A. Sensitivity Analysis

A.1. Parameter Sensitivity

As shown in Table 1, we analyze the sensitivity in terms of the adaptation intensity $\lambda_1$, $\lambda_2$, where $\lambda_1$ works on the node classification loss and $\lambda_2$ controls the intensity of structure-aware matching loss. We first try a group of consistent parameters \{0.05, 0.1, 0.2\} for $\lambda_1, \lambda_2$ (1st to 3rd lines), finding that decreasing the values leads to a significant performance drop compared with our main settings ($\lambda_1, \lambda_2 = 0.1$). By fixing $\lambda_1$, increasing and decreasing $\lambda_2$ slightly decrease the overall performance, demonstrating that our setting ($\lambda_2 = 0.1$) is optimal. By fixing $\lambda_2$, decreasing $\lambda_1$ shows a significant negative impact on the framework while increasing it gives some further improvements. These results demonstrate that the larger intensity on the node loss contributes to establishing a better graphical space for the graph-matching-based adaptation.

<table>
<thead>
<tr>
<th>$\lambda_1$</th>
<th>$\lambda_2$</th>
<th>mAP0.5</th>
<th>mAP0.5</th>
<th>mAP0.75</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>0.05</td>
<td>22.8</td>
<td>42.2</td>
<td>21.4</td>
</tr>
<tr>
<td>0.1</td>
<td>0.1</td>
<td>24.0</td>
<td>43.5</td>
<td>23.5</td>
</tr>
<tr>
<td>0.2</td>
<td>0.2</td>
<td><strong>24.2</strong></td>
<td>43.3</td>
<td>23.3</td>
</tr>
<tr>
<td>0.1</td>
<td>0.05</td>
<td>23.2</td>
<td>42.9</td>
<td>23.0</td>
</tr>
<tr>
<td>0.1</td>
<td>0.2</td>
<td>23.5</td>
<td>43.3</td>
<td>23.1</td>
</tr>
<tr>
<td>0.05</td>
<td>0.1</td>
<td>22.3</td>
<td>42.0</td>
<td>21.8</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1</td>
<td><strong>24.2</strong></td>
<td><strong>43.7</strong></td>
<td>23.1</td>
</tr>
</tbody>
</table>

Table 1. Comparison results on Cityscapes→Foggy Cityscapes (%) of different settings of $\lambda_1$ and $\lambda_2$. We set $\lambda_1, \lambda_2 = 0.1$ in the experiments of the manuscript as 2nd line.

A.2. Position Sensitivity

We further investigate the position to deploy the Node Discriminator (ND) to align the matched nodes, and record the comparison results in Table 2. We compare three settings for the node alignment, i.e., P1: semantic-complete nodes $V_{s/t}$ (without the hallucination nodes), P2: enhanced nodes after graph convolution $\hat{V}_{s/t}$, and P3: the nodes after Cross Graph Interaction (CGI) $\hat{V}_{s/t}$. It can be observed that performing the alignment on the semantic-complete nodes (P1) achieves the best results with well-aligned node pairs. Besides, we find a significant performance drop on P3 because the proposed CGI will exchange information across domains, confusing the discriminator and harming the adversarial alignment. Hence, aligning nodes in P1 is optimal in the proposed method as the setting in our manuscript.

<table>
<thead>
<tr>
<th>Pos.</th>
<th>prsn</th>
<th>rider</th>
<th>car</th>
<th>truc</th>
<th>bus</th>
<th>train</th>
<th>Moto</th>
<th>bike</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td><strong>46.9</strong></td>
<td><strong>48.4</strong></td>
<td><strong>63.7</strong></td>
<td>27.1</td>
<td>50.7</td>
<td>35.9</td>
<td><strong>34.7</strong></td>
<td><strong>41.4</strong></td>
<td><strong>43.5</strong></td>
</tr>
<tr>
<td>P2</td>
<td>43.9</td>
<td>46.0</td>
<td>57.0</td>
<td><strong>29.7</strong></td>
<td><strong>53.9</strong></td>
<td><strong>39.7</strong></td>
<td>34.6</td>
<td>39.6</td>
<td>43.0</td>
</tr>
<tr>
<td>P3</td>
<td>44.0</td>
<td>45.4</td>
<td>57.2</td>
<td>25.2</td>
<td>48.4</td>
<td>26.8</td>
<td>27.5</td>
<td>38.7</td>
<td>39.2</td>
</tr>
</tbody>
</table>

Table 2. Comparison results on Cityscapes→Foggy Cityscapes (%) by deploying the ND on different nodes, i.e., semantic-complete nodes $V_{s/t}$ (P1), enhanced nodes after graph convolution $\hat{V}_{s/t}$ (P2), and the nodes after cross graph interaction $\hat{V}_{s/t}$ (P3).

A.3. Normalization Sensitivity

The proposed method transforms the visual feature to the graphical space (V2G) with a projection module (Fc-Norm-ReLU-Fc). Hence, we present a comparison among different projection strategies with different normalization (Norm) tricks, including Group Normalization (commonly used in the FCOS [16] detection head), Batch Normalization (commonly used in the ResNet [4] backbone network), and Layer Normalization [1], as shown in Table 3. Our projection design with Layer Normalization works better on node embedding than other common settings, preserving node-based correspondence and achieving the best adaptation result (43.5% mAP).

<table>
<thead>
<tr>
<th>Pos.</th>
<th>prsn</th>
<th>rider</th>
<th>car</th>
<th>truc</th>
<th>bus</th>
<th>train</th>
<th>Moto</th>
<th>bike</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td><strong>46.9</strong></td>
<td><strong>48.4</strong></td>
<td><strong>63.7</strong></td>
<td>27.1</td>
<td>50.7</td>
<td>35.9</td>
<td><strong>34.7</strong></td>
<td><strong>41.4</strong></td>
<td><strong>43.5</strong></td>
</tr>
<tr>
<td>P2</td>
<td>43.9</td>
<td>46.0</td>
<td>57.0</td>
<td><strong>29.7</strong></td>
<td><strong>53.9</strong></td>
<td><strong>39.7</strong></td>
<td>34.6</td>
<td>39.6</td>
<td>43.0</td>
</tr>
<tr>
<td>P3</td>
<td>44.0</td>
<td>45.4</td>
<td>57.2</td>
<td>25.2</td>
<td>48.4</td>
<td>26.8</td>
<td>27.5</td>
<td>38.7</td>
<td>39.2</td>
</tr>
</tbody>
</table>

Table 3. Comparison results on Cityscapes→Foggy Cityscapes (%) by deploying the ND on different nodes, i.e., semantic-complete nodes $V_{s/t}$ (P1), enhanced nodes after graph convolution $\hat{V}_{s/t}$ (P2), and the nodes after cross graph interaction $\hat{V}_{s/t}$ (P3).
Table 3. Comparison results on Cityscapes→Foggy Cityscapes (%) of different normalization strategies in the vision-to-graph (V2G) transformation.

<table>
<thead>
<tr>
<th>Pos.</th>
<th>prsn</th>
<th>rider</th>
<th>car</th>
<th>truc</th>
<th>bus</th>
<th>train</th>
<th>moto</th>
<th>bike</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>GN</td>
<td>45.7</td>
<td>44.9</td>
<td>63.1</td>
<td>24.8</td>
<td>48.3</td>
<td>43.2</td>
<td>32.6</td>
<td>40.9</td>
<td>42.9</td>
</tr>
<tr>
<td>BN</td>
<td>46.1</td>
<td>42.8</td>
<td>61.7</td>
<td>27.6</td>
<td>45.5</td>
<td>34.8</td>
<td>32.0</td>
<td>38.0</td>
<td>41.0</td>
</tr>
<tr>
<td>LN</td>
<td>46.9</td>
<td>48.4</td>
<td>63.7</td>
<td>27.1</td>
<td>50.7</td>
<td>35.9</td>
<td>34.7</td>
<td>41.4</td>
<td>43.5</td>
</tr>
</tbody>
</table>

B. Discussion

B.1. Baseline Selection

Two-stage v.s. single-stage baselines. Two-stage object detectors, e.g., Faster RCNN [10], consist of a feature extractor, a Region Proposal Network (RPN) and a detection head for classification and regression. These approaches first adopt RPN on image features to obtain Region of Interests (RoIs), and then perform detection based on these region proposals. Differently, single-stage object detectors [9,16] only contain a feature extractor and detection head, and these approaches directly make prediction on image features without RPN.

Reasons for the single-stage baseline. In this paper, we mainly focus on the domain adaptation for single-stage object detectors as lots of recently published works [2,5–8,15], and we select the single-stage detector as the baseline because of the following two main reasons.

1) Discarding RPN. Most adaptation works [18,20,21] perform adaptation on both image features and RoI representations, which highly rely on the RPN and are limited to the two-stage detectors. In contrast, our method achieves fine-grained adaptation only using image features and totally discards the RPN, yielding enormous potentials to be generalized to different baselines. Hence, we use the single-stage baseline free of RPN in our method to demonstrate the advantages without bells and whistles.

2) Fair comparison. The fairness and agreement of the benchmark comparison have been proven in recently published literature [2,5,7,8,15] for single-stage object detectors due to the comparable source only results and adaptation gains. Besides, we also report the fair adaptation gains in benchmark comparison to demonstrate our effectiveness in terms of domain adaptation. Moreover, most of the latest adaptation works [2,5,7,8,15] are based on the single-stage detectors [9,16], and we aim to present a comparison with them using same baseline model.

Potentials for the two-stage extension. We propose a Graph-embedded Semantic Completion module (GSC) to complete the mismatched semantics and leverage a Bipartite Graph Matching adaptor (BGM) to achieve fine-grained adaptation on image features. These two modules are totally independent of the detection baseline types and can be effortlessly extended to different baselines by deploying on the features extracted from backbone networks.

B.2. Limitation

Though the proposed model could achieve state-of-the-art results, it may have some failure cases (Figure 1) due to the limited visual features. As shown in 1st and 2nd row, we find that our method may miss and wrongly detect some distant objects obscured by heavy fog, e.g., the missing truck (1st row) and the wrongly detected person (2nd row), due to the poor visual features caused by the tiny scale (long distance) and low-quality appearance (heavy fog). This problem can be solved from two aspects, i.e., improving visual representations and compensating for visual features with other cues. On the one hand, we can use more robust backbone networks, e.g., ResNet-101 [4], to obtain better features than the VGG-16 backbone [12]. On the other hand, we can establish graph matching between visual and linguistic cues [19] to compensate for the limited visual features with extra semantics.

Figure 1. Illustration of some failure examples compared between (a) the proposed SIGMA framework and (b) ground-truth.

C. Implementation Details

C.1. Discriminator Architecture

As shown in Table 4, we present the detailed architecture of the adversarial alignment module in our SIGMA framework, which includes the loss terms $\mathcal{L}_{GA}$ and $\mathcal{L}_{VA}$. We adopt image-level global alignment [3] using the Global Discriminator as [3, 5–7, 15, 17, 18, 20]. Then, we introduce a node discriminator to align well-match graph nodes,
as illustrated in the bottom part of Table 4. Considering the graph nodes refactor the image-level spatial correspondence with edge connections, we replace the convolution layers with fully-connected layers. Besides, we change the Group Normalization (GroupNorm) with Layer Normalization (LayerNorm) due to the advantage of operating the node-based representation, as in Sec. A.3.

<table>
<thead>
<tr>
<th>Global Discriminator [5]</th>
<th>Node Discriminator (ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient Reversal Layer (GRL)</td>
<td>Gradient Reversal Layer (GRL)</td>
</tr>
<tr>
<td>Conv 256 × 3 × 3, stride 1 → GroupNorm → ReLU</td>
<td>Fc 256 → LayerNorm → ReLU</td>
</tr>
<tr>
<td>Conv 256 × 3 × 3, stride 1 → GroupNorm → ReLU</td>
<td>Fc 256 → LayerNorm → ReLU</td>
</tr>
<tr>
<td>Conv 256 × 3 × 3, stride 1 → GroupNorm → ReLU</td>
<td>Fc 256 → LayerNorm → ReLU</td>
</tr>
<tr>
<td>Conv 1 × 3 × 3, stride 1</td>
<td>Fc 1 → LayerNorm → ReLU</td>
</tr>
</tbody>
</table>

Table 4. Architectures of the adversarial alignment modules.

### C.2. Implementation and Training

1) **Different blocks.** The non-linear projection layer used in the vision-to-graph (V2G) transformation is deployed with a Fc-LayerNorm-ReLU-Fc block, and the classifier for node classification is Fc-ReLU-Fc.

2) **Dropout rate.** The dropout rate is set 0.1 for the edge-drop [11] to avoid the potential visual bias.

3) **Spectral clustering.** For the learning of the graph-guided memory bank, we perform spectral clustering if the number of nodes is larger than 5 to ensure the clustering reliability. Besides, we replace the Laplacian affinity [14] with K-Nearest Neighbor (K=5) in the clustering algorithm, which reduces the time-consuming significantly.

4) **End-to-end training.** Our method can achieve end-to-end training without the warm-up stage. We utilize halved source nodes as the placeholders if no nodes appear in the target domain to train our matching module and introduce extra 10,000 iterations for training, which can achieve the same results as the warm-up-included strategy.

5) **Multiple matching.** The detailed implementation of the multiple-matching ablation study (in Table 5 of our manuscript) is as follows,

\[ \mathcal{L}_{mat} = \text{Loss}[\text{sigmoid}(\mathbf{M}_{\text{aff}}), \mathbf{Y}_\Pi], \]

where \( \mathbf{M}_{\text{aff}} \) is the node affinity without adopting Instance Normalization and the Sinkhorn [13] layer, and \( \text{Loss}[A, B] \) can be selected as the BCE and MSE loss to evaluate the difference between \( A \) and \( B \).

### Algorithm 1 Semantic-complete Graph Matching

**Input:**
- \( \mathcal{I}_{s/t} \): source and target images
- \( \mathcal{Y}_s \): source annotations
- \( \lambda_{1,2} \): hyperparameters in the loss function

**Output:**
- Domain adaptive object detector \( \Theta \)

1: for \( l = 1 \) to \( \text{maxiter} \) do
2: extract image features \( \mathcal{F}_{s/t} \) with backbone networks;
3: generate global alignment loss \( \mathcal{L}_{GA} \) on \( \mathcal{F}_{s/t} \);
4: send \( \mathcal{F}_{s/t} \) to the detection head to generate \( \mathcal{L}_{det} \) with \( \mathcal{F}_s \) and classification maps \( \mathcal{M}_t \) with \( \mathcal{F}_t \);
5: perform V2G transformation to obtain nodes \( \mathcal{V}_{s/t}^{raw} \);
6: generate node alignment loss \( \mathcal{L}_{NA} \);
7: perform DNC for semantic-complete nodes \( \mathcal{V}_{s/t} \);
8: establish graphs \( \mathcal{G}_{s/t} \) and perform GCN for \( \mathcal{V}_{s/t} \);
9: update GMB with enhanced nodes \( \mathcal{V}_{s/t} \);
10: **Bipartite Graph Matching (BGM)**
11: perform CGI obtaining \( \mathcal{V}_{s/t} \) and generate loss \( \mathcal{L}_{node} \);
12: generate graph matching \( \mathcal{L}_{mat} \);
13: use \( \mathcal{L} = \lambda_1 \mathcal{L}_{node} + \lambda_2 \mathcal{L}_{mat} + \mathcal{L}_{NA} + \mathcal{L}_{GA} + \mathcal{L}_{det} \) to update network parameters with backpropagation;
14: end for
15: return Domain adaptive object detector \( \Theta \);

### C.3. Optimization Pipeline

The overall optimization pipeline of the proposed SIGMA framework is shown in Algorithm 1. Given the source and target images \( \mathcal{I}_{s/t} \), source annotations \( \mathcal{Y}_s \), and some predefined hyperparameters \( \lambda_{1,2} \), we implement the SIGMA framework to obtain a domain adaptive object detector \( \Theta \) with \( \text{maxiter} \) iterative training.

### D. Qualitative Results

**D.1. Matching Visualization**

As shown in Figure 2, we visualize the learned doubly stochastic node affinity matrix \( \mathbf{M}_{\text{aff}} \) and the ground-truth matrix \( \mathbf{Y}_\Pi \) (Refer to Figure 2 of the manuscript for better understanding). Each activated entry \( \mathbf{M}_{\text{aff}}^{ij} \) represents a matched node pair across domains, and each activated entry \( \mathbf{Y}_\Pi^{ij} = 1 \) (marked in red) indicates that the source node \( v_i^s \) and the target counterpart \( v_j^t \) are in the same category. Based on the proposed structure-aware matching loss, each source node successfully find an optimal target node in the same category adaptively and match it to achieve graph-matching-based adaptation.
D.2. Qualitative Comparison

We present more qualitative comparisons among (a) source only, (b) EPM [5], (c) the proposed SIGMA, and (d) ground-truth in Figure 3-6. Our method can eliminate some missing errors (false-negative cases) and avoid some wrong classification cases (false-positive cases) compared with the class-agnostic method EPM [5], which verifies the effectiveness of aligning class-conditional distributions.

References


Figure 3. Qualitative results on the Cityscapes→Foggy Cityscapes adaptation scenario of (a) the source only model, (b) EPM [5], (c) the proposed SIGMA, and (d) Ground-truth. (Zooming in for best view.)
Figure 4. Qualitative results on the Cityscapes→Foggy Cityscapes adaptation scenario of (a) the source only model, (b) EPM [5], (c) the proposed SIGMA, and (d) Ground-truth. (Zooming in for best view.)
Figure 5. Qualitative results on the Cityscapes→Foggy Cityscapes adaptation scenario of (a) the source only model, (b) EPM [5], (c) the proposed SIGMA, and (d) Ground-truth. (Zooming in for best view.)
Figure 6. Qualitative results on the Cityscapes→Foggy Cityscapes adaptation scenario of (a) the source only model, (b) EPM [5], (c) the proposed SIGMA, and (d) Ground-truth. (Zooming in for best view.)