SIGMA++: Improved Semantic-complete Graph Matching for Domain Adaptive Object Detection

Wuyang Li, Xinyu Liu, and Yixuan Yuan, *Member, IEEE*

Abstract—Domain Adaptive Object Detection (DAOD) generalizes the object detector from an annotated domain to a label-free novel one. Recent works estimate prototypes (class centers) and minimize the corresponding distances to adapt the cross-domain class conditional distribution. However, this prototype-based paradigm 1) fails to capture the class variance with agnostic structural dependencies, and 2) ignores the domain-mismatched classes with a sub-optimal adaptation. To address these two challenges, we propose an improved SemantIc-complete Graph MAtching framework, dubbed SIGMA++, for DAOD, completing mismatched semantics and reformulating adaptation with hypergraph matching. Specifically, we propose a Hypergraphical Semantic Completion (HSC) module to generate hallucination graph nodes in mismatched classes. HSC builds a cross-image hypergraph to model class conditional distribution with high-order dependencies and learns a graph-guided memory bank to generate missing semantics. After representing the source and target batch with hypergraphs, we reformulate domain adaptation with a hypergraph matching problem, i.e., discovering well-matched nodes with homogeneous semantics to reduce the domain gap, which is solved with a Bipartite Hypergraph Matching (BHM) module. Graph nodes are used to estimate semantic-aware affinity, while edges serve as high-order structural constraints in a structure-aware matching loss, achieving fine-grained adaptation with hypergraph matching. The applicability of various object detectors verifies the generalization, and extensive experiments on nine benchmarks show its state-of-the-art performance on both AP_{50} and adaptation gains. Code is available at https://github.com/CityU-AIM-Group/SIGMA/tree/SIGMA++.

Index Terms—Domain Adaptive Object Detection, Prototypes, Class Conditional Distribution, Hypergraph Matching.

1 Introduction

Generic object detection [8], [52], [59], [60], [71] has achieved great success when trained and evaluated in the consistent domain. However, directly implementing a well-trained object detector in a novel domain suffers from a severe performance drop due to the inherent domain gap [60]. This limitation significantly limits real-world applications, e.g., autonomous driving in variant weather [2] and disease diagnosis with different medical machines [1].

To address this limitation, recent works have introduced Unsupervised Domain Adaptation [26], [29], [72] in object detection, using a labeled source domain to generalize to an unlabeled target domain. Most current works focus on the feature-level representation and align the latent feature space of cross-domain scenes. Some works [12], [13], [33], [63], [65] align the marginal distribution of the whole image, conducting a pixel-to-pixel adaptation across image-level features. Moreover, the instance-level representation raises many interests [13], [35], [90], [95], serving for adapting the foreground distribution. Recently, some approaches [9], [10], [46], [70], [84], [92] consider the semantic-level knowledge and align class conditional distribution, which adapt the domain-shared class space [26]. As shown in Fig 1(a), these works estimate class prototypes (class centers) with object instances, and then minimize the distance of cross-domain prototypes to bridge the domain gap at the category level.

Though great success, there are two overlooked deficiencies in the aforementioned prototype-based paradigm [9], [46], [70], [84], [92]. Firstly, these works directly align class centers and neglect the essential within-class variance, leading to a sub-optimal adaptation. Due to the diverse size and appearance of instances, the within-class variance [4] contains the necessary information to model distribution inherently, e.g., the scale and shape [16], [23], which is critical for domain adaptation and generalization. Overlooking the within-class variance brings about many non-adapted

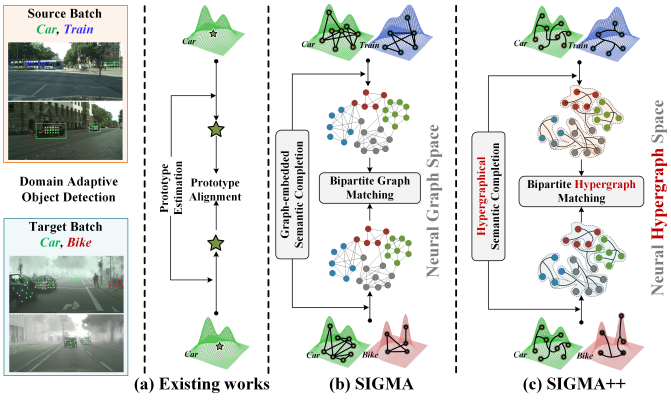


Fig. 1. Illustration of the category-level DAOD paradigms. (a) Existing works [9], [46], [70], [84], [92] first estimate class prototypes (marked as stars) to model class conditional distribution, and then align prototypes to adapt the cross-domain distribution. (b) Our previous work, SIGMA [47], leverages structural graphs to model distribution and conducts node-to-node graph matching to align distribution in a fine-grained manner. (c) The proposed SIGMA++ models and aligns distribution in the more representative hypergraph space, reducing the redundant and ineffective low-order edges for better adaptation.

- *This work was supported by Hong Kong Research Grants Council (RGC) General Research Fund 11211221. (Corresponding author: Yixuan Yuan)*
- *W. Li is with the Department of Electrical Engineering, City University of Hong Kong. X. Liu and Y. Yuan are with the Department of Electronic Engineering, The Chinese University of Hong Kong. (E-mail: wuyangli2-c@my.cityu.edu.hk; jeffreylxy97@gmail.com; yxyuan@ee.cuhk.edu.hk)*

objects, yielding the potential overlapping of different class conditional distributions with false-positive detection cases. Although some works [9], [47], [84] have introduced graph structure to improve prototype-based measurements, they only consider the second-order edge (each edge connects two nodes) to model the semantic dependency, as shown in Fig. 1(b). The redundant and low-order edges lead to an ineffective distribution estimation with inevitable optimization difficulty [8] and sub-optimal adaptation[1]. Hence, these observations motivate us to seek a more effective graph structure, hypergraph, to model and adapt the distribution, as shown in Fig. 1(c). With the measurable high-order dependency among several graph nodes, the hypergraph model captures the inherent high-order relationships among objects for better adaptation, significantly reducing the optimization difficulty on redundant low-order edges.

The second challenge lies in the domain-mismatched semantics within the training batch. Most existing approaches [9], [70], [92] only adapt the co-occurred classes in two domains, ignoring mismatched classes appearing in a single domain. Neglecting missing classes leads to a non-effective adaptation due to the loss of semantic knowledge. Moreover, the missing semantics in the target domain even result in the potential risk toward source-specific direction since the supervised source classification could inherently generate a source-biased distribution [77]. Our preliminary work [47] relieves this problem with a node completion strategy, which generates hallucination nodes [89] in the missing classes to ensure the matched and unbiased adaptation signals. As the node completion highly relies on the well-modeled graph embedding, it can be improved and characterized better in a more representative graph space, e.g., with measurable high-order dependencies and scalable structure. Hence, we aim to enhance node completion with hypergraph, exploring more effective message propagation with hypergraph learning. Then, hallucination nodes could be extracted in the informative hypergraph space to adapt class conditional distribution effectively.

To overcome the aforementioned challenges, we propose an improved SemantIc-complete Graph MAtching framework, dubbed SIGMA++, for DAOD, which completes mismatched semantics with hypergraph learning and reformulates the adaptation with hypergraph matching, i.e., discovering the suitable matching between cross-domain graphs to bridge the domain gap. Specifically, we design a Hypergraphical Semantic Completion (HSC) module to complete the mismatched semantics, which utilizes domain-level statistics to generate hallucination nodes in the missing classes. Then, we establish a hypergraph to model class conditional distribution in each domain, and conduct hypergraph convolution to learn a graph-guided memory bank for better semantic completion in turn. Based on the hypergraphs with inherent distribution knowledge, we propose a Bipartite Hypergraph Matching (BHM) module to solve the graph matching between the source and target graph, achieving a fine-grained domain alignment. We utilize graph nodes to learn semantic-aware node affinity and introduce graph edges in a structure-aware matching loss for the Quadratic Assignment Problem (QAP) [54]. This matching-based domain alignment enables a fine-grained adaptation with well-matched semantics and relieves the biased adaptation in existing prototype-based methods. To be summarized, our contributions are as follows.

- We propose an improved SemantIc-complete Graph MAtching (SIGMA++) framework for DAOD, which aligns class conditional distribution with hypergraph matching. To the best of our knowledge, this work represents the first attempt to leverage hypergraph matching to bridge the domain gap.
- We design a Hypergraphical Semantic Completion (HSC) module to complete mismatched semantics with hypergraph learning. It models the distribution with high-order dependencies and generates hallucination nodes in the mismatched classes.
- We achieve a fine-grained adaptation in the Bipartite Hypergraph Matching (BHM) module, which reformulates domain adaptation in hypergraph space and solves it with high-order hypergraph matching.
- The proposed SIGMA++ achieves state-of-the-art performance on nine benchmarks. Moreover, its implementation of various detectors demonstrates satisfactory applicability on different detectors.

Compared with our preliminary work [47] published in CVPR 2022, we have made numerous extensions with the following contributions. (1) The original Graph-embedded Semantic Completion module [47] is extended to the hypergraph space, termed HSC, for better semantic completion, capturing inherent high-order dependencies with hypergraph learning. (2) BHM module is proposed to leverage the high-order structural dependency between cross-domain hypergraphs for more effective domain adaptation. (3) The benchmark comparison of nine different adaptation settings is introduced to thoroughly analyze our algorithm, covering almost all standard settings in existing DAOD literature. (4) Tons of sensitivity analyses are given to verify the effectiveness and explore more insights regarding the algorithm design. (5) We implement our method on different object detectors to explore its transferability, verifying its good applicability for varied detection pipelines.

2 Related Work

2.1 Domain Adaptive Object Detection

Domain adaptive object detection (DAOD) leverages a labeled source domain and an unlabeled target domain to train robust object detectors [8], [52], [59], [60], [71] applicable for the target domain [12], [13], [35], [58], [76]. Numerous efforts have been conducted, generally categorizing into input-level [11], [20], [32], [36], [41], [48], [58], [62], [91], output-level [20], [36], [56], [58], [62], model-level [20], [32], [48], and feature-level [12], [13], [33], [35], [38], [46], [47], [61], [65], [73], [76], [84], [90], [92], [93], [95] approaches. Input-level methods leverage data augmentation strategies [20], [32], [48] and extra style-transferring networks [94] to generate interpolated data [11], [91] on the original image space for better adaptation. Output-level works are interested in the pseudo labels and conduct a self-training [36], [56], [58]. Model-level methods [20], [32],

1. The second-order dependencies are not enough [24] to model the complex semantic relationships in the real-world scenarios.

[48] rely on an extra teacher model to conduct distillation-based learning with augmented image samples. Kindly note this trend of works should be compared separately with the other single-model counterparts since they need two object detection models.

As one of the main research streams, feature-level approaches leverage adversarial learning [72] and explicit metric learning [29] for adaptation. Some works focus on the feature distribution from a spatial perspective, aligning the global feature hierarchically [12], [13], [21], [33], [63], [65], with spatial attention [35], [44] and region proposals [13], [90], [95]. Moreover, some works delve into class conditional distribution to achieve a category-level adaptation. These works tend to estimate and align class prototypes within the training batch [9], [84], and domain context [46], [92]. However, existing works model the distribution with limited class prototypes and align prototypes with explicit metric learning, which are sub-optimal for aligning the non-convex deep feature distribution. In this work, we model class conditional distribution with hypergraph and align the distribution with hypergraph matching in a fine-grained manner, preserving and aligning the deep feature with high-order semantic dependencies for better model generalization.

2.2 Graph Learning

The graph model can capture the second-order dependency by modeling node-to-node edge connections and building a structural embedding space with geometric knowledge. With the superior capacity in representing non-euclidean graph data, neural graph learning [42] could establish layer-wise knowledge propagation among adjacent nodes, and achieve feature aggregation with graph convolution. Beyond the application with graph data, graph learning has been widely extended in 1) natural language processing [80] for natural language generation [5], question answering [68], and semantic parsing [37], etc., and 2) computer vision of image classification [15], [88], semantic segmentation [74], object detection [45], [46], [47], and scene graph generation [87], etc. In these downstream tasks, the learnable structural edges are established to enhance the structural feature space without explicit edge annotations.

Hypergraph Learning. Going beyond the pair-wise second-order dependency, neural hypergraphs set up hyperedges to group a set of nodes and learn higher-order dependencies with hypergraph convolution [24]. Hypergraph has unique advantages in modeling more complex correlations, serving for various downstream tasks, e.g., multi-modal learning [24], [40], trajectory prediction [82], and deep metric learning [50]. Instead of following the main tread of hypergraph learning and focusing on the feature fusion [40], [50] within a single domain, we leverage feature-point-based hypergraph to model structural cross-domain distribution and generate hallucination nodes to complete the mismatched semantics for domain adaptation.

2.3 Graph Matching

Different from graph learning focusing on an individual graph, graph matching explores the one-to-one matching of graph nodes belonging to different graph entities, which learns the second-order correspondence of matched node-pairs [75]. As a Quadratic Assignment Problem (QAP) [54] with combinational nature, graph matching solvers [54], [86] optimize a cross-graph permutation matrix to encode matched node pairs, considering both node and edge affinities. Recently, graph matching has been extended to visual correspondence detection [27], multi-object tracking [30], point cloud registration [25] and transfer learning [18] to model pair-wise relationships in the graph space. Gao, *et al*. [27] model key-point-based graphs on images and establish graph matching between images covering the same objects. Fu *et al*. [25] model graphs on the 3D rigid point cloud and perform graph matching on two homogeneous point sets to achieve robust point cloud registration. The authors in [30] perform graph matching across the tracklet and detection space to achieve high-quality object tracking.

Hypergraph Matching. Going beyond the graph matching with second-order affinity, hypergraph matching delves into the more complex correlation of hypergraph entities, from the third-order [43], [75], [85] to even higher order [57] correspondences with hyperedges. The advantages lie in the more robust matching with hyperedge affinity learning, aligning inherent dependencies in the high-order structural space. Recently, hypergraph matching has been used and solved in a learnable manner [49] for better visual correspondence detection [75]. In this work, we are the first to model class conditional distribution with hypergraph and achieve the domain adaptation by solving the hypergraph matching problem, which captures the high-order semantic dependency of variant objects across domains.

3 Motivation and Preliminaries

In domain adaptive object detection, we have the batch-wise labeled source $\mathcal{S} = \{(x_s^i, y_s^i)\}_{i=1}^B$ and unlabeled target data samples $\mathcal{T} = \{x_t^i\}_{i=1}^B$ drawn from the inconsistent domain distribution P_s and P_t ($P_s \neq P_t$), respectively. The two domains share the category space with the same class set $\Omega = \{0, 1, ..., |\Omega|\}$, and the randomly sampled image batches $\mathcal{S}$ and $\mathcal{T}$ always contain mismatched classes $\Omega_{s/t}^B \subset \Omega$ of the observable object instances, i.e., $\Omega_s^B \neq \Omega_t^B$. Existing works [70], [84], [92] aim to model and align the class conditional distribution $P_{X|Y}(\phi(x_{s/t})|y)$ using a prototype-based paradigm, where $\phi(\cdot)$ is the feature extractor. They first estimate the class centers $\mu_{s/t}^y = \mathbb{E}_{X|Y}[\phi(x)|y]$ with handcraft priors, e.g., mean features of appeared Region of Interests (RoI): $\mu_{s/t}^y = \frac{1}{N_{s/t}} \sum_i^{N_{s/t}} \text{RoI}_i^y$, and then minimize the domain-discrepancy between μ_s^y and μ_t^y to bridge the domain gap. However, these methods potentially achieve a biased adaptation depending only on center-based knowledge, and fail to adapt mismatched classes $\Omega_{s/t}^{miss}$ appearing in a single domain due to the intractable $\mu_{s/t}^{y=\Omega_{s/t}^{miss}}$.

To address these issues, we generate novel samples in the missing classes $\Omega_{s/t}^{miss}$ to complete the mismatched semantic and establish a cross-image graph $\mathcal{G}_{s/t}$[2] to model the class

2. The cross-image graph $\mathcal{G}_{s/t}$ consists of the second-order common graph $\mathcal{G}_{s/t}^c$ in our previous work [47] and high-order hypergraph $\mathcal{G}_{s/t}^h$ in this extended version. We follow [75] to use the common graph for theoretical analysis without losing the extensibility.

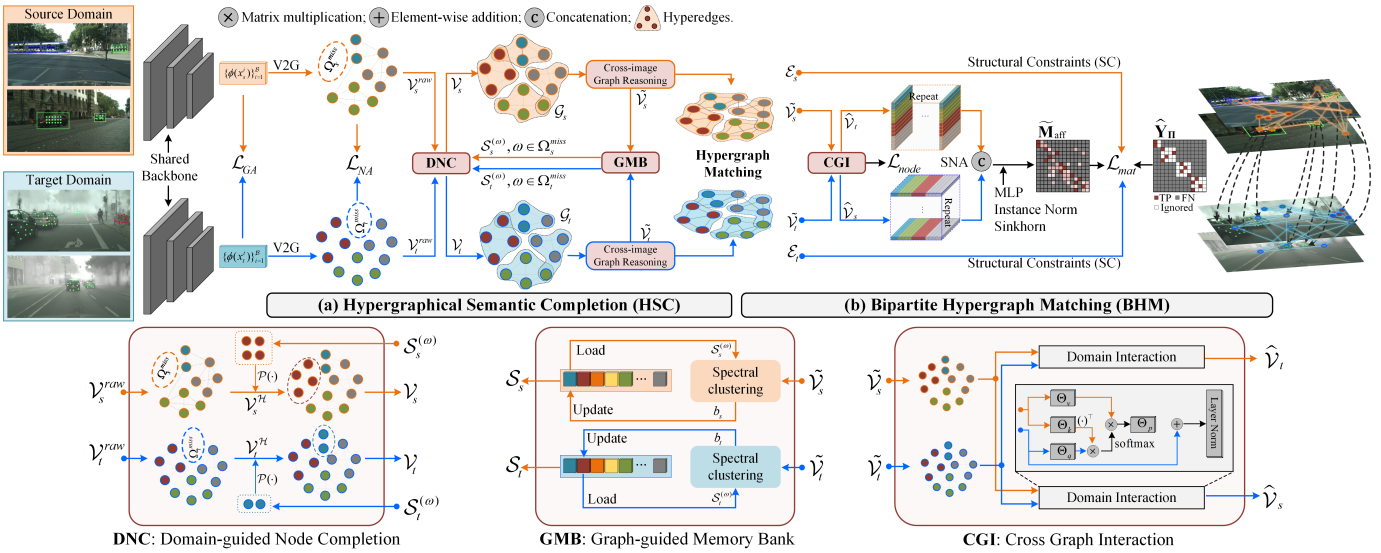


Fig. 2. Overview of the proposed SIGMA++ framework for DAOD. V2G represents vision-to-graph transformation.

conditional distribution $P_{X|Y}(\phi(x_{s/t})|y)$ for each domain. Then, we reformulate domain adaptation as a graph matching problem between $\mathcal{G}_s$ and $\mathcal{G}_t$, which can be solved with a differential QAP [25], [27], [30] as follows,

$$\begin{aligned} &\min_{\mathbf{\Pi}} \mathcal{F}(\mathbf{\Pi}) = ||\mathbf{A}_s - \mathbf{\Pi}\mathbf{A}_t\mathbf{\Pi}^\top||_{\mathrm{F}}^2 - tr(\mathbf{X}_u^\top \mathbf{\Pi}), \\ &\mathbf{\Pi} \in [0,1]^{N_s \times N_t}, \mathbf{\Pi}\mathbf{1}_{N_s} \leq \mathbf{1}_{N_t}, \mathbf{\Pi}^\top \mathbf{1}_{N_t} \leq \mathbf{1}_{N_s}, \end{aligned} \tag{1}$$

where $\mathbf{A}_s \in \mathbb{R}^{N_s \times N_s}$ and $\mathbf{A}_t \in \mathbb{R}^{N_t \times N_t}$ is the adjacent matrix encoding structure information of the graph $\mathcal{G}_s$ and $\mathcal{G}_t$ respectively, $N_{s/t}$ is the number of graph nodes, $||\cdot||_{\mathrm{F}}$ is the Frobenius norm, $\mathbf{X}_u \in \mathbb{R}^{N_t \times N_s}$ is the unary affinity matrix and generally specified as the node affinity $\mathbf{M}_{\text{aff}}$ [27], and $\mathbf{\Pi}$ is the relaxed permutation matrix encoding node-to-node assignment ($\mathbf{\Pi}_{i,j} = 1$ indicates that the node $v_s^i \in \mathcal{G}_s$ is matched with the node $v_t^j \in \mathcal{G}_t$). We follow [27] to relax the one-hot $\mathbf{\Pi}$ with continuous values to meet the differential requirement of model training.

Different from existing works [84], [90], [92] overlooking mismatched classes, we complete missing semantics and effectively align the distribution for each appeared class. Besides, our method achieves a fine-grained adaptation guided by graph matching, breaking the barrier of existing center-based methods adopting sub-optimal alignment.

4 Proposed Method

The overall workflow of the proposed SIGMA++ framework is shown in Fig. 2. Given batch-wise annotated source images $\{(x_s^i, y_s^i)\}_{i=1}^B$ and unlabeled target images $\{x_t^i\}_{i=1}^B$, we use a shared feature extractor $\phi(\cdot)$ to extract image-level features $\{\phi(x_{s/t}^i)\}_{i=1}^B$, which are sent to Hypergraphical Semantic Completion (HSC) module (Fig. 2(a)). In the proposed HSC module, we first transform visual features to the graph space and perform domain-guided node completion to complete mismatched semantics, obtaining semantic-complete node sets $\mathcal{V}_{s/t}$. Then, we establish cross-image graphs $\mathcal{G}_{s/t}$ to model class conditional distribution with enhanced nodes $\tilde{\mathcal{V}}_{s/t}$, which also serve to update a graph-guided memory bank to support the semantic completion in turn. Afterward, the well-modeled graphs $\mathcal{G}_{s/t}$ are sent to the Bipartite Hypergraph Matching (BHM) module (Fig. 2(b)). We use graph nodes $\tilde{\mathcal{V}}_{s/t}$ for cross-domain graph interaction and learn a semantic-aware node affinity matrix $\tilde{\mathbf{M}}_{\text{aff}}$. Besides, we leverage graph edges $\mathcal{E}_{s/t}$ to serve as structural constraints to optimize the graph matching permutation, achieving fine-grained adaptation with well-aligned graph entities.

4.1 Hypergraphical Semantic Completion

Given batch-wise labeled source images $\{(x_s^i, y_s^i)\}_{i=1}^B$ and unlabeled target images $\{x_t^i\}_{i=1}^B$ drawn from the $|\Omega|$-class category space, we first adopt a shared backbone $\phi(\cdot)$ to extract image-level visual features $\{\phi(x_{s/t}^i)\}_{i=1}^B$, $\phi(x_{s/t}^i) \in \mathbb{R}^{D \times H \times W}$, and transform them to the graph space with sampling and projection. In the source domain, we perform spatial-uniformed sampling to collect the pixels inside ground-truth boxes as class-aware foreground feature points. Then, a ratio $\frac{1}{|\Omega|+1}$ of the points outside foreground boxes are sampled as backgrounds to capture the inherent style information hidden in the informative scene. For the target domain, we forward-propagate target features in the detection head $\hat{\phi}(\cdot)$ to obtain classification score maps $\mathcal{M}_t \in \mathbb{R}^{|\Omega| \times H \times W}$ as the surrogate sampling principle. Similarly, the pixels satisfying $\max_{|\Omega|}(\mathcal{M}_t^i) > \tau_{fg}$ are sampled as class-aware foreground points, and a ratio $\frac{1}{|\Omega|+1}$ of low-score pixels ($\max_{|\Omega|}(\mathcal{M}_t^i) < \tau_{bg}$) are treated as backgrounds.

After sampling fine-grained visual feature points, we conduct a linear projection followed by Layer Normalization [3] to obtain raw graph nodes $\mathcal{V}_{s/t}^{raw} = \{v_{s/t}^i\}_{i=1}^{N_{s/t}}$. This projection extends the activated visual feature space with positive numerical distribution $[0, +\infty)$ to the embedding space $(-\infty, +\infty)$ for better graphical representation.

4.1.1 Domain-guided Node Completion

In DAOD, with a randomly sampled source and target image batch, the object classes $\Omega^B_{s/t} \subset \{0, 1, ..., |\Omega|\}$ within a training batch are always mismatched between the source and the target domain ($\Omega^B_s \neq \Omega^B_t$), limiting the adaptation of class conditional distributions. Hence, we propose a semantic completion strategy to generate hallucination nodes in missing classes $\Omega^{miss}_s = \{\omega | \omega \in \Omega^B_t, \omega \notin \Omega^B_s\}$, $\Omega^{miss}_t = \{\omega | \omega \in \Omega^B_s, \omega \notin \Omega^B_t\}$, obtaining semantic-complete nodes $\mathcal{V}_{s/t}$. To generate additional nodes containing non-existing semantics, we define a graph-guided memory bank $\mathcal{S}_{s/t} \in \mathbb{R}^{C\times D}$ to save the classes-specific knowledge of inner-domain semantics, and we will explain the learning strategy of this memory bank in Sec. 4.1.3. Considering the source and target domains share a similar category space [13], we utilize the semantic cues from the counterpart domain to guide the node generation, providing a joint measurement of the cross-domain distribution.

Formally, to complete the source-missing class $\omega \in \Omega^{miss}_s$, we first calculate the variance of target-domain nodes in class ω to obtain a variant vector $\sigma^{(\omega)}_t \in \mathbb{R}^D$, approximating the distribution scale for the missing class ω at the current training stage. Then, the memory seed $\mathcal{S}^{(\omega)}_s$ is loaded from the memory bank as the class-specific expectation $\mu^{(\omega)}_s$. Based on $\mu^{(\omega)}_s$ and $\sigma^{(\omega)}_t$, we conduct Gaussian sampling followed with a linear projection $\mathcal{P}(\cdot)$ to obtain the hallucination nodes in missing classes:

$$\mathcal{V}^{\mathcal{H}}_s = \{v^h_s | v^h_s = \mathcal{P}(x^h_s), x^h_s \sim Gaussian(\mu^{(\omega)}_s, \sigma^{(\omega)}_t)\}. \quad (2)$$

The same completion is also conducted in the target domain to obtain the nodes $\mathcal{V}^{\mathcal{H}}_t$ in the target-missing classes Ω^{miss}_t. Instead of aligning these statistic-based estimations directly [84], [90], [92], we fully utilize domain knowledge to generate novel and unbiased samples, avoiding the biased and sub-optimal alignment. Finally, both existing nodes and hallucination ones constitute the semantic-complete node set $\mathcal{V}_{s/t}$ for the following graph modeling.

4.1.2 Cross-image Graph Reasoning

As the fine-grained nodes $\mathcal{V}_{s/t}$ contain rich knowledge of class conditional distribution, we establish a graph $\mathcal{G}_{s/t}$ in each domain to model this inherent knowledge by capturing cross-image semantic dependencies. Our previous work [47] establishes a common graph $\mathcal{G}^c_{s/t}$ by modeling the node-to-node second-order dependencies. Then, the distribution could be represented with this cross-image structural model for better domain adaptation.

Hypergraph Learning Since the second-order graph $\mathcal{G}^c_{s/t}$ cannot model robust and abundant semantic relationships in the real-world scenarios [24], we extend it to the hypergraph $\mathcal{G}^h_{s/t}$ to capture high-order correlations, which serves as a critical role in modeling class conditional distribution. The hypergraph $\mathcal{G}^h_{s/t}$ is formulated with three variables, i.e., the graph node set $\mathcal{V}_{s/t}$, hyperedge set $\mathcal{E}^h_{s/t} = \{e_i | e_i = \{v_j\}^K_{j=1}\}^{|\mathcal{E}|}_{i=1}$, and a diagonal weight matrix $\mathbf{W} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ with the diagonal entries $\mathbf{W}(e) \in [0, 1]$ as edge weight [24]. Formally, the hyperedge is encoded in an incidence matrix $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{E}|}$, where the entry $\mathbf{H}(v, e) = 1$ indicates the node $v \in e$ and $\mathbf{H}(v, e) = 0$ indicates $v \notin e$. Based on $\mathbf{H}$, two kinds of degree matrices in a diagonal format could be justified for effective hypergraph learning: 1) the hyperedge degree matrix $\mathbf{D}_e \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ where each diagonal entry $\mathbf{D}_e(e) = \Sigma_{v\in\mathcal{V}}\mathbf{W}(e)\mathbf{H}(v, e)$ represents the degree of hyperedge $e \in \mathcal{E}$, and 2) the node degree matrix $\mathbf{D}_v \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ formulated as $\mathbf{D}_v(v) = \Sigma_{e\in\mathcal{E}}\mathbf{W}(e)\mathbf{H}(v, e)$. The degree matrix encodes the connecting frequency on each node/hyperedge for effective hypergraph reasoning.

To build the hypergraph with hyperedges $\mathbf{H}$, we leverage the distance [24] among the node embedding and consider K Nearest Neighbors (KNN). Specifically, for each node, we calculate the Euclidean distance with other nodes and then connect it with its $K - 1$ nearest neighbors as a hyperedge. Based on this, $|\mathcal{V}|$ hyperedges in K^{th} order are obtained. For the edge weight matrix $\mathbf{W}$, we allocate the same weight $\mathbf{W}(e) = 1$ to prevent the biased learning of different classes. Based on the justified hypergraph $\mathcal{G}^h_{s/t}$, we conduct a single-layer hypergraph convolution [24] to propagate the cross-image information as follows,

$$\tilde{\mathbf{V}} = \mathbf{D}^{-1/2}_v \mathbf{H}\mathbf{W}\mathbf{D}^{-1}_e \mathbf{H}^\top \mathbf{D}^{-1/2}_v \mathbf{V}\Theta_h, \quad (3)$$

where Θ_h is learnable for hypergraph reasoning. Hence, the high-order dependencies are well modeled, yielding a more precise distribution estimation. Moreover, compared with optimizing $|\mathcal{V}| \times |\mathcal{V}|$ second-order connections [47], hypergraph significantly reduces redundant edges by modeling $|\mathcal{V}|$ hyperedges, capturing K^{th}-order dependencies with scalable hypergraph structure.

4.1.3 Graph-guided Memory Bank

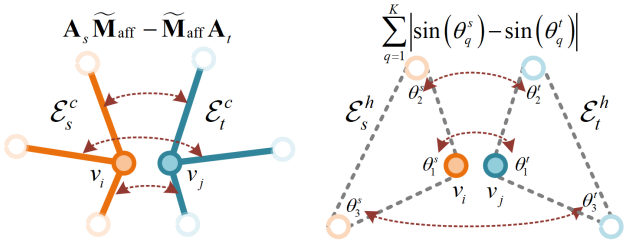
To preserve the necessary cues for hallucination node generation, we propose a memory bank to save class-specific graph embedding and design a cluster-based update strategy. Specifically, we randomly initialize a memory bank $\mathcal{S}_{s/t} \in \mathbb{R}^{|\Omega|\times D}$ at the training start and gradually update memory seeds with appeared graph nodes. For each appeared class ω within a training batch, we collect nodes $\{\tilde{v}^{(\omega)}_{s/t}\}, \tilde{v}^{(\omega)}_{s/t} \in \mathbb{R}^D$ in class ω and load the memory seed $\mathcal{S}^{(\omega)}_{s/t} \in \mathbb{R}^D$ from the memory bank $\mathcal{S}_{s/t}$. Then, we get them together $\{\mathcal{S}^{(\omega)}_{s/t}, \tilde{v}^{(\omega)}_{s/t}\}$ and conduct spectral clustering [69] in the graph space to generate two clusters, i.e., a seed-included cluster $\pi^{seed}_{s/t} = \{\mathcal{S}^{(\omega)}_{s/t}, \tilde{v}^{(\omega)}_{s/t}\}$ and an "else" cluster $\pi^{else} = \{\tilde{v}^{(\omega)}_{s/t}\}$. Since the domain-level knowledge provides a more robust estimation than batch-wise observation, we only use the nodes in $\pi^{seed}_{s/t}$ to update, which relieves the impact of noisy nodes in the early training stage:

$$\mathcal{S}^{(\omega)}_{s/t} \leftarrow sim(b_{s/t}, \mathcal{S}^{(\omega)}_{s/t})\mathcal{S}^{(\omega)}_{s/t} + [1 - sim(b_{s/t}, \mathcal{S}^{(\omega)}_{s/t})]b_{s/t}, \quad (4)$$

where $sim(b_{s/t}, \mathcal{S}^{(\omega)}_{s/t}) = \frac{b_{s/t}\cdot\mathcal{S}^{(\omega)}_{s/t}}{\|b_{s/t}\|_2 \cdot \|\mathcal{S}^{(\omega)}_{s/t}\|_2}$ indicates the adaptive momentum for better gradient-free learning [73], [92], and $b_{s/t} = \frac{1}{|\pi^{seed}_{s/t}|-1}\sum_{\tilde{v}^{(\omega)}_{s/t}\in\pi^{seed}_{s/t}} \tilde{v}^{(\omega)}_{s/t}$. Note that the hallucination nodes will not participate in this update to avoid the potential negative impact of handcraft Gaussian priors.

4.2 Bipartite Hypergraph Matching

Based on the well-modeled graphs $\mathcal{G}_{s/t}$ in two domains, we reformulate the domain alignment as a graph matching



(a) Second-order graph matching with **THREE edges.** (b) Third-order (K=3) hypergraph matching with **ONE hyperedge.**

Fig. 3. Illustration of the (a) second-order graph matching [47] with three edges and (b) third-order hypergraph matching with one hyperedge (the hyperedge connects three nodes), between the source ($\mathcal{E}_s$ in orange) and target ($\mathcal{E}_t$ in blue) edges. The dark nodes indicate the matched ones, and the nodes in light color represent their neighbor nodes formulated with graph edges.

problem, i.e., solving the QAP between $\mathcal{G}_s$ and $\mathcal{G}_t$ for fine-grained adaptation. Specifically, we use graph nodes $\tilde{\mathcal{V}}_{s/t}$ to establish cross-domain graph interaction and learn a node affinity $\tilde{\mathbf{M}}_{\text{aff}}$. Besides, graph edges $\mathcal{E}_{s/t}$ are utilized to bridge the domain gap with a structure-aware matching loss.

4.2.1 Cross-domain Graph Interaction

Since graph matching is a collaborative optimization problem between two graphs, message propagation across graphs is essential for the optimal solution in affinity learning. Hence, we introduce the knowledge exchange between $\mathcal{G}_s$ and $\mathcal{G}_t$ with cross-domain semantic interaction:

$$\begin{aligned}\hat{\mathbf{V}}_s &= \text{Norm}\{\text{softmax}[(\tilde{\mathbf{V}}_s\Theta_q)(\tilde{\mathbf{V}}_t\Theta_k)^\top](\tilde{\mathbf{V}}_t\Theta_v)\Theta_p + \tilde{\mathbf{V}}_s\},\\ \hat{\mathbf{V}}_t &= \text{Norm}\{\text{softmax}[(\tilde{\mathbf{V}}_t\Theta_q)(\tilde{\mathbf{V}}_s\Theta_k)^\top](\tilde{\mathbf{V}}_s\Theta_v)\Theta_p + \tilde{\mathbf{V}}_t\},\end{aligned} \tag{5}$$

where $\hat{\mathbf{V}}_{s/t}$ is the formatted node embedding matrix from the node set $\hat{\mathcal{V}}_{s/t} = \{\hat{v}_{s/t}^i\}_{i=1}^{N_{s/t}}$ with cross-domain perception, Norm is Layer Normalization [3], and $\Theta_{(\cdot)}$ are learnable parameters. To enhance the graphical semantics, we introduce an auxiliary node classification task by adopting a classifier f_{cls} with the Cross-Entropy loss:

$$\mathcal{L}_{node} = -\sum_{i=1}^{N_s+N_t} \hat{y}_i log\{softmax[f_{cls}(\hat{v}_{s/t}^i)]\}, \tag{6}$$

where $\hat{y}_i$ is the ground-truth labels for source nodes and the pseudo labels (obtained from score maps $\mathcal{M}_t$) for target nodes. Dense relationships with an inherent node matching [75] can be established, serving for the sparse and fine-grained adaptation with interactive semantic cues.

4.2.2 Semantic-aware Node Affinity

Given the graph nodes $\hat{\mathcal{V}}_{s/t}$ with cross-domain perception, we further learn an affinity matrix to model the node correspondence between $\mathcal{G}_s$ and $\mathcal{G}_t$. Different from existing graph matching approaches [25], [27], [30] using local visual representations, we leverage the category-level semantic with inherent relationships to learn a semantic-aware affinity matrix. Specifically, we define the entry of the node affinity matrix as follows: $\mathbf{M}_{\text{aff}}^{i,j} = f_{mlp}\{f_p(\hat{v}_s^i)\ \copyright\ f_p(\hat{v}_t^j)\}$, $\mathbf{M}_{\text{aff}} \in \mathbb{R}^{N_s\times N_t}$, where © is the concatenation operation, f_p indicates a linear projection and f_{mlp} is a multi-layer perceptron layer (MLP) with a single output channel. This MLP layer learns inherent semantic relationships between two graph nodes and encodes them into affinity representations. $\mathbf{M}_{\text{aff}}$ is then sent to the Instance Normalization layer as [25] and the differential Sinkhorn layer [67] to obtain a double-stochastic affinity matrix $\tilde{\mathbf{M}}_{\text{aff}}$ with maximum 20 iterations [25]. Finally, each positive entry in the affinity matrix $\tilde{\mathbf{M}}_{\text{aff}}$ indicates a matched node pair across two graphs for fine-grained domain adaptation.

4.2.3 Structure-aware Matching Loss

Since the well-modeled graphs contain inherent distribution knowledge, we propose a structure-aware matching loss to adapt the graphs in different domains ($\mathcal{G}_s$ and $\mathcal{G}_t$). This loss function establishes the inherent matching of the cross-domain graph entities, giving an effective distribution alignment between the nodes with homogeneous semantics. Based on the graph matching objective (Eq. 1), our previous work [47] relies on the node-to-node adjacent matrix $\mathbf{A}_{s/t}$ to optimize the graph matching of the second-order common graph $\mathcal{G}_{s/t}^c$, minimizing the neighbor difference of matched nodes, as shown in Fig. 3(a).

Hypergraph Matching. As the hypergraph models diverse high-order dependencies inherently, it plays a critical role in representing and aligning the cross-domain distribution, adapting the high-order knowledge within hyperedges. Hence, we generalize the original second-order graph matching [47] to the high-order hypergraph space, adapting the class conditional distribution thoroughly and robustly [75]. Formally, this hypergraph-based matching objective is formulated as follows,

$$\begin{aligned}\mathcal{L}_{mat} = &\sum_i \frac{1}{N_s}[\max_j(\tilde{\mathbf{M}}_{\text{aff}} \odot \mathbf{Y}_{\mathbf{\Pi}})_{i,j} - \mathbf{1}]^2\\ &+ \sum_{i,j} \frac{1}{||\mathbf{1} - \mathbf{Y}_{\mathbf{\Pi}}||_1}[\tilde{\mathbf{M}}_{\text{aff}} \odot (\mathbf{1} - \mathbf{Y}_{\mathbf{\Pi}})]_{i,j}^2\\ &+ \frac{1}{N_m}\sum_{e\in\mathcal{E}^h}[\exp(-\sum_{\substack{q=1\\ \theta\in e}}^{K}|sin(\theta_q^s) - sin(\theta_q^t)|)/\epsilon],\end{aligned} \tag{7}$$

where the (i,j) entry in $\mathbf{Y}_{\mathbf{\Pi}} \in \mathbb{R}^{N_s\times N_t}$ is $\mathbf{1}$ if $v_s^i \in \mathcal{G}_s$ and $v_t^j \in \mathcal{G}_t$ are in the same class ω, otherwise $\mathbf{0}$, $N_s = |\hat{\mathcal{V}}_s|$ and N_m=$min(|\mathcal{E}_s|, |\mathcal{E}_t|)$ are used for loss reduction, θ_q is the angle of two nodes within a hyperedge (as in Fig. 3(b)), and ϵ is a scaling factor set 0.001 as [75]. The first item focuses on correctly matched node pairs and enhances the best-matching of true-positive cases, while the second term suppresses wrongly activated false-positive cases. These two items work together and use the node-level knowledge to ensure the basic learning [75] of graph matching.

The third item leverages edge connections to establish the high-order matching between hypergraphs, which matches the structural dependencies embedded in the hypergraph as a structural constraint [27]. We present a third-order example with $K = 3$ in Fig. 3(b). Specifically, we first use the node affinity $\tilde{\mathbf{M}}_{\text{aff}}$ to obtain a coarse matching with node-level correspondence. Then, given each coarsely matched node-pair, i.e., a source node $v_i \in \mathcal{G}_s^h$ and a target counterpart $v_j \in \mathcal{G}_t^h$, the hyperedges $e_s \in \mathcal{E}_s^h$ and $e_t \in \mathcal{E}_t^h$ could be justified, modeling its $K-1$ neighbor

nodes, respectively. As shown in Fig. 3(b), we leverage inner angles θ_q to represent hyperedges and minimize the difference of cross-domain hyperedges as [75] for hypergraph matching. However, representing and matching each hyperedge require permuting K^2 pairs of angles, leading to the failure [75] in generalizing to the higher orders due to the optimization difficulty. To relieve this problem, we sort the Euclidean distance between the matched node and the other nodes within the hyperedge and only measure the angles of adjacent nodes instead of every two nodes to represent the hyperedge. This modification leverages the inherent KNN-based geometric priors within hyperedges for efficient affinity learning.

With the hypergraph matching objective, we break through the limitation of the second-order graph matching in our previous work [47] by adapting high-order structural knowledge, which preserves scaling and rotation invariance [75] of the fully-adapted distribution. Moreover, going beyond third-order hypergraph matching [75], our method could be applied for the matching with higher orders, fully using the inherent geometric prior in the hyperedge space.

4.3 Model Optimization

4.3.1 Loss Function

During the training period of SIGMA++, we preserve the vanilla image-level global alignment [35] as the baseline, which inputs visual features $\{x^i_{s/t}\}^B_{i=1}$ and generates adversarial loss $\mathcal{L}_{GA}$ at the image level. Moreover, we introduce Node Alignment (NA) to align the marginal distribution of the well-matched nodes to prevent potential semantic bias. Specifically, this module ($f_{NA}(\cdot)$) contains a gradient reversed layer [26], three stacked discrimination blocks (FC-LayerNrom-ReLU), and a domain classification loss:

$$\mathcal{L}_{NA} = -\sum_{i}^{N_s} \mathcal{D} log[f_{NA}(v^i_s)] - \sum_{i}^{N_t} (1-\mathcal{D}) log[f_{NA}(v^i_t)], \tag{8}$$

where $\mathcal{D}$ is the domain label as [13] and $v^i_{s/t}$ are existing graph nodes. Then, the overall loss function is denoted as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{node} + \lambda_2 \mathcal{L}_{mat} + \mathcal{L}_{NA} + \mathcal{L}_{GA} + \mathcal{L}_{det}, \tag{9}$$

where $\mathcal{L}_{node}$ is the node classification loss, $\mathcal{L}_{mat}$ is the graph matching loss, $\mathcal{L}_{NA}$ is the node alignment loss, $\mathcal{L}_{GA}$ is the global alignment loss [35] and $\mathcal{L}_{det}$ is the detection loss. $\lambda_{1/2}$ are set to 1.0 and 0.1 respectively to control the loss weight.

4.3.2 Optimization Pipeline

The overall optimization pipeline is shown in Algorithm 1. SIGMA++ leverages source and target images $\{x^i_{s/t}\}^B_{i=1}$, source labels $\{y^i_s\}^B_{i=1}$, and an initialized object detector [71], consisting of a feature extractor $\phi(\cdot)$ and a detection head $\hat{\phi}(\cdot)$. Then, it uses the overall objective function $\mathcal{L}$ to train the framework with *maxiter* iterations, which outputs a domain adaptive object detector. Kindly note that 2-4 *lines* represent the operations in the visual space, and 5-12 *lines* indicate the strategies adopted in the graph space for adaptation.

Algorithm 1 Semantic-complete Graph Matching

Input:
$\{x^i_{s/t}\}^B_{i=1}$: source and target images
$\{y^i_s\}^B_{i=1}$: source annotations
$\phi(\cdot), \hat{\phi}(\cdot)$: feature extractor and detection head
Output:
Domain adaptive object detector
1: **for** $l = 1$ **to** $maxiter$ **do**
2: extract image-level visual features $\{\phi(x^i_{s/t})\}^B_{i=1}$;
3: generate global alignment loss $\mathcal{L}_{GA}$ on $\{\phi(x_{s/t})\}^B_{i=1}$;
4: generate task-loss $\mathcal{L}_{det}$ and classification score maps $\mathcal{M}_t$ with forward-propagation $\{\hat{\phi}(\phi(x^i_{s/t}))\}^B_{i=1}$;
Hypergraphical Semantic Completion (HSC)
5: perform V2G transformation to obtain nodes $\mathcal{V}^{raw}_{s/t}$;
6: generate the node alignment loss $\mathcal{L}_{NA}$;
7: perform domain-guided node completion for $\mathcal{V}_{s/t}$;
8: conduct cross-image graph reasoning for $\tilde{\mathcal{V}}_{s/t}$;
9: update graph-guided memory bank;
Bipartite Hypergraph Matching (BHM)
10: perform cross-domain graph interaction to obtain $\hat{\mathcal{V}}_{s/t}$ and generate the node classification loss $\mathcal{L}_{node}$;
11: learn semantic-aware affinity matrix $\tilde{\mathbf{M}}_{\text{aff}}$;
12: generate structure-aware matching loss $\mathcal{L}_{mat}$;
Network Parameter Updating
13: update $\phi(\cdot), \hat{\phi}(\cdot)$ with backward-propagation;
14: **end for**
15: **return** Domain adaptive object detector Θ;

5 Experiments

5.1 Experimental Setup

5.1.1 Domain Adaptation Settings

(1) City Landscapes Adaptation. This adaptation scenario focuses on diverse street scenes to push forward the research on self-driving, consisting of five adaptation settings.

Cityscapes→Foggy Cityscapes. Cityscapes [17] dataset contains real-world street scenes under the normal weather condition, which are captured with an onboard camera. The standard data splitting for DAOD contains a *train* set (2975 images) and a *validation* set (500 images), and the bounding boxes are labeled with eight categories. Based on Cityscapes, the authors in [64] introduce heavy foggy noise and generate a synthesized dataset, Foggy Cityscapes [64], which consists of 0.005, 0.01, and 0.02, three levels of fog. We follow the literature by using *the worst condition (0.02)* as the target domain and explore the weather-based domain gap in this normal-to-foggy adaptation scenario.

Sim10k→Cityscapes. Sim10k [39] is a simulated dataset generated from the video game Grand Theft Auto V, which has an inherent domain gap with real-world Cityscapes [17]. Sim10k contains 10,000 images of the labeled bounding boxes in the class *car*. We follow existing works to conduct adaptation under this synthetic-to-real scenario and report the performance of the car class.

KITTI⇆Cityscapes. KITTI [28] is a real-world traffic scene dataset collected from vehicle-mounted cameras, which has the inherent domain gap with the Cityscapes [17] captured from onboard cameras. KITTI contains annotated cars in 7,481 images with a cross-viewpoint domain gap. We follow

TABLE 1
Results (%) on Cityscapes→Foggy Cityscapes with VGG16. For a fair comparison with single-model-based methods, we report the usage of extra models, e.g., the style-transfer model and auxiliary teacher model. Res50 represents the results from a ResNet50 backbone.

Method	Extra model	person	rider	car	truck	bus	train	motor	bike	AP_{50}	Gain	SO	Reference
CFFA [92]		34.0	46.9	52.1	30.8	43.2	29.9	34.7	37.4	38.6	17.8	20.8	CVPR' 20
HTCN [11]	CycleGAN	33.2	47.5	47.9	31.6	47.4	40.9	32.3	37.1	39.8	19.5	20.3	CVPR' 20
GPA (Res50) [84]		32.9	46.7	54.1	24.7	45.7	41.1	32.4	38.7	39.5	16.7	22.8	CVPR' 20
EPM [35]		41.9	38.7	56.7	22.6	41.5	26.8	24.6	35.5	36.0	17.6	18.4	ECCV' 20
RPA [90]		33.6	43.8	49.6	32.9	45.5	46.0	35.7	36.8	40.5	19.7	20.8	CVPR' 21
UMT [20]	Teacher	33.0	46.7	48.6	**34.1**	**56.5**	46.8	30.4	37.4	41.7	19.9	21.8	CVPR' 21
MeGA-CDA [73]		37.7	49.0	52.4	25.4	49.2	46.9	34.5	39.0	41.8	17.4	24.4	CVPR' 21
SSAL [56]		45.1	47.4	59.4	24.5	50.0	25.7	26.0	38.7	39.6	19.2	20.4	NeurIPS' 21
ICCR-VDD [79]		33.4	44.0	51.7	33.9	52.0	34.7	34.2	36.8	40.0	17.2	22.8	ICCV' 21
DSS (Res50) [77]		42.9	**51.2**	53.6	33.6	49.2	18.9	36.2	41.8	40.9	18.1	22.8	CVPR' 21
SDA (Res50) [61]		38.8	45.9	57.2	29.9	50.2	**51.9**	31.9	40.9	43.3	20.5	22.8	ICCV' 21
KTNet [70]		46.4	43.2	60.6	25.8	41.2	40.4	30.7	38.8	40.9	22.5	18.4	ICCV' 21
DBGL [9]		33.5	46.4	49.7	28.2	45.9	39.7	34.8	38.3	39.6	19.3	20.3	ICCV' 21
SCAN [46]		41.7	43.9	57.3	28.7	48.6	48.7	31.0	37.3	42.1	23.7	18.4	AAAI' 22
TDD [32]	Teacher	39.6	47.5	55.7	33.8	47.6	42.1	37.0	41.4	43.1	17.3	25.8	CVPR' 22
TIA [91]	CycleGAN	34.8	46.3	49.7	31.1	52.1	48.6	37.7	38.1	42.3	22.0	20.3	CVPR' 22
MGA [93]		43.9	49.6	60.6	29.6	50.7	39.0	**38.3**	**42.8**	44.3	25.2	18.8	CVPR' 22
FGRR [10]		33.5	46.4	49.7	28.2	45.9	39.7	34.8	38.3	39.6	19.3	20.3	TPAMI' 22
SIGMA [47]		**46.9**	48.4	**63.7**	27.1	50.7	35.9	34.7	41.4	43.5	25.1	18.4	CVPR' 22
SIGMA++		46.4	45.1	61.0	32.1	52.2	44.6	34.8	39.9	**44.5**	**26.1**	18.4	

TABLE 2
Comparison results (%) on Cityscapes→BDD100k with VGG16.

Method	prsn	rider	car	truc	bus	moto	bike	AP_{50}
DAF [13]	26.9	22.1	44.7	17.4	16.7	17.1	18.8	23.4
ICR-CCR [81]	31.4	31.3	46.3	19.5	18.9	17.3	23.8	26.9
SWDA [63]	30.2	29.5	45.7	15.2	18.4	17.1	21.2	25.3
SCDA [95]	29.3	29.2	44.4	20.3	19.6	14.8	23.2	25.8
EPM [35]	39.6	26.8	55.8	18.8	19.1	14.5	20.1	27.8
SFA [76]	40.2	27.6	57.5	19.1	23.4	15.4	19.2	28.9
TDD [32]	39.6	**38.9**	53.9	**24.1**	25.5	**24.5**	**28.8**	33.6
SIGMA [47]	46.9	29.6	64.1	20.2	23.6	17.9	26.3	32.7
SIGMA++	**47.5**	30.4	**65.6**	21.1	**26.3**	17.8	27.1	**33.7**

the literature to explore the dual-directional adaptations between the two datasets to fully verify our algorithm.

Cityscapes→BDD100K. BDD100K [83] is a large-scale city landscapes dataset with vital diversity of scenes. This dataset contains 100K videos from different cities with ten kinds of annotated objects, encompassing various weather conditions, six different scenarios, and three kinds of time of the data. We follow existing works [76], [79] to use the daytime subset with 36,728 for training and 5,258 images for evaluation and report the performance on the seven commonly seen and overlapped classes with Cityscapes.

(2) Medical Scene Adaptation. To evaluate the proposed algorithm in heterogeneous scenes, we explore the cross-clinic domain gap in the consolidated video frames [1].

CVC-ClinicDB→Abnormal Symptoms. CVC-ClinicDB [6] is a dataset in the colonoscopy scene from the Hospital Clinic in Barcelona, Spain, which contains 612 standard definition frames in 384×288 resolution. Abnormal Symptoms in Endoscopic Images dataset [34] (abbr. Abnormal Symptoms) contains endoscopy images from Vestre Viken Hospital Trust in Norway. These two datasets contain annotated bounding boxes in polyp class and have the inherent domain gap of inconsistent medical clinics.

(3) Natural Image Adaptation. This scenario aims to adapt the images with large style differences for generalizing generic object detection, covering three sub-tasks as follows.

Pascal VOC→Clipart/Watercolor/Comic. Pascal VOC [22] is a real-world dataset with annotated commonly seen object instances, which contains 2007 and 2012 two subsets and a total of 20 categories. Following the standard data splitting, we use Pascal VOC 2007 and 2012 *trainval* split with a total of 16,551 images for training. Clipart, Watercolor, and Comic [36] are collected from the website with images in abstract, artistic, and comical styles. Clipart shares the same 20 classes with Pascal VOC and contains 1,000 images, while Watercolor and Comic share 6 categories with Pascal VOC, i.e., bike, bird, car, cat, dog, and person, and contain 2000 images, respectively. For Pascal VOC→Clipart, we follow the mainstream setting [9], [24], [63] to leverage all Clipart images as the unlabeled target domain for both training and testing. For Pascal VOC→Watercolor/Comic, we use the *train* set of 1,000 images for training and utilize *test* set with 1,000 images for evaluation as existing DAOD works.

5.1.2 Implementation Details

The feature extractors $\phi(\cdot)$ are deployed with *BN-free* VGG16 [66] and *BN-frozen* ResNet [31], which are pre-trained on ImageNet [19] for a fair comparison with the mainstream of DAOD [13]. Our model is trained with the Stochastic Gradient Descent (SGD) optimizer with a 0.0025 learning rate, 4 batch-size, the momentum of 0.9, and weight decay of 5×10^{-4}. We sample at most 100 graph nodes for each feature map in each domain. Considering the graph matching may fail if no nodes appear in the target domain, we follow [35] to pre-train the framework as a warm-up stage before introducing the BHM module. Since state-of-the-art counterparts [11], [20], [32], [63], [91] leverage extra interpolated data generated from CycleGAN [94] in *natural image adaptation*, we follow their data usage and pre-train the model with source data for a fair comparison. The score-threshold τ_{fg} for sampling target-domain foreground nodes

TABLE 3
Comparison results (%) on Sim10K→Cityscapes, KITTI→Cityscapes, and Cityscapes→KITTI with VGG16. Res50 indicates the results from ResNet50. * reveals the reproduced results with better hyperparameters, higher than the results from the original paper.

Method	Sim10K→Cityscapes			KITTI→Cityscapes			Cityscapes→KITTI			Reference
	AP_{50}	**Gain**	SO	**AP_{50}**	**Gain**	SO	**AP_{50}**	**Gain**	SO	
EPM [35]	49.0	9.2	39.8	43.2	8.8	34.4	72.9	18.7	54.2	ECCV' 20
DSS (Res50) [77]	44.5	9.8	34.7	42.7	8.1	34.6	-	-	-	CVPR' 21
MEGA [73]	44.8	10.5	34.3	43.0	12.8	30.2	75.5	22.0	53.5	CVPR' 21
RPNPA [90]	45.7	11.1	34.6	-	-	-	75.1	21.6	53.5	CVPR' 21
UMT [20]	43.1	8.8	34.3	-	-	-	-	-	-	CVPR' 21
SSAL [56]	51.8	13.8	38.0	45.6	10.7	34.9	-	-	-	NeurIPS' 21
KTNet [70]	50.7	10.9	39.8	45.6	11.2	34.4	-	-	-	ICCV' 21
SCAN [46]	52.6	12.8	39.8	45.8	11.4	34.4	-	-	-	AAAI' 22
MGA [93]	49.8	-	-	45.2	-	-	-	-	-	CVPR' 22
TIA [91]	-	-	-	44.0	13.8	30.2	75.9	22.4	53.5	CVPR' 22
FGRR [10]	44.5	9.9	34.6	-	-	-	-	-	-	TPAMI' 22
TDD [32]	53.4	15.6	37.8	47.4	**17.2**	30.2	-	-	-	CVPR' 22
SIGMA* [47]	55.2	15.3	39.8	48.2	13.8	34.4	75.2	21.0	54.2	CVPR' 22
SIGMA++	**57.7**	**17.9**	39.8	**49.5**	15.1	34.4	**76.9**	**22.7**	54.2	

TABLE 4
Results (%) on CVC-ClinicDB→Abnormal Symptoms with ResNet101.

Method	$AP_{50:95}$	AP_{75}	**AP_{50}**	**Gain**	SO
FRCNN [60]	44.3	47.7	72.7	-	-
FCOS [71]	47.9	51.0	70.5	-	-
EPM [35]	52.1	56.8	74.4	3.9	70.5
SADA [14]	53.3	59.1	75.7	3.0	72.7
PolypDA [1]	52.3	57.0	72.9	0.2	72.7
SCAN [46]	53.2	57.4	75.6	5.1	70.5
SIGMA++	**55.8**	**60.4**	**79.8**	**9.3**	70.5

is empirically set to 0.5 to satisfy the active condition of the non-linear $sigmoid$, and the τ_{bg} for the background is set to 0.05 following the commonly used score-threshold setting in object detectors [52], [59], [60], [71]. Moreover, the adaption-unrelated settings about the object detector strictly follow previous works [35], [46], [56], [70].

5.1.3 Evaluation Metrics

For a fair comparison with DAOD works, we use COCO [51] evaluation metric for the city landscapes adaptation and medical scene adaptation and Pascal VOC [22] metric for natural image adaptation. Mean Average Precision with different IoU thresholds (AP_{IoU}) is used for evaluation. Considering the inconsistent baseline implementation, we report source-only (SO) [13] results, i.e., training only with source data and testing with target data. Moreover, the adaptation gain (Gain) compared with SO is also reported, serving as a critical role in comparing the adaptation capacity fairly.

5.2 Comparison with State-of-the-arts

(1) Cityscapes→Foggy Cityscapes. As shown in Table 1, SIGMA achieves the best 44.5% AP_{50} with the best adaptation gains (26.1%). Compared with category-level adaptation approaches, e.g., CFFA [92] (38.6%), MeGA-CDA [73] (41.8%), KTNet [70] (40.9%), GPA [84] (39.5%), and SIGMA [47] (43.5%), SIGMA++ achieves 5.9%, 2.7%, 3.6%, 5.0%, and 1.0% AP_{50} gains respectively, verifying the better capability of modeling class conditional distribution with hypergraph than prototype-based measurements. Besides, SIGMA++ surpasses EPM [35], KTNet [70], SSAL [56], SCAN [46], and SIGMA [47] with 8.5%, 3.6%, 4.9%, 2.4%, and 1.0% AP_{50} using the same FCOS [71] object detector.

(2) Cityscapes→BDD100K. We present the comparison in Table 2. The proposed SIGMA++ achieves state-of-the-art performance with 33.7% AP_{50}, outperforming the latest work SIGMA [47] (32.7%) with 1.0%. Notably, although SIGMA++ doesn't use extra interpolated data as [32], [63], it also surpasses these two counterparts obviously, verifying the effectiveness of the proposed method.

(3) Sim10k→Cityscapes. The comparison is shown in 1^{st} *col.* of Table 3. SIGMA++ achieves a 57.7% AP_{50} with the best adaptation gain (17.9% AP_{50}), outperforming existing works significantly. Moreover, 7.0%, 5.9%, 5.1% , and 2.5% improvements are achieved by SIGMA++ compared with KTNet [70] (50.7%), SSAL [56] (51.8%) SCAN [46] (52.6%) and SIGMA [47] (55.2%) with same baseline [35].

(4) KITTI⇆Cityscapes. The comparison results are shown in 2^{nd} and 3^{rd} *col.* of Table 3. SIGMA outperforms existing works with a 49.5% AP_{50} and 76.9% AP_{50}, achieving the second best (15.1%) and the best best adaptation gain (22.7%) respectively. For KITTI→Cityscapes, SIGMA++ (49.5%) surpasses EPM [35] (43.2%), KTNet [70] (45.6%), SSAL [56] (45.6%), SCAN (45.8%), and SIGMA [47] (75.2%) significantly with the same object detector baseline.

(5) CVC-ClinicDB→Abnormal Symptoms. The comparison results are shown in Table 4. We find that the proposed SIGMA++ framework still achieves satisfactory performance in more challenging medical scenes [1], giving a 79.8% AP_{50}. Compared with state-of-the-art method (PolypDA [1]) in endoscopic DAOD (72.9% AP_{50}), the proposed SIGMA++ also surpasses it with 6.9% gains, verifying our generalization and robustness.

(6) Pascal VOC→Clipart. The benchmark comparison is recorded in Table 5. The proposed SIGMA++ achieves the best results on both AP_{50} (46.7%) and adaptation gains (21.4%), outperforming the previous best entry TIA [91] (46.3% AP_{50} and 18.5% Gain) with 0.4% and 2.9% respectively. Compared with SIGMA [47], the proposed SIGMA++ further gives 2.2% improvements, demonstrating the effec-

TABLE 5
Comparison results (%) on PASCAL VOC→Clipart with ResNet101 backbone.

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	hrs	bike	prsn	plnt	sheep	sofa	train	tv	**AP$_{50}$**	**Gain**	SO
SWDA [63]	26.2	48.5	32.6	33.7	38.5	54.3	37.1	18.6	34.8	58.3	12.5	12.5	33.8	65.5	54.5	52.0	9.3	24.9	54.1	49.1	38.1	6.9	27.8
IIDA [78]	41.5	52.7	34.5	28.1	43.7	58.5	41.8	15.3	40.1	54.4	26.7	28.5	37.7	75.4	63.7	48.7	16.5	30.8	54.5	48.7	42.1	14.3	27.8
ICR-CCR [81]	28.7	55.3	31.8	26.0	40.1	63.6	36.6	9.4	38.7	49.3	17.6	14.1	33.3	74.3	61.3	46.3	22.3	24.3	49.1	44.3	38.3	11.3	27.0
ATF [33]	41.9	67.0	27.4	36.4	41.0	48.5	42.0	13.1	39.2	**75.1**	33.4	7.9	41.2	56.2	61.4	50.6	42.0	25.0	53.1	39.1	42.1	27.8	14.3
SCL [65]	**44.7**	50.0	33.6	27.4	42.2	55.6	38.3	19.2	37.9	69.0	30.1	26.3	34.4	67.3	61.0	47.9	21.4	26.3	50.1	47.3	41.5	10.3	27.8
HTCN [11]	33.6	58.9	34.0	23.4	45.6	57.0	39.8	12.0	39.7	51.3	20.1	20.1	39.1	72.8	61.3	43.1	19.3	30.1	50.2	51.8	40.3	12.5	27.8
SAPN [44]	27.4	**70.8**	32.0	27.9	42.4	63.5	47.5	14.3	**48.2**	46.1	31.8	17.9	43.8	68.0	68.1	49.0	18.7	20.4	55.8	51.3	42.2	14.4	27.8
UMT [20]	39.6	59.1	32.4	35.0	45.1	61.9	48.4	7.5	46.0	67.6	21.4	**29.5**	**48.2**	75.9	**70.5**	**56.7**	25.9	28.9	39.4	43.6	44.1	16.3	27.8
DBGL [9]	28.5	52.3	34.3	32.8	38.6	66.4	38.2	25.3	39.9	47.4	23.9	17.9	38.9	78.3	61.2	51.7	26.2	28.9	56.8	44.5	41.6	13.8	27.8
FGRR [10]	30.8	52.1	35.1	32.4	42.2	62.8	42.6	21.4	42.8	58.6	**33.5**	20.8	37.2	81.4	66.2	50.3	21.5	29.3	**58.2**	47.0	43.3	15.5	27.8
TIA [91]	42.2	66.0	36.9	**37.3**	43.7	**71.8**	**49.7**	18.2	44.9	58.9	18.2	29.1	40.7	**87.8**	67.4	49.7	27.4	27.8	57.1	50.6	46.3	18.5	27.8
SIGMA [47]	40.1	55.4	37.4	31.1	54.9	54.3	46.6	23.0	44.7	65.6	23.0	22.0	42.8	55.6	67.2	55.2	32.9	**40.8**	45.0	58.6	44.5	19.2	25.3
SIGMA++	36.3	54.6	**40.1**	31.6	**58.0**	60.4	46.2	**33.6**	44.4	66.2	25.7	25.3	44.4	58.8	64.8	55.4	**36.2**	38.6	54.1	**59.3**	**46.7**	**21.4**	25.3

TABLE 6
Comparison results (%) on Pascal VOC→Watercolor (left) and Pascal VOC→Comic (right) with ResNet101 backbone.

Method	Pascal VOC→Watercolor									Pascal VOC→Comic								
	bike	bird	car	cat	dog	prsn	**AP$_{50}$**	**Gain**	SO	bike	bird	car	cat	dog	prsn	**AP$_{50}$**	**Gain**	SO
DAF [13]	75.2	40.6	48.0	31.5	20.6	60.0	46.0	1.4	44.6	-	-	-	-	-	-	-	-	-
SWDA [63]	82.3	**55.9**	46.5	32.7	35.5	66.7	53.3	8.7	44.6	36.0	18.3	29.3	9.3	22.9	48.4	27.4	3.0	24.4
I^{3}Net [12]	81.1	49.3	46.2	35.0	31.9	65.7	51.5	4.4	47.1	**47.5**	19.9	33.2	11.4	19.4	49.1	30.1	8.2	21.9
DBGL [9]	83.1	49.3	50.6	**39.8**	38.7	61.3	53.8	9.2	44.6	35.6	20.3	33.9	16.4	26.6	45.3	29.7	5.5	24.4
SAPNet [44]	81.1	51.1	53.6	34.3	39.8	71.3	55.2	10.6	44.6	-	-	-	-	-	-	-	-	-
SCL [65]	82.2	55.1	51.8	39.6	38.4	64.0	55.2	10.6	44.6	-	-	-	-	-	-	-	-	-
FGRR [10]	86.1	54.8	48.9	36.6	40.4	67.5	55.7	11.1	44.6	42.2	21.1	30.2	21.9	**30.0**	50.5	32.7	8.3	24.4
UMT [20]	**88.2**	55.3	51.7	**39.8**	**43.6**	69.9	**58.1**	11.7	46.4	-	-	-	-	-	-	-	-	-
SIGMA [47]	76.3	52.9	56.9	34.7	38.5	69.1	54.8	11.0	43.8	-	-	-	-	-	-	-	-	-
SIGMA++	79.0	55.1	**58.2**	38.4	41.0	**71.4**	57.1	**13.3**	43.8	45.7	**24.4**	**36.9**	**25.5**	27.9	**62.5**	**37.1**	**13.2**	23.9

tiveness of the hypergraph-based modeling.

(7) Pascal VOC→Watercolor. The comparison is shown in the left part of Table 6. SIGMA++ achieves a comparable 57.1% AP$_{50}$ as the second-best entry of state-of-the-art. We can also observe a significant 13.3% adaptation gain achieved by SIGMA++, which outperforms the second-best counterpart UMT [20] with a 1.6% improvement. Moreover, compared with SIGMA [47], the significant 2.3% performance improvements clearly illustrate the advantage of the hypergraph structure with high-order dependencies.

(8) Pascal VOC→Comic. As shown in the right part of Table 6. SIGMA++ achieves the best adaptation performance with a 37.1% AP$_{50}$, surpassing the previous best entry FGRR [10] (32.7%) by a large margin with 4.4% improvements. Moreover, the best adaptation gain (13.2%) can be achieved, verifying the effectiveness of our method.

5.3 Ablation Analysis

5.3.1 Analysis on Each Module

As shown in Table 7, we conduct comprehensive experiments to verify the effect on each individual module of the proposed SIGMA++, which is evaluated on four heterogeneous adaptation scenarios. First of all, as illustrated in 1-3 *rows*, a general comparison among baseline [35] (42.5%), graph-based SIGMA [47] (49.9%) and hypergraph-based SIGMA++ (51.5%) are given, showing the consistent improvement on our improved adaptation algorithm. Then, we gradually remove each module in SIGMA++ to explore its individual effect, consisting of the following sub-settings.

(1) Remove HSC module and conduct $\mathcal{L}_{node}$ on $\tilde{\mathcal{V}}_{s/t}$ to ensure the parameter trainable. (2) Remove the whole BHM module. (3) Eliminate the domain-guided node completion (DNC) and leverage semantic-mismatched nodes to establish graphs. (4) Remove the single-layer hypergraph convolution in cross-image graph reasoning (CGR). (5) Remove the graph-guided memory bank (GMB) and collect class seeds online. (6) Abort the node discriminator in aligning the marginal distribution of the node embedding. (7) Remove the cross-domain graph interaction (CGI). (8) Replace the semantic-aware node affinity (SNA) with the inner-product-based affinity [25]. (9) Remove all of the structure-aware matching loss (SML). (10) Remove the structure constraint (SC) term in SML. (11) Replace the *sinkhorn* layer (sparse matching) with a *sigmoid* layer and generate matching signals on all the correct/wrong matching (dense matching). We use the more robust average results (Avg.) for comparison. Based on the above experiments, we have the following observations and analyses.

Firstly, as shown in (1) and (2), compared with the full model (51.5%), removing HSC (46.7%) and BHM (47.8%) respectively leads to 4.8% and 3.7% performance drops, verifying the individual effectiveness of semantic completing and matching-based adaptation. *Secondly*, according to the results from (3) to (6) in the HSC module, removing each part will give some performance drops, revealing

TABLE 7
Ablation study results (%) on four adaptation scenarios, including Cityscapes→Foggy Cityscapes (C→F), Sim10k→Citscapes (S→C), Pascal VOC→Clipart (P→C), and Pascal VOC→Watercolor (P→W).

ID	Setting	C→F	S→C	P→C	P→W	**Avg.**
-	GA-baseline [35]	35.3	45.9	37.8	51.2	42.5
-	SIGMA (full) [47]	43.5	55.2	44.5	56.0	49.9
-	SIGMA++ (full)	**44.5**	**57.7**	**46.7**	**57.1**	**51.5**
(1)	*w/o.* HSC	41.2	53.0	40.2	52.3	46.7
(2)	*w/o.* BHM	42.0	54.1	41.2	53.9	47.8
(3)	*w/o.* DNC	42.0	55.9	43.3	55.0	49.1
(4)	*w/o.* CGR	41.8	53.7	42.9	54.0	48.1
(5)	*w/o.* GMB	42.3	53.2	41.9	53.2	47.7
(6)	*w/o.* ND	44.2	56.2	43.7	55.2	49.8
(7)	*w/o.* CGI	42.8	55.8	45.0	55.7	49.9
(8)	*w/o.* SNA	43.9	56.9	44.2	55.9	50.2
(9)	*w/o.* SML	42.2	56.2	44.9	56.1	49.9
(10)	*w/o.* SML-SC	44.0	57.3	46.4	56.4	51.0
(11)	*w/o.* SML-Sinkhorn	43.9	56.9	45.7	57.0	50.9

TABLE 8
Results (%) on Cityscapes→Foggy Cityscapes with different graph node settings. $N^f_{s/t}$ represents the maximum sampled nodes from source and target domains on each image-level feature map.

(a) Vision-to-Graph Transformation.

Setting	prsn	rider	car	truc	bus	train	moto	bike	$\mathbf{AP}_{50}$
w/o. Proj	46.9	43.5	**62.8**	**32.1**	51.3	44.2	35.7	39.4	**44.5**
Linear	46.4	45.1	61.0	**32.1**	**52.2**	44.6	34.8	**39.9**	**44.5**
Non-L	**47.0**	**46.4**	61.3	27.0	52.0	**44.7**	**36.4**	39.5	44.3
w/o. Norm	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
GN	46.0	44.8	60.8	27.6	49.0	35.8	**36.8**	39.4	42.5
BN	45.6	43.9	**62.0**	29.5	49.1	30.3	33.9	**41.5**	42.0
LN	**46.4**	**45.1**	61.0	**32.1**	**52.2**	**44.6**	34.8	39.9	**44.5**

(b) Node Number.

N^f_s	N^f_t	prsn	rider	car	truc	bus	train	moto	bike	$\mathbf{AP}_{50}$
200	0	43.7	41.3	58.7	29.4	43.4	21.4	32.1	37.8	38.5
0	200	44.6	40.9	60.3	24.1	38.6	25.5	33.1	37.4	38.1
20	20	45.5	43.3	60.6	28.4	47.0	55.6	33.0	39.7	44.1
100	100	**46.4**	**45.1**	61.0	**32.1**	**52.2**	44.6	**34.8**	**39.9**	**44.5**
200	200	**46.4**	44.9	**61.4**	29.5	45.8	**48.3**	29.7	38.5	43.1

its essential role in semantic completion. The substantial performance drop in CGR (3.4%) and GMB (3.8%) shows that the graph-based long-distance dependencies play a critical role in adapting class conditional distribution, ranging from batch-level (CGR) to domain-level (GMB) knowledge. *Thirdly*, the experimental results from (7) to (11) verify our optimal design in the BHM module. We observe that CGI (7) and SNA (8) make similar contributions with the same performance (49.9%), implying the inherent collaborative and necessary effectiveness in modeling the cross-domain semantic interaction. Moreover, as shown in (11), we observe a slight performance drop after deploying the spare graph matching in a dense manner as the singe-matching aligns primary node pairs and relives noisy adaptation on ambiguous nodes in the dense matching. See Sec. 5.4.5 for a more clear qualitative analysis of the proposed modules.

5.3.2 Analysis on Graph Node

Vision-to-Graph Transformation. We analyze different transformation designs in Table 8(a), which consists of the node projection (top) and normalization (bottom).

1) Node Projection. Replacing the linear projection with a non-linear variation (Fc-ReLU-Fc) [47] will lead to an ignorable 0.2% performance degradation. This phenomenon reveals that the proposed method is robust to the vision-to-graph projection designs. Compared with the linear structure, we observe that directly building the hypergraph on visual features (w/o. Proj) can achieve exactly the same 44.5% AP_{50}, revealing a well-aligned high-dimensional space between the graph and visual representation. These results verify the technical rationality of building graphs on visual features directly, implying the inherent adaptation between structured graph space and grid-based visual space.

2) Node Normalization. Considering the unstable numerical distribution across feature levels, we further analyze the normalization strategy on the node embedding, including Group Normalization (used in FCOS [71] detection head), Batch Normalization (used in the ResNet [31] feature extractor), and Layer Normalization [3] (widely used in transformer detectors [8]). BN and GN are deployed on the whole feature maps, while LN is conducted on nodes. We observe that the normalization layer is necessary, without which the model cannot converge due to the significant numerical gap among different feature levels of the BN-free backbone [66]. Moreover, our design (LN) is verified to work better than BN (42.0% AP_{50}) and GN (42.5% AP_{50}), preserving node correspondence with the best 44.5% AP_{50}.

Node Number. Delving into the scale of the graph space, we analyze the influence on the node number in both domains ($N^f_{s/t}$ indicates the maximum number of nodes sampled from each feature map), as shown in Table 8(b). Only leveraging source and target nodes (1^{st} and 2^{nd} *rows*) severely affects the adaptation with 38.5% and 38.1% AP_{50}, and the reasons may come from two aspects. 1) Using source nodes will encourage the embedding space learning toward the source-specific direction, leading to the severe optimization bias of the source domain. 2) Due to the inherent noisy property of target nodes, only using target nodes is easy to suffer from the local optimum without reliable guidance from labels, leading to the overfitting of noisy target nodes. Moreover, we observe that the hypergraph-based adaptation is relatively robust with limited node number, e.g., achieving a comparable 44.1% AP_{50} only using 20 nodes, due to the more effective and robust hypergraph structure. In contrast, sampling too many nodes (e.g., 200) results in optimization difficulty in both graph learning and matching, yielding a sub-optimal performance (43.1% AP_{50}) and a 1.4% decline compared with our default 100 nodes.

5.3.3 Analysis on Hypergraph Structure

Hypergraph Architecture. As shown in Table 9(a), a high-level comparison between common graph architecture [47] and the improved hypergraph-based one is given, showing that the hypergraph (44.5%) works better than the graph-based counterpart (43.5%) with an obvious 1.0% AP_{50} gain. After delving into per-class precision, we can observe that the hypergraph works consistently better on these *rare classes*, e.g., trunk (32.1% vs. 27.1%) and bus (52.2% vs. 50.7%), and works significantly well on the rarest class train (44.6% vs. 35.9%) in the dataset [17]. Based on this observation, we assume that the high-order semantic correlations

TABLE 9
Results (%) on Cityscapes → Foggy Cityscapes with different (a) graph architectures, (b) hyperedge orders, and (c) graph reasoning layers.

(a) Hypergraph Architecture.									
Setting	prsn	rider	car	truc	bus	train	moto	bike	AP_{50}
Graph	**46.9**	**48.4**	**63.7**	27.1	50.7	35.9	34.7	**41.4**	43.5
Hyper-G	46.4	45.1	61.0	**32.1**	**52.2**	**44.6**	**34.8**	39.9	**44.5**
(b) Hyperedge Order.									
Setting	prsn	rider	car	truc	bus	train	moto	bike	AP_{50}
2	**46.9**	48.4	63.7	27.1	50.7	35.9	34.7	**41.4**	43.5
3	46.4	45.1	61.0	**32.1**	52.2	44.6	34.8	39.9	44.5
4	46.2	46.2	60.9	23.9	**53.6**	48.4	**35.0**	40.4	44.3
6	46.7	**46.4**	62.6	26.5	50.9	**53.6**	31.3	39.3	**44.7**
8	45.4	44.7	**63.7**	31.9	51.1	46.0	32.7	40.0	44.4
(c) Hypergraph Reasoning Layer.									
Setting	prsn	rider	car	truc	bus	train	moto	bike	AP_{50}
1	46.4	45.1	61.0	**32.1**	**52.2**	44.6	**34.8**	39.9	**44.5**
2	**46.6**	**46.7**	61.0	30.7	50.5	**47.1**	32.6	40.5	**44.5**
3	46.0	45.0	**62.0**	29.8	49.9	44.2	32.9	**41.6**	43.9

TABLE 10
Comparison results (%) on Cityscapes → Foggy Cityscapes with ResNet50. Tow sub-table: the comparison between Faster RCNN C4 [60] based SIGMA++ and the latest works using the same baseline. Bottom sub-table: the comparison between FCOS [71] based SIGMA++ and the latest methods using the same baseline. GA-FRCNN and GA-FCOS indicate the consistent baseline implementation [55] with image-level alignment.

Detector	$AP_{50:95}$	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
GA-FRCNN	19.3	38.4	18.2	2.0	17.3	40.7
SA-DAF [14]	20.8	41.1	19.4	2.3	18.5	43.9
VISGA [61]	21.8	43.3	19.2	**2.5**	19.1	45.9
NLTE [53]	21.0	41.8	19.4	2.2	18.5	45.1
SIGMA++	**23.2**	**43.5**	**21.0**	2.4	**20.7**	**47.0**
GA-FCOS	18.9	36.7	16.6	3.2	19.0	36.2
EPM [35]	21.7	39.5	20.2	4.2	20.5	40.1
SCAN [46]	23.0	42.3	21.2	4.0	21.0	**42.8**
SIGMA++	**24.1**	**45.1**	**22.8**	**5.6**	**23.9**	**42.8**

can compensate for the rare objects of limited visual cues, working well in modeling and adapting the biased class conditional distribution with notable advantages.

Hyperedge Order. We explore the order of the hypergraph (i.e., the number of graph nodes connected in a hyperedge) and analyze the orders of the semantic dependencies, as shown in Table 9(b). We implement the second-order edge as our preliminary work [47] as the hyperedge will degenerate into the pair-wise edge. From the experimental results, we observe that the proposed SIGMA++ is extremely robust to the order of edge connections, giving a slight frustration between 44.0% and 45.0% on AP_{50}. More interestingly, we find that the accuracy of the rarest class, train, is relatively sensitive to the order of the hyperedge, which verifies the necessary role of the high-order semantic connections in these rare object instances.

Hyperedge Reasoning Layer. Graph reasoning highly relies on edge connections and plays a critical role in establishing the long-distance semantic dependencies among graph nodes, which is deployed with hypergraph convolutional layers. Hence, as shown in Table 9(c), we explore the effect with more HGCN layers. Specifically, we implement the one-layer HGCN as Eq. 3 and the more-layers following [24] by adding a non-linear ReLU activation. The experimental results retrieve that the one-layer HGCN has already achieved the optimal performance (44.5%) while introducing deeper HGCN won't give more benefits (2-layer with 44.5% AP_{50}) and even results in negative impact due to the redundant parameters (3-layer with 43.9% AP_{50}).

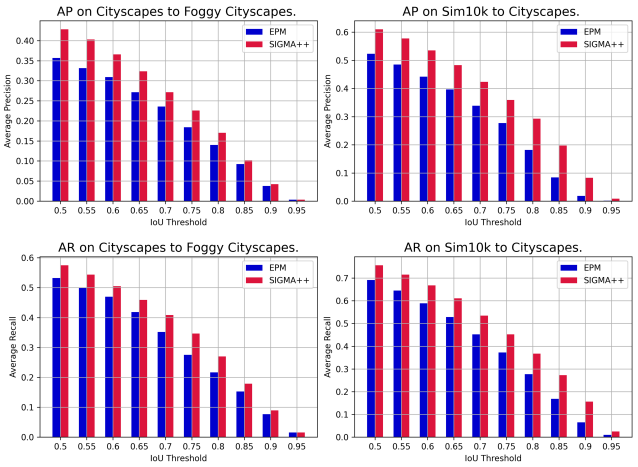


Fig. 4. Average Precision (AP) (tow *row*) and Average Recall (AR) (bottom *row*) comparison between the proposed SIGMA++ and EPM [35] with different IoU thresholds on Cityscapes→Foggy Cityscapes (left *col.*) and Sim10k→Cityscapes (right *col.*).

5.3.4 *Extension to Different Detection Pipelines*

To further investigate the transferability of the proposed method, we implement SIGMA++ on both two-stage Faster RCNN C4 [60] and single-stage FCOS [71] object detectors and report the comparison with the latest works using the strictly consistent detection framework [55], as shown in Table 10. For the FRCNN-based deployment, we randomly collect a subset of off-the-shelled region proposals obtained from the Region Proposal Network (RPN) in the source and target domain equally, which serve as the graph nodes. Then, we conduct hypergraphical semantic completion and bipartite hypergraph matching to achieve domain adaptation. As shown in Table 10, it can be observed that our SIGMA++ achieves the best results on almost all evaluation metrics on two kinds of detectors. SIGMA++ gives the best 43.5% and 45.1% AP_{50}, respectively, surpassing the baseline model (38.4% and 36.7% AP_{50}) with significant 5.1% and 8.4%. The experiments clearly verify our effectiveness and applicability on various detection pipelines.

5.3.5 *Comparison with Different IoU Thresholds*

As shown in Fig. 4, to better illustrate our strength in terms of high-quality object detection, we compare the average precision (AP) and average recall (AR) (100 maximum detection) difference between the proposed SIGMA++ and EPM [35] using different IoU thresholds (from 0.5 to 0.95 with 0.05 intervals). The experiments are conducted on two adaptation scenarios, distinguished by the left and right sub-figures. The results show that our method consistently improves the detection performance on both AP and AR evaluation across various IoU thresholds, comprehensively boosting object detection under cross-domain scenarios. Moreover, we observe that SIGAM++ consistently works

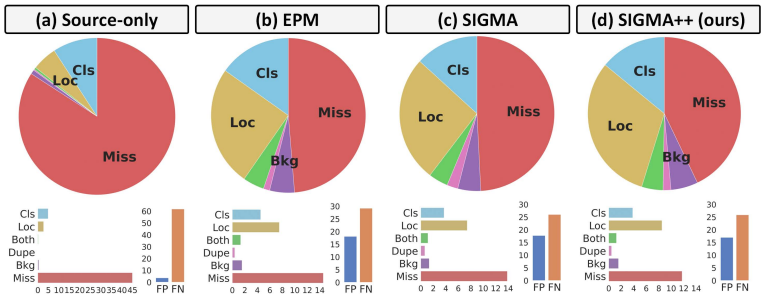


Fig. 5. Detailed error analysis using TIDE [7] toolbox of (a) source-only, (b) EPM [35], (c) SIGMA [47], and (d) the proposed SIGMA++.

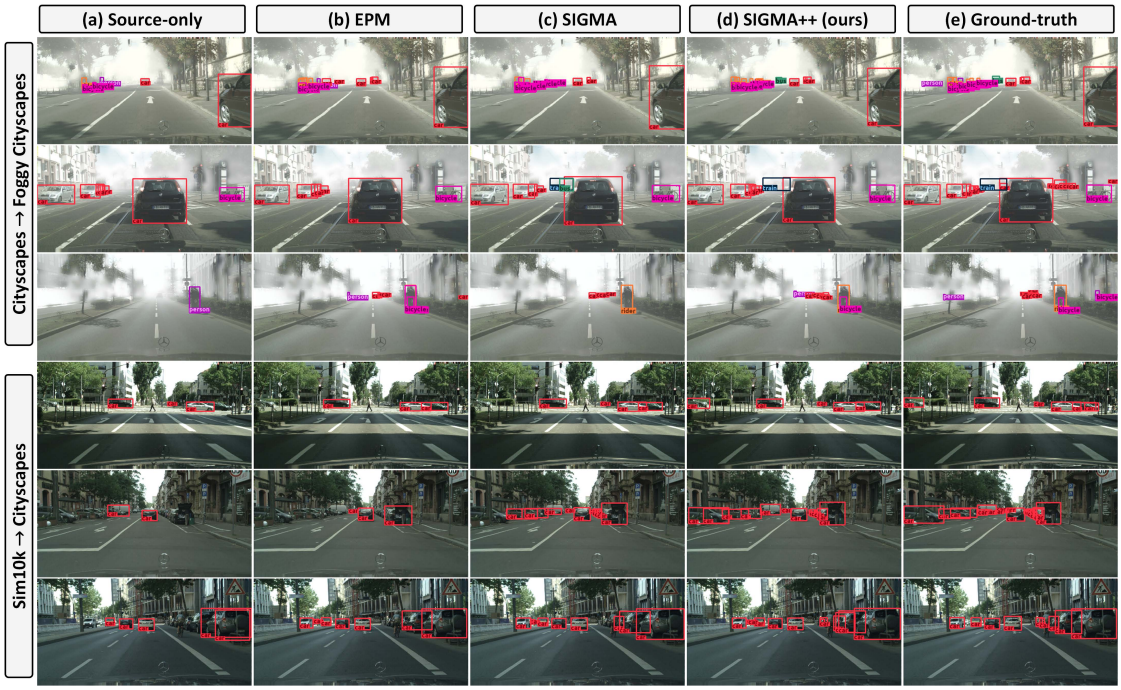


Fig. 6. Detection results comparison on Cityscapes→Foggy Cityscapes (top *rows*) and Sim10k→Cityscapes (bottom *rows*) of (a) source-only [71], (b) EPM [35], (c) SIGMA [47], (d) our SIGMA++, and (e) ground-truth. The different colors of bounding boxes indicate varied object classes.

better on multi-class (Cityscapes → Foggy Cityscapes) and single-class (Sim10k → Cityscapes) adaptation scenarios, verifying the superior robustness in terms of object classes.

5.4 Qualitative Results

5.4.1 Analysis on Detection Errors

We conduct a detailed error analysis using TIDE [7] toolbox and report the comparison among (a) source-only model, (b) EPM [35], (c) SIGMA [47], (d) the proposed SIGMA++, as shown in Fig. 5. From the experimental results, we have the following observations. (1) Compared with source-only results, existing DAOD methods can significantly reduce the missing errors (*Miss*). (2) There is a consistent decline in *Miss* from the source-only to SIGMA++ (from 45% to 12%), clearly retrieving the effectiveness of the proposed SIGMA++ in domain adaptation. (3) The clear reduction in *Miss* between SIGMA (14%) and SIGMA++ (12%) illustrates our improved algorithm's advantage. (4) Compared with the source-only, the classification errors (*Cls*) can be reduced in both SIGMA and SIGMA++ due to the effective graph-based learning, clearly verifying the effectiveness of the proposed adaptation framework.

5.4.2 Analysis on Detection Results

We present the comparison among (a) source-only, (b) EPM [35], (c) SIGMA [47], (d) the proposed SIGMA++, and (d) ground-truth in the city landscapes adaptation (Fig. 6) and natural scene adaptation (Fig. 7). As shown in Fig. 6, we observe that SIGMA++ can detect some extremely hard objects, e.g., the bus (in green) in 1^{st} row, which is missed by both EPM [35] and SIGMA [47]. Moreover, as shown in 2^{nd} row, although SIGMA can detect the train missed by EPM, it still generates an error box of the bus (in green) due to the limited graph matching. Fortunately, this issue is successfully solved by SIGMA++, which leverages the hypergraph to model robust high-order correspondence with limited visual cues. Similarly, we can observe a similar phenomenon in Fig. 7 and find a consistent performance improvement from (a) to (d), clearly verifying the superior advantages of

Fig. 7. Detection results comparison on Pascal VOC→Clipart (top *rows*) and Pascal VOC→Watercolor (bottom *rows*) among (a) the source-only model [71], (b) EPM [35], (c) SIGMA [47], (d) SIGMA++, and (e) ground-truth. The red and navy boxes indicate the correct and wrong detection.

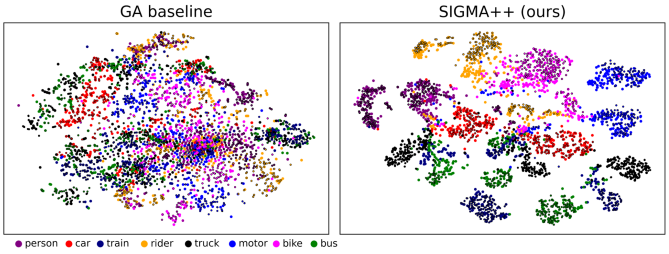


Fig. 8. TSNE feature comparison between the GA baseline [35] and SIGMA++ (ours). The circles with black contours are sampled from the target domain, while the ones without edges are from the source domain. We randomly sample 250 feature points inside bounding boxes for each class in each domain equally.

the proposed hypergraph matching paradigm. For example, with the well-modeled high-order semantic dependencies, SIGMA++ can correctly detect the dog in 1^{st} row, which is wrongly classified as a sheep in EPM and SIGMA due to the unobservable and sub-optimal semantic correlations.

5.4.3 Analysis on T-SNE Feature Visualization

As shown in Fig. 8, to illustrate the effectiveness of adapting class conditional distribution, we present a feature-level visualization via T-SNE and make a comparison with the class-agnostic GA baseline [35]. For each class, we randomly sample 250 ResNet50 feature points inside bounding boxes for each domain. The circles in different colors represent the varied classes, and the circles with black contours indicate the target-domain samples. From the experimental comparison, we have the following observations. (1) Our baseline can only adapt the marginal distribution but fail to align the class conditional distribution. Differently, SIGMA++ can align the distribution for each class with clearly separatable feature embedding, which significantly benefits the followed detection head for better detection. (2) We observe that similar classes, e.g., *person*, *rider*, and *bike*, are more separable in SIGMA++, playing a critical role in achieving high-quality object detection. (3) More interestingly, we find that the features in rare classes, e.g., *train*, *truck*, and *bus* can be disentangled into several sub-clusters, which may be caused by the hyperedges with inherent grouping properties [24]. With the limited data number and visual cues of rare classes, these sub-clusters can capture more representative local patterns and achieve an inherent interaction with those similar sub-clusters, which significantly improves the performance of these rare classes. Moreover, this observation is consistent with the numerical analysis of Table 9(a) in Sec. 5.3.3, showing the obvious advantages of adapting class conditional distribution.

5.4.4 Analysis on Graph Matching Visualization

We further visualize the matched points in Fig. 9, which is conducted on the semantic-rich Cityscapes→Foggy Cityscapes setting with the batch size of 2 (one source image and one target image). For each source-target image pair,

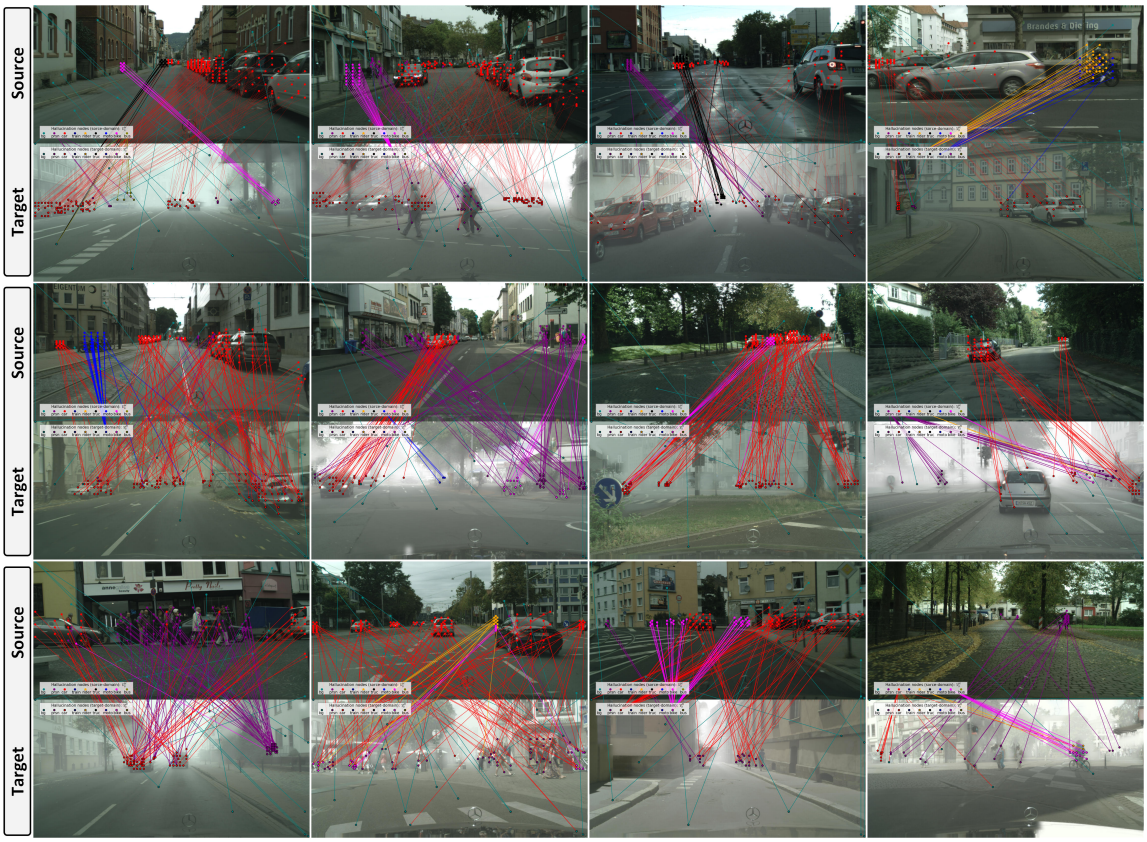


Fig. 9. Graph matching visualization with Cityscapes→Foggy Cityscapes adaptation. For the domain-mismatched classes, we illustrate hallucination nodes $\mathcal{V}_{s/t}^{\mathcal{H}}$ in the lower-left/upper-left corner (the white-region legend) of source/target-domain images, respectively. (Zooming in for a better view.)

the points are sampled graph nodes[3], and straight lines indicate the predicted matching between the source and target nodes. Moreover, to thoroughly visualize the matching, we illustrate hallucination nodes $\mathcal{V}_{s/t}^{\mathcal{H}}$ beyond the current image content in the lower-left/upper-left corner of source/target-domain images, respectively. The color of graph nodes represents the ground-truth label (source-domain) and pseudo-label (target-domain), while straight lines are colored as the source-domain node for a better view. According to the matching visualization (Fig. 9), we have the following observations. 1) For the classes appearing in both domains, most nodes in the same class can be matched correctly, serving as cross-domain sample pairs to align the class conditional distribution in a fine-grained manner. 2) For the mismatched classes, most nodes can be successfully matched to the hallucination nodes $\mathcal{V}_{s/t}^{\mathcal{H}}$ in the correct class, revealing that the matching can generate unbiased adaptation signals with domain-mismatched semantic content. 3) Some nodes matched to the cross-domain counterparts in inconsistent classes do exist, which is reasonable and explainable. For example, in 3^{rd} sample of the first row, we observe that there is a *truck* (in black) node matched to the *car* (in red). The reason lies in that different classes also share similar local patterns, e.g., the wheel shared by both *truck* and *car*, thereby enabling fine-grained pattern learning and matching across various classes.

3. We re-project the sampled feature points to the original image scale according to the corresponding convolution stride.

5.4.5 *Qualitative Ablation Study*

We further conduct a qualitative analysis on the proposed modules, as shown in Fig. 10. Comparing Fig. 10(b) with the baseline results (Fig. 10(a)), we observe that BHM eliminates several missing errors by detecting more objects, e.g., the more correctly detected cars in 1^{st} and 2^{nd} rows of images. The baseline model cannot work well on the regular class *car* due to its sizeable within-class diversity and significant class variance, e.g., different scales, shapes, colors, etc. Differently, BHM is able to detect diverse cars with better recall, revealing the satisfactory discriminability of diversity and addressing the poor perception of class variance. Moreover, we observe that BHM eliminates some false detection against common sense, e.g., the wrongly detected person on top of the truck in 3^{rd} row of the image. This phenomenon implies that the structural dependence among classes can format a more reasonable feature space with rich relationships, yielding a more transferable distribution with high-order variance cues. For the proposed Hypergraphical Semantic Completion (HSC), the comparison between Fig. 10(c) and Fig. 10(a) shows that HSC can improve the classification accuracy (the correctly classified rider in 1^{st} image), and reduce some missing errors on rare classes (the correctly detected bus in 2^{nd} image). These results imply that HSC is able to encourage an unbiased distribution adaptation, relieving the influence of domain-mismatched semantic content. After combing HSC and BHM (Fig. 10 (d)), we observe that the full SIGMA++ model addresses both

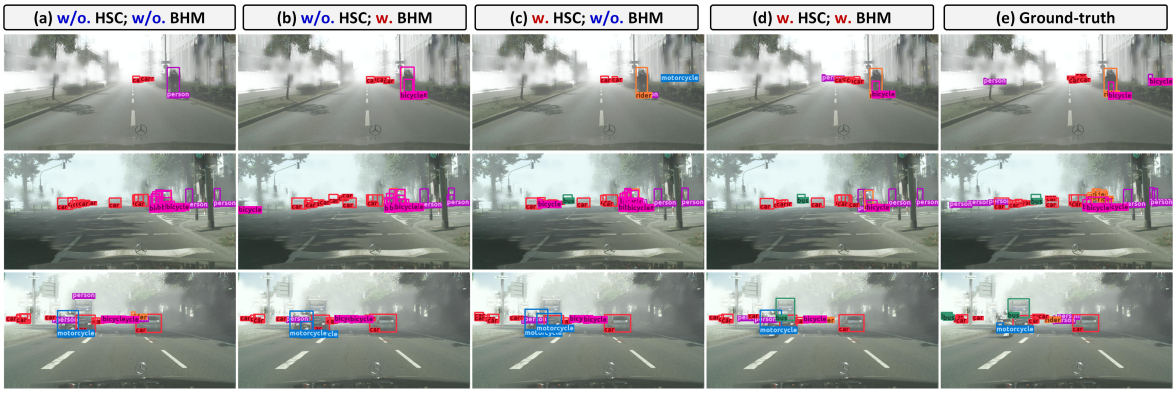


Fig. 10. Qualitative comparison results on the proposed modules among (a) the GA baseline model without both HSC and BHM, (b) SIGMA++ without HSC, (c) SIGMA++ without BHM, (d) the full SIGMA++ model, and (e) the ground-truth labels. (Zooming in for a better view.)

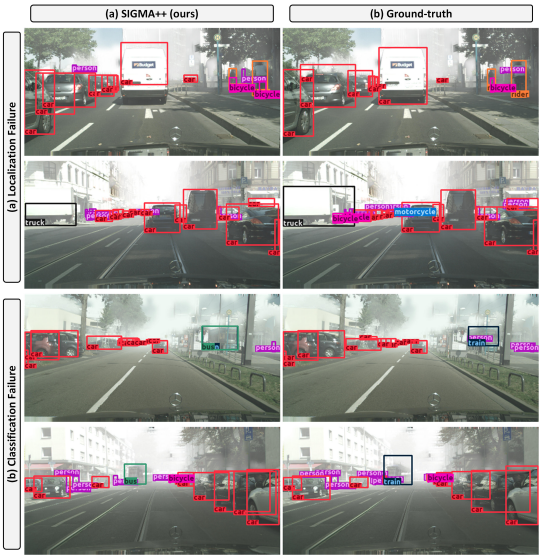


Fig. 11. Illustration of (a) failure cases of the proposed SIGMA++ and (b) the ground-truth label in terms of localization and classification errors.

aforementioned corner cases, showing the complementary effect of the proposed two modules.

5.4.6 Analysis on Failure Cases

We further conduct a detailed analysis of the failure examples and summarize the failure patterns in Fig. 11. As shown in Fig. 11(a), we observe that some of the detected bounding boxes fail to localize the ground-truth object precisely enough, e.g., the *car* in 1^{st} image and the *truck* in 2^{nd} image. The reason lies in that our method is developed to align the semantic-level distribution instead of improving the spatial-level regression. Introducing more localization-friendly knowledge, e.g., coordinate-related cues, into node embedding could relieve this limitation. Moreover, as shown in Fig. 11(b), we find that there are still some occluded objects misclassified by the proposed SIGMA++, e.g., the *train* wrongly classified as *bus*. The visual cues for these occluded and rare objects are extremely limited, which are not enough to support a reliable class separation. Hence, it will be a reasonable solution to introduce extra-linguistic knowledge, e.g., text embedding, to compensate for the limited visual knowledge, establishing the cross-modality graph matching between the visual and language space.

6 Conclusion

This paper proposes an improved Semantic-complete Graph Matching framework for DAOD, coined SIGMA++. It represents domain knowledge via establishing semantic-complete hypergraphs, and models domain adaptation as a graph-matching problem, which breaks the barrier of existing category-level approaches in terms of the sub-optimal prototype alignment. It adopts a Hypergraphical Semantic Completion (HSC) module to complete mismatched semantics and models class conditional distribution with hypergraphs. Then, it leverages a Bipartite Hypergraph Matching (BHM) module to achieve fine-grained alignment with hypergraph matching. Extensive experiments on nine benchmarks show that the proposed method outperforms existing approaches by a large margin.

References

[1] Consolidated domain adaptive detection and localization framework for cross-device colonoscopic images. *MIA*, 71:102052, 2021. 1, 8, 9

[2] Vinicius F Arruda, Thiago M Paixão, Rodrigo F Berriel, Alberto F De Souza, Claudine Badue, Nicu Sebe, and Thiago Oliveira-Santos. Cross-domain car detection using unsupervised image-to-image translation: From day to night. In *IJCNN*, pages 1–8, 2019. 1

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4, 6, 11

[4] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021. 1

[5] Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. Graph convolutional encoders for syntax-aware neural machine translation. In *EMNLP*, pages 1957–1967, 2017. 3

[6] Jorge Bernal, F Javier Sánchez, Gloria Fernández-Esparrach, Debora Gil, Cristina Rodríguez, and Fernando Vilariño. Wm-dova maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. *Comput. Med. Imaging Graph*, 43:99–111, 2015. 8

[7] Daniel Bolya, Sean Foley, James Hays, and Judy Hoffman. Tide: A general toolbox for identifying object detection errors. In *ECCV*, 2020. 13

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020. 1, 2, 11

[9] Chaoqi Chen, Jiongcheng Li, Zebiao Zheng, Yue Huang, Xinghao Ding, and Yizhou Yu. Dual bipartite graph learning: A general approach for domain adaptive object detection. In *ICCV*, pages 2703–2712, 2021. 1, 2, 3, 8, 10

[10] Chaoqi Chen, Jiongcheng Li, Hong-Yu Zhou, Xiaoguang Han, Yue Huang, Xinghao Ding, and Yizhou Yu. Relation matters: Foreground-aware graph-based relational reasoning for domain adaptive object detection. *arXiv preprint arXiv:2206.02355*, 2022. 1, 8, 9, 10

[11] Chaoqi Chen, Zebiao Zheng, Xinghao Ding, Yue Huang, and Qi Dou. Harmonizing transferability and discriminability for adapting object detectors. In *CVPR*, pages 8869–8878, 2020. 2, 8, 10

[12] Chaoqi Chen, Zebiao Zheng, Yue Huang, Xinghao Ding, and Yizhou Yu. I3net: Implicit instance-invariant network for adapting one-stage object detectors. In *CVPR*, pages 12576–12585, 2021. 1, 2, 3, 10

[13] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *CVPR*, pages 3339–3348, 2018. 1, 2, 3, 5, 7, 8, 9, 10

[14] Yuhua Chen, Haoran Wang, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Scale-aware domain adaptive faster r-cnn. *IJCV*, pages 2223–2243, 2021. 9, 12

[15] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *CVPR*, pages 5177–5186, 2019. 3

[16] Jiacheng Cheng and Nuno Vasconcelos. Learning deep classifiers consistent with fine-grained novelty detection. In *CVPR*, pages 1664–1673, June 2021. 1

[17] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016. 7, 11

[18] Debasmit Das and C. S. George Lee. Graph matching and pseudo-label guided deep unsupervised domain adaptation. In *ICANN*, pages 342–352, 2018. 3

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 8

[20] Jinhong Deng, Wen Li, Yuhua Chen, and Lixin Duan. Unbiased mean teacher for cross-domain object detection. In *CVPR*, pages 4091–4101, June 2021. 2, 8, 9, 10

[21] Jinhong Deng, Xiaoyue Zhang, Wen Li, and Lixin Duan. Cross-domain detection transformer based on spatial-aware and semantic-aware token alignment. *arXiv preprint arXiv:2206.00222*, 2022. 3

[22] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111:98–136, 2015. 8, 9

[23] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Stefano Soatto. Empirical study of the topology and geometry of deep networks. In *CVPR*, pages 3762–3770, 2018. 1

[24] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *AAAI*, volume 33, pages 3558–3565, 2019. 2, 3, 5, 8, 12, 14

[25] Kexue Fu, Shaolei Liu, Xiaoyuan Luo, and Manning Wang. Robust point cloud registration framework based on deep graph matching. In *CVPR*, pages 8893–8902, 2021. 3, 4, 6, 10

[26] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, pages 1180–1189, 2015. 1, 7

[27] Quankai Gao, Fudong Wang, Nan Xue, Jin-Gang Yu, and Gui-Song Xia. Deep graph matching under quadratic constraint. In *CVPR*, pages 5069–5078, 2021. 3, 4, 6

[28] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361, 2012. 7

[29] Bo Geng, Dacheng Tao, and Chao Xu. Daml: Domain adaptation metric learning. *TIP*, 20(10):2980–2989, 2011. 1, 3

[30] Jiawei He, Zehao Huang, Naiyan Wang, and Zhaoxiang Zhang. Learnable graph matching: Incorporating graph partitioning with deep feature learning for multiple object tracking. In *CVPR*, pages 5299–5309, 2021. 3, 4, 6

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 8, 11

[32] Mengzhe He, Yali Wang, Jiaxi Wu, Yiru Wang, Hanqing Li, Bo Li, Weihao Gan, Wei Wu, and Yu Qiao. Cross domain object detection by target-perceived dual branch distillation. In *CVPR*, 2022. 2, 8, 9

[33] Zhenwei He and Lei Zhang. Domain adaptive object detection via asymmetric tri-way faster-rcnn. In *ECCV*, pages 309–324, 2020. 1, 2, 3, 10

[34] Trung-Hieu Hoang, Hai-Dang Nguyen, Viet-Anh Nguyen, Thanh-An Nguyen, Vinh-Tiep Nguyen, and Minh-Triet Tran. Enhancing endoscopic image classification with symptom localization and data augmentation. In *ACM MM*, pages 2578–2582, 2019. 8

[35] Cheng-Chun Hsu, Yi-Hsuan Tsai, Yen-Yu Lin, and Ming-Hsuan Yang. Every pixel matters: Center-aware feature alignment for domain adaptive object detector. In *ECCV*, pages 733–748, 2020. 1, 2, 3, 7, 8, 9, 10, 11, 12, 13, 14

[36] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *CVPR*, pages 5001–5009, 2018. 2, 8

[37] Tao Ji, Yuanbin Wu, and Man Lan. Graph-based dependency parsing with graph neural networks. In *ACL*, pages 2475–2485, 2019. 3

[38] Junguang Jiang, Baixu Chen, Jianmin Wang, and Mingsheng Long. Decoupled adaptation for cross-domain object detection. *arXiv preprint arXiv:2110.02578*, 2021. 2

[39] Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? In *ICRA*, pages 746–753, 2017. 7

[40] Eun-Sol Kim, Woo Young Kang, Kyoung-Woon On, Yu-Jung Heo, and Byoung-Tak Zhang. Hypergraph attention networks for multimodal learning. In *CVPR*, pages 14581–14590, 2020. 3

[41] Taekyung Kim, Minki Jeong, Seunghyeon Kim, Seokeon Choi, and Changick Kim. Diversify and match: A domain adaptive representation learning paradigm for object detection. In *CVPR*, pages 12456–12465, 2019. 2

[42] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 3

[43] Jungmin Lee, Minsu Cho, and Kyoung Mu Lee. Hyper-graph matching via reweighted random walks. In *CVPR*, pages 1633–1640, 2011. 3

[44] Congcong Li, Dawei Du, Libo Zhang, Longyin Wen, Tiejian Luo, Yanjun Wu, and Pengfei Zhu. Spatial attention pyramid network for unsupervised domain adaptation. In *ECCV*, pages 481–497. Springer, 2020. 3, 10

[45] Wuyang Li, Zhen Chen, Baopu Li, Dingwen Zhang, and Yixuan Yuan. Htd: Heterogeneous task decoupling for two-stage object detection. *TIP*, 30:9456–9469, 2021. 3

[46] Wuyang Li, Xinyu Liu, Xiwen Yao, and Yixuan Yuan. Scan: Cross domain object detection with semantic conditioned adaptation. In *AAAI*, 2022. 1, 2, 3, 8, 9, 12

[47] Wuyang Li, Xinyu Liu, and Yixuan Yuan. Sigma: Semantic-complete graph matching for domain adaptive object detection. In *CVPR*, 2022. 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14

[48] Yu-Jhe Li, Xiaoliang Dai, Chih-Yao Ma, Yen-Cheng Liu, Kan Chen, Bichen Wu, Zijian He, Kris Kitani, and Peter Vajda. Cross-domain adaptive teacher for object detection. In *CVPR*, pages 7581–7590, 2022. 2

[49] Xiaowei Liao, Yong Xu, and Haibin Ling. Hypergraph neural networks for hypergraph matching. In *ICCV*, pages 1266–1275, 2021. 3

[50] Jongin Lim, Sangdoo Yun, Seulki Park, and Jin Young Choi. Hypergraph-induced semantic tuplet loss for deep metric learning. In *CVPR*, pages 212–222, June 2022. 3

[51] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 9

[52] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016. 1, 2, 9

[53] Xinyu Liu, Wuyang Li, Qiushi Yang, Baopu Li, and Yixuan Yuan. Towards robust adaptive object detection under noisy annotations. In *CVPR*, 2022. 12

[54] Eliane Maria Loiola, Nair Maria Maia de Abreu, Paulo Oswaldo Boaventura-Netto, Peter Hahn, and Tania Querido. A survey for the quadratic assignment problem. *Eur. J. Oper. Res.*, pages 657–690, 2007. 2, 3

[55] Francisco Massa and Ross Girshick. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. https://github.com/facebookresearch/maskrcnn-benchmark, 2018. 12

[56] Muhammad Akhtar Munir, Muhammad Haris Khan, M Saquib Sarfraz, and Mohsen Ali. Synergizing between self-training and adversarial learning for domain adaptive object detection. 2021. 2, 8, 9

[57] Quynh Nguyen, Antoine Gautier, and Matthias Hein. A flexible tensor block coordinate ascent scheme for hypergraph matching. In *CVPR*, pages 5270–5278, 2015. 3

[58] Rindra Ramamonjison, Amin Banitalebi-Dehkordi, Xinyu Kang, Xiaolong Bai, and Yong Zhang. Simrod: A simple adaptation method for robust object detection. In *ICCV*, pages 3570–3579, 2021. 2

[59] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1, 2, 9

[60] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015. 1, 2, 9, 12

[61] Farzaneh Rezaeianaran, Rakshith Shetty, Rahaf Aljundi, Daniel Olmeda Reino, Shanshan Zhang, and Bernt Schiele. Seeking similarities over differences: Similarity-based domain alignment for adaptive object detection. In *ICCV*, pages 9204–9213, 2021. 2, 8, 12

[62] Adrian Lopez Rodriguez and Krystian Mikolajczyk. Domain adaptation for object detection via style consistency. *arXiv preprint arXiv:1911.10033*, 2019. 2

[63] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *CVPR*, pages 6956–6965, 2019. 1, 3, 8, 9, 10

[64] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *IJCV*, 126(9):973–992, 2018. 7

[65] Zhiqiang Shen, Harsh Maheshwari, Weichen Yao, and Marios Savvides. Scl: Towards accurate domain adaptive object detection via gradient detach based stacked complementary losses. *arXiv preprint arXiv:1911.02559*, 2019. 1, 2, 3, 10

[66] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 8, 11

[67] Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *Ann. math. stat.*, 35:876–879, 1964. 6

[68] Daniil Sorokin and Iryna Gurevych. Modeling semantics with gated graph neural networks for knowledge base question answering. In *COLING*, pages 3306–3317, 2018. 3

[69] X Yu Stella and Jianbo Shi. Multiclass spectral clustering. In *ICCV*, volume 2, pages 313–313. IEEE Computer Society, 2003. 5

[70] Kun Tian, Chenghao Zhang, Ying Wang, Shiming Xiang, and Chunhong Pan. Knowledge mining and transferring for domain adaptive object detection. In *ICCV*, pages 9133–9142, October 2021. 1, 2, 3, 8, 9

[71] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, pages 9627–9636, 2019. 1, 2, 7, 9, 11, 12, 13, 14

[72] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, pages 7167–7176, 2017. 1, 3

[73] Vibashan VS, Vikram Gupta, Poojan Oza, Vishwanath A. Sindagi, and Vishal M. Patel. Mega-cda: Memory guided attention for category-aware unsupervised domain adaptive object detection. In *CVPR*, pages 4516–4526, June 2021. 2, 5, 8, 9

[74] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In *CVPR*, June 2019. 3

[75] Runzhong Wang, Junchi Yan, and Xiaokang Yang. Neural graph matching network: Learning lawler's quadratic assignment problem with extension to hypergraph and multiple-graph matching. *TPAMI*, 2021. 3, 6, 7

[76] Wen Wang, Yang Cao, Jing Zhang, Fengxiang He, Zheng-Jun Zha, Yonggang Wen, and Dacheng Tao. Exploring sequence feature alignment for domain adaptive detection transformers. In *ACM MM*, page 1730–1738, 2021. 2, 8

[77] Yu Wang, Rui Zhang, Shuo Zhang, Miao Li, Yangyang Xia, Xishan Zhang, and Shaoli Liu. Domain-specific suppression for adaptive object detection. In *CVPR*, pages 9603–9612, June 2021. 2, 8, 9

[78] Aming Wu, Yahong Han, Linchao Zhu, and Yi Yang. Instance-invariant domain adaptive object detection via progressive disentanglement. *TPAMI*, 2021. 10

[79] Aming Wu, Rui Liu, Yahong Han, Linchao Zhu, and Yi Yang. Vector-decomposed disentanglement for domain-invariant object detection. *ICCV*, 2021. 8

[80] Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, and Bo Long. Graph neural networks for natural language processing: A survey. *arXiv preprint arXiv:2106.06090*, 2021. 3

[81] Chang-Dong Xu, Xing-Ran Zhao, Xin Jin, and Xiu-Shen Wei. Exploring categorical regularization for domain adaptive object detection. In *CVPR*, pages 11724–11733, 2020. 8, 10

[82] Chenxin Xu, Maosen Li, Zhenyang Ni, Ya Zhang, and Siheng Chen. Groupnet: Multiscale hypergraph neural networks for trajectory prediction with relational reasoning. In *CVPR*, 2022. 3

[83] Huazhe Xu, Yang Gao, Fisher Yu, and Trevor Darrell. End-to-end learning of driving models from large-scale video datasets. In *CVPR*, pages 2174–2182, 2017. 8

[84] Minghao Xu, Hang Wang, Bingbing Ni, Qi Tian, and Wenjun Zhang. Cross-domain detection via graph-induced prototype alignment. In *CVPR*, pages 12355–12364, 2020. 1, 2, 3, 4, 5, 8, 9

[85] Junchi Yan, Changsheng Li, Yin Li, and Guitao Cao. Adaptive discrete hypergraph matching. *IEEE Trans Cybern*, 48(2):765–779, 2017. 3

[86] Junchi Yan, Xu-Cheng Yin, Weiyao Lin, Cheng Deng, Hongyuan Zha, and Xiaokang Yang. A short survey of recent advances in graph matching. In *ACM ICMR*, pages 167–174, 2016. 3

[87] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *ECCV*, pages 670–685, 2018. 3

[88] Jin Ye, Junjun He, Xiaojiang Peng, Wenhao Wu, and Yu Qiao. Attention-driven dynamic graph convolutional network for multi-label image recognition. In *European Conference on Computer Vision*, pages 649–665, 2020. 3

[89] Weilin Zhang and Yu-Xiong Wang. Hallucination improves few-shot object detection. In *CVPR*, pages 13008–13017, 2021. 2

[90] Yixin Zhang, Zilei Wang, and Yushi Mao. Rpn prototype alignment for domain adaptive object detector. In *CVPR*, pages 12425–12434, June 2021. 1, 2, 3, 4, 5, 8, 9

[91] Liang Zhao and Limin Wang. Task-specific inconsistency alignment for domain adaptive object detection. In *CVPR*, 2022. 2, 8, 9, 10

[92] Yangtao Zheng, Di Huang, Songtao Liu, and Yunhong Wang. Cross-domain object detection through coarse-to-fine feature adaptation. In *CVPR*, pages 13766–13775, 2020. 1, 2, 3, 4, 5, 8, 9

[93] Wenzhang Zhou, Dawei Du, Libo Zhang, Tiejian Luo, and Yanjun Wu. Multi-granularity alignment domain adaptation for object detection. In *CVPR*, 2022. 2, 8, 9

[94] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2223–2232, 2017. 2, 8

[95] Xinge Zhu, Jiangmiao Pang, Ceyuan Yang, Jianping Shi, and Dahua Lin. Adapting object detectors via selective cross-domain alignment. In *CVPR*, pages 687–696, 2019. 1, 2, 3, 8