IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 May 2023, accepted 17 July 2023, date of publication 24 July 2023, date of current version 1 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3298369

RESEARCH ARTICLE

EBCDet: Energy-Based Curriculum for Robust Domain Adaptive Object Detection

AMIN BANITALEBI-DEHKORDI[1], ABDOLLAH AMIRKHANI[2], (Senior Member, IEEE), AND ALIREZA MOHAMMADINASAB[2]

[1]Big Data and Intelligence Platform Laboratory, Huawei Technologies Canada Company Ltd., Markham, ON L3R 5A4, Canada
[2]School of Automotive Engineering, Iran University of Science and Technology, Tehran 16846-13114, Iran

Corresponding author: Abdollah Amirkhani (amirkhani@iust.ac.ir)

ABSTRACT This paper proposes a new method for addressing the problem of unsupervised domain adaptation for robust object detection. To this end, we propose an energy-based curriculum for progressively adapting a model, thereby mitigating the pseudo-label noise caused by domain shifts. Throughout the adaptation process, we also make use of spatial domain mixing as well as knowledge distillation to improve the pseudo-labels reliability. Our method does not require any modifications in the model architecture or any special training tricks or complications. Our end-to-end pipeline, although simple, proves effective in adapting object detector neural networks. To verify our method, we perform an extensive systematic set of experiments on: synthetic-to-real scenario, cross-camera setup, cross-domain artistic datasets, and image corruption benchmarks, and establish a new state-of-the-art in several cases. For example, compared to the best existing baselines, our Energy-Based Curriculum learning method for robust object Detection (EBCDet), achieves: 1-3 % AP50 improvement on Sim10k-to-Cityscapes and KITTI-to-Cityscapes, 3-4 % AP50 boost on Pascal-VOC-to- Comic, WaterColor, and ClipArt, and 1-5% relative robustness improvement on Pascal-C, COCO-C, and Cityscapes-C (1-2 % absolute mPC). Code is available at: https://github.com/AutomotiveML/EBCDet.

INDEX TERMS Object detection, domain adaptation, energy, model robustness, curriculum learning.

I. INTRODUCTION

Deep learning has helped create many strong object detector neural networks over the past decade. State-of-the-art detectors are being used in real-world applications, all the way from small efficient models on smartphones to large ensemble models on cloud clusters [1], [2]. In training neural networks for such applications, engineers and practitioners for the most part utilize existing public datasets or collect a limited labeled dataset for supervised training. In practice however, the labeled data collected for supervised training are often diverted from the environment that the model will eventually be deployed at, especially when there is little control on the environment such as outdoors, or when there are changes in the weather, location, lighting, or capturing sensors. In other words, target data distribution is shifted away from the source distribution. This is referred to as domain shift [3], [4], [5].

There are a wide range of existing techniques to mitigate the domain shift for object detection. Broadly speaking, these methods can be classified into a number of categorizes: augmentation based, domain alignment, reconstructions based, and self labeling. Augmentation based methods such as [6], [7], and [8] are only suitable for certain kinds of domain changes where the visual shifts can be manifested by image augmentation operations [9], [10]. Domain alignment methods are further divided to branches such as divergence-based [11], [12] or adversarial-based domain adaptation [13], [14], [15]. These methods are widely used for object detection domain adaptation [16], [17], [18], [19], [20], [21]. The goal in these methods is to align the intermediate representations of the source and target domains. However in doing so, they require non-trivial design changes or specialized modules such as gradient reversal layers, adversarial

The associate editor coordinating the review of this manuscript and approving it for publication was Xiangxue Li.

domain classifiers, etc. Reconstruction-based approaches use an auxiliary reconstruction task (e.g. Generative Adversarial Networks (GANs) [22] or Auto-Encoders) to synthesize a look of target domain images [23], [24], [25], [26], [27]. Just like the augmentation approaches, these methods also have a limited capability since target domain distribution can't exactly be reconstructed from the limited available information. Self labeling methods generate pseudo-labels for the unlabeled target data, and use those to adapt a model [28], [29], [30], [31]. However, depending on the severity of the domain shifts, pseudo-labels can become noisy and therefore unreliable.

In this paper, we propose an energy-based curriculum for progressively adapting a model, thereby mitigating the pseudo-label noise and domain shifts. Throughout the adaptation process, we also make use of our earlier ideas of spatial domain mixing as well as teacher guided knowledge distillation to improve the pseudo-labels reliability [32]. The benefits of our approach are three-fold: First, our method is data-centric, and therefore does not require any modifications in the model architecture or any special training tricks. Second, it is architecture agnostic so that it can be applied on different types of object detectors. Third, it can effectively adapt detectors without the need for labeled data from the target domain.

The main contributions of this paper can be summarized as:

- We introduce an energy-based method to partition the unlabeled target domain data to subsets in the order of divergence from the source distribution. This creates a curriculum that can further be used by domain adaptive object detection techniques.
- We build an end-to-end pipeline using our energy-based curriculum, teacher guided pseudo-label distillation, and spatial source-target domain mixing. Our framework progressively adapts a model from source domains to target domains, and results in adapted detectors with high accuracy.
- We conduct an extensive set of experiments to show the effectiveness of our method. Our results demonstrate the efficacy of the method on: synthetic-to-real scenario, cross-camera setup, cross-domain artistic datasets, and image corruption benchmarks, and establish a new state-of-the-art in several cases. Moreover, we perform extensive ablation studies and provide insights and discussions around different aspects of our method.

II. RELATED WORKS

This section reviews the relevant literature to our method.

A. IMAGE AUGMENTATION

Data augmentation is in general used to improve the generalization of neural networks. In the context of domain adaptation however, data augmentation techniques are also used to improve robustness against image corruptions. Authors in [7], [33], and [8] investigate the use of augmentations for image corruptions in image classification, and [6], [34] look at object detection. These efforts have resulted in partial success but fail to address the general problem. The main reason is due to inability of augmentation operations to generalize to various kinds of unseen corruptions. This was confirmed by [9] where the authors found that the perceptual similarity between augmentations and corruptions is a strong indicator of the corruptions error. In another study, authors in [10] observed that augmentations designed for synthetic corruptions do not necessarily work well for natural corruptions.

B. UNSUPERVISED DOMAIN ADAPTATION (UDA) FOR DETECTION

As mentioned in Section I, object detection UDA methods span over a diversity of approaches including domain alignment or reconstruction based techniques. Unlike image augmentation methods that do not use unlabeled target data, UDA attempts to leverage information available in the unlabeled target data for a better adaptation. Domain alignment methods (e.g. divergence-based or adversarial-based) learn to align the semantic representations of the source and target domains. For example, Domain Adaptive Faster-RCNN (DAF) [16] or Adversarial Feature Learning (AFL) [35] learn domain-invariant representations through adversarial training of object detectors. There are also methods that build on top of the traditional domain-invariant feature learning strategies. To this end, Selective Cross-Domain Alignment (SCDA) [18] incorporated a hierarchical alignment module; Coarse-to-Fine (C2F) [19] and C2FDA [36] performed coarse-to-fine feature adaptation; Strong-Weak Distribution Alignment (SWDA) [20] enforced strong local but weak global alignment; Every Pixel Matters (EPM) [21] designed a center-aware alignment framework for anchor-free FCOS model [37]. The alignment based methods have made a great progress, however, they usually come at a cost of adding extra modules and require non-trivial architectural manipulations.

In addition to the domain aligning, there is the family of reconstruction based methods in which a model learns to generate images similar to the unlabeled target examples. These methods usually involve a Conditional GAN [23], [38], [39], [40] or a stack of auto-encoders [41] for image synthesis, and transfer the source images to target-like images. The main issue with such methods is that image generation and style transfer have their own limitations. Although improvements have been observed, but some of the domain gap still remains.

Other than domain aligning and reconstruction based methods, batch-norm adaptation techniques are also worth a special mention. Batch-normalization (BN) layers play an important role in training convolutional neural networks for object detection. They reduce over-fitting, accelerate training, and allow a better convergence for deeper neural networks [42], [43]. In addition to the utilizations of BN layers mentioned above, BN-adaptation has also shown to be effective against image corruptions [44] and adversarial attacks [45], and in general useful in UDA [46], [47]. As we

will show in Section IV, BN-adaptation alone won't entirely close the domain gap in object detection UDA. That being said, we do use it as a part of our iterative gradual adaptation framework.

C. SELF PSEUDO-LABELING IN OBJECT DETECTION UDA

A variety of UDA methods use pseudo-labels for adaptation [29], [30], [31], [48], [49]. To this end, they generate pseudo-labels from the unlabeled target images and use those to train an adapted model, or to fine-tune a source-trained model. For object detection, pseudo-labels come in the form of bounding boxes and a class category assigned to each box [29]. Due to domain shifts however, often times the pseudo-labels are noisy, unreliable, and not confident. This may hurt the performance of the adapted model. To address this issue MTOR [48] used consistency regularization terms on region parts and graph-structures between a student and a mean teacher model. In another work, RLDA [30] modeled the region proposal distribution of Faster RCNN detector to reduce the pseudo-label noise (which makes RLDA tied to a specific architecture). Moreover, in [31], pseudo-labeling was combined with a method of style transfer. Overall, using high quality pseudo-labels is a promising direction since it can be combined with other techniques and be implemented in an architecture-agnostic manner.

D. ENERGY-BASED MODELS

Our method generates an adaptation curriculum based on the energy distribution of unlabeled target samples. Here, we provide a brief overview of the energy models. Energy-based models are alternatives to probabilistic decision-making, as they don't have a requirement for standard normalizations. As a result, energy-based methods avoid the problems associated with estimating the normalization constant in probabilistic models. The absence of the normalization condition further allows for a higher degree flexibility in designing machine learning algorithms [50], [51].

Energy-based analysis has been adopted in many applications. For example, [52] proposes to use energy values as a measure of out-of-distribution detection for image classification. Moreover, [53] designs a joint inference mechanism based on the energy values. In another recent work, [54] leveraged energy to build a framework for open-set object detection. In Section III, we provide an energy formulation based on the popular Helmholtz free energy definition [50], [51]. In our work, we relate the domain shifts to energy distributions and propose a method of creating a training curriculum from the unlabeled target images.

E. CURRICULUM LEARNING FOR OBJECT DETECTION

There exists a body of work related to curriculum learning for image classification [55], object detection [56], [57], [58], [59] and image segmentation [60], [61], [62], [63]. A general theme in these works is to partition the training examples to easy-vs-hard cases, start training with easy examples, and then continue the training with harder samples. The idea is that the models learn the detection task holistically from the easy examples, but then later learns to detect harder cases such as small or occluded objects from the difficult samples. In our work, instead of classifying each example as hard or easy, we partition the unlabeled target data to a number of subsets based on their energy distribution, and gradually adapt the model according to how much domain shift they exhibit with respect to the labeled source dataset.

In summary, compared to existing works, our method is simpler as it does not require training GANs for synthetic data generation, does not enforce an architecture change (data-centric), and does not pose a change in the loss definitions. Despite its simplicity, our method shows to be very effective on several domain adaptation scenarios and benchmarks.

III. METHOD

In this section we first provide a problem formulation and then explain the details of our method.

A. PROBLEM STATEMENT AND SETUP

As a whole, our problem follows an unsupervised domain adaptation setting, in which a source model θ^S, trained on source data $\mathcal{D}$ with distribution $p_S : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^+$, has a target test data $\overline{\mathcal{D}}$ with a distribution $p_T : \overline{\mathcal{X}} \times \mathcal{Y} \rightarrow \mathbb{R}^+$. In this case, $p_S(y|x) = p_T(y|x)$ but $p_S(x) \neq p_T(x)$. Note that in the case of images, both data distributions are still coming from the domain of images i.e. data point from either will be an image with d pixels, with each pixel value falling within 0 and 1; this part is the same even if the images are different.

For the task of object detection, dataset $\mathcal{D}$ will contain images and their corresponding bounding boxes and object classes/categories. Dataset $\overline{\mathcal{D}}$ however, contains only images from a target domain, which have a distribution shift/drift compared to images in $\mathcal{D}$. A common technique in such situations is to generate and use pseudo-labels from $\overline{\mathcal{D}}$. However, due to domain shift, pseudo-labels will be noisy and thus less reliable, and may result in an under-performing adaptation.

We propose a method for estimating the distribution shift, and then quantize the shift into a number of bins (See Fig. 1). These bins are calculated based on the free energy of representation vectors, and are then used for our curriculum-based gradual adaptation approach, which will be explained later in this section.

The energy function is defined as $E(x) : \mathbb{R}^d \rightarrow \mathbb{R}$ for a d-dimensional data example x and a scalar non-probabilistic energy value. Note that the energies are uncalibrated, as in they are measured in arbitrary units, and energies of two separately trained models cannot be combined. As such, it is a common practice to turn the energy values to probabilities (positive values between 0-1). A simple way to do that is via the Gibbs distribution. Other ways are also possible, but can be reduced to Gibbs by a suitable redefinition of energy [51]. The probability distribution of the set of energy values for an energy-based model according to the Gibbs distribution [50],

FIGURE 1. An example curriculum for the case of vehicle detection during different light conditions. Domain shift increases from left to right.

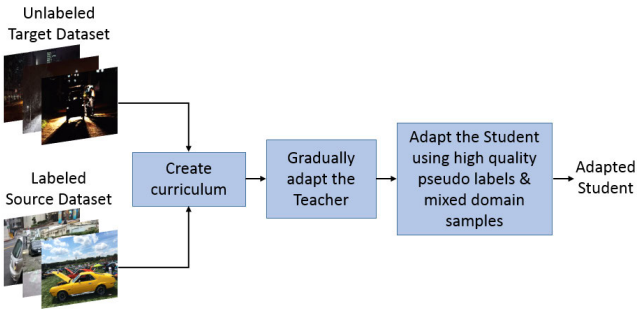

FIGURE 2. System overview: we take a labeled source dataset $\mathcal{D}$, an unlabeled target dataset $\overline{\mathcal{D}}$, and use our energy-based technique to create a curriculum, with which we then gradually adapt a model. For illustration purposes, we choose vehicle detection in daylight images as the source task, and vehicle detection at night as the target task.

[51], [52] is defined as follows:

$$p(y|x) = \frac{e^{-E(x,y)}}{\int e^{-E(x,y')}dy'}. \tag{1}$$

The denominator is referred to as the partition function and marginalizes over y (For simplicity, we dropped a temperature parameter that can otherwise exist in the exponents). We use the Helmholtz free energy [51], [52] in our method, which for a data point x is defined as:

$$F(x) = -log\Big(\int e^{-E(x,y')}dy'\Big). \tag{2}$$

With this setup and definitions, we describe our method next.

B. ENERGY-BASED CURRICULUM FOR OBJECT DETECTION

Fig. 2 shows a big picture overview of our approach. As observed, we take a labeled source dataset $\mathcal{D}$ and an unlabeled target dataset $\overline{\mathcal{D}}$ and use our energy-based technique to create a curriculum, with which we then gradually adapt a model. Our system supports self-adaptation as well as teacher-student adaptation. In self-adaptation, we use one fixed model architecture throughout all steps of our method. However, as we show in our experiments, smaller models with low learning capacity may not have enough power to adapt on their own. For such 'Student' models, we first use a larger 'Teacher' architecture with higher capacity to adapt and generate higher quality pseudo-labels, and then use these pseudo-labels to train the 'Student' model. Next, we describe the three major components of our method: creating curriculum, gradually adapting the Teacher, and adapting the Student.

1) ENERGY-BASED CURRICULUM

As shown in the literature [52], [53], the negative free energy can be used for out-of-distribution (OOD) detection. This is achieved by interpreting data samples with a low likelihood in the data density function as OOD, i.e.:

$$p(x) = \frac{e^{-F(x;S)}}{Z}, \tag{3}$$

where $F(x; S)$ denotes the free energy for the student model θ^S, and $Z = \int e^{-F(x;S)}dx$ defines the normalized densities, which can be intractable to compute or estimate [52], [53]. Taking the logarithm of (3) yields:

$$log\big(p(x)\big) = -F(x; S) - log(Z). \tag{4}$$

In (4), $log(Z)$ is constant with respect to x, therefore, the likelihood of a sample being OOD becomes directly related to its negative free energy. In other words, the negative free energy can be used as an indicator of distribution shift. To this end, we first train a Teacher neural network model using an ordinary supervised training algorithm over the source data, and then compute the energy over the representation vector of the last layer (logits) for the unlabeled target examples. Based on their free energy values, we divide them into a number of n_c equally sized bins. The early bins attain low negative energy and are thus considered as highly shifted samples. Towards the end bins on the other hand have a distribution that is closest to the source distribution. Fig. 1 provides a visualization of a curriculum generated for the example of vehicle detection during day (source) versus night (target). Fig. 3 shows a flow-diagram of the curriculum generation procedure.

a: CLASSIFICATION TASKS

It is also worth noting that the formulation of free energy will slightly change depending on the task. For a classification network with C target classes, a categorical distribution with softmax is used. Therefore:

$$p(y|x) = \frac{e^{S_y^c(x)}}{\sum_i^C e^{S_i^c(x)}}, \tag{5}$$

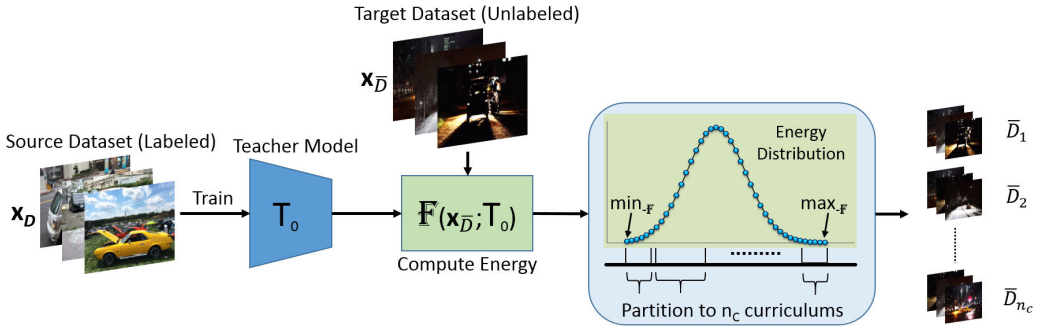

FIGURE 3. Creating curriculum: We use a source-trained Teacher model, T_0, to compute the energy values over the unlabeled target dataset. Then, we quantize the sorted energy values to n_c bins, thereby sorting the unlabeled data according to the domain shift with respect to the source data.

where $S^c_y(x)$ denotes the logit (probability) of the y-th class and $S^c(x) : \mathbb{R}^d \rightarrow \mathbb{R}^C$. The energy for a given input (x, y) in this case is defined as $E(x, y) = -S^c_y(x)$, and the free energy function $F^c(x; S^c)$ is then expressed similar to (2) as:

$$F^c(x; S^c) = -log \sum_i^C e^{S^c_i(x)}. \tag{6}$$

b: REGRESSION TASKS

In this case, $S^r(x) : \mathbb{R}^d \rightarrow \mathbb{R}$, and $E(x, y) = -S^r(x, y)$. The conditional density can be expressed by:

$$p(y|x; S^r) = \frac{e^{S^r(x,y)}}{\int e^{S^r(x,y')}dy'}. \tag{7}$$

And the free energy is defined by:

$$F^r(x; S^r) = -log \left(\int e^{S^r(x,y')}dy' \right). \tag{8}$$

It is worth noting that the above formulations follow the typical definitions of the literature [52], [53]. However, other variations can also be incorporated.

c: OBJECT DETECTION TASKS

Object detection architectures usually involve with a bounding box regression output and a class/category prediction output [64], [65], [66], [67]. As such, it will involve both the classification and regression formulations mentioned above. The total energy is therefore expressed by:

$$\begin{aligned} F^o(x; S^c, S^r) &= F^c(x; S^c) + F^r(x; S^r) \\ &= \frac{-\sum_b^{n_b} log \sum_i^C e^{S^c_{b,i}(x)}}{n_b} \\ &\quad + \frac{-\sum_b^{n_b} \sum_j^4 log \int e^{S^r_{b,j}(x,y')}dy'}{4n_b}, \end{aligned} \tag{9}$$

where $S^c_{b,i}$ is the classifier's output for the i-th class label of the b-th bounding box, $S^r_{b,j}$ is the regression output for the j-th value of the b-th bounding box, $b \in [1, n_b]$, $i \in [1, C]$, $j \in [1, 4]$, and n_b denotes the total number of bounding boxes over C categories.

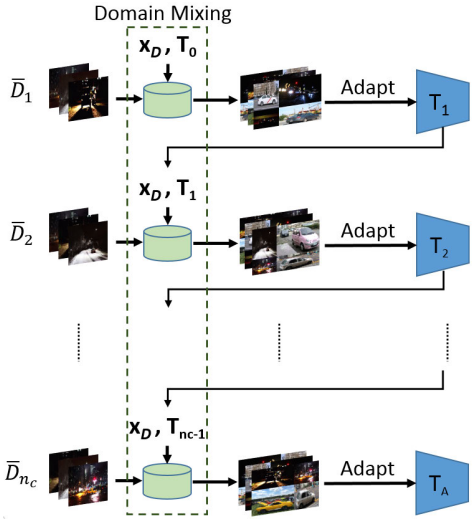

FIGURE 4. Adapting teacher: within each iteration, first we extract pseudo-labels of the corresponding data partition. Then, we generate mixed domain examples from source and target data. At last, batch normalization layers of the Teacher are updated, before going to the next iteration.

2) GRADUAL ADAPTATION USING THE GENERATED CURRICULUM

Once the unlabeled data $\overline{\mathcal{D}}$ is partitioned according to our energy-based curriculum, we then use partitions in a sequential manner to gradually adapt the Teacher model. To this end, as illustrated in Fig. 4, we start adapting the teacher with the least shifted data partition, $\overline{\mathcal{D}}_1$. In doing so, we first generate pseudo-labels of $\overline{\mathcal{D}}_1$ using the current Teacher, T_0. Then, we create mixed domain collages of both labeled and unlabeled images. Next, we adapt the batch normalization (BN) layers of T_0 using the mixed collages and their labels/pseudo-labels. The updated teacher is called T_1. This process will be repeated for the rest of the curriculum until the final teacher T_A is achieved. T_A has been gradually updated with the target domain images, and its pseudo-labels are gradually improved compared to the preceding teachers.

a: DOMAIN MIXED COLLAGES

We created these collages from source and target images in order to organically mix their distributions, thus making it

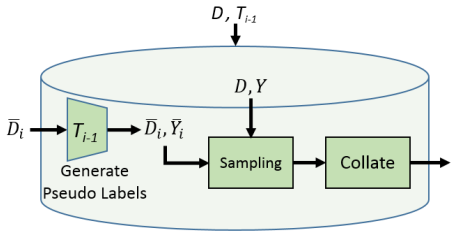

FIGURE 5. Mixing domains: For an improved adaptation, we create a mixed collage of source and target images and corresponding labels/pseudo-labels.

FIGURE 6. Domain mixed examples: In each case, bounding box labels of the source data (in yellow) are mixed with the bounding box pseudo-labels of the target data (in red), to create a collage image.

easier for models to adapt. In our experiments, we use a 2×2 collage. To obtain the training labels for a collage image, we use the ground-truth labels of source examples, and pseudo-labels of target examples. Moreover, we leverage a weighted balanced sampling strategy to take into account the fact that source data size might be much different than target data size. Fig. 5 shows a schematic of how these mixed domain examples are generated and Fig. 6 demonstrates several examples. Note that our sample mixing is more effective than [32] in that we mix samples iteratively for each data split (due to model updates pseudo-labels get more accurate), versus [32] mixes samples only once.

b: ADAPTING BN LAYERS

In each round of Teacher adaptation, we freeze all parameters, except the batch normalization (BN) layers, and use the mixed domain examples to fine-tune the BN layers. This helps the Teacher to adapt better to target examples. Note that this is not a new idea and has been used for domain adaptation before [44], [46], [47], [68], [69]. The key idea is that in UDA (unsupervised domain adaptation), the model was already trained with ground truth labels on the source data, and "knows" about the task of object detection, but has a difficulty of handling images from the shifted distribution. In other words, batch-norm parameters are used as proxy parameters to shift back the distribution to an interval similar to the source distribution so the model can do as good of a job as on the source data. This approach while simplistic, is effective and hence it is adopted in the literature. A side benefit of updating only BN layers instead of the whole NN is to save on computations. The implementations of domain mixing and BN adaptation described above have been adopted from our earlier work [32].

3) ADAPTING THE STUDENT USING MIXED DOMAIN EXAMPLES AND HIGH QUALITY PSEUDO-LABELS

The final adapted Teacher T_A in general will have a strong adaptation performance due to its high capacity and the gradual adaptation explained previously in this section. We use T_A to generate high quality pseudo-labels over $\overline{\mathcal{D}}$, and then mix those with $\mathcal{D}$. The Student S_0 is next fine-tuned with these mixed examples to generate the final adapted Student S_A. Fig. 7 shows these steps, and Algorithm 1 summarizes the proposed method.

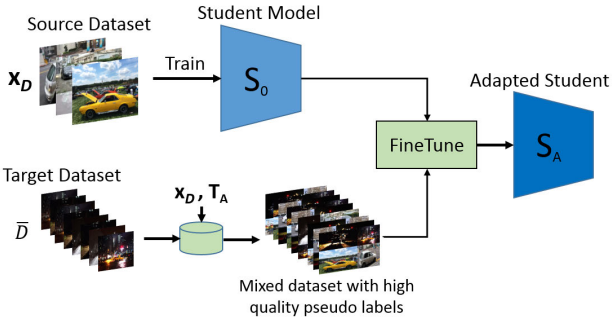

FIGURE 7. Adapting student: We first generate the pseudo-labels of the unlabeled target dataset using the adapted Teacher model T_A. Then, we fine-tune the source-trained Student with the resulting domain-mixed examples to achieve the final adapted Student model S_A.

IV. EXPERIMENTS RESULTS

We review the experiment results and ablation studies in this section. The main results are organized in three parts:

- Synthetic-to-real & cross-camera domain shifts.
- Cross-domain artistic domain adaptation.
- Robustness against image corruptions.

These results are followed by ablation studies to understand different aspects of our approach. Note that for real-world applications, it is particularly challenging to collect real data that generalizes to diverse situations. With the datasets chosen for the experiments, we show that our method can not only adapt from real data (captured with different setup/sensors), but can also adapt from synthetic data, and even paintings, cartoons, or clip-arts.

A. TRAINING PROCEDURE AND HYPER-PARAMETERS

For the experiments, we used the YOLOv5 [1] architecture, at different scales or input resolutions. For the most part we used the default hyper-parameters in the YOLOv5 repo [1], but tuned the learning rate. We employed a standard SGD optimizer with momentum 0.937, weight decay

Algorithm 1 Energy-Based Curriculum for Robust Obj. Detection

Inputs: Labeled source data $\mathcal{D}$, unlabeled target data $\overline{\mathcal{D}}$, Student model θ^{S_0} (trained on $\mathcal{D}$), adaptation epochs e_a, fine-tuning epochs e_{ft}, number of curriculum partitions n_c.
Output: Adapted Student model θ^{S_A}

procedure EBCDet ($\theta^{S_0}, \mathcal{D}, \overline{\mathcal{D}}, e_a, e_{ft}, n_c$)
  # Create a curriculum
    Choose a Teacher architecture θ^T (with capacity $\geqslant \theta^S$)
    $\theta^{T_0} \leftarrow$ Train θ^T supervised on $\mathcal{D}$
    $F^o(\overline{\mathcal{D}}) \leftarrow$ `ComputeEnergy`$(\overline{\mathcal{D}}; \theta^{T_0})$ according to (9)
    $\{\overline{\mathcal{D}}_i\}_{i=1}^{n_c} \leftarrow$ `PartitionBasedOnEnergy`$(\overline{\mathcal{D}}; F^o(\overline{\mathcal{D}}))$
  # Adapt the Teacher model
    for $l \in \theta^{T_0}$.layers **do**
      if l is not BatchNorm **then** `FreezeLayer`(l)
    for $i \in \{1, 2, \ldots, n_c\}$ **do**
      $\mathcal{D}^i_{mixed} \leftarrow$ `MixSamples`$(\mathcal{D}, \overline{\mathcal{D}}_i, \theta^{T_{i-1}})$ w.r.t. Fig. 5
      $\theta^{T_i} \leftarrow$ `MinimizeLoss`$(\theta^{T_{i-1}}, \mathcal{D}^i_{mixed}, e_a)$
    $\theta^{T_A} \leftarrow \theta^{T_{n_c}}$ ▷ Adapted Teacher
  # Adapt the Student
    $\mathcal{D}_{mixed} \leftarrow$ `MixSamples`$(\mathcal{D}, \overline{\mathcal{D}}, \theta^{T_A})$ w.r.t. Fig. 5
    $\theta^{S_A} \leftarrow$ `MinimizeLoss`$(\theta^{S_0}, \mathcal{D}_{mixed}, e_{ft})$

$5e^{-4}$, warmup, and cosine decay. For the object detection hyper-parameters, we used a NMS IoU threshold of 0.65, and confidence threshold of 0.001 for training and 0.4 for pseudo-label generation. Following [1], we incorporated the generalized IoU (GIoU), focal, and objectness losses.

We provide various experiment results on several datasets including Pascal-VOC [70], Microsoft COCO [71], Cityscapes [72], Sim10k [73], KITTI [74], and the ClipArt1k, WaterColor2k and Comic2k datasets [31]. In all cases, we used a default $n_c = 5$ partitioning, and 40 epochs for each adaptation iteration (an ablation study on n_c is available later in this section). Moreover, the fine-tuning of Student models was done in 100 epochs with a batch size of 128 and learning rate of $4e^{-5}$. In addition, for baseline COCO models, we trained for 300 epochs (similar to [1]) and used a learning rate of 0.01. Pascal and Cityscapes baseline models were transfer learned on top of the COCO models with the same strategy as the Student fine-tuning.

It is also worth noting that we opted for single resolution training for simplicity. Unless otherwise specified, the default resolution was set at 416 (max width). However, we also examined higher/lower resolutions to explore the potential of our models at different capacities. For example, YOLOv5 S320 denotes a small scale model at 320 resolution, whereas YOLOv5 X1280 represents a larger model with 1280 size [1].

B. SYNTHETIC-TO-REAL & CROSS-CAMERA EXPERIMENTS

1) DATASETS

For these experiments, we employ the widely used settings of Sim10k [73] and KITTI [74] datasets adapted to the Cityscapes [72] dataset. Sim10K-to-Cityscapes signifies the synthetic to real domain adaptation, and KITTI-to-Cityscapes evaluates the cross-camera adaptation. We followed the common practice and used the `car` class for comparison with existing methods. In these two experiments, we used the training set of KITTI and Sim10K as labeled source datasets, training set of Cityscapes as unlabeled target dataset, and validation set of Cityscapes as the target test set.

2) METRICS OF PERFORMANCE

To facilitate a fair comparison, methods are grouped based on their 'Source' performance/capacity (i.e. AP50 of the source-trained non-adapted model). We then compare the performance of the adapted models on the target test set. Metrics of comparison are the AP50, absolute gain τ, and effective gain ρ defined as follows:

$$\tau = \text{AP}^{50}(\theta^{S_A}) - \text{AP}^{50}(\theta^{S_0}), \tag{10}$$

$$\rho = 100 \times \frac{\text{AP}^{50}(\theta^{S_A}) - \text{AP}^{50}(\theta^{S_0})}{\text{AP}^{50}(\text{Oracle}) - \text{AP}^{50}(\theta^{S_0})}, \tag{11}$$

where 'Oracle' (upper-bound) is a model that is directly trained on the target data with ground-truth labels. Note that the AP50 (or mAP in general) is the standard metric used for the assessment of object detection models. We followed [19], [32] in using the τ metric to measure the absolute gains. This metric was designed to enable a comparison between models that had a different source performance (due to the diversity of architectures, many different models have been developed for similar tasks). τ evaluates the gains achieved only due to the adaptation algorithm. Furthermore, ρ was introduced in [32], to measure effective/relative adaptation gains. This metric gives a perspective of how much of the domain gap can be closed by an adaptation algorithm. A $\rho = 0\%$ denotes no adaptation gains, and a $\rho = 100\%$ means the target performance is as high as the Oracle.

3) RESULTS

Table 1 shows the results of the Sim10K-to-Cityscapes experiment. We observe from Table 1 that our Energy-Based Curriculum learning method for object Detection, EBCDet , performs competitively compared to the state-of-the-art approaches. We used model scales that achieve similar source AP50 to groups of existing methods, to be able to draw a fair comparison. Similarly, Table 2 shows the results of the KITTI-to-Cityscapes experiment. We observe that EBCDet outperforms the baselines across the Adapted AP50, absolute, and effective gains.

C. CROSS-DOMAIN ARTISTIC EXPERIMENTS

1) DATASETS

These experiments include three datasets of WaterColor2k (watercolor paintings), Comic2k (comic strips), and ClipArt1k (clipart images), introduced in [31]. We investigated the adaptation from a natural image dataset such as the VOC to each one of these datasets. To this end, VOC07 trainval is used as the labeled source data, and the training set of WaterColor2k, Comic2k, and ClipArt1k datasets are used as the unlabeled target data. Models are evaluated on the test set of these three datasets.

TABLE 1. Sim10K-to-Cityscapes results: to facilitate a fair comparison, methods are grouped based on their 'Source' performance. We compare the performance of the adapted models on the target test data. Metrics of comparison are the AP50, absolute gain τ as in (10), and effective gain ρ as in (11). In addition, we report the AP50 of an upper-bound 'Oracle', a model that is directly trained on the target data with ground-truth labels. 'S320', 'M416', 'X640', 'X1280' represent different scales of Yolov5 architecture with increasing depth, width, and input resolution. Note that some methods such as [57] can operate in multiple ways (variations on how curriculum is defined in this case). In such cases, we added a separate entry in the table for each variation.

Method	Arch.	Backbone	Source	Adapted AP50	Oracle	τ	ρ
DAF [16]	F-RCNN	VGG16	30.10	39.00	-	8.90	-
MAF [17]	F-RCNN	VGG16	30.10	41.10	-	11.00	-
DALocNet [49]	F-RCNN	CustomVGG16	**31.71**	42.34	-	10.63	-
RLDA [30]	F-RCNN	Inception-v2	31.08	42.56	68.10	11.48	31.01
Curriculum Self-Paced [57]	F-RCNN	ResNet50	30.67	47.68	62.73	17.01	53.05
[57] w/ Random Curriculum	F-RCNN	ResNet50	30.67	46.84	62.73	16.17	50.04
[57] w/ Difficulty Predictor Curriculum	F-RCNN	ResNet50	30.67	47.02	62.73	16.35	50.99
[57] w/ Domain Discriminator Curriculum	F-RCNN	ResNet50	30.67	45.76	62.73	15.09	47.06
EBCDet (teacher ResNet152)	F-RCNN	ResNet50	30.67	**48.43**	62.73	**17.76**	**55.39**
SCDA [18]	F-RCNN	VGG16	34.00	43.00	-	9.00	-
MDA [75]	F-RCNN	VGG16	34.30	42.80	-	8.50	-
SWDA [20]	F-RCNN	VGG16	34.60	42.30	-	7.70	-
C2FDA [36]	F-RCNN	ResNet50	34.60	**48.90**	57.30	13.1	57.71
UaDAN [76]	F-RCNN	ResNet50	34.6	48.6	-	**14**	-
RPA [77]	F-RCNN	VGG16	34.6	45.7	60	11.1	43.7
SDA [40]	CenterNet	DLA-34	**35.80**	45.80	60.0	10	41.3
Coarse-to-Fine [19]	F-RCNN	VGG16	35.00	43.80	59.90	8.80	35.34
SimROD [32]	YOLOv5	S320	33.62	38.73	48.81	5.11	33.66
EBCDet (self-adapt)	YOLOv5	S320	33.62	41.94	48.81	8.32	54.77
EBCDet (w. teacher X640)	YOLOv5	S320	33.62	45.88	48.81	12.26	**80.71**
MTOR [48]	F-RCNN	ResNet50	39.40	46.60	-	7.20	-
EveryPixelMatters [21]	FCOS	VGG16	**39.80**	49.00	69.70	9.20	30.77
SIGMA [78]	FCOS	VGG16	**39.80**	53.7	69.70	13.9	46.49
SimROD [32]	YOLOv5	S416	39.57	44.21	56.49	4.63	27.37
EBCDet (self adapt)	YOLOv5	S416	39.57	47.05	56.49	7.48	44.21
EBCDet (w. teacher X1280)	YOLOv5	S416	39.57	**53.98**	56.49	**14.41**	**85.17**
SimROD [32]	YOLOv5	X1280	71.66	75.94	82.90	4.28	38.08
Ours: EBCDet (self-adapt)	YOLOv5	X1280	71.66	**77.31**	82.90	**5.65**	**50.27**

TABLE 2. KITTI-to-cityscapes adaptation results.

Method	Arch.	Backbone	Source	Adapted AP50	Oracle	τ	ρ
DAF [16]	F-RCNN	VGG	30.20	38.50	-	8.30	-
MAF [17]	F-RCNN	VGG	30.20	41.00	-	10.80	-
RLDA [30]	F-RCNN	Inception-v2	31.10	42.98	68.10	11.88	32.11
PDA [25]	F-RCNN	VGG	30.20	43.90	55.80	13.70	53.52
SGA-S [59]	F-RCNN	ResNet101	30.22	43.07	NA	12.85	NA
Curriculum Self-Paced [57]	F-RCNN	ResNet50	31.52	43.86	62.73	12.34	39.53
[57] w/ Random Curriculum	F-RCNN	ResNet50	31.52	41.52	62.73	10.00	32.04
[57] w/ Difficulty Predictor Curriculum	F-RCNN	ResNet50	31.52	42.13	62.73	10.61	33.99
[57] w/ Domain Discriminator Curriculum	F-RCNN	ResNet50	31.52	40.22	62.73	8.7	27.87
SimROD [32]	YOLOv5	S416	31.61	35.94	56.15	4.33	17.65
EBCDet (self-adapt)	YOLOv5	S416	31.61	38.22	56.15	6.61	26.94
EBCDet (w. teacher X1280)	YOLOv5	S416	**31.61**	**47.11**	56.15	**15.50**	**63.16**
SCDA [18]	F-RCNN	VGG	37.40	42.60	-	5.20	-
EveryPixelMatters [21]	FCOS	ResNet50	35.30	45.00	70.40	9.70	27.64
SIGMA [78]	FCOS	ResNet50	35.30	45.8	70.40	10.5	29.91
SimROD [32]	YOLOv5	M416	36.09	42.94	59.29	6.85	29.51
C2FDA [36]	F-RCNN	ResNet50	**37.60**	48.00	57.30	10.40	**52.79**
EBCDet (self adapt)	YOLOv5	M416	36.09	43.88	59.29	7.79	33.58
EBCDet (w. teacher X1280)	YOLOv5	M416	36.09	**48.30**	59.29	**12.21**	52.63
SimROD [32]	YOLOv5	X1280	52.07	58.25	82.50	6.18	20.31
Ours: EBCDet (self-adapt)	YOLOv5	X1280	52.07	**59.67**	82.50	**7.60**	**24.98**

2) METRICS OF PERFORMANCE

Similar to Section IV-B, we used the adapted AP50, τ, and ρ for performance assessment.

3) RESULTS

Table 3-5 show the results of these experiments for the WaterColor2k, ClipArt1k, and Comic2k datasets, respectively. Our method shows a solid performance on the three benchmarks, reaching effective gains of around 98%, 75%, and 67%, for WaterColor, ClipArt, and Comic datasets, respectively. This corresponds to the adapted AP50 improvements of +1.64%, +2.49%, and +2.16% over the state-of-the-art [32].

D. IMAGE CORRUPTION EXPERIMENTS

1) DATASETS

Image corruption datasets were introduced to understand how robust neural networks are against common corruptions [6], [10], [33], [82]. To this end, over a dozen common distortions

TABLE 3. Real (VOC)-to-WaterColor2K results: Methods implemented on the same architecture achieve a same source AP50.

Method	Arch.	Backbone	Source	Adapted AP50	Oracle	τ	ρ
DAF [16]	F-RCNN	VGG	39.80	34.30	NA	-5.50	NA
DAM [24]	F-RCNN	VGG	**39.80**	52.00	NA	12.20	NA
DeepAugment [8]	YOLOv5	S416	37.46	45.19	56.07	7.73	41.54
BN-Adapt [42]	YOLOv5	S416	37.46	45.72	56.07	8.26	44.39
Stylize [79]	YOLOv5	S416	37.46	46.26	56.07	8.80	47.29
STAC [29]	YOLOv5	S416	37.46	49.83	56.07	12.37	66.47
DT+PL [31]	YOLOv5	S416	37.46	44.86	56.07	7.40	39.77
SimROD [32]	YOLOv5	S416	37.46	52.58	56.07	15.12	81.26
EBCDet (self-adapt)	YOLOv5	S416	37.46	53.76	56.07	16.3	87.59
EBCDet (teacher X416)	YOLOv5	S416	37.46	**55.87**	56.07	**18.41**	**98.93**
ADDA [80]	SSD	VGG	49.60	49.80	58.40	0.20	2.27
DT+PL [31]	SSD	VGG	**49.60**	54.30	58.40	4.70	53.41
SWDA [20]	F-RCNN	VGG	44.60	56.70	58.60	12.10	86.43
AFL [35]	YOLOv3	DarkNet53	44.3	49.2	-	4.9	-
SGA-S [59]	F-RCNN	ResNet101	44.78	55.30	NA	10.52	NA
DeepAugment [8]	YOLOv5	M416	46.95	54.02	66.34	7.07	36.47
BN-Adapt [42]	YOLOv5	M416	46.95	55.75	66.34	8.80	45.39
Stylize [79]	YOLOv5	M416	46.95	55.24	66.34	8.29	42.76
STAC [29]	YOLOv5	M416	46.95	57.82	66.34	10.87	56.07
DT+PL [31]	YOLOv5	M416	46.95	49.14	66.34	2.19	11.30
SimROD [32]	YOLOv5	M416	46.95	60.08	66.34	13.13	67.72
EBCDet (self-adapt)	YOLOv5	M416	46.95	61.72	66.34	14.77	76.17
Ours: EBCDet (teacher X416)	YOLOv5	M416	46.95	**64.03**	66.34	**17.08**	**88.09**

TABLE 4. Real (VOC)-to-ClipArt1k adaptation results.

Method	Arch.	Backbone	Source	Adapted AP50	Oracle	τ	ρ
ADDA [80]	SSD	VGG	26.80	27.40	55.40	0.60	2.10
DT+PL [31]	SSD	VGG	**26.80**	**46.00**	55.40	**19.20**	**67.13**
DAF [16]	F-RCNN	VGG	26.20	22.40	50.00	-3.80	-15.97
DT+PL [31]	F-RCNN	VGG	26.20	34.90	50.00	8.70	36.55
DALocNet [49]	F-RCNN	CustomVGG16	25.12	31.55	-	6.43	-
SWDA [20]	F-RCNN	VGG	**27.80**	38.10	50.00	10.30	46.40
SUDA [81]	F-RCNN	ResNet101	-	39.2	-	-	-
Curriculum Self-Paced [57]	F-RCNN	ResNet50	26.14	37.83	33.89	11.69	**150.8**
DAM [24]	F-RCNN	VGG	24.90	**41.80**	50.00	**16.90**	67.33
C2FDA [36]	F-RCNN	ResNet50	27.0	40.90	-	13.9	-
DeepAugment [8]	YOLOv5	S416	29.32	31.65	56.07	2.33	8.71
BN-Adapt [42]	YOLOv5	S416	29.32	37.43	56.07	8.11	30.32
Stylize [79]	YOLOv5	S416	29.32	38.80	56.07	9.48	35.44
STAC [29]	YOLOv5	S416	29.32	39.64	56.07	10.32	38.58
DT+PL [31]	YOLOv5	S416	29.32	39.49	56.07	10.17	38.02
SimROD [32]	YOLOv5	S416	29.32	41.28	56.07	11.96	44.72
EBCDet (self-adapt)	YOLOv5	S416	29.32	43.77	56.07	14.45	54.02
Ours: EBCDet (teacher X416)	YOLOv5	S416	29.32	**49.61**	56.07	**20.29**	**75.85**
UaDAN [76]	F-RCNN	ResNet50	31	40.2	-	9.2	-
AFL [35]	YOLOv3	DarkNet53	37.6	43.7	-	6.1	-

such as Gaussian/shot/impulse noise, Defocus/motion/zoom blur, snow, frost, fog, rain, brightness/contrast modifications or JPEG compression were emulated and applied on top of existing datasets. It was shown that models trained on clean data (i.e. source data) are not very robust against the corruptions. We treat these datasets as a kind of domain/distribution shift, and apply our method to adapt the models trained on clean data, to corrupt target data.

We performed experiments on Pascal-C, COCO-C, and Cityscapes-C datasets with the 15 standard types of corruptions [6], [33]. We used a similar experiment setting as [32] where for Pascal-C, the VOC07 trainval was used as labeled source data, corrupt VOC12 trainval as unlabeled target data, and corrupt VOC07 test set as the target test data. For COCO-C, we used the first half of the train set as the labeled source data, the second half (corrupt) as unlabeled target data, and the validation set (corrupt) as the target test data. We follow a similar approach of halving the train set for Cityscapes as well. We choose images from the following cities as source labeled set: *'aachen, bremen, cologne, darmstadt, hanover, jena, krefeld, stuttgart, and tubingen'*. The rest of the cities constitute the (corrupt) unlabeled target data, and the Cityscapes validation set (corrupt) is used as the target test set.

2) METRICS OF PERFORMANCE

Since image corruption benchmarks use different kinds of corruptions, the experiments are reptead for different kinds of corruptions and severity levels and are therefore very extensive. The metrics of performance are also slightly different

TABLE 5. Real (VOC)-to-Comic2k adaptation results.

Method	Arch.	Backbone	Source	Adapted AP50	Oracle	τ	ρ
ADDA [80]	SSD	VGG	24.90	23.80	46.40	-1.10	-5.12
DT [31]	SSD	VGG	24.90	29.80	46.40	4.90	22.79
DT+PL [31]	SSD	VGG	**24.90**	**37.20**	46.40	**12.30**	**57.21**
DAF [16]	F-RCNN	VGG	21.40	23.20	-	1.80	-
DT [31]	F-RCNN	VGG	21.40	29.80	-	8.40	-
SWDA [20]	F-RCNN	VGG	21.40	28.40	-	7.00	-
DAM [24]	F-RCNN	VGG	**21.40**	**34.50**	-	**13.10**	-
DeepAugment [8]	YOLOv5	S416	18.19	21.39	39.81	3.20	14.80
BN-Adapt [42]	YOLOv5	S416	18.19	25.53	39.81	7.34	33.95
Stylize [79]	YOLOv5	S416	18.19	27.57	39.81	9.38	43.39
STAC [29]	YOLOv5	S416	18.19	26.40	39.81	8.21	37.97
DT+PL [31]	YOLOv5	S416	18.19	25.66	39.81	7.47	34.55
SimROD [32]	YOLOv5	S416	18.19	29.54	39.81	11.35	52.50
EBCDet (self-adapt)	YOLOv5	S416	18.19	**32.69**	39.81	**14.50**	**67.07**
DeepAugment [8]	YOLOv5	M416	23.58	27.65	49.13	4.07	15.93
BN-Adapt [42]	YOLOv5	M416	23.58	32.04	49.13	8.46	33.11
Stylize [79]	YOLOv5	M416	23.58	34.56	49.13	10.98	42.97
STAC [29]	YOLOv5	M416	23.58	32.76	49.13	9.18	35.93
DT+PL [31]	YOLOv5	M416	23.58	33.53	49.13	9.95	38.94
SimROD [32]	YOLOv5	M416	23.58	37.93	49.13	14.35	56.15
Ours: EBCDet (self-adapt)	YOLOv5	M416	23.58	**40.09**	49.13	**16.51**	**64.62**

than the the ones used in Section IV-B-IV-C. Following [10], [33], and [32], we use the following metrics:

- mPC: mean performance under corruption
- rPC: relative performance under corruption
- τ_c: relative robustness

averaged over K corruption types, and defined as:

$$\text{mPC} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{N_s}\sum_{s=1}^{N_s}\text{AP}_{k,s}, \tag{12}$$

$$\text{rPC} = \frac{\text{mPC}}{\text{AP}_{\text{clean}}}, \tag{13}$$

$$\tau_c = \text{mPC}(\theta^{S_A}) - \text{mPC}(\theta^{S_0}), \tag{14}$$

where $\text{AP}_{k,s}$ denotes the test average precision with corruption type k at severity level s.

3) RESULTS

Table 6-8 show the results of the image corruption experiments. In addition to our method, we adopt the results of the baselines from [32]. Note that the BN-adapt method only adapts the batch normalization layers, and we observe from the results that this is not enough. Also, augmentation methods such as DeepAugment [8] can't address the corruption domain shifts as shown in the results. For our method, we employed a YOLOv5X Teacher in these experiments for high quality pseudo-label generation on the unlabeled target dataset, and a YOLOv5M Student. Results of the corruption experiments show a consistent strength for our method across Pascal-C, COCO-C, and Cityscapes-C datasets.

E. ABLATION STUDIES

In this section we study different aspects and components of our method, and provide insights/discussions on the results.

TABLE 6. Results on the Pascal-C benchmark: metrics defined in IV-D.

Method	$\text{AP}^{50}_{\text{clean}}$	mPC^{50}	rPC	τ_c	ρ
Source	83.13	53.78	64.69	0.00	0
Stylize	84.79	62.92	74.21	9.14	36.62
BN-Adapt	83.01	64.60	77.82	10.82	43.35
DeepAugment	85.05	64.88	76.28	11.10	44.47
STAC	**87.00**	66.88	76.87	13.10	52.48
SimROD	86.97	75.40	86.70	21.62	86.62
EBCDet (ours)	86.97	**76.34**	**87.78**	**22.56**	**90.38**
Oracle	86.75	78.74	90.77	24.96	100

TABLE 7. Results on the COCO-C benchmark.

Method	AP_{clean}	mPC	rPC	τ_c	ρ
Source	**36.85**	22.03	59.78	0.00	0
Stylize	35.75	23.82	66.63	1.79	22.02
BN-Adapt	36.24	24.79	68.41	2.76	33.95
DeepAugment	35.51	24.33	68.52	2.30	28.29
STAC	36.76	24.80	67.46	2.77	34.07
SimROD	36.79	28.46	77.36	6.43	79.09
EBCDet (ours)	36.79	**29.13**	**79.18**	**7.10**	**87.33**
Oracle	36.23	30.16	83.25	8.13	100

TABLE 8. Results on the Cityscapes-C dataset.

Method	AP_{clean}	mPC	rPC	τ_c	ρ
Source	19.48	11.53	59.19	0.00	0
Stylize	21.77	14.62	67.16	3.09	25.81
DeepAugment	20.28	14.79	72.93	3.26	27.23
STAC	**24.54**	15.39	62.71	3.86	32.25
SimROD	24.06	18.01	74.85	6.48	54.14
EBCDet (ours)	24.06	**19.27**	**80.09**	**7.74**	**64.66**
Oracle	26.58	23.50	88.41	11.97	100

1) CONTRIBUTION OF DIFFERENT COMPONENTS

Table 9 provides an ablation study on the different components of our method, on the Pascal-C with YOLOv5M. We observe that:

TABLE 9. Ablation study on the Pascal-C dataset with yolov5m. Results show the added benefits of different components. TG, GA, DM, and FT refer to Teacher Guidance, Gradual Adaption, Domain Mixing, and Fine Tuning. '1-cycle' adaptation means no curriculum learning was used.

Method	TG	DM	GA	FT	mPC^{50}	τ_c
Source					53.78	0.0
Curriculum only			curriculum		63.39	9.6
BN-Adapt			1-cycle		64.60	10.8
BN-A + DMX		✓	1-cycle		66.78	13.0
EBCDet w/o TG		✓	1-cycle	✓	71.81	18.0
EBCDet w/o GA	✓	✓		✓	73.45	19.7
EBCDet 1-cycle	✓	✓	1-cycle	✓	75.40	21.7
EBCDet	✓	✓	curriculum	✓	76.34	22.6

TABLE 10. Image corruption experiments for different model sizes.

EBCDet-mPC	YOLOv5-S	YOLOv5-M	YOLOv5-X
Params (M)	7.3	21.4	87.7
FLOPs (B)	17.0	51.3	218.8
COCO-C	25.32±0.08	29.13±0.06	32.36±0.05
CityScapes-C	16.88±0.11	19.27±0.08	22.66±0.09

1) Batch norm adaptation helps reduce the domain shifts, as much as +10.8%, but it's not enough on its own.
2) Mixing source and target domain images into collages, and using teacher models both improve the adaptation performance by +2.2% and +8.6%, respectively.
3) Adaptation based on the curriculum outperforms the non-curriculum case (single pass over the entire unlabeled data), even with domain mixing or teacher guidance.
4) Overall, different components are organically combined to mitigate the domain shift and pseudo-label noise.

2) RESULTS FOR DIFFERENT MODEL CAPACITIES/SIZES

Table 10 contains the results of image corruption experiments on COCO-C and Cityscapes-C datasets, at different scales of YOLOv5 model. As we expect, models with higher capacity achieve better results.

3) PER-CORRUPTION TYPE RESULTS

Next, we study the performance of our method for the image corruptions benchmark, on a per-corruption basis. This is to ensure our method is not biased towards a certain kind of corruption. To this end, Table 11 shows the results of image corruption experiment on Pascal-C with YOLOv5M model. We observe that our results are for the most part consistent across different corruptions.

4) THE NUMBER OF CURRICULUM PARTITIONS

As mentioned earlier in this section, we use a default $n_c = 5$ number of partitions in our curriculum generation step. Experiments in Fig. 8 show an ablation on the number of partitions. We observe that increasing the number of partitions improves the performance, although it saturates at some point.

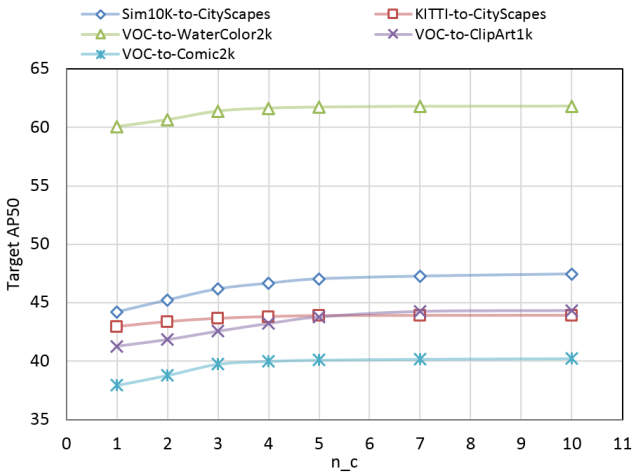

FIGURE 8. Ablation on the number of curriculum partitions. Different datasets/experiments seem to saturate at a different n_c value.

5) ENERGY VS SOFTMAX VS ENTROPY VS RANDOM VS NONE

Besides the free energy, there are other alternatives that can be used to create a learning curriculum. Here, we examine softmax and entropy of logits. In addition, we evaluate a random baseline were unlabeled examples are randomly partitioned into n_c groups. We run the Pascal-C experiment with YOLOv5M and report the results in Table 12. Note that in this table, we also report the case where no curriculum was used, i.e. the rest of the method except unlabeled examples were not partitioned. We observe from this table that the energy criterion achieves better results, suggesting that it can better distinguish the domain shift. Nonetheless, softmax and entropy functions also work well. It is also worth noting that a bad curriculum can hurt the results, which is due the fact that the batch normalization layers are updated with unrepresentative smaller set of examples frequently, and therefore pseudo-label quality degrades. Also note that the comparisons in Table 12 are based on choosing different curriculum data splitting criteria in our method. We provided end-to-end comparisons with other curriculum and non-curriculum strategies in Table 1-8. Among others, we provided results on the following curriculum-based strategies: Curriculum Self-Paced [57] (including the original method of #objects/average-size, random curriculum, image difficulty predictor and domain discriminator curriculums), and SGA-S [59] on multiple datasets.

6) ENERGY VERSUS ACCURACY

Fig. 9 shows the progress of gradually adapting a YOLOv5S model in the Sim10K-to-Cityscapes scenario (from Table 1). In this figure, we show the energy intervals used for curriculum generation on the horizontal axis, and the corresponding adapted teacher's accuracy on the vertical axis. We observe that the model is gradually adapting better to the target domain.

TABLE 11. Results of image corruptions experiment for different types of corruption on Pascal-C and YOLOv5M.

			Noise			Blur				Weather				Digital			
Method	AP^{50}_{clean}	mPC^{50}	Gauss.	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG
Source	83.13	53.78	47.44	51.35	44.98	53.87	42.17	48.61	36.64	51.77	56.29	71.74	78.82	55.81	56.43	54.52	56.17
Stylize	84.79	62.92	53.44	57.56	52.62	60.18	57.42	57.53	45.32	63.02	67.50	78.02	81.91	65.64	69.69	66.10	67.86
DeepAugment	85.05	64.88	61.75	64.06	60.64	63.74	57.95	56.18	44.75	62.31	68.27	79.36	82.69	68.34	61.92	71.40	69.78
BN Adapt	83.01	64.60	61.06	63.83	60.54	62.33	55.29	58.77	46.71	65.44	67.88	78.34	81.62	69.48	68.81	62.15	66.75
STAC	**87.00**	66.88	61.46	64.77	60.73	67.17	55.54	61.35	49.57	68.41	71.20	82.52	85.90	71.83	69.92	65.61	67.25
SimROD	86.97	75.40	72.00	74.11	73.01	72.65	70.25	72.85	60.65	77.81	77.47	84.03	86.17	79.66	80.49	72.54	77.36
EBCDet (Ours)	86.97	**76.34**	**72.81**	**74.78**	**73.84**	**73.52**	**71.12**	**73.60**	**61.72**	**78.96**	**78.66**	**85.20**	**87.08**	**80.58**	**81.78**	**73.69**	**77.76**
Oracle	86.75	78.74	76.35	76.68	76.42	75.63	75.12	77.10	70.31	80.07	79.56	84.25	86.15	80.60	82.88	78.73	81.22

TABLE 12. Comparing mPC^{50} of various techniques for curriculum generation; results are on YOLOv5M and Pascal-C.

Source	None	Random	Softmax	Entropy	Energy	Oracle
53.78	75.40	73.87	75.59	75.73	76.34	78.74

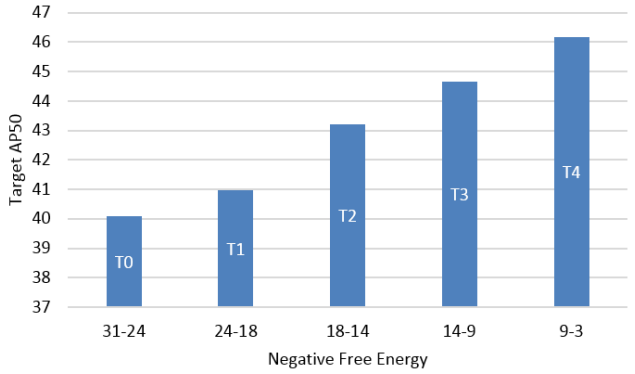

FIGURE 9. Energy vs accuracy: gradual adaptation of teachers across different energy intervals - Sim10K-to-Cityscapes scenario with YOLOv5S.

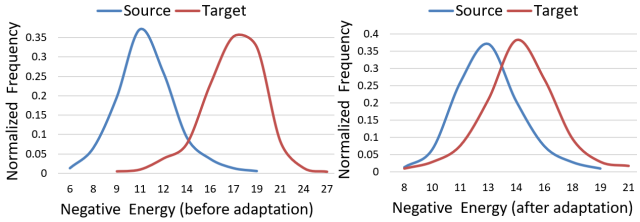

FIGURE 10. Energy distribution before and after adaptation on Pascal-C. For a better visualization, the frequencies are normalized for each dataset.

7) ENERGY DISTRIBUTION BEFORE AND AFTER ADAPTATION

Fig. 10 shows the distribution of negative free energy values for the Pascal-C experiment, before and after adaptation over the source and target datasets. We observe from this figure that before adaptation, source and target datasets exhibit somewhat separate distributions. This domain shift is indeed the cause for the low performance of source-trained models evaluated on target test set. On the other hand, after adaptation, the model identifies less of the data examples as out-of-distribution.

8) BASELINE [32] WITH TEACHER ADAPTATION

It is worth noting that like our method, [32] can also take advantage of pseudo-labels generated by an external teacher model. In the result tables of Section IV, for brevity and simplicity, we reported the self-adapted results of this method, and compared with the self-adapted version of ours. Here, we provide a comparison for the teacher-adapted case in Table 13.

FIGURE 11. Adaptation from COCO to real videos captured in/out of city.

TABLE 13. AP^{50} for our method and [32], when adapting with teachers. Results on Sim10k-to-Cityscapes with YOLOv5 at different scales and resolutions.

Teacher	Student	$AP50_{Source}$	[32]	EBCDet
X640	S320	33.62	44.70	45.88
X1280	S416	39.57	52.05	53.98
X1280	M640	55.86	64.40	65.53

9) ABLATION ON DOMAIN MIXING

In this work, we created domain mixed examples by creating collages from source and target domains. However, there could be various ways of mixing image samples and their bounding box annotations. We explored a simple alpha blending (0.5 coefficient) of two randomly sampled images from the source and target domains in the SIM10k-to-Cityscapes task. The annotations, in this case bounding boxes, were the union of all present objects. While this resulted in an improvement over the source-only method (adapted AP50 of 41.63 vs 39.57) it was still considerably lower than our current method (47.05). We suspect this is because the mixed images don't look like either of the domains and look artificial. Nonetheless, this is an interesting area for future research.

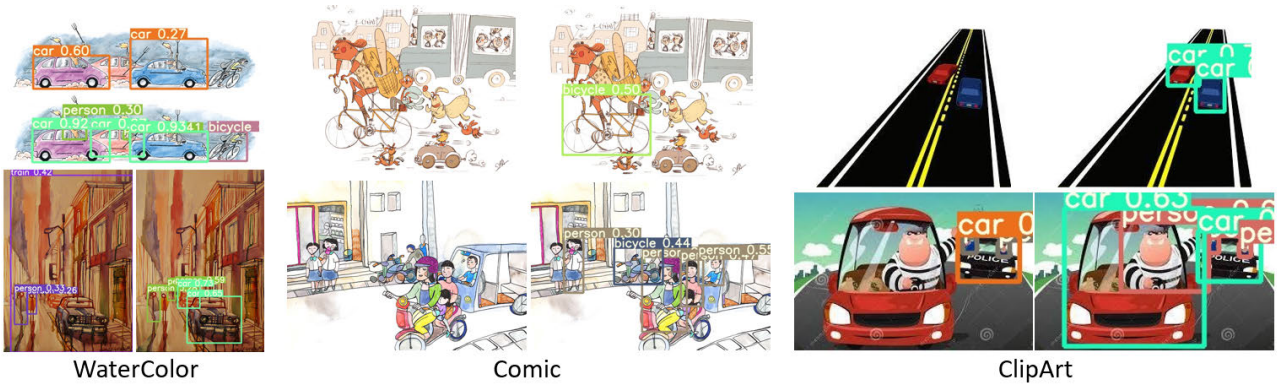

FIGURE 12. Examples of domain adaptation from VOC to WaterColor2k, Comic2k, and ClipArt1k, datasets. Each pair shows source-trained model output (left) versus the adapted model output (right). We observe a better detection across adapted models.

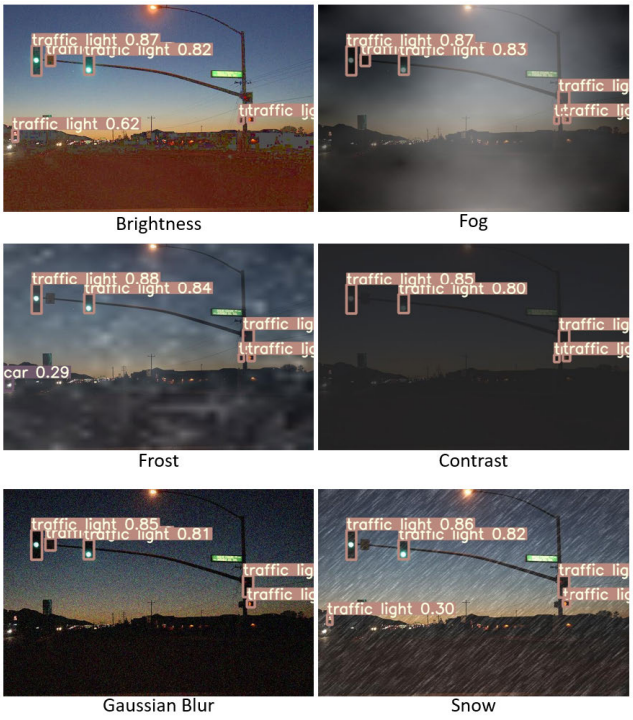

FIGURE 13. Examples of image corruptions from COCO-C: Adapted model performs well in the presence of different kinds of corruptions.

TABLE 14. Results on Cityscapes-to-KITTI. For other methods, results are directly used from the original papers, where available.

Method	Source	Adapted AP50	Oracle	τ	ρ
RPA [77]	34	44.8	85.6	10.8	20.9
SGA-S [59]	53.5	71.4	-	17.9	-
MAF [17]	53.5	72.1	-	18.6	-
DAF [16]	56.2	73.7	90.1	17.5	51.6
DT+PL [31]	56.2	73.8	90.1	17.6	51.9
RLDA [30]	56.2	**77.6**	90.1	21.4	63.1
EBCDet	54.7	76.9	82.1	**22.2**	**81**

TABLE 15. Results with and without curriculum adaptation.

Task	W/O Cur.	W/ Cur.	Oracle	Gain	ρ
SIM10k-to-City	44.21	47.47	56.49	3.26	**26.6**
KITTI-to-City	35.94	38.22	56.49	2.28	**11.1**
VOC-WaterColor	60.08	61.8	66.34	1.72	**27.5**
VOC-ClipArt1k	41.28	44.31	56.07	3.03	**20.5**
VOC-Comic2k	37.93	40.21	49.13	2.28	**20.4**

10) REVERSE ADAPTATION

The datasets chosen in the main experiments follow the literature in evaluating the performance of adapted models for the purposes of: 1) synthetic-to-real adaptation to measure the usefulness of synthetic data, 2) cross-camera shifts, 3) natural to artistic (watercolor, comic strips, cliparts) to measure the usefulness of models trained on natural everyday images for artistic use-cases, 4) clean to distorted setup to measure the robustness of lab-trained models in practical situations. With the exception of the cross-camera scenario, the literature has followed the above mentioned evaluation pipelines. For the cross-camera setup however, it is possible and interesting to evaluate in the reverse direction of adaptation. We provide the results for this experiment in Table 14, where we evaluate different methods on Cityscapes-to-KITTI adaptation.

11) RESULTS WITH AND WITHOUT THE CURRICULUM

Here, we provide the results with and without curriculum adaptation. In other words, we use our end to end pipeline in both cases, but report the impact of using curriculum along with the rest of the pipeline. This will show how much of the "remaining" gap the curriculum can close. Note that these results are not new, but are gathered from Fig. 8 and the results tables (we used the self-adaptation results from Tables 1-5). Table 15 shows the results ($n_c = 10$). We observe that the impact varies depending on the dataset/task, and how large the gap with the oracle is, but in most cases 20+% of the gap can be closed.

12) QUALITATIVE VISUALIZATIONS

In this subsection, we provide visualizations to better evaluate the performance of our method qualitatively. To this end, Fig. 11 demonstrates examples of adapting from COCO

FIGURE 14. Example curriculum: images from ClipArt1k are sorted according to their similarity to VOC.

models to videos captured in/out of city streets during the day/night. Fig. 12 demonstrates qualitative results of adapting the VOC dataset to WaterColor2k, ClipArt1k, and Comic2k. We observe that the adapted model can better identify objects in the target datasets than the source-trained model. Next, we demonstrate examples of image corruptions from the COCO-C dataset. Fig. 13 shows examples of images with different kinds of corruptions. We can see that the adapted model can successfully detect most of objects in these images, even in the presence of severe corruptions. Finally, Fig. 14 shows an example curriculum.

13) A NOTE ON COMPUTATIONAL COMPLEXITY

Previously we mentioned that our method is simple as it does not require architecture changes or modifications in the loss functions. We also note that although our method adds an overhead in terms of computational complexity, this overhead is usually not significant. A naive baseline that uses a source-trained model to generate pseudo-labels on unlabeled target data, requires training the Student for $e^S_{\mathcal{D}}$ epochs, generating pseudo-labels i.e. 1 epoch inference on $\overline{\mathcal{D}}$, and training the Student for $e^S_{\overline{\mathcal{D}}}$ epochs. Our method requires training the Student for $e^S_{\mathcal{D}}$ epochs, training the Teacher for $e^T_{\mathcal{D}}$ epochs, computing the energy values for $\overline{\mathcal{D}}$ i.e. 1 epoch of inference, adapting (training) the Teachers for e_a epochs on each $\overline{\mathcal{D}}_i$ partition, generating pseudo-labels on $\overline{\mathcal{D}}_i$, generating pseudo-labels with the adapted teacher on $\overline{\mathcal{D}}$, and fine-tuning the Student for e_{ft} epochs with $\overline{\mathcal{D}}$. Assuming that an inference epoch is significantly cheaper than training a full model, the difference between computational complexity of our method and the naive baseline rounds up to be as much as training the Teacher model for $e^T_{\mathcal{D}}$ epochs on $\mathcal{D}$ plus $e^a_{\overline{\mathcal{D}}}$ epochs on $\overline{\mathcal{D}}$. In case of self adaptation (i.e. no Teacher), this difference reduces to only $e^a_{\overline{\mathcal{D}}}$ epochs on $\overline{\mathcal{D}}$, which is much less than training a full model. In case the teacher model is significantly larger than the student however, the computational overhead can be meaningful and in some cases may be significant.

V. CONCLUSION

In this paper, we introduced an energy-based curriculum generation method for robust object detection. In an unsupervised domain adaptation setting, our method partitions the unlabeled target domain data into a number of subsets based on their energy values. This way, the partitions are sorted based on their distribution shift with respect to the source dataset. Next, a bigger Teacher model is iteratively adapted over these partitions. In each iteration, the Teacher gradually adapts its BN layers using domain mixed samples. Finally, the Student model is fine-tuned with high quality pseudo-labels provided by the Teacher. Our method showed a competitive performance against the existing baselines, at times by a large margin.

REFERENCES

[1] G. Jocher. (2023). *YOLOv5 Repository*. [Online]. Available: https://github.com/ultralytics/yolov5

[2] W. Fang, L. Wang, and P. Ren, "Tinier-YOLO: A real-time object detection method for constrained environments," *IEEE Access*, vol. 8, pp. 1935–1944, 2020.

[3] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain generalization: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4396–4415, Apr. 2023.

[4] J. Wang, C. Lan, C. Liu, Y. Ouyang, T. Qin, W. Lu, Y. Chen, W. Zeng, and P. Yu, "Generalizing to unseen domains: A survey on domain generalization," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 8, pp. 8052–8072, Aug. 2022.

[5] L. Zhang and X. Gao, "Transfer adaptation learning: A decade survey," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 21, 2022, doi: 10.1109/TNNLS.2022.3183326.

[6] C. Michaelis, B. Mitzkus, R. Geirhos, E. Rusak, O. Bringmann, A. S. Ecker, M. Bethge, and W. Brendel, "Benchmarking robustness in object detection: Autonomous driving when winter is coming," in *Proc. Mach. Learn. Auto. Driving Workshop 33rd Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2019, pp. 1–21.

[7] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, "AugMix: A simple data processing method to improve robustness and uncertainty," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–15.

[8] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, D. Song, J. Steinhardt, and J. Gilmer, "The many faces of robustness: A critical analysis of out-of-distribution generalization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8320–8329.

[9] E. Mintun, A. Kirillov, and S. Xie, "On interaction between augmentations and corruptions in natural corruption robustness," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 3571–3583, 2021.

[10] R. Taori, A. Dave, and V. Shankar, "Measuring robustness to natural distribution shifts in image classification," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 18583–18599.

[11] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 443–450.

[12] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, 2016, vol. 30, no. 1, pp. 2058–2065.

[13] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1180–1189.

[14] D. Yoo, N. Kim, S. Park, A. S. Paek, and I. S. Kweon, "Pixel-level domain transfer," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 517–532.

[15] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 469–477.

[16] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster R-CNN for object detection in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3339–3348.

[17] Z. He and L. Zhang, "Multi-adversarial faster-RCNN for unrestricted object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6667–6676.

[18] X. Zhu, J. Pang, C. Yang, J. Shi, and D. Lin, "Adapting object detectors via selective cross-domain alignment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 687–696.
[19] Y. Zheng, D. Huang, S. Liu, and Y. Wang, "Cross-domain object detection through coarse-to-fine feature adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13763–13772.
[20] K. Saito, Y. Ushiku, T. Harada, and K. Saenko, "Strong-weak distribution alignment for adaptive object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6949–6958.
[21] C.-C. Hsu, Y.-H. Tsai, Y.-Y. Lin, and M.-H. Yang, "Every pixel matters: Center-aware feature alignment for domain adaptive object detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 733–748.
[22] I. Goodfellow, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.
[23] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 95–104.
[24] T. Kim, M. Jeong, S. Kim, S. Choi, and C. Kim, "Diversify and match: A domain adaptive representation learning paradigm for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12448–12457.
[25] H.-K. Hsu, C.-H. Yao, Y.-H. Tsai, W.-C. Hung, H.-Y. Tseng, M. Singh, and M.-H. Yang, "Progressive domain adaptation for object detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 738–746.
[26] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2242–2251.
[27] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5967–5976.
[28] I. Triguero, S. García, and F. Herrera, "Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study," *Knowl. Inf. Syst.*, vol. 42, no. 2, pp. 245–284, Feb. 2015.
[29] K. Sohn, Z. Zhang, C.-L. Li, H. Zhang, C.-Y. Lee, and T. Pfister, "A simple semi-supervised learning framework for object detection," 2020, *arXiv:2005.04757*.
[30] M. Khodabandeh, A. Vahdat, M. Ranjbar, and W. Macready, "A robust learning approach to domain adaptive object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 480–490.
[31] N. Inoue, R. Furuta, T. Yamasaki, and K. Aizawa, "Cross-domain weakly-supervised object detection through progressive domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5001–5009.
[32] R. Ramamonjison, A. Banitalebi-Dehkordi, X. Kang, X. Bai, and Y. Zhang, "SimROD: A simple adaptation method for robust object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3550–3559.
[33] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–16.
[34] S. Cygert and A. Czyzewski, "Toward robust pedestrian detection with data augmentation," *IEEE Access*, vol. 8, pp. 136674–136683, 2020.
[35] K. Fujii, H. Kera, and K. Kawamoto, "Adversarially trained object detector for unsupervised domain adaptation," *IEEE Access*, vol. 10, pp. 59534–59543, 2022.
[36] H. Zhang, G. Luo, J. Li, and F.-Y. Wang, "C2FDA: Coarse-to-fine domain adaptation for traffic object detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 12633–12647, Aug. 2022.
[37] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9626–9635.
[38] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.
[39] J. Hoffman, E. Tzeng, T. Park, J. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, "CyCADA: Cycle-consistent adversarial domain adaptation," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, Stockholm, Sweden, Jul. 2018, pp. 1989–1998.
[40] G. Li, Z. Ji, and X. Qu, "Stepwise domain adaptation (SDA) for object detection in autonomous vehicles using an adaptive CenterNet," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17729–17743, Oct. 2022.
[41] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, 2010.
[42] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2015, pp. 448–456.
[43] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?" in *Proc. Adv. Neural Inf. Process. Syst., (NeurIPS)*, 2018, pp. 2488–2498.
[44] S. Schneider, E. Rusak, L. Eck, O. Bringmann, W. Brendel, and M. Bethge, "Improving robustness against common corruptions by covariate shift adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 11539–11551.
[45] C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le, "Adversarial examples improve image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 816–825.
[46] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou, "Revisiting batch normalization for practical domain adaptation," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, Toulon, France, Apr. 2017, pp. 1–12.
[47] W.-G. Chang, T. You, S. Seo, S. Kwak, and B. Han, "Domain-specific batch normalization for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7346–7354.
[48] Q. Cai, Y. Pan, C.-W. Ngo, X. Tian, L. Duan, and T. Yao, "Exploring object relation in mean teacher for cross-domain detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11449–11458.
[49] Y. Yu, X. Xu, X. Hu, and P.-A. Heng, "DALocNet: Improving localization accuracy for domain adaptive object detection," *IEEE Access*, vol. 7, pp. 63155–63163, 2019.
[50] G. E. Hinton and R. S. Zemel, "Autoencoders, minimum description length, and Helmholtz free energy," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 6, 1994, pp. 3–10.
[51] Y. LeCun, S. Chopra, and R. Hadsell, "A tutorial on energy-based learning," *Predicting Structured Data*, vol. 1, pp. 1–59, Aug. 2006.
[52] W. Liu, X. Wang, J. Owens, and Y. Li, "Energy-based out-of-distribution detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 21464–21475.
[53] M. Akbari, A. Banitalebi-Dehkordi, and Y. Zhang, "EBJR: Energy-based joint reasoning for adaptive inference," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2021, pp. 1–14.
[54] K. J. Joseph, S. Khan, F. S. Khan, and V. N. Balasubramanian, "Towards open world object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5826–5836.
[55] J. Choi, M. Jeong, T. Kim, and C. Kim, "Pseudo-labeling curriculum for unsupervised domain adaptation," 2019, *arXiv:1908.00262*.
[56] J. Wang, X. Wang, and W. Liu, "Weakly- and semi-supervised faster R-CNN with curriculum learning," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 2416–2421.
[57] P. Soviany, R. T. Ionescu, P. Rota, and N. Sebe, "Curriculum self-paced learning for cross-domain object detection," *Comput. Vis. Image Understand.*, vol. 204, Mar. 2021, Art. no. 103166.
[58] D. Zhang, J. Han, L. Zhao, and D. Meng, "Leveraging prior-knowledge for weakly supervised object detection under a collaborative self-paced curriculum learning framework," *Int. J. Comput. Vis.*, vol. 127, no. 4, pp. 363–380, Apr. 2019.
[59] C. Zhang, Z. Li, J. Liu, P. Peng, Q. Ye, S. Lu, T. Huang, and Y. Tian, "Self-guided adaptation: Progressive representation alignment for domain adaptive object detection," *IEEE Trans. Multimedia*, vol. 24, pp. 2246–2258, 2022.
[60] D. Dai, C. Sakaridis, S. Hecker, and L. Van Gool, "Curriculum model adaptation with synthetic and real data for semantic foggy scene understanding," *Int. J. Comput. Vis.*, vol. 128, no. 5, pp. 1182–1204, May 2020.
[61] C. Sakaridis, D. Dai, and L. Van Gool, "Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7373–7382.
[62] Q. Lian, L. Duan, F. Lv, and B. Gong, "Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6757–6766.

[63] Y. Zhang, P. David, and B. Gong, "Curriculum domain adaptation for semantic segmentation of urban scenes," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2039–2049.
[64] S. Wu, Y. Xu, B. Zhang, J. Yang, and D. Zhang, "Deformable template network (DTN) for object detection," *IEEE Trans. Multimedia*, vol. 24, pp. 2058–2068, 2022.
[65] H. Motorcu, H. F. Ates, H. F. Ugurdag, and B. K. Gunturk, "HM-Net: A regression network for object center detection and tracking on wide area motion imagery," *IEEE Access*, vol. 10, pp. 1346–1359, 2022.
[66] S. Zhai, D. Shang, S. Wang, and S. Dong, "DF-SSD: An improved SSD object detection algorithm based on DenseNet and feature fusion," *IEEE Access*, vol. 8, pp. 24344–24357, 2020.
[67] M. Heisler, A. Banitalebi-Dehkordi, and Y. Zhang, "SemAug: Semantically meaningful image augmentations for object detection through language grounding," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 610–626.
[68] P. Benz, C. Zhang, A. Karjauv, and I. S. Kweon, "Revisiting batch normalization for improving corruption robustness," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 494–503.
[69] A. Merchant, B. Zoph, and E. D. Cubuk, "Does data augmentation benefit from split BatchNorms," 2020, *arXiv:2010.07810*.
[70] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
[71] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.
[72] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.
[73] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan, "Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?" in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 746–753.
[74] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.
[75] R. Xie, F. Yu, J. Wang, Y. Wang, and L. Zhang, "Multi-level domain adaptive learning for cross-domain detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 3213–3219.
[76] D. Guan, J. Huang, A. Xiao, S. Lu, and Y. Cao, "Uncertainty-aware unsupervised domain adaptation in object detection," *IEEE Trans. Multimedia*, vol. 24, pp. 2502–2514, 2022.
[77] Y. Zhang, Z. Wang, and Y. Mao, "RPN prototype alignment for domain adaptive object detector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12420–12429.
[78] W. Li, X. Liu, and Y. Yuan, "SIGMA: Semantic-complete graph matching for domain adaptive object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5281–5290.
[79] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness," in *Proc. 7th Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–22.
[80] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2962–2971.
[81] J. Zhang, J. Huang, Z. Tian, and S. Lu, "Spectral unsupervised domain adaptation for visual recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 9819–9830.
[82] A. Amirkhani and A. H. Barshooi, "DeepCar 5.0: Vehicle make and model recognition under challenging conditions," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 541–553, Jan. 2023.

AMIN BANITALEBI-DEHKORDI received the Ph.D. degree in electrical and computer engineering from The University of British Columbia (UBC), Canada, in 2014. He is currently a Principal Researcher of machine learning and technical lead with the Vancouver Research Centre, Huawei Technologies Canada Company, Ltd. His academic career has resulted in publications in the fields of computer vision and pattern recognition, visual attention modeling, video quality assessment, and high dynamic range video. His industrial experience expands to areas in machine learning, deep learning, computer vision, NLP, and signal/image/video processing.

ABDOLLAH AMIRKHANI (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees (Hons.) in electrical engineering from the Iran University of Science and Technology (IUST), Tehran, in 2012 and 2017, respectively. He is currently an Assistant Professor with the School of Automotive Engineering, IUST. He has been actively involved in several national research and development projects, related to the development of new methodologies and learning algorithms based on AI technologies. His research interests include machine vision, fuzzy cognitive maps, data mining, and machine learning. In 2015, he received the Outstanding Student National Award from the Vice President of Iran. In 2016, he was conferred the award by the Ministry of Science, Research and Technology. He is an Associate Editor of the *Engineering Science and Technology, an International Journal*.

ALIREZA MOHAMMADINASAB received the B.Sc. degree in electrical engineering from the University of Sistan Baluchestan (USB), Iran, in 2012. He is currently pursuing the M.Sc. degree with the School of Automotive Engineering, Iran University of Science and Technology (IUST). His research interests include autonomous vehicles, computer vision, and deep learning.