be considered. As shown in Figure E. 1) Training a larger UDA model using source/target domain ( $D_s \cup D_t$ ) and then performing inference ( $D_t$ ) to get the pseudo ground truth. However, this approach needs expensive training costs. 2) Using the promising zero-shot segmentation model, SAM [16]. However, we need to provide extra prompts, i.e. location of *points* or *boxes* when we



**Figure D:** An illustration of comparative results using different strategies. Here, we present more visualization results to verify the effectiveness of our strategies. From the results, we can find that, when adopting 0.1% samples ( $1^{st}$  round), our strategies can select informative samples for the ADA-ID task, not only effectively detecting the typical Intrusion/No-Intrusion behaviors and intruder category, but also giving correct Intrusion/No-Intrusion labels ('Y'/'N'). These advantages indicate that our proposed sampling strategies are effective.



**Figure E:** Comparison of three different strategies. Here, the first strategy (Larger pre-trained UDA model) is expensive for training costs. The second strategy is to use SAM [16] to obtain the Pseudo GT. However, this strategy can cause spatial information leakage due to extra prompts. The aforementioned two strategies are not suited for our ADA-ID task. In the third strategy, we use a simple condition diffusion model to get the translation images (Target domain  $\rightarrow$  Source-like target domain). Then, using the pre-trained model ADAID-YOLO for inference to obtain the Pseudo GT.

use the SAM model, which causes spatial information leakage to some extent. This is because we do not know the spatial location of the restricted AoI in the picture. 3) Therefore, to get pseudo ground truth, we propose a new efficient and low-cost strategy, Dynamic Diffusion Pseudo-Labeling. Specifically, We first train a well-designed condition diffusion model for learning the translation of the source domain ( $D_s$ ) and the target domain ( $D_t$ ). Then, we use the pre-trained Denoising UNet model to infer input data from the target domain for obtaining the source-like target data, and use it to obtain the  $\tilde{GT}$ . The results of the ablation experiments show the effectiveness of our method, which not only generates correct pseudo ground truth but maintains a high inference speed.