AdveRsarial Calibration between Modalities

Yutian Lei

CMU-RI-TR-22-78

December 21, 2022

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Prof. Fernando De La Torre, *chair*
Dr. Dong Huang, *chair*
Professor Alexander G. Hauptmann
Dr. Zhengyi Luo

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Robotics.

To all advisors, friends and family who has supported me unconditionally

Abstract

Advances in computer vision and machine learning techniques have led to flourishing success in RGB-input perception tasks, which has also opened unbounded possibilities for non-RGB-input perception tasks, such as object detection from wireless signals, point clouds, and infrared light. However, compared to the matured development pipeline of RGB-input (source modality) models, developing non-RGB-input (target-modality) models from scratch poses excessive challenges in the modality-specific networks/training-tricks design and labor in the target-modality data collection/annotation.

In this thesis, the AdveRsarial Calibration (ARC) is proposed as an efficient pipeline for calibrating target-modality inputs to matured DNN models developed on the source modality. Under ARC, a target-modality-input model is simply composed by adding a small calibrator module ahead of an existing source-modality model. Our ARC training techniques require as little as zero manual annotation on the target modality while producing comparable or better metrics than baseline target models that require 100% manual annotations. We present the ARC components that enable us to achieve the above goals: (1) model inversion to synthesize inverted images from the source-modality model, (2) Foreground Semantics Reconstruction, (3) Decayed Semantic Supervision, and (4) Skipped Inverted Attention,

We demonstrate the effectiveness of ARC by composing the WiFi-input, Lidar-input, and Thermal-Infrared-input models upon the pre-trained RGB-input models respectively.

Acknowledgments

I would like to express my heartfelt gratitude to my advisor, Dr. Dong Huang, for his invaluable guidance, support, and encouragement throughout the course of this research. His expertise and insight have been invaluable resources, and I am deeply grateful for the opportunity to work with him.

I would also like to thank Prof. Fernando, Prof. Alexander, and Dr. Zhengyi Luo for their valuable contributions and helpful feedback. I am grateful for their support and for the opportunity to collaborate with them.

I am also thankful to all of my friends and family for their love and support. Their encouragement and belief in me have meant the world to me, and I could not have completed this work without their help.

Finally, I would like to express my gratitude to all of the individuals who have contributed to this research in any way, whether directly or indirectly. Your efforts and support have been greatly appreciated and have helped to make this work possible.

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

Introduction

Although research on the DNN-based perception problem has mainly focused on RGB-input models, non-RGB sensors still have clear advantages over RGB cameras in in a wide range of scenarios. For instance, wireless signals [56, 65] can easily penetrate furniture occlusion and identify human bodies for their Dielectric properties, while being lighting-free, occlusion-resistant, and privacy-friendly compared to cameras. Lidar scans [12, 52] contain depth information, enabling more accurate and robust object localization than RGB under low light condition or bad weather. Thermal InfRared (TIR) cameras [24, 25] capture near-infrared ($0.75 - 1.4\mu m$) or long-wavelength infrared ($8 - 15\mu m$) signals, which makes, in particular, human bodies more visible than in RGB images and more robust to the visible spectrum interference.

Thanks to the decade of work on RGB-input DNN models, researchers has accumulated extensive resources on image-based architectures (e.g., MaskRCNN [17], Yolo5 [26], Swin-Transformer [40]), pre-train-datasets (ImageNet [6], MS-COCO [36], OpenImages [31]), pre-train weights, training tricks [13, 18, 64] and code repos (e.g., Detectron2 [60], mmDetection[4]). Unfortunately, non-RGB-input models cannot be **directly** built upon above RGB resources. Instead, one usually needs to design and train a new network from scratch [45, 56, 65], and collect/annotate data from non-RGB sensors at the comparable scale of RGB databases above to achieve similar performance on perception tasks with non-RGB modality.

In this thesis, we propose AdveRsarial Calibration (ARC) for calibrating target-

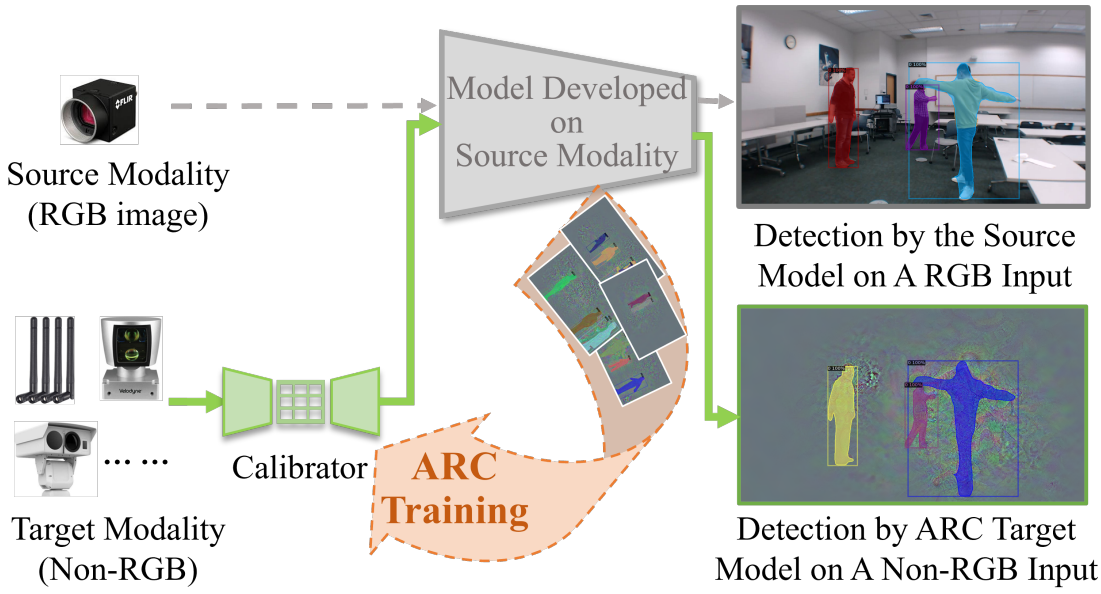

Figure 1.1: We propose AdveRsarial Calibration (ARC) for calibrating input modalities of DNN models. Our ARC model is constructed and trained upon a pre-trained source-modality model by "ARC Training". A target-modality input is processed along the thick solid green arrows "→".

modality inputs to a DNN model developed on the source modality. Figure 1.1 shows the main idea of ARC. Ahead of a source-modality model, we add a small target-modality-input calibrator module, composing a [**Calibrator**|**Source**]-structured target-modality model. The **Calibrator** transforms a target-modality input into a source-modality-like tensor highlighting the foreground, which is then mapped to object detection results by the **Source** module. Trained on {source, target} input pairs of **zero** manual annotation, the ARC target-modality models reach comparable or better metrics on WiFi, Lidar, and Thermal Infrared than the baselines that requires 100% manual annotation. This is achieved by our ARC training techniques that learn prior knowledge from the enclosed **Source** module and iteratively regularize gradients on the **Calibrator** layers. ARC training is an adversarial learning process between the **Calibrator** module that mimics source modality inputs and the **Source** module that penalizes detection errors.

Chapter 2

Realated Works

For conciseness, we only list object detection work on the three Non-RGB target modalities (WiFi, Lidar and Thermal) involved in our experiments.

2.1 Modality-specific Perception

In two-stage approaches, Non-RGB inputs are converted to RGB images before feeding to an RGB-input model. Researchers of [9, 28, 29] only convert Wifi signals to low-resolution ($< 160 \times 120$) RGB images by over-fitting a few antenna layouts. No codes or data are available. Points2Pix [42] translates Lidar points to RGB images with a conditional GAN. The infrared images were re-colored to RGB in [35, 47] by image translation [27].

In single-stage approaches, the whole model is specifically designed and trained on a non-RGB modality. Due to the lack of spatial representation, the WiFi-input models are mostly focused on coarse-granularity tasks such as crowd counting [7, 38] or single-person activity recognition [33, 59]. [56] develop pioneer WiFi-specific DNNs for multi-person segmentation and pose estimation. Lidar-input models are specific to point-clouds representations, such as the Point View [44, 45, 50], the Bird's Eye View [32, 61] and the Range View(LaserNet [41], RangeDet [11]). The range view is popular for its low quantization error and computational costs. Thermal images are close to RGB spatially, enabling [24, 25] to train RGB-input models on the TIR inputs.

Unlike the two-stage approaches, we work on perception tasks that learn foreground representation instead of RGB appearances. By enclosing the pre-train source model in the target model, we simplify the design effort and reduce target-modality annotations in the single-stage approaches.

2.2 Knowledge Distillation between Models

To leverage one well-trained model in training another model, teacher-to-student Knowledge Distillation (KD) is a popular approach [8, 19]. The student networks mimic the teacher networks on predictive probabilities [19, 34], intermediate features [49, 58], or attention maps [3, 39, 55, 62]. When the teacher and student models have different input modalities [15], KD requires the same amount of annotated source-target data as those in the teacher model training. All KD methods run the teacher and student in parallel during training.

Unlike KD, our [Calibrator|Source] target model is directly initialized and supervised by the enclosed Source module, which requires neither an independent teacher inference dataflow nor fully annotated target-modality data. Ablation study shows that ARC clearly performs better.

2.3 Adversarial Training

To improve robustness on imbalanced datasets, many adversarial training strategies explicitly produce hard features/samples: auto-augmentation [68], co-mixup [30], random erasing [66], representation self-challenging [21, 22], reverse attention [5]. Regulators such as cross-layer consistency [20, 57] and self-distillation mechanism [23] were also very effective. Generative Adversarial Networks(GAN) implicitly produce hard samples from a discriminator and are recently extended to the object detection tasks by [37, 46].

We improve target model robustness by synthesizing image-like foreground representation and regularising gradients of the enclosed source model.

Vector Quantized Representation. VQ-VAE [48, 53] and VQ-GAN [10] show that a quantized latent space provides a compact representation of natural

images, language, and audio/video sequence while using a relatively small number of parameters, making them efficient to train and use. We extend VQVAE to learn the foreground representation shared between modalities.

Chapter 3

AdveRsarial Calibration (ARC)

Problem Definition: ARC for the object detection task:

- **Source model**: $S(\cdot) : \mathbf{I} \rightarrow \mathbf{Y}$ maps one RGB image $\mathbf{I} \in \Re^{[width \times height \times 3]}$ to the object locations and categories $\mathbf{Y} = [object\_bboxes, object\_mask, object\_class]$.
- **Target model**: $T(\cdot) : \mathbf{X} \rightarrow \mathbf{Y}$ maps one target modality tensor $\mathbf{X} \in \Re^{[width\_T \times height\_T \times channel\_T]}$ to $\mathbf{Y}$.
- **ARC target model**: a [Calibrator|Source] model $T(\cdot) : \{C(\cdot)|S(\cdot)\}$, where the "Calibrator" module $C(\cdot) : \mathbf{X} \rightarrow \mathbf{J}$ produces an image-like tensor $\mathbf{J} \in \Re^{[width \times height \times 3]}$. The "Source" module (with the same network structure as the Source model $S(\cdot)$) maps $\mathbf{J}$ to $\mathbf{Y}$.

The goal of ARC is to train a ARC target model $\{C(\cdot)|S(\cdot)\}$ given a pre-trained source model $S(\cdot)$ and a set of $\{\mathbf{X}, \mathbf{I}\}$ pairs. (See the framework in Figure 3.1). For simplicity, we assume that the source and target modality data are sampled from the same $\mathbf{Y}$ (foreground/background) distributions, such that ARC is only focused on reducing the discrepancy between modalities.

3.1 Reasoning for the $\{C(\cdot)|S(\cdot)\}$ Target Model

If we follow the development procedure of $S(\cdot)$ to develop a new $T(\cdot)$, we need a modality-specific multi-resolution feature extractor (comparable to ResNet), which is coupled with a task-specific output head (Such as the anchor-based or transformer-

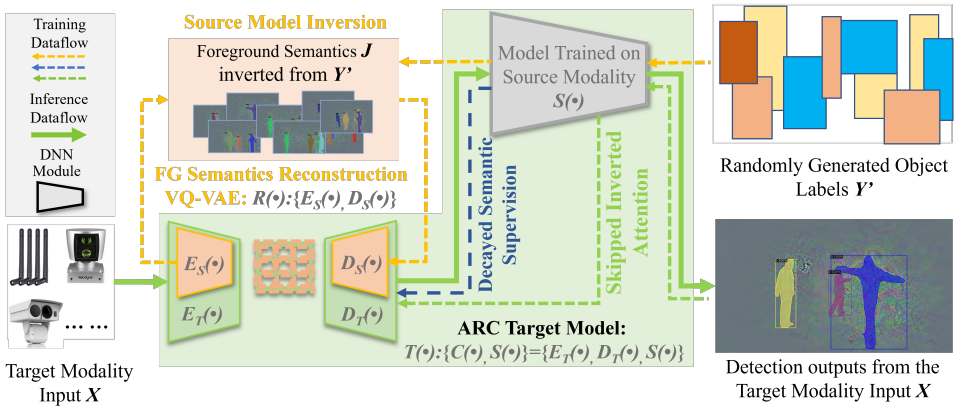

Figure 3.1: AdveRsarial Calibration(ARC) Framework. Our ARC target model $T(\cdot) : \{E_T, D_T, S\}$ is composed from calibrator $C(\cdot) : \{E_T, D_T\}$ and source model $S(\cdot)$, with its inference dataflow along the solid green arrows ("$\longrightarrow$"). The training of ARC leverages three resources of supervision(along the dashed arrows "$\dashleftarrow$") : **(a)** Foreground(FG) Semantic Reconstruction(**FSR**), which incorporates the source model prior: the foreground(object) semantics $\mathbf{J}_S$ synthesized from $S(\cdot)$ and self-reconstructed by an auxiliary VQ-VAE, $R(\cdot) : \{E_S, D_T\}$. **(b)** Decayed Semantic Supervision(**DSS**), which regularizes gradients from $S(\cdot)$ by foreground semantics $\mathbf{J}_T$ inverted on target-modality training data. **(c)** Skipped Inverted Attention(**SIA**), which improve the attention of $C(\cdot)$ output using high-level $S(\cdot)$ layer gradients. See details in the **ARC Training** section.

based bounding box regressors) by multi-resolution skip connections. One also needs annotated $\{\mathbf{X}, \mathbf{Y}\}$ data comparable to the amount of the $\{\mathbf{I}, \mathbf{Y}\}$ data used in the source model $S(\cdot)$ training.

Under ARC(Figure 3.1), we only design calibrator $C(\cdot)$ that produces a *single-resolution* tensor $\mathbf{J}$ feeding to the source module $S(\cdot)$ enclosed in $T(\cdot)$.

Foreground encoding in $C(\cdot)$: Since it is $S(\cdot)$'s expertise to locate and classify objects, $C(\cdot)$ only needs to pass to $S(\cdot)$ some foreground-sensitive features $\mathbf{J}$, i.e., the edges/textures highlighting all the object categories. To generate such foreground features, we revisit the insight of the image-wise Class Activation Maps [67]: all pixels of each object category can be mapped to an element of a probability vector by Softmax activation, which highlights foreground pixels assuming category-wise features follow a multi-modal distribution. In order to preserve the internal spatial layout of objects, we encode pixel patches(object parts) under the multi-modal distribution. This is done by a encoder-decoder structure $C(\cdot) : \{E_T, D_T\}$ with a

quantized latent space inspired by VQ-VAE[53] (see Figure 3.1).

VQVAEs are a variant of the standard variational autoencoder (VAE) architecture, which uses a probabilistic model to learn the latent representation of the input data. In a VQVAE, the latent representation is obtained by quantizing the continuous latent space of the VAE into a finite set of discrete "codebook" vectors. These codebook vectors are learned during training and are used to represent the latent representation of the input data.

One advantage of VQVAE is that it can produce high-quality reconstructions of the input data while using a relatively small number of parameters, making it efficient to train and use as a calibrator in our model.

We set the latent space tensor size to $[width/8, height/8, channel]$, in which every $[1 \times 1 \times channel]$ vector is hard-coded to one of the multi-modal centers of local patches, denoted as codebook $\{B_i \in \Re^{1\times1\times channel}\}|_{i=1,...,p}$. Mapping such a VQ-VAE-like latent space to image-like feature $\mathbf{J}$, the calibrator decoder D_T simply takes the same structure of the standard VQ-VAE decoder. The calibrator encoders E_T for target-modality inputs are similar to the standard VQ-VAE encoder with minor adjustment below.

Target-Modality Encoder E_T: The WiFi signal corresponding to one synchronized RGB image [56], is represented as the Channel State Information (CSI) [16] tensor [samples, transmitters, receivers, sub-carriers]. The CSI tensor elements has no spatial dependence as RGB pixels. In this case, we construct E_T by adding Wi2Vi [29] layers in front of a VQ-VAE encoder. For other target modalities (infrared and Lidar range images) that have the image-like tensor shape, we directly use the VQ-VAE encoder structure for E_T.

Remarks: To enable the $S(\cdot)$ module of $T(\cdot)$ to produce the same $\mathbf{Y}$ as that of the source model, the latent space of $C(\cdot)$ has to contain the same semantics as those encoded in the source model. This opens the probability to learn $C(\cdot)$ from a pre-trained source model$S(\cdot)$, without requiring a huge set of $\{\mathbf{X}, \mathbf{Y}\}$ training data. We only demonstrate the basic case of direct concatenation $\{C(\cdot)|S(\cdot)\}$ in this paper.The overheads of $C(\cdot)$ could be reduced by connecting $C(\cdot)$ to the smaller feature maps of $S(\cdot)$.

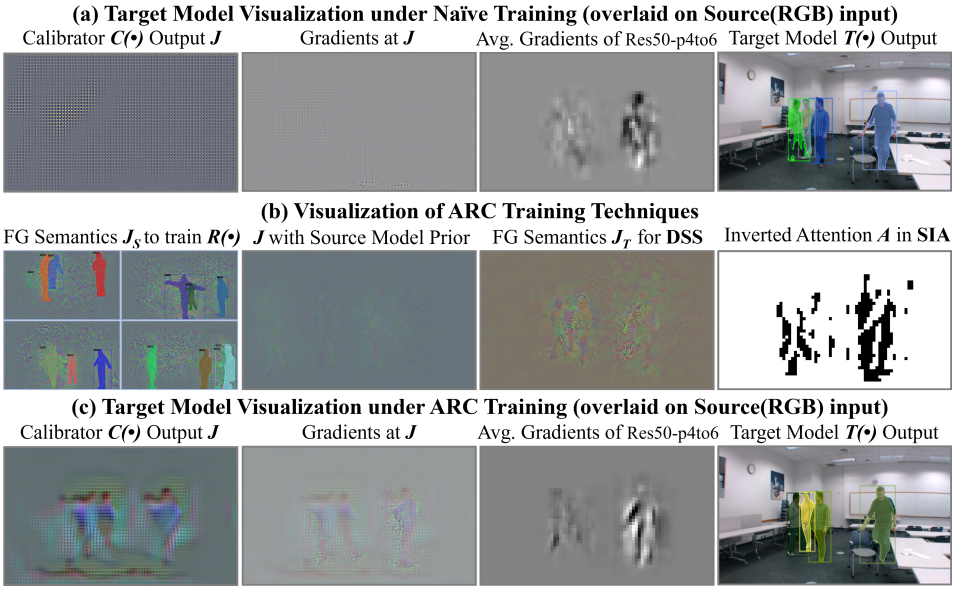

Figure 3.2: Training strategy examples for a WiFi-input target model $T(\cdot)$: $\{C(\cdot), S(\cdot)\}$. Here, the source model $S(\cdot)$ is an RGB-input ResNet50-FPN-Mask-RCNN. **(a)** Target model visualization under *Naïve* training: neither foreground-sensitive features in $C(\cdot)$ output **J** nor any clear gradients at **J** comparable to "Avg. Gradients of Res50-p4to6" (high-level gradients resized to the image size and averaged on one channel). **(b)** Visualization of ARC training techniques (See the **ARC Training** section). **(c)** Target model visualization under ARC training: clear foreground-sensitive features, strong gradients at **J** and better detection.

3.2 ARC Training

The devil lies in the training of the ARC target model $T(\cdot)$. There are two *naïve* sources of supervision: **(1)** Given the $\{\mathbf{X}, \mathbf{I}\}$ pairs, one may pre-train $C(\cdot)$ to approximate **I**. Since there are usually more background pixels than foreground pixels, the pre-trained $C(\cdot)$ outputs **J** may largely be background textures that are irrelevant to the detection tasks. **(2)** Given abundant $\{\mathbf{X}, \mathbf{Y}\}$ training pairs, one may randomly initialize $C(\cdot)$ and update all $T(\cdot)$ layers using the gradients back-propagated from the source model losses. Such a strategy suffers from the vanishing gradient problem [1, 14] and is contradictory to the common DNN training practice (for instance, ImageNet-pretrained Resnet + randomly initialized Mask-RCNN). In fact, the randomly initialized $C(\cdot)$ should receive the strong gradients for updating, while

the pre-trained weights of $S(\cdot)$ should be preserved by updating with weak gradients. Figure 3.2(a) shows that the *naïve* training produces neither foreground-sensitive features nor any clear gradients at $\mathbf{J}$ comparable to the high-level gradients, e.g., "Avg. Gradients of Res50-p4to6".

To address the aforementioned problem, we propose the four ARC training techniques. Firstly, we conduct Source Model Inversion to generate foreground semantics features to provide clean and strong supervision for the calibrator. Secondly, we initialize the calibrator through Foreground Semantics Reconstruction to provide a compact latent space representation of the inverted images and help the calibrator converge. Thirdly, we use Decayed Semantic Supervision to regularize the model gradients with image-space semantic supervision while at the same time amending the acute details in the calibrator training. Finally, we fine-tune our model with Skipped Inverted Attention to amplify and balance the gradients of the source model loss. (see the graphic description in Figure 3.1)

3.2.1 Source Model Inversion

We first guide the calibrator $C(\cdot)$ to produce foreground-sensitive features leveraging a pre-trained source model $S(\cdot)$. Given the pre-trained $S(\cdot)$ and the object detection GTs ($\mathbf{Y}$), we conduct image-wise model inversion [2, 54] to generate $\mathbf{J}_S$ by minimizing the $S(\cdot)$ losses,

$$\mathcal{L}_S(S(\mathbf{J}_S), \mathbf{Y}) = \lambda_{bbox}\mathcal{L}_{bbox} + \lambda_{cls}\mathcal{L}_{cls} + \lambda_{mask}\mathcal{L}_{mask}, \tag{3.1}$$

We solve this problem by image-wise optimization: freezing all the $S(\cdot)$ layers, computing gradient from $\mathcal{L}_S(\mathbf{J}_S, \mathbf{Y})$ and only updating $\mathbf{J}_S$ from its random initialization. After the optimization converges, $\mathbf{J}_S$ is synthesized as a most probable and style-invariant "image" from which $S(\cdot)$ can detect $\mathbf{Y}$. $\mathbf{J}_S$ only captures the edge-like patterns of the foreground (See pixels marked in colors Figure 3.2(b)). We call $\mathbf{J}_S$ the **Foreground Semantics**.

Compared with the original RGB pixels, we see $\mathbf{J}_S$ as clean supervision to guide $C(\cdot)$ training. Compared with the gradient amplitude from $\mathcal{L}_S$, the pixel intensity of $\mathbf{J}_S$ is comparable to $\mathbf{I}$ resulting in stronger gradients on the $C(\cdot)$ layers. To increase the diversity of $\mathbf{J}_S$, we also randomly generate the object layouts in $\mathbf{Y}'$ of different

bbox locations (instance masks for Mask-RCNN) and class labels.

3.2.2 Foreground Semantics Reconstruction(FSR)

Next, to inject the prior knowledge in J_S to calibrator $C(\cdot)$, we train an auxiliary VQ-VAE, $R(\cdot) = \{E_S, D_S\} : \mathbf{J}_S \rightarrow \mathbf{J}_S$ that encodes and reconstructs J_S. Then we share the $R(\cdot)$'s VQ codebook with $C(\cdot)$, and initialize the $C(\cdot)$'s decoder $D_T(\cdot)$ by the $R(\cdot)$'s decoder $D_S(\cdot)$ weights. We call the initialization method of $C(\cdot)$ as **Foreground Semantics Reconstruction(FSR)**.

In addition, our target model $T(\cdot)$ encloses $S(\cdot)$ as a module, which can explicitly inherit the source model knowledge by initializing with the pre-trained source model weights.

Finally, to accommodate both above priors incorporated into $C(\cdot)$ and $S(\cdot)$, we train $T(\cdot)$ in a **two-stage update** strategy: (i) fix $S(\cdot)$ and only update $C(\cdot)$ until it converges; (ii) continue training by updating both $C(\cdot)$ and $S(\cdot)$.

Remarks: (a) $\mathbf{J}_S$ is synthesized from a pre-trained source model over synthesized $\mathbf{Y}'$, which does not requires manually annotated source-inputs $\mathbf{I}$. **(b)** Given that the $R(\cdot)$ training requires no annotation on target data $\mathbf{X}$, and the $S(\cdot)$ module can provide pseudo ground truth $\mathbf{Y}_{pseudo}$ for the $\{\mathbf{X}, \mathbf{I}\}$ pairs, the overall training of $T(\cdot)$ is **self-supervised** (i.e., **zero** manual annotation on $\mathbf{X}$).

3.2.3 Decayed Semantic Supervision (DSS)

The Decayed Semantic Supervision is used to regularize $\mathcal{L}_S$ gradients with image-space semantic supervision. Given a pre-trained $S(\cdot)$ and either $\{\mathbf{X}, \mathbf{Y}_{pseudo}\}$ (the self-supervised ARC) or $\{\mathbf{X}, \mathbf{Y}\}$ (the supervised ARC) as the target model training data, we invert $S(\cdot)$ to generated $\mathbf{J}_T$ as GT to directly supervise $C(\cdot)$. The $C(\cdot)$ output $\mathbf{J}$ is an image-like tensor, therefore $C(\cdot)$ can be trained with image-based losses against J_T along with the source loss $\mathcal{L}_S$, leading to the Semantic Supervision (**SS**) loss,

$$\mathcal{L}_{SS}(\mathbf{J}, \mathbf{J}_T) = SSIM(\mathbf{J}, \mathbf{J}_T) + L_1(\mathbf{J}, \mathbf{J}_T) + \mathcal{L}_S \tag{3.2}$$

where $SSIM(\cdot, \cdot)$ is the structural similarity loss and $L_1(\cdot, \cdot)$ is the L1-norm loss (Both losses provide **higher amplitudes gradients** than gradients back-propagated

from $\mathcal{L}_S$). Due to the image-wise optimization nature of model inversion, each $\mathbf{J}_T$ over-fit different noise of the source model $S(\cdot)$. When training on all $\mathbf{J}_T$ samples, $C(\cdot)$ tends to produce averaged foreground semantics smoothing out sample-specific details that may be critical to detection.

We propose a simple fix, called Decayed Semantic Supervision(**DSS**), to such a problem:

$$\mathcal{L}_{DSS} = \lambda_{DSS}(SSIM(\mathbf{J}, \mathbf{J}_T) + L_1(\mathbf{J}, \mathbf{J}_T)) + \mathcal{L}_S, \tag{3.3}$$

where λ_{DSS} is a scalar that continuously decays with the increase of iterations (see supplementary materials). We initially let $\mathbf{J}_T$ provide strong gradients/supervision on $\mathbf{J}$, then let the weaker gradients from the source model losses $\mathcal{L}_S$ amend the acute details missing in averaging $\mathcal{L}_{DSS}$s overall $\{\mathbf{X}, \mathbf{J}_T\}$ pairs. Effectiveness of **DSS** is qualitatively shown in Figure 3.2(b) and quantitatively evaluated in Table 4.4.

3.2.4 Skipped Inverted Attention (SIA)

The Skipped Inverted Attention (SIA) is applied to amplify and balance the gradients from $\mathcal{L}_S$. The strong gradients at the high-level layers of $S(\cdot)$ (see "Avg. Gradients of Res50-p4to6 " in Figure 3.2(a) of a ResNet-FPN-MaskRCNN $S(\cdot)$ module) does not propagate into strong "Gradients at **J**".

To address this issue, we generate a 2D inverted attention mask $\mathbf{A}$ from the above gradients $\mathbf{G}$ and skip the earlier ResNet layers backward to supervise $C(\cdot)$. Formally, from $\mathbf{G} \in \Re^{width \times height \times channel}$, $\mathbf{A}(\mathbf{G}) \in \Re^{width \times height}$ is computed as, $\mathbf{a}(i) = \begin{cases} 0, & \text{if} \quad \sum_{j=1}^{channel} g(i,j) \geq q \\ 1, & \text{otherwise} \end{cases}$ where g_p is a scalar of the $(100-p)^{\text{th}}$ percentile of $\sum_{j=1}^{channel} \mathbf{G}(:,j)$, $(i \in [1, \cdots, width \times height])$. Low $\mathbf{G}$ and high $\mathbf{A}$ values denote under-represented regions by $S(\cdot)$ marked by white pixels in Figure 3.2(b)"Inverted Attention $\mathbf{A}$ in SIA". Forwarding the element-wise masked feature $\mathbf{A} \odot \mathbf{J}$ through $S(\cdot)$, we then update $T(\cdot)$ by the **SIA** loss

$$\mathcal{L}_{SIA} = \mathcal{L}_S(S(\mathbf{A} \odot \mathbf{J}), \mathbf{Y}). \tag{3.4}$$

Training with the $\mathbf{A} \odot \mathbf{J}$-induced loss $\mathcal{L}_{SIA}$, $C(\cdot)$ is forced to balance feature learning in all regions (See strong foreground gradients at $\mathbf{J}$ in Figure 3.2(c)).

In summary, the ARC Training procedure consists of initializing $T(\cdot)$ with **(1)** and updating $T(\cdot)$ with **(2-3)**. Figure 3.2(c) shows that ARC training produces foreground-focused features in **J**, stronger gradients at **J** and better instance mask detection than the *Naïve* training in Figure 3.2(a).

Chapter 4

Experiments

4.1 Keywords Explanation

"Standard" refers to the source model training strategy (backbone pretrained on Imagenet and $S(\cdot)$ fully updated with $\mathcal{L}_S$).

"ARC training" refers to our techniques.

"ARC-Self-supervised": Training on Pseudo-GT generated by inference $S(\cdot)$ with the source inputs of the target-source pairs in the target-modality training set.

"ARC-Supervised": Training on manually annotated GT of the target-modality training set.

"ARC-Semi-Supervised": Training on Pseudo-GT and a subset of manually annotated GT. All hyper-parameters of ARCare described in the supplementary materials.

Model (on **x101-GT**)	Input	Training Strategy	Box mAP ↑	Mask mAP ↑	Target-Modality Annot. (%) ↓	Inference Flops|#Para.
Source models $S(\cdot)$						
R50-FPN-MaskRCNN	RGB	Coco-pretrain [60] + Standard	82.36	87.94	-	61.42G|44.30M
Target models baselines						
PiW [56]|$S(\cdot)$	CSI	Source Init. $S(\cdot)$+ Standard	59.86	45.08	100	62.27G|45.0M
Wi2Vi [29]|$S(\cdot)$	CSI	RGB-FG-pretrained Wi2Vi [29]	0.12	0.09	100	63.22G|49.82M
Wi2Vi [29]|$S(\cdot)$	CSI	Source Init. $S(\cdot)$+Standard	68.12	54.79	100	63.22G|49.82M
Ours target models $C(\cdot)\|S(\cdot)$						
ARC-Self-supervised	CSI	ARC training	71.21	63.67	**0**	70.10G|51.34M
ARC-Semi-Supervised	CSI	ARC training	74.65	65.86	10	70.10G|51.34M
ARC-Supervised	CSI	ARC training	**77.38**	**66.49**	100	70.10G|51.34M

Table 4.1: WiFi CSI-input model results on the Person-in-Wifi(PiW) [56] dataset (all 16 layouts). All models were trained and evaluated against the **X101-GT** generated by X101-FPN-MaskRCNN(×3) in Detectron2 [60] model zoo. The best target model metrics are in bold.

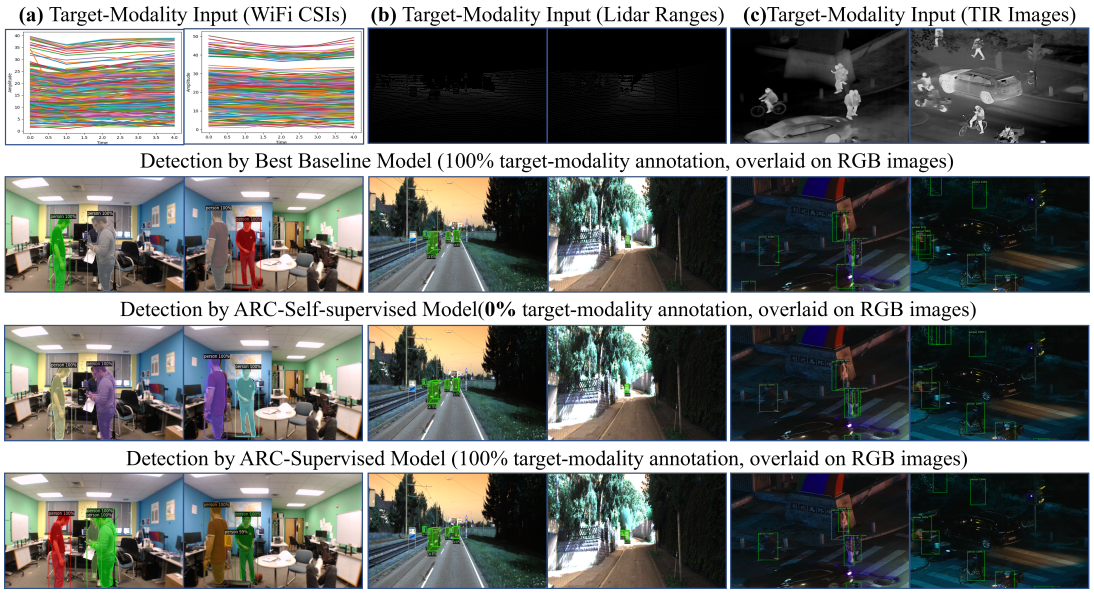

Figure 4.1: Qualitative results on three target-modality inputs **(a)** WiFi CSIs (amplitude sequences corresponding to each frame), **(b)** Lidar Ranges (sparse depth points projected on the pixel grid) and **(c)** Thermal InFraRed (TIR) images.

4.2 Implementation Details

WiFi-input Model For WiFi-input Model, we use R50-FPN-MaskRCNN [17] as our source detector and Wi2Vi [29] + VQ-VAE2 [48] as calibrator. We used two-level latent maps which are approximately 16x, 2x times smaller than the original image. The codebook size is set as 1024. The decay factor for Decayed Semantic Supervision λ_{DSS} is set as 0.9999. The percentile threshold p used in Skipped Inverted Attention is 0.1.

We use 8 GPUs with a batch size 64 for training. Following the R50-FPN-MaskRCNN training codes, the ARC target-input model is trained using Adam optimizer for 150k iterations with weight decay 0.0001 and an initial learning rate of 0.0001. The training process is warmed up by a linear warm-up-scheduler with 0.001 factor for 1k iterations and then the learning rate is decayed by 10x times at 90k iteration.

Lidar-input Model For Lidar-input Model, we use DD3D [43] as our source-

detector and VQVAE2 as calibrator. We used three-level latent maps which are approximately 32x, 4x, 2x, times smaller than the original image. The codebook size is set as 4096. We use 8 GPUs with a batch size 64 for training. The decay factor for Decayed Semantic Supervision λ_{DSS} is set as 0.9995. The percentile threshold p used in Skipped Inverted Attention is 0.1.

Following DD3D training codes, the ARC target-input model is trained using SGD optimizer for 24k iterations with weight decay of 0.0001, momentum of 0.9, and an initial learning rate of 0.002. The training process is warmed up by a linear warm-up-scheduler with 0.001 factor for 2k iterations and then the learning rate is decayed by 10x times at 21.5k and 25k iterations.

Infrared-input Model For Infrared-input Model, we use R50-FPN-FastRCNN as our source detector and VQVAE2 as the calibrator. We used three-level latent maps which are approximately 32x, 4x, 2x, times smaller than the original image respectively. The codebook size is set as 4096. We use 8 GPUs with a batch size 64 for training. The decay factor for Decayed Semantic Supervision λ_{DSS} is set as 0.9999. The percentile threshold p used in Skipped Inverted Attention is 0.1.

Following the R50-FPN-FastRCNN training codes, the ARC target-input model is trained using the Adam optimizer for 50k iterations with a weight decay 0.0001 and an initial learning rate of 0.0001. The training process is warmed up by a linear warm-up-scheduler with a 0.001 factor for 1k iterations and then the learning rate is decayed by 10x times at 30k iterations.

4.3 Image-wise Model Inversion

Our Image-wise model inversion following [2, 54] is respectively conducted on the three different source models to generate foreground semantics $\mathbf{J}_S$. Examples are shown in Figure A.1, Figure A.2, and Figure A.3 on the appendix. In all cases, $\mathbf{J}_S$ contains clear foreground-sensitive features.

4.4 WiFi-input Target Model

In Table 4.1, we build WiFi-input models upon a RGB-input MaskRCNN model(source model) on the Person-in-Wifi(**PiW**) [56] dataset. The PiW dataset contains synchronized RGB videos (20FPS) and Channel State Information(CSI) sequences (100Hz) of the Wifi signal. There are 16 3-transmitter|3-receiver antenna indoor layouts with 1 to 5 person captured. One RGB frame (resized to $[3, 640, 384]$) corresponds to a CSI tensor ([samples, transmitters, receivers, sub-carriers]=$[5, 3, 3, 30]$). We obtained a copy of the PiW Dataset from the authors[1], and follow their protocols on 80% randomly selected frames for training and the remaining 20% frames for testing. The model in [56] only produces image-wise semantic mask and body-joint heatmaps and cannot be directly compared on person detection metrics. No common ground truth was annotated to compare the RGB-input model and WiFi-input model.

In Table 4.1, we evaluate ARC under the following setting: **(1)** Common ground truth **X101-GT**, generated by a MS-COCO-pre-trained ResNeXt101-FPN-MaskRCNN-32x8d(x3) model in Detectron2 [60] on RGB inputs. All models are trained on and evaluated against X101-GT. **(2)** Source model $S(\cdot)$: an RGB-input R50-FPN-MaskRCNN pre-trained on MS-COCO and fine-tuned on the PiW data. **(3)** Target model baselines: We add the CSI-to-RGB modules of PiW [56] and Wi2Vi [29] to $S(\cdot)$ respectively, resulting in two baselines target-input models: "PiW|$S(\cdot)$" and "Wi2Vi|$S(\cdot)$". We train target baselines by randomly initializing their CSI-to-RGB modules, initializing $S(\cdot)$ with source model weights, and training with the "Standard" strategy. Pre-training the Wi2Vi module to synthesize the foreground-cropped RGB pixels [29], denoted by "RGB-FG-pre-trained WiVi", does not work well on the multi-person and multi-layout PiW data.

"Our target models $C(\cdot)|S(\cdot)$" were trained by "ARC" under three configurations: "ARC-Self-supervised" produces box and mask mAP of $[71.21, 63.67]$ using **0%** target-modality annotation, which outperforms the best target baseline metrics $[68.12, 54, 79]$ on 100% target-modality annotation. This shows that ARC effectively transferred priors from the strong RGB-input model to the CSI-input model. Comparing to the best baseline(Wi2Vi), the overheads of ARC models are 10% in Flops and 3% in #Para. "ARC-Semi-supervised" and "ARC-Supervised" outperforms all other target

[1]https://www.donghuang-research.com/publications

models using 10% and 100% target-modality annotation, respectively. Figure 4.1(a) visualizes the results.

Models	Input	Training	Car BEV $AP\|_{40}$ [Easy, Med., Hard] ↑	Car 3D-Bbox $AP\|_{40}$ [Easy, Med., Hard] ↑	Target-Modality Annot. (%) ↓	Inference Flops|#Para.
Source models $S(\cdot)$						
DD3D-DLA34 [43](paper)	RGB	Standard	[31.0, 22.6, 20.0]	[23.2, 16.3, 14.2]	-	109.9G|25.6M
DD3D-DLA34 (github)	RGB	Standard	[31.7, 24.4, 21.7]	[22.6, 17.0, 14.9]	-	109.9G|25.6M
Target model baselines						
DD3D-DLA34	Range	Standard	[40.7, 25.4, 22.0]	[29.4, 17.9, 14.9]	100	109.9G|25.6M
Ours target models $C(\cdot)\|S(\cdot)$						
ARC-Self-supervised	Range	ARC	[41.5, 26.1, 23.2]	[30.2, 18.2, 15.4]	**0**	123.6G|28.1M
ARC-Semi-Supervised	Range	ARC	[43.6, 27.3, 25.5]	[32.1, 19.8, 16.8]	10	123.6G|28.1M
ARC-Supervised	Range	ARC	**[46.3, 33.4, 30.9]**	**[35.4, 21.8, 19.9]**	100	123.6G|28.1M

Table 4.2: Lidar Range-input model results on KITTI [12]. Metrics(Car metrics only) are evaluated on the frontal sector of Lidar scans matching the RGB Camera 2 field-of-view. The best target model metrics are in bold.

4.5 Lidar-input Target Model

We build Lidar Range-input models upon a RGB-input DD3D model [43] on the Kitti-3D dataset [12]. Since there is no 360-degree RGB coverage that matches the 360-degree Lidar scans, we only evaluate results on the **frontal-view sector** of the Lidar scans[2] overlapped the RGB Camera#2 field-of-view. The 32-beam Lidar scans at 1-degree horizontal resolution, creating 32×90 points in the frontal-view sector, which are then projected to the 384×1270 pixel grid to match the RGB pixels. Following [11, 41], the missing range pixels are filled by a fixed depth value of 80 (meters). Following [43], we report $AP|_{40}$ [51] computed on the training|testing split of 3712|3769 samples. As shown in Figure 4.1(b), the range image has very sparse depth pixels with no visual appearance.

In Table 4.2, the "Target model baseline", directly training a DD3D on Range-inputs, produces higher metrics than the RGB-input Source models, which indicates the advantage of the Lidar over the RGB camera. It appears that the weak RGB-input model cannot provide good prior knowledge to develop the Lidar-input model. However, the ARC-Self-supervised target model, with **0%** target-modality annotation, still outperforms the target baseline trained on 100% annotation. Trained on 10% and 100% target-modality annotations respectively, our ARC-Self-Supervised and

[2]LaserNet [41], RangeDet [11] only reported results on 360-degree Range-inputs with no pre-trained model released to evaluate the frontal sector range data.

ARC-Supervised target models easily outperform all other models in all $AP|_{40}$ metrics. Figure 4.1(b) visualizes the results.

4.6 Infrared-input Target Model

On the LLVIP dataset [25], we show how ARC works when the source modality(RGB) contains far less information than the target modality(Thermal InfRared (TIR)) under low-light vision. There are 15488 RGB-TIR pairs with manually labeled 2D bounding boxes provided in the official repo [3]. We used the training/testing split proposed in [25].

In Table 4.3, we used the R50-FPN-FastRCNN (input size 1024×1280) for both the source model (RGB input) and target model baseline (TIR input). Both models were pre-trained on MS-COCO and fine-tuned on the RGB and TIR inputs respectively. The ARC target models are trained with the ARC training algorithm. The source model produces Bbox Average Precision(Box AP) of 43.83, which is clearly inferior to the target model baseline (Box AP 55.58), therefore may not provide strong priors or correct Pseudo labels for the target model. However, the ARC-Self-supervised target model still gets Box AP of 55.63 using **0%** target-modality annotation, which is comparable to the target baseline trained on 100% annotation. With only 10% annotation, the ARC-Semi-supervised model easily outperforms the target baseline. With 100% annotation, the ARC-supervised target model outperforms all other models. Figure 4.1(c) visualizes the results. In this experiment, the higher ARC overheads are due to the large input size, which could be reduced by concatenating $C(\cdot)$ with the smaller feature maps of $S(\cdot)$.

Models	Input	Training	Box AP ↑	Target-Modality Annot. (%) ↓	Inference Flops|#Para.
Source model $S(\cdot)$					
R50-FPN-FasterRCNN	RGB	Standard	43.83	-	255G|41.70M
Target model baseline					
R50-FPN-FasterRCNN	TIR	Standard	55.58	100	255G|41.70M
Ours target models $C(\cdot)\|S(\cdot)$					
ARC-Self-supervised	TIR	ARC	55.63	**0**	302G|44.35M
ARC-Semi-Supervised	TIR	ARC	57.08	10	302G|44.35M
ARC-Supervised	TIR	ARC	**58.05**	100	302G|44.35M

Table 4.3: Thermal InfRared TIR-input model results on the LLVIP dataset [25]. Best in bold.

[3] https://github.com/bupt-ai-cz/LLVIP

4.7 Ablation Study

In Table 4.4, we conduct an ablation study of the ARC training techniques on the "2018_10_17_2" subset of the PiW dataset. We start with the baseline training strategy: "RandInit+$\mathcal{L}_S$", and add ARC or alternative techniques in four groups. The cumulative relation among groups is denoted by their indents of "+"s. Each group is added upon its previous ARC techniques. For instance, "Feature-based KD..." and "Source Init. $S(\cdot)$..." are both added upon "FSR pretrained $C(\cdot)$...".

Training Strategies on $C(\cdot)\|S(\cdot)$	Box AP ↑	Mask AP ↑	Target-Modality Annot. (%)↓
Baseline			
Rand. Init.+Standard	75.28	66.26	100
ARC (bold rows) vs. Alternative techniques			
+ FSR pretrained $C(\cdot)$	**78.83**	**68.73**	100
+ w/o FSR pretrained $C(\cdot)$	76.69	67,03	100
+ FSR pretrained $C(\cdot)$	**78.83**	**68.73**	100
+ Feature-based KD [63] from $S(\cdot)$	70.90	54.43	100
+ Source Init. $S(\cdot)$ **and two-stage update**	**80.36**	**70.55**	100
+ SS loss $\mathcal{L}_{SS}$ (Eq.(3.2))	81.03	71.09	100
+ DSS loss $\mathcal{L}_{DSS}$ (Eq.(3.3))	**81.60**	**71.62**	100
+ RSC [22] on the $C(\cdot)$ output layer	80.94	71.21	100
+ RSC [22] on the Res-50-p5 layer	81.89	71.92	100
+ SIA(Eq.(3.4)	**82.15**	**72.33**	100
ARC-Self-supervised	75.14	65.96	**0**

Table 4.4: Ablation study of training techniques on ARC target model $C(\cdot)|S(\cdot)$ on the "2018_10_17_2" subset of PiW [56]. The cumulative relation among groups of "ARC vs.Alternative techniques" are denoted by their indents of "+"s. Each group of techniques is added upon the ARC techniques (**bold rows**) of the previous group.

Within each group of comparison, the ARC technique (bold rows) produces better metrics than the alternative counterparts. In particular, the ARC techniques provide better priors to the target model than "Feature-based KD from Source model". SIA is better than RSC [22] that directly computes and applies attention on the same layer ("the C(·) output layer" or "the Res-50-p5 layer"). Using 100% target-modality annotation, our final model (after applying SIA) produces [82.15, 72.33], which is significantly better than the baseline ([75.28, 66.26]). Trained on the Pseudo GT generated by $S(\cdot)$, ARC-Self-supervised produces [75.14, 65.96] comparable to the baseline that is trained on 100% manual target-modality annotation.

Also, we take the experiments with different numbers of pseudo images inverted from random GT Y'. In Table 4.4, our results [Box AP, Mask AP]=[82.15, 72.33] were reported with pseudo images of 100% real images. Using 0% and 50% pseudo images, the results are [77.13,68.25] and [79.83,70.12] respectively.

Chapter 5

Conclusions

In this thesis, we have presented ARC, a novel approach for improving the performance of machine learning models in the context of cross-modality transfer learning. Our approach is based on the idea of using an adversarial learning process between the Calibrator module and the Source module to bridge the gap between different modalities.

To ensure that the Calibrator module receives strong and clean supervision, we have developed the technique of source model inversion. This involves image-wise optimization using a pre-trained source model. By using this process to generate foreground semantics features, we are able to provide the Calibrator module with strong supervision that helps it learn more meaningful and useful features.

To help the Calibrator module converge more quickly and efficiently, we have developed the technique of foreground semantics reconstruction. This involves using the inverted latent representation of the input data to initialize the Calibrator module, which helps it to learn a more compact and meaningful latent space.

To regularize the model gradients and ensure that the Calibrator module learns useful and generalizable features, we have developed the technique of decayed semantic supervision. This involves using a decayed image-space semantic supervision to guide the training of the Calibrator module, while at the same time amending the acute details in the calibrator training by the Source model loss.

Finally, to amplify and balance the gradients of the Source module loss, we have developed the technique of skipped inverted attention. This involves fine-tuning the

Calibrator module with a combination of skipped connections and inverted attention mechanisms, which helps to improve the model's performance and generalizability.

Through extensive experiments, we have demonstrated the effectiveness of ARCin various modalities, and have shown that it can leverage prior knowledge of the source-modality model and as few as **zero** target-modality annotations to achieve the comparable performance of our baseline model.

Overall, one of the main goals of this work is to bridge researchers specialized in non-RGB-modalities with the matured RGB-input DNN models. While there has been significant progress in the development of RGB-input models, researchers who work with other types of modalities, such as WiFi signals, lidar points, or thermal images, often face unique challenges and have developed specialized approaches to address them. By developing techniques that enable the integration of non-RGB modalities with RGB-input models, this work aims to bridge the gap between these two groups of researchers and facilitate the sharing of ideas and best practices across different modalities. This, in turn, is expected to lead to more robust and effective machine-learning models that can be applied to a wide range of real-world applications.

Future work: One potential future direction for the ARCapproach is to extend it to text modalities by leveraging the powerful pre-trained language models that are widely used in natural language processing tasks. This would allow the ARCapproach to be applied to a wider range of cross-modality transfer learning scenarios, including tasks such as machine translation, dialogue systems, and text classification. Additionally, to further reduce the overhead of the Calibrator module, it may be possible to concatenate it with the smaller feature maps of the Source module, which would allow the Calibrator module to learn more efficient and compact representations of the input data. These improvements would make the ARCapproach even more efficient and effective, and could enable its use in a wider range of real-world applications.

Appendix A

Appendix

A.1 Image-wise Model Inversion Results

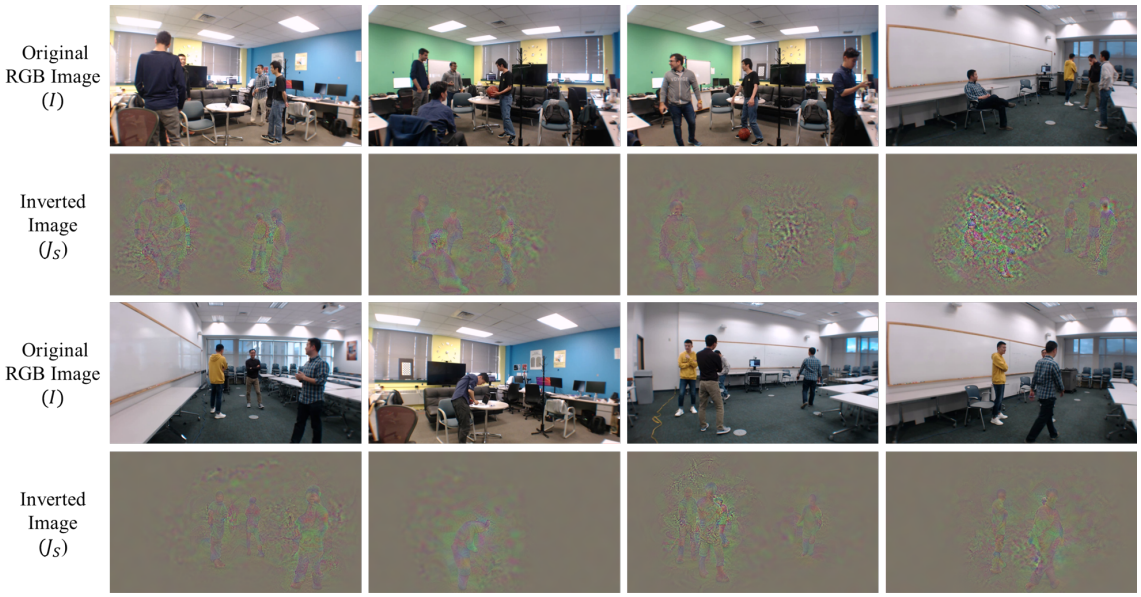

Figure A.1: Model inversion examples of the R50-FPN-MaskRCNN source model in the "WiFi-input Target Model" experiments

Figure A.2: Model inversion examples of the DD3D source model in the "Lidar-input Target Model" experiments.

Figure A.3: Model inversion examples of the R50-FPN-FasterRCNN source model in the "Infrared-input Target Model" experiments.

A.2 More Qualitative Results

In Figure A.4, Figure A.5, and Figure A.6, we report more qualitative results on three target-modality input models. In all cases, the ARC-Self-supervised models that are trained on 0% target-modality annotations produces comparable or better than the best baseline models trained on 100% annotations. Trained on the same 100% target-modality annotations, the ARC-Supervised models clearly outperform the best baseline models.

Figure A.4: More Qualitative results on WiFi-input models (overlaid on RGB images).

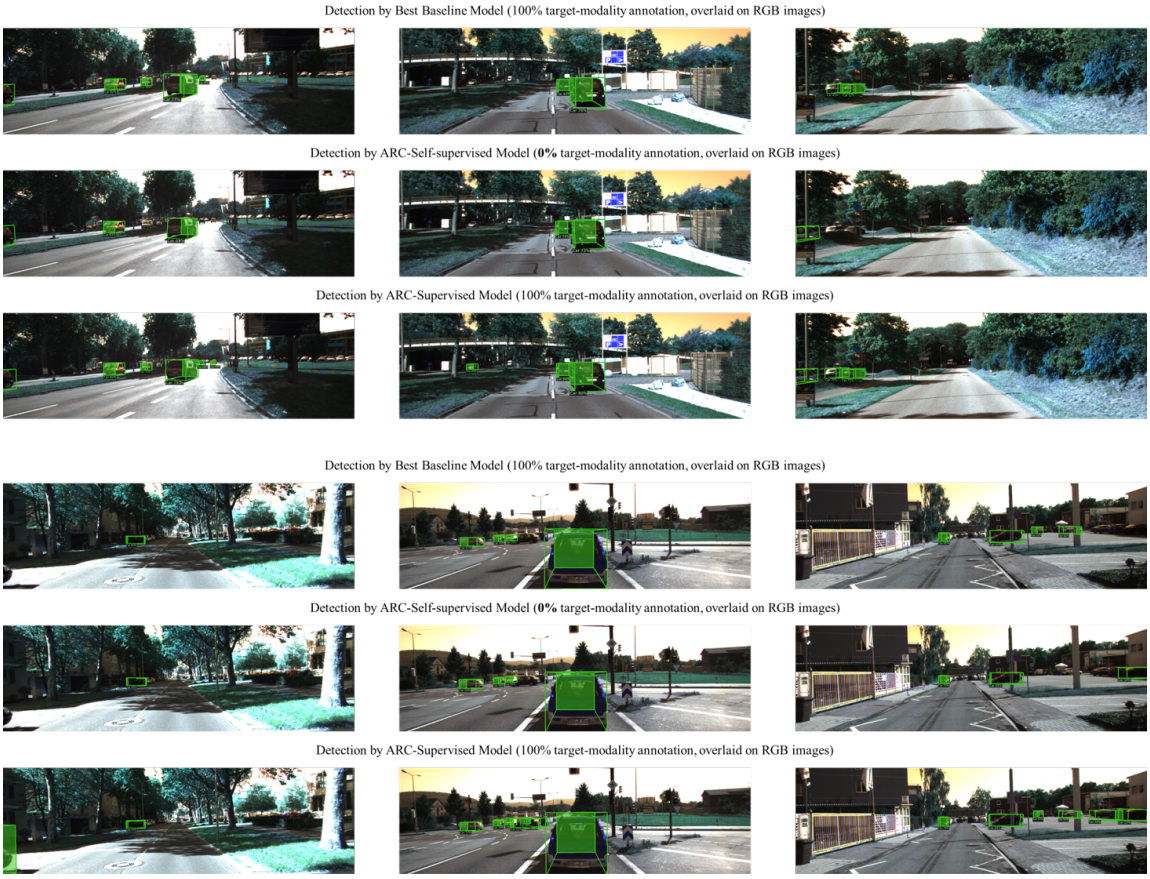

Figure A.5: More Qualitative results on Lidar-input models (overlaid on RGB images).

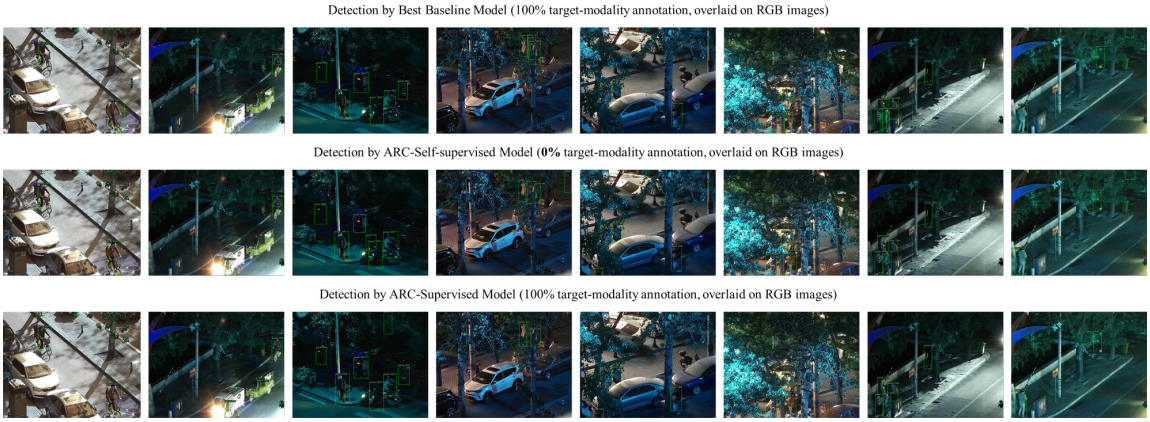

Figure A.6: More Qualitative results on Infrared-input models (overlaid on RGB images).

Bibliography

[1] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2): 157—166, 1994. 3.2

[2] Ang Cao and Justin Johnson. Inverting and understanding object detectors. *arXiv preprint arXiv:2106.13933*, 2021. 3.2.1, 4.3

[3] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *NIPS*, pages 742–751, 2017. 2.2

[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 1

[5] Shuhan Chen, Xiuli Tan, Ben Wang, Huchuan Lu, Xuelong Hu, and Yun Fu. Reverse attention based residual network for salient object detection. *TIP*, 29: 3763–3776, 2020. 2.3

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 1

[7] S. Depatla and Y. Mostofi. Crowd counting through walls using wifi. In *Proceedings of IEEE International Conference on Pervasive Computing and Communications*, 2018. 2.1

[8] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *CVPR*, pages 5138–5146, 2019. 2.2

[9] Michael Drob. Rf pix2pix unsupervised wi-fi to video translation. In *arXiv:2102.09345*, 2021. 2.1

[10] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. *arXiv preprint arXiv:2012.09841*, 2021. 2.3

[11] Lue Fan, Xuan Xiong, Feng Wang, Naiyan Wang, and ZhaoXiang Zhang. Rangedet: In defense of range view for lidar-based 3d object detection. In *ICCV*, October 2021. 2.1, 4.5, 2

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. (document), 1, 4.2, 4.5

[13] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. *arXiv preprint arXiv:2012.07177*, 2020. 1

[14] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010. 3.2

[15] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In *CVPR*, 2016. 2.2

[16] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Tool release: Gathering 802.11 n traces with channel state information. *ACM SIGCOMM Computer Communication Review*, 41(1):53–53, 2011. 3.1

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. URL http://arxiv.org/abs/1703.06870. 1, 4.2

[18] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. *arXiv preprint arXiv:1812.01187*, 2018. 1

[19] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2.2

[20] Yuenan Hou, Zheng Ma, Chunxiao Liu, and Chen Change Loy. Learning lightweight lane detection cnns by self attention distillation. In *ICCV*, pages 1013–1021, 2019. 2.3

[21] Zeyi Huang, Wei Ke, and Dong Huang. Improving object detection with inverted attention. In *WACV*, 2020. 2.3

[22] Zeyi Huang, Haohan Wang, Eric P. Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *ECCV*, 2020. 2.3, **??**, **??**, 4.7

[23] Zeyi Huang, Yang Zou, BVK Kumar, and Dong Huang. Comprehensive attention self-distillation for weakly-supervised object detection. *NIPS*, 33, 2020. 2.3

[24] Soonmin Hwang, Jaesik Park, Namil Kim, Yukyung Choi, and In So Kweon. Multispectral pedestrian detection: Benchmark dataset and baselines. In *CVPR*, 2015. 1, 2.1

[25] Xinyu Jia, Chuang Zhu, Minzhen Li, Wenqi Tang, and Wenli Zhou. Llvip: A

visible-infrared paired dataset for low-light vision. In *Proceedings of the ICCV*, pages 3496–3504, 2021. (document), 1, 2.1, 4.6, 4.3

[26] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, ChristopherSTAN, Liu Changyu, Laughing, tkianai, Adam Hogan, lorenzomammana, yxNONG, AlexWang1900, Laurentiu Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Francisco Ingham, Frederik, Guilhen, Hatovix, Jake Poznanski, Jiacong Fang, Lijun Yu, changyu98, Mingyu Wang, Naman Gupta, Osama Akhtar, PetrDvoracek, and Prashant Rai. ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements, October 2020. URL https://doi.org/10.5281/zenodo.4154370. 1

[27] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020. 2.1

[28] Sorachi Kato, Takeru Fukushima, T. Murakami, H. Abeysekera, Yusuke Iwasaki, T. Fujihashi, Takashi Watanabe, and S. Saruwatari. Csi2image: Image reconstruction from channel state information using generative adversarial networks. *IEEE Access*, 9:47154–47168, 2021. 2.1

[29] Mohammad Hadi Kefayati, Vahid Pourahmadi, and Hassan Aghaeinia. Wi2vi: Generating video frames from wifi csi samples. *IEEE Sensors Journal*, 20(19): 11463–11473, 2020. 2.1, 3.1, **??**, **??**, **??**, 4.2, 4.4

[30] JangHyun Kim, Wonho Choo, Hosan Jeong, and Hyun Oh Song. Co-mixup: Saliency guided joint mixup with supermodular diversity. In *International Conference on Learning Representations*, 2021. 2.3

[31] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020. 1

[32] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, pages 12697–12705, 2019. 2.1

[33] Heju Li, Xin He, Xukai Chen, Yinyin Fang, and Qun Fang. Wi-motion: A robust human activity recognition using wifi signals. *IEEE Access*, 7:153287 – 153299, 2019. 2.1

[34] Zhizhong Li and Derek Hoiem. Learning without forgetting. *PAMI*, 40(12): 2935–2947, 2017. 2.2

[35] Matthias Limmer and Hendrik P.A. Lensch. Infrared colorization using deep convolutional neural networks. *arXiv preprint arXiv:1604.02245*, 2019. 2.1

[36] 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*, Zürich, 2014. URL /se3/wp-content/uploads/2014/09/coco_eccv.pdf,http://mscoco.org. 1

[37] Lanlan Liu, Michael Muelly, Jia Deng, Tomas Pfister, and Jia Li. Generative modeling for small-data object detection. In *ICCV*, 2019. 2.3

[38] Shangqing Liu, Yanchao Zhao, Fanggang Xue, Bing Chen, and Xiang Chen. Deepcount: Crowd counting with wifi via deep learning. *arXiv preprint arXiv:1903.05316*, 2019. 2.1

[39] Xialei Liu, Hao Yang, Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Continual universal object detection. *arXiv preprint arXiv:2002.05347*, 2020. 2.2

[40] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 1

[41] Gregory P Meyer, Ankit Laddha, Eric Kee, Carlos VallespiGonzalez, and Carl K Wellington. Lasernet: An efficient probabilistic 3d object detector for autonomous driving. In *CVPR*, page 12677–12686, 2019. 2.1, 4.5, 2

[42] Stefan Milz, Martin Simon, Kai Fischer, and Maximillian Pöpperl. Points2pix: 3d point-cloud to image translation using conditional generative adversarial networks. *arXiv preprint arXiv:1901.09280*, 2019. 2.1

[43] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *ICCV*, 2021. 4.2, **??**, 4.5

[44] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2016. 2.1

[45] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 1, 2.1

[46] Jakaria Rabbi, Nilanjan Ray, Matthias Schubert, Subir Chowdhury, and Dennis Chao. Small-object detection in remote sensing images with end-to-end edge-enhanced gan and object detector network. *Remote Sensing*, 12(9):1432, 2020. 2.3

[47] Rahul Rajendran, Thaweesak Trongtirakul, Thaweesak Trongtirakul, Karen Panetta, and Sos Agaian. A pixel-based color transfer system to recolor nighttime imagery. In *Mobile Multimedia/Image Processing, Security, and Applications*, 2019. 2.1

[48] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *arXiv preprint arXiv:1906.00446*, 2019. 2.3, 4.2

[49] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 2.2

[50] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, June 2019. 2.1

[51] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel Lopez Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection: From single to multiclass recognition. *PAMI*, 2020. 4.5

[52] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, and et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 1

[53] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *arXiv preprint arXiv:1711.00937*, 2017. 2.3, 3.1

[54] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba. HOGgles: Visualizing Object Detection Features. *ICCV*, 2013. 3.2.1, 4.3

[55] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *CVPR*, pages 3156–3164, 2017. 2.2

[56] Fei Wang, Sanping Zhou, Stanislav Panev, Jinsong Han, and Dong Huang. Person-in-wifi: Fine-grained person perception using wifi. In *ICCV*, 2019. (document), 1, 2.1, 3.1, **??**, 4.1, 4.4, 4.4

[57] Lezi Wang, Ziyan Wu, Srikrishna Karanam, Kuan-Chuan Peng, Rajat Vikram Singh, Bo Liu, and Dimitris N Metaxas. Sharpen focus: Learning with attention separability and consistency. In *ICCV*, pages 512–521, 2019. 2.3

[58] Tao Wang, Li Yuan, Xiaopeng Zhang, and Jiashi Feng. Distilling object detectors with fine-grained feature imitation. In *CVPR*, pages 4933–4942, 2019. 2.2

[59] Wei Wang, Alex X. Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. Device-free human activity recognition using commercial wifi devices. *IEEE Journal on Selected Areas in Communications*, 35(5):1118–1131, 2017. 2.1

[60] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019. (document), 1, **??**, 4.1, 4.4

[61] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 2018. 2.1

[62] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention:

Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016. 2.2

[63] Linfeng Zhang and Kaisheng Ma. Improve object detection with feature-based knowledge distillation: Towards accurate and efficient detectors. In *ICLR*, 2021. ??

[64] Zhi Zhang, Tong He, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of freebies for training object detection neural networks. *arXiv preprint arXiv:1902.04103*, 2019. 1

[65] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. Through-wall human pose estimation using radio signals. In *CVPR*, pages 7356–7365, 2018. 1

[66] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*, 2017. 2.3

[67] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016. 3.1

[68] Barret Zoph, Ekin D. Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V. Le. Learning data augmentation strategies for object detection. In *ECCV*, 2020. 2.3