Align and Distill: Unifying and Improving Domain Adaptive Object Detection

Anonymous authors
Paper under double-blind review

Abstract

Object detectors often perform poorly on data that differs from their training set. Domain adaptive object detection (DAOD) methods have recently demonstrated strong results on addressing this challenge. Unfortunately, we identify systemic benchmarking pitfalls that call past results into question and hamper further progress: (a) Overestimation of performance due to underpowered baselines, (b) Inconsistent implementation practices preventing transparent comparisons of methods, and (c) Lack of generality due to outdated backbones and lack of diversity in benchmarks. We address these problems by introducing: (1) A unified benchmarking and implementation framework, Align and Distill (ALDI), enabling comparison of DAOD methods and supporting future development, (2) A fair and modern training and evaluation protocol for DAOD that addresses benchmarking pitfalls, (3) A new DAOD benchmark dataset, CFC-DAOD, enabling evaluation on diverse real-world data, and (4) A new method, ALDI++, that achieves state-of-the-art results by a large margin. ALDI++ outperforms the previous state-of-the-art by +3.5 AP50 on Cityscapes → Foggy Cityscapes, +5.7 AP50 on Sim10k → Cityscapes (where ours is the *only* method to outperform a fair baseline), and +0.6 AP50 on CFC Kenai → Channel. ALDI and ALDI++ are architecture-agnostic, setting a new state-of-the-art for YOLO and DETR-based DAOD as well without additional hyperparameter tuning. Our framework[†], dataset[‡], and state-of-the-art method offer a critical reset for DAOD and provide a strong foundation for future research.

1 Introduction

The challenge of DAOD. Modern object detector performance, though excellent across many benchmarks (Lin et al., 2014; Weinstein et al., 2021b;a; Bondi et al., 2018; Schneider et al., 2018; Rodriguez et al., 2011), often severely degrades when test data exhibits a distribution shift with respect to training data (Oza et al., 2023). For instance, detectors do not generalize well when deployed in new environments in environmental monitoring applications (Kay et al., 2022; Weinstein et al., 2021b). Similarly, models in medical applications perform poorly when deployed in different hospitals or on different hardware than they were trained (Xue et al., 2023; Guan & Liu, 2021). Unfortunately, in real-world applications it is often difficult, expensive, or time-consuming to collect the additional annotations needed to address such distribution shifts in a supervised manner. An appealing option in these scenarios is *unsupervised domain adaptive object detection (DAOD)*, which attempts to improve detection performance when moving from a "source" domain (used for training) to a "target" domain (used for testing) (Koh et al., 2021; Kalluri et al., 2023) without the use of target-domain supervision.

The current paradigm. The research community has established a set of standard benchmark datasets and methodologies that capture the deployment challenges motivating DAOD. Benchmarks consist of labeled data that is divided into two sets: a source and a target, each originating from different domains. DAOD methods are trained with source-domain images and labels, as in traditional supervised learning, and have access to unlabeled target domain images. The target-domain labels are not available for training.

To measure DAOD methods' performance, researchers use *source-only models* and *oracle models* as points of reference. Source-only models—sometimes also referred to as *baselines*—are trained with source-domain data

[†] https://anonymous.4open.science/r/aldi-0C88 [‡] https://anonymous.4open.science/r/cfc-daod-6D16

only, representing a lower bound for performance without domain adaptation. Oracle models are trained with *supervised* target-domain data, representing a fully-supervised upper bound. The goal in DAOD is to close the gap between source-only and oracle performance without target-domain supervision.

Impediments to progress. Recently-published results indicate DAOD is exceptionally effective, doubling the performance of source-only models and even outperforming fully-supervised oracles (Li et al., 2022b; Chen et al., 2022; Cao et al., 2023). However, upon close examination we discover problems with current benchmarking practices that call these results into question:

P1: Improperly constructed source-only and oracle models, leading to overestimation of performance gains. We find that source-only and oracle models are consistently constructed in a way that does not properly isolate domain-adaptation-specific components, leading to misattribution of performance improvements. We show that when source-only and oracle models are fairly constructed—*i.e.* use the same architecture and training settings as DAOD methods—no existing methods outperform oracles and *many methods do not even outperform source-only models* (Fig. 1), in stark contrast to claims made by recent work. These results mean we do not have an accurate measure of the efficacy of DAOD.

P2: Inconsistent implementation practices preventing transparent comparisons of methods. We find existing DAOD methods are built using a variety of different object detection libraries with inconsistent training settings, making it difficult to determine whether performance improvements come from new DAOD methods or simply improved hyperparameters. We find that tweaking these hyperparameters—whose values often differ between methods yet are not reported in papers—can lead to a *larger change in performance than the proposed methods themselves* (see Section 6.3), thus we cannot take reported advancements at face value. Without the ability to make fair comparisons we cannot transparently evaluate contributions nor make principled methodological progress.

P3: (a) Lack of diverse benchmarks and (b) outdated ~~model~~backbone architectures, leading to overestimation of methods' generality. DAOD benchmarks have focused largely on urban driving scenarios with synthetic distribution shifts (Sakaridis et al., 2018; Johnson-Roberson et al., 2016), and methods continue to use outdated detector ~~architectures~~backbones for comparison with prior work (Chen et al., 2018). ~~The underlying assumption is that methods will perform equivalently across application domains and backbone architectures.~~ We show that in fact the *ranking of methods changes across benchmarks and architectures*, revealing that published results may be uninformative for practitioners using modern architectures and real-world data.

A critical reset for DAOD research. DAOD has the potential for impact in a range of real-world applications, but these systemic benchmarking pitfalls impede progress. We aim to address these problems and lay a solid foundation for future progress in DAOD with the following contributions:

1. *Align and Distill (ALDI)*, a unified benchmarking and implementation framework for DAOD. In order to enable fair comparisons, we first identify key themes in prior work (Section 2) and unify common components into a single state-of-the-art framework, *ALDI* (Section 3). ALDI facilitates detailed study of prior art and streamlined implementation of new methods, supporting future research.

2. A fair and modern training protocol for DAOD methods, enabled by ALDI. We provide quantitative evidence of the benchmarking pitfalls we identify and propose an updated training and evaluation protocol to address them (Section 6.1). This enables us to set more realistic and challenging targets for the DAOD community and perform the first fair comparison of prior work in DAOD (Section 6.2).

3. A new benchmark dataset, CFC-DAOD, sourced from a real-world adaptation challenge in environmental monitoring (Section 5). CFC-DAOD increases the diversity of DAOD benchmarks and is notably larger than existing options. We show that the ranking of methods changes across different benchmarks (Section 6.2), thus the community will benefit from an additional point of comparison.

4. A new method, ALDI++, that achieves state-of-the-art results by a large margin. Using the same model settings across all benchmarks, ALDI++ outperforms the previous state-of-the-art by +3.5 AP50 on Cityscapes → Foggy Cityscapes, +5.7 AP50 on Sim10k → Cityscapes (where ours is the *only* method to outperform a fair source-only model), and +2.0 AP50 on CFC Kenai → Channel.

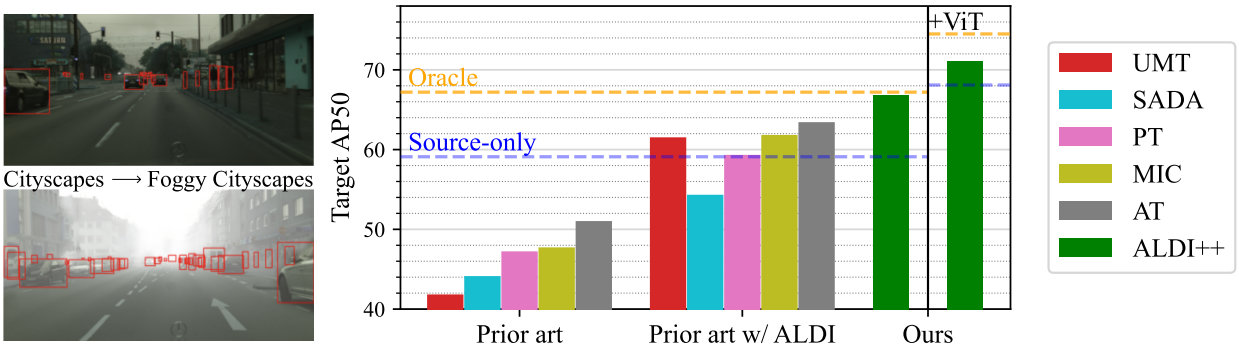


Figure 1: **ALDI provides a unified framework for fair comparison of domain adaptive object detection (DAOD) methods, and ALDI++ achieves state-of-the-art performance.** We show: **(1)** Inconsistent implementation practices give the appearance of steady progress in DAOD (left bars (Deng et al., 2021; Chen et al., 2021; 2022; Hoyer et al., 2023; Li et al., 2022b)); reimplementation and fair comparison with ALDI shows less difference between methods than previously reported (middle bars); **(2)** A fairly constructed source-only model (blue line) outperforms many existing DAOD methods, indicating less progress has been made than previously reported; and a proper oracle (orange line) outperforms *all* existing methods, in contrast to previously-published results; and **(3)** Our proposed method ALDI++ (green bars) achieves state-of-the-art performance on DAOD benchmarks such as Cityscapes → Foggy Cityscapes and is complementary to ongoing advances in object detection like VitDet (Li et al., 2022a).

2 Related Work

Our work concerns domain adaptive 2D object detection (DAOD). Two methodological themes have dominated recent DAOD research: *feature alignment* and *self-training/self-distillation*. We first give an overview of these themes and previous efforts to combine them, and then use commonalities to motivate our unified framework, *Align and Distill*, in Section 3.

Feature alignment in DAOD. Feature alignment methods aim to make target-domain data "look like" source-domain data, reducing the magnitude of the distribution shift. The most common approach utilizes an adversarial learning objective to align the feature spaces of source and target data (Ganin & Lempitsky, 2015; Chen et al., 2021; 2018; Zhu et al., 2019). Faster R-CNN in the Wild (Chen et al., 2018) utilizes adversarial networks at the image and instance level. SADA (Chen et al., 2021) extends this to multiple adversarial networks at different feature levels. Other approaches propose mining for discriminative regions (Zhu et al., 2019), weighting local and global features differently (Saito et al., 2019), incorporating uncertainty (Nguyen et al., 2020), and using attention networks (Vs et al., 2021). Alignment at the pixel level has also been proposed using image-to-image translation techniques to modify input images directly (Deng et al., 2021).

Self-training/self-distillation in DAOD. Self-training methods use a "teacher" model to predict pseudo-labels on target-domain data that are then used as training targets for a "student" model. Self-training can be seen as a type of *self-distillation* (Pham et al., 2022; Caron et al., 2021), which is a special case of knowledge distillation (Hinton et al., 2015; Chen et al., 2017) where the teacher and student models share the same architecture. Most recent self-training approaches in DAOD are based on the Mean Teacher (Tarvainen & Valpola, 2017) framework, in which the teacher model is updated as an exponential moving average (EMA) of the student model's parameters. Extensions to Mean Teacher for DAOD include: MTOR, which utilizes graph structure to enforce student-teacher feature consistency (Cai et al., 2019), Probabilistic Teacher (PT), which uses probabilistic localization prediction and soft distillation losses (Chen et al., 2022), and Contrastive Mean Teacher (CMT), which uses MoCo (He et al., 2020) for student-teacher consistency (Cao et al., 2023).

Combining feature alignment and self-training. Several approaches utilize *both* feature alignment and self-training/self-distillation, motivating our unified framework. Unbiased Mean Teacher (UMT) (Deng et al., 2021) uses mean teacher in combination with image-to-image translation to align source and target data at

the pixel level. Adaptive Teacher (AT) (Xue et al., 2023) uses mean teacher with an image-level discriminator network. Masked Image Consistency (MIC) (Hoyer et al., 2023) uses mean teacher, SADA, and a masking augmentation to enforce teacher-student consistency. Because these methods were implemented in different codebases using different training recipes and hyperparameter settings, it is unclear which contributions are most effective and to what extent feature alignment and self-training are complementary. We address these issues by reimplementing these approaches in the ALDI framework and perform fair comparisons and ablation studies in Section 6.

DAOD implementations. There are two components to an object detector design: the detection architecture (*e.g.* Faster R-CNN Ren et al. (2015), YOLO Redmon et al. (2016), DETR Carion et al. (2020)) and the backbone (*e.g.* VGG Simonyan & Zisserman (2014), ResNet He et al. (2016), ViT Dosovitskiy et al. (2020)). Current state-of-the-art methods in DAOD predominantly use Faster R-CNN architectures. DOAD methods for YOLO and DETR backbones have recently received some attention Zhou et al. (2023); Yu et al. (2022); Jia et al. (2023), but have yet to surpass Faster R-CNN-based methods' performance. For this reason, our main experiments also utilize the Faster R-CNN architecture. Existing methods differ in their choice of backbone, making comparisons difficult; we address this by consistently utilizing ResNet-50 backbones for all experiments and in our re-implementations of prior work. However, the ALDI framework is architecture and backbone agnostic, and we provide additional experiments using YOLO and DETR architectures, as well as ViT and ConvNeXt Liu et al. (2022) backbones.

DAOD datasets. Cityscapes (CS) → Foggy Cityscapes (FCS) (Cordts et al., 2016; Sakaridis et al., 2018) is a popular DAOD benchmark that emulates domain shift caused by changes in weather in urban driving scenarios. The dataset contains eight vehicle and person classes. Sim10k → CS (Johnson-Roberson et al., 2016) poses a Sim2Real challenge, adapting from video game imagery to real-world imagery. The benchmark focuses on a single class, "car". Other common tasks include adapting from real imagery in PascalVOC (Everingham et al., 2010) to clip art and watercolor imagery (Inoue et al., 2018). We report results on CS → FCS and Sim10k → CS due to their widespread popularity in the DAOD literature and focus on real applications. We note that existing benchmarks reflect a relatively narrow set of potential DAOD applications. To study whether methods generalize outside of urban driving scenarios, in Section 5 we introduce a novel dataset sourced from a real-world adaptation challenge in environmental monitoring, where imagery is much different from existing benchmarks.

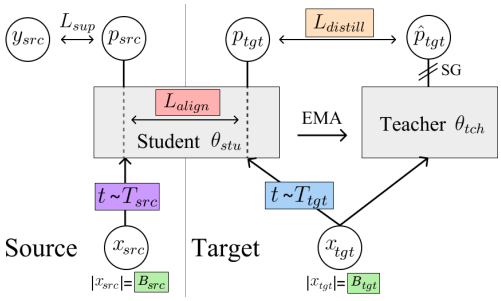
3 Align and Distill (ALDI): Unifying DAOD

We first introduce *Align and Distill (ALDI)*, a new benchmarking and implementation framework for DAOD. ALDI unifies feature alignment and self-distillation approaches in a common framework, enabling fair comparisons and addressing P2. Inconsistent implementation practices, while also providing the foundation for development of a new method ALDI++ that achieves state-the-art performance (Section 4, Section 6.2). The framework is visualized in Fig. 2. All components are ablated in Section 6.3.

Data. DAOD involves two datasets: a labeled source dataset X_{src} and an unlabeled target dataset X_{tgt}. Each training step, a minibatch of size B is constructed containing both B_{src} source images and B_{tgt} target images, $B = B_{\text{src}} + B_{\text{tgt}}$.

Models. ALDI is designed as a student-teacher framework to facilitate algorithms utilizing self-training/self-distillation. When enabled, both a student model θ_{stu} and a teacher model θ_{tch} are initialized with the same weights, typically obtained through supervised pretraining on ImageNet or X_{src}. Pretraining on X_{src} is often referred to as "burn-in." The student is trained via backpropagation, while the teacher's weights are updated each training step to be the EMA of the student's weights (Tarvainen & Valpola, 2017), *i.e.* $\theta_{\text{tch}} = \alpha\theta_{\text{tch}} + (1-\alpha)\theta_{\text{stu}}$ with $\alpha \in [0, 1]$. After training, we keep θ_{tch} and discard θ_{stu}. Algorithms that do not use self-training/self-distillation (*e.g.* SADA (Chen et al., 2021)) simply disable θ_{tch}.

In this paper we focus predominantly on two-stage detectors based on Faster R-CNN (Ren et al., 2015) as they are currently the state-of-the-art in DAOD, though we note that our framework ~~can be extended to other styles of detector in future work.~~is architecture-agnostic and also supports YOLO and DETR-based detectors. We provide additional YOLO and DETR results in Appendix A.1.



ALDI module	Example settings
Network init. $\theta_{stu}, \theta_{tch}$	ImageNet (Chen et al., 2021; Deng et al., 2021; Hoyer et al., 2023), Source-domain burn-in (Chen et al., 2022; Li et al., 2022b)
T_{src}, T_{tgt}	Flip and scale, SimCLR augs (Chen et al., 2022; Li et al., 2022b), Image masking (Hoyer et al., 2023)
$B_{src} : B_{tgt}$	1 : 1 (Chen et al., 2021; Deng et al., 2021; Hoyer et al., 2023), 2 : 1 (Chen et al., 2022; Li et al., 2022b)
$L_{distill}$	Pseudo-label (Deng et al., 2021; Hoyer et al., 2023; Li et al., 2022b), Knowledge distillation (Chen et al., 2022)
L_{align}	Adversarial (Li et al., 2022b; Chen et al., 2021; Hoyer et al., 2023), Image-to-image translation (Deng et al., 2021)

Figure 2: **(Left) The ALDI framework.** Each training step (moving left to right and bottom to top): **(1)** Sample B_{src} labeled source images x_{src}; transform by $t \sim T_{\text{src}}$; pass to student; compute supervised loss L_{sup} using ground-truth labels y_{src}. **(2)** Sample B_{tgt} unlabeled target images x_{tgt}; transform by $t \sim T_{\text{tgt}}$; pass to student to get preds p_{tgt}. Compute alignment objectives L_{align} using x_{src} and x_{tgt}. **(3)** Pass same unlabeled target data x_{tgt}, weakly transformed, to teacher; postprocess to obtain teacher predictions $\hat{p}_{\text{tgt}}$. Compute distillation loss L_{distill} between teacher and student predictions. Use stop gradient (SG) on teacher model; update teacher to the EMA of student's weights. **(Right) Example settings for each component of ALDI.** ALDI supports a range of existing methods off-the-shelf while providing a general implementation framework for new methods.

Training involves one or more of the following three objectives. We note that each objective is optional in order to support a range of algorithmic approaches.

1. Supervised training with source data. For each labeled source sample $x_{\text{src},i}$, we apply a transformation $t \sim T_{\text{src}}$, where T_{src} is the set of possible source-domain transformations. The transformed sample is passed through the student model to compute the supervised loss $\mathcal{L}_{\text{sup}}$ given the ground truth targets $y_{\text{src},i}$:

$$\mathcal{L}_{\text{sup}} = \frac{1}{B_{\text{src}}} \sum_{i=1}^{B_{\text{src}}} \mathcal{L}\left(\theta_{\text{stu}}(t(x_{\text{src},i})), y_{\text{src},i}\right) \tag{1}$$

where $\mathcal{L}(\cdot,\cdot)$ are standard object detection loss functions, *e.g.* those of Faster R-CNN (Ren et al., 2015).

2. Self-distillation with target data. For each unlabeled target sample $x_{\text{tgt},i}$, we transform the input using $\hat{t} \sim T_{\text{weak}}$ (a set of weak transformations) for the teacher model and $t \sim T_{\text{tgt}}$ (stronger transformations) for the student model. The teacher's predictions $\hat{p}_{\text{tgt},i}$ serve as distillation targets for the student's predictions $p_{\text{tgt},i}$, and we compute distillation loss $\mathcal{L}_{\text{distill}}$:

$$\hat{p}_{\text{tgt},i} = \theta_{\text{tch}}(\hat{t}(x_{\text{tgt},i})) \quad (2) \qquad p_{\text{tgt},i} = \theta_{\text{stu}}(t(x_{\text{tgt},i})) \quad (3) \qquad \mathcal{L}_{\text{distill}} = \frac{1}{B_{\text{tgt}}} \sum_{i=1}^{B_{\text{tgt}}} \mathcal{L}_{\text{distill}}\left(p_{\text{tgt},i}, \hat{p}_{\text{tgt},i}\right) \quad (4)$$

where teacher outputs $\hat{p}_{\text{tgt},i}$ are postprocessed to be either soft (*e.g.*, logits or softmax outputs) or hard (*e.g.*, thresholded pseudo-label) targets and the choice of distillation loss is method-specific. This formulation unifies different distillation techniques into a common objective, supporting a range of approaches.

3. Feature alignment. The source samples $x_{\text{src},i}$ and target samples $x_{\text{tgt},i}$ are optionally "aligned" using an alignment objective $\mathcal{L}_{\text{align}}$ that enforces invariance across domains at either the image or feature level. This formulation is general; however, in this paper, we focus on two common alignment losses: domain-adversarial training and image-to-image alignment.

Domain-adversarial training (*i.e.* DANN (Ganin & Lempitsky, 2015)) trains a domain classifier D to distinguish between source and target features, while the feature extractor aims to confuse D:

$$\mathcal{L}_{\text{align,DANN}} = -\frac{1}{B} \sum_{i=1}^{B} \left[y_{\text{dom},i} \log(D(\theta(x_i))) + (1 - y_{\text{dom},i}) \log(1 - D(\theta(x_i)))\right] \tag{5}$$

where $y_{\text{dom},i}$ is the domain label (source = 0, target = 1) and $\theta(x_i)$ is a feature representation of x_i.

Image-to-image alignment instead pursues domain invariance in the pixel space. Given image-to-image generative models G_{src}, G_{tgt} (*e.g.*, a CycleGAN (Zhu et al., 2017)), images are "translated" (a pixel-level transformation) from the source domain to the target domain and vice versa. We then obtain $x_{\text{tgt-like},i} = G_{\text{src}}(x_{\text{src},i})$, $x_{\text{src-like},i} = G_{\text{tgt}}(x_{\text{tgt},i})$, and substitute into Eq. (1), Eq. (2), and Eq. (3).

Unification of prior work. We demonstrate the generality of our framework by reimplementing five recently-proposed methods on top of ALDI for fair comparison: UMT (Deng et al., 2021), SADA (Chen et al., 2021), PT (Chen et al., 2022), MIC (Hoyer et al., 2023), and AT (Li et al., 2022b). We enumerate the settings required to reproduce each method in Appendix C.

4 ALDI++: Improving DAOD

We next propose a set of simple but effective enhancements to the *Align and Distill* approach. We call the resulting method ALDI++. We show in Section 6.2 that these enhancements lead to state-of-the-art results, and ablate each component in Section 6.3.

1. Robust burn-in. A key challenge in student-teacher methods is improving target-domain pseudo-label quality. We point out that pseudo-label quality in the early stages of self-training is largely determined by the *out-of-distribution (OOD) generalization* capabilities of the initial teacher model θ_{tch}^{init}, and thus propose a pre-training ("burn-in") strategy aimed at improving OOD generalization *before* self-training.

We add strong data augmentations including random resizing, color jitter, and cutout (DeVries & Taylor, 2017; Chen et al., 2020), and keep an EMA copy of the model during burn-in, two strategies that have previously been shown to improve OOD generalization (Morales-Brotons et al., 2024; Arpit et al., 2022; Gao et al., 2022), *i.e.* we pre-train a model θ with the loss from Eq. (1), where $t \sim T_{\text{src}}$ and L_{sup} are still the standard Faster R-CNN losses. Each iteration we update an EMA copy of the model,

$$\theta^{\text{EMA}} = \alpha\theta^{\text{EMA}} + (1-\alpha)\theta \tag{6}$$

with $\alpha \in [0, 1]$. After pre-training, we initialize $\theta_{\text{stu}} = \theta_{\text{tch}} = \theta^{\text{EMA}}$. We are the first to utilize these strategies for DAOD burn-in, and we show in Section 6.3 that this pre-training strategy leads to faster convergence time and better results.

2. Multi-task soft distillation. Most prior work utilizes confidence thresholding and non-maximum suppression to generate "hard" pseudo-labels from teacher predictions $\hat{p}_{tgt}$. However in object detection this strategy is sensitive to the confidence threshold, leading to false positive and false negative errors that harm self-training (Kay et al., 2023; RoyChowdhury et al., 2019). Inspired by the knowledge distillation literature we propose instead using "soft" distillation losses—*i.e.* using teacher prediction scores as targets without thresholding—allowing us to eliminate the confidence threshold hyperparameter.

We describe here our approach for two-stage (Faster R-CNN-based) object detection. Distillation implementation details for YOLO and DETR architectures can be found in Appendix B. We distill each task of Faster R-CNN—Region Proposal Network localization (*rpn*) and objectness (*obj*), and Region-of-Interest Heads localization (*roih*) and classification (*cls*)—independently. At each stage, the teacher provides distillation targets for the same set of input proposals used by the student—*i.e.* anchors A in the first stage, and *student* region proposals p_{tgt}^{rpn} in the second stage:

$$p_{tgt}^{rpn,obj} = \theta_{stu}^{rpn,obj}(A, x_{tgt}^{t}) \tag{7}$$

$$\hat{p}_{tgt}^{rpn,obj} = \theta_{tch}^{rpn,obj}(A, x_{tgt}^{\hat{t}}) \tag{8}$$

$$p_{tgt}^{roih,cls} = \theta_{stu}^{roih,cls}(p_{tgt}^{rpn}, x_{tgt}^{t}) \tag{9}$$

$$\hat{p}_{tgt}^{roih,cls} = \theta_{tch}^{roih,cls}(p_{tgt}^{rpn}, x_{tgt}^{\hat{t}}) \tag{10}$$

At each iteration, student distillation losses $L_{distill}$ are computed as:

$$L_{distill}^{rpn} = \lambda_0 L_{rpn}(p_{tgt}^{rpn}, \hat{p}_{tgt}^{rpn}) + \lambda_1 L_{obj}(p_{tgt}^{obj}, \hat{p}_{tgt}^{obj}) \tag{11}$$

$$L_{distill}^{roih} = \lambda_2 L_{roih}(p_{tgt}^{roih}, \hat{p}_{tgt}^{roih}) + \lambda_3 L_{cls}(p_{tgt}^{cls}, \hat{p}_{tgt}^{cls}) \tag{12}$$

$$L_{distill} = L^{rpn}_{distill} + L^{roih}_{distill} \tag{13}$$

Where L_{rpn} and L_{roih} are the smooth L1 loss and L_{obj} and L_{cls} are the cross-entropy loss, and $\lambda_{0...3} = 1$ by default. See Fig. 2 for a visual depiction, and the appendix for implementation details.

One prior DAOD work, PT (Chen et al., 2022), has also used soft distillation losses. Our method addresses two shortcomings: (1) PT requires a custom "Probabilistic R-CNN" architecture for distillation, while our approach is general and can work with any two-stage detector, and (2) PT uses $\hat{p}^{cls}$ as an indirect proxy for distilling p^{obj}, while our approach distills each task directly.

3. Revisiting DAOD training recipes. We also re-examine common design choices in DAOD in order to establish strong baseline settings for ALDI++. In particular, we find that two simple changes consistently improve domain adaptation results: (1) Using strong regularization on both target *and* source data during self-training~~(*i.e.* $T_{\text{src}}, T_{\text{tgt}} \supseteq T_{\text{strong}}$)~~, and (2) Training with equal amounts of source and target supervision in each minibatch (*i.e.* $B_{src} = B_{tgt}$). We also opt to disable all feature alignment in ALDI++ to stabilize training and find that the effects on accuracy are minimal (see Section 6.3).

5 The CFC-DAOD Dataset

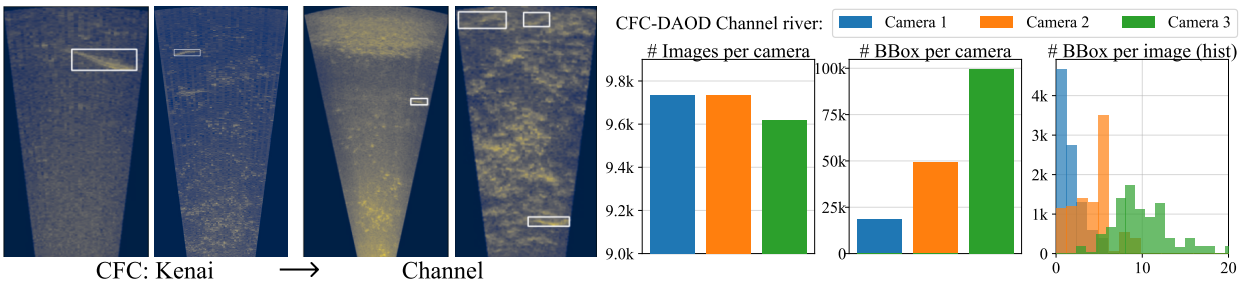


Figure 3: **The CFC-DAOD benchmark** focuses on detecting fish (white bounding boxes) in sonar imagery under domain shift caused by environmental differences between the training location (Kenai) and testing location (Channel). Our dataset contains 168k bounding boxes in 29k frames sampled from 150 new videos captured over two days from 3 different sonar cameras on the Channel river, enabling DAOD experiments. Here we visualize the distribution of images and annotations from each camera.

Next we introduce our dataset contribution, CFC-DAOD, addressing P3: (a) Lack of diverse benchmarks leading to overestimation of methods' generality.

CFC. The Caltech Fish Counting Dataset (CFC) (Kay et al., 2022) is a *domain generalization* benchmark sourced from fisheries monitoring, where sonar video is used to detect and count migrating salmon. The detection task consists of a single class ("fish") and domain shift is caused by real-world environmental differences between camera deployments. We identify this application as an opportunity to study the generality of DAOD methods due to its stark differences with existing DAOD benchmarks—specifically, sonar imagery is grayscale, has low signal-to-noise ratios, and foreground objects are difficult to distinguish from the background—however CFC focuses on generalization rather than adaptation and *does not include the data needed for DAOD.*

CFC-DAOD We introduce an extension to CFC, deemed CFC-DAOD, to enable the study of DAOD in this application domain. The task is to adapt from a source location—"Kenai", *i.e.* the default training set from CFC—to a difficult target location, "Channel". We collected an additional 168k bounding box annotations in 29k frames sampled from 150 new videos captured over two days from 3 different sensors on the "Channel" river (see Fig. 3). For consistency, we closely followed the video sampling protocol used to collect the original CFC dataset as described by the authors (see (Kay et al., 2022)). Our addition to CFC is crucial for DAOD as it adds an unsupervised training set for domain adaptation methods and a supervised training set to train oracle methods. We keep the original supervised Kenai training set from CFC (132k annotations in 70k images) and the original Channel test set (42k annotations in 13k images). We note

this is substantially larger than existing DAOD benchmarks (CS contains 32k instances in 3.5k images, and Sim10k contains 58k instances in 10k images). See Appendix D for more dataset statistics and Appendix G for qualitative visualizations. We make the dataset public.

6 Experiments

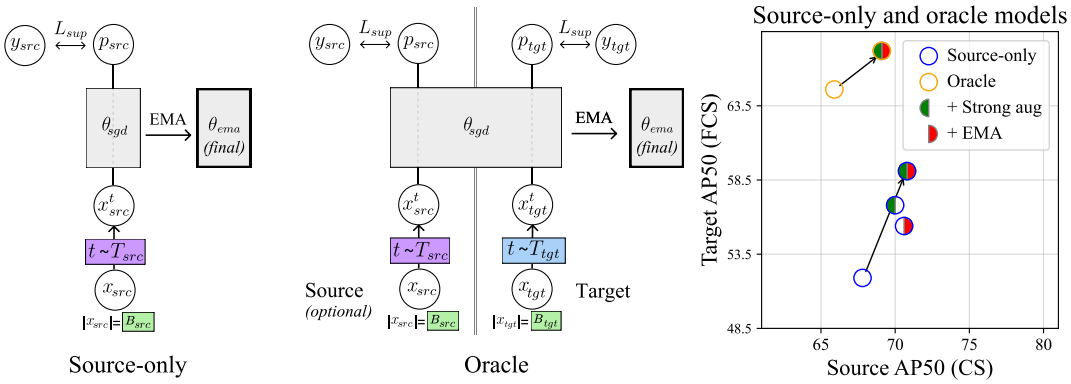


Figure 4: **Revisiting source-only and oracle models in DAOD.** We argue that in order to provide a fair measure of *domain adaptation* performance in DAOD, source-only and oracle models must utilize the same *non-adaptive* architectural and training components as methods being studied. In the case of *Align and Distill*-based approaches, this means source-only and oracle models must have access to the same set of source augmentations and EMA as DAOD methods. We see that these upgrades significantly improve source-only performance on target-domain data (+7.2 AP50 on Foggy Cityscapes), *even though the source-only model has never seen any target-domain data*, and these upgrades also improve oracle performance. Overall, these results set more challenging and realistic performance targets for DAOD methods.

In this section we propose an updated benchmarking protocol for DAOD (Section 6.1) that allows us to fairly analyze the performance of ALDI++ compared to prior work (Section 6.2) and conduct extensive ablation studies (Section 6.3).

6.1 Benchmarking Protocol

Datasets. We perform experiments on Cityscapes → Foggy Cityscapes, Sim10k → Cityscapes, and CFC Kenai → Channel. In addition to being consistent with prior work, these datasets represent three common adaptation scenarios capturing a range of real-world challenges: weather adaptation, Sim2Real, and environmental adaptation, respectively. We note that there have been inconsistencies in prior work in terms of which ground truth labels for Cityscapes are used. We use the Detectron2 version, which includes three intensity levels of fog $\{0.005, 0.01, 0.02\}$.

Metrics. For all experiments we report the PascalVOC metric of mean Average Precision with IoU ≥ 0.5 ("AP50") (Everingham et al., 2010). This is consistent with prior work on Cityscapes, Foggy Cityscapes, Sim10k, and CFC.

Revisiting source-only and oracle models. Here we address P1: Improperly constructed source-only and oracle models, leading to overestimation of performance gains. The goal of DAOD is to develop adaptation techniques that use unlabeled target-domain data to improve target-domain performance. Thus, in order to properly isolate *adaptation-specific* techniques, **any technique that does not need target-domain data to run should also be used by source-only and oracle models**. This means that source-only and oracle models should also utilize the same strong augmentations and EMA updates as DAOD methods.

In Fig. 4 we illustrate the resulting source-only and oracle models, and show that including these components significantly improves both source-only and oracle model performance (+7.2 and +2.6 AP50 on Foggy Cityscapes, respectively). This has significant implications for DAOD research: because source-only and

oracle models have not been constructed with equivalent components, performance gains stemming from better generalization have until now been *misattributed* to DAOD. With properly constructed source-only and oracle models, the gains from DAOD are much more modest (see Fig. 5). Note that for clarity we compare all methods in Fig. 1 and Fig. 5 against a single source-only and oracle model, however it would be more appropriate to compare each method to its own bespoke source-only and oracle models that use the same training components; see Appendix A.7 for the full comparison.

Fixed training settings. Prior work has used inconsistent backbones and image sizes, making head-to-head comparisons less fair. Using ALDI we instead compare using the same training settings, offering a fair comparison. As a starting point we utilize reasonably modern settings likely to be used by a practitioner: the Cityscapes defaults in the Detectron2 codebase. All methods in our comparisons, including source-only and oracle models, utilize Faster R-CNN architectures with ResNet-50 (Ren et al., 2015) backbones with FPN (Lin et al., 2017), COCO (Lin et al., 2014) pre-training, and an image size of 1024px on the shortest side. See Appendix C for more details.

6.2 Fair Comparison and State-of-the-Art Results

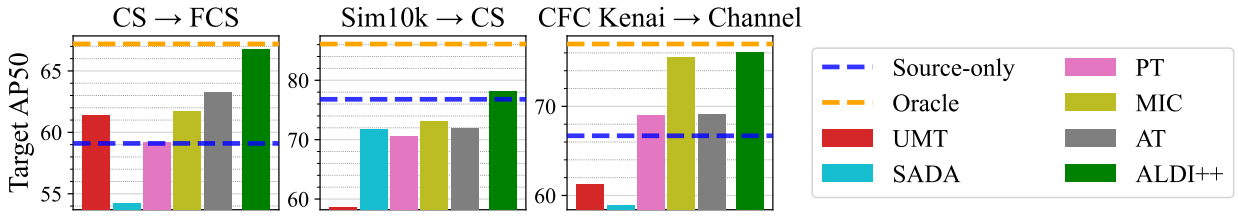


Figure 5: **Fair comparison of ALDI++ with existing state-of-the-art approaches using the ALDI framework and modern training recipes.** Some prior methods show consistent benefit but others lag behind fair source-only models. Our method ALDI++ outperforms prior work on all datasets studied by a significant margin: +3.5 AP50 on CS → FCS, +5.7 AP50 on Sim10k → CS, and +0.6 AP50 on CFC Kenai → Channel. Notably, ALDI++ is the *only* method to outperform a source-only model on Sim10k → CS.

We compare ALDI++ with reimplementations of five state-of-the-art DAOD methods on top of our framework: UMT (Deng et al., 2021), SADA (Chen et al., 2021), PT (Chen et al., 2022), MIC (Hoyer et al., 2023), and AT (Li et al., 2022b); see Appendix C for the ALDI settings used to reproduce them. We use the fair benchmarking protocol proposed in Section 6.1. Results are shown in Fig. 5. All methods (including ALDI++) use the same settings for all benchmarks.

Comparison with state-of-the-art. ALDI++ outperforms all prior work and sets a new state-of-the-art on all benchmarks studied, outperforming the next-best methods by +3.5 AP50 on CS → FCS, +5.7 AP50 on Sim10k → CS (where ours is the *only* method to outperform a fair source-only model), and +0.6 AP50 on CFC Kenai → Channel. ALDI++ achieves near-oracle level performance on CS → FCS and CFC Kenai → Channel (0.4 and 0.9 AP50 away, respectively), while other methods close less than half the gap between source-only and oracle models.

Comparison across datasets. We compare all methods on CS → FCS, Sim10k → CS, and CFC Kenai → Channel, in Fig. 5. We find the ranking of methods differs across datasets. ALDI++, MIC and AT are consistently the top-performing methods across all datasets. MIC performs noticeably better on CFC Kenai → Channel than other prior work, nearly matching the performance of ALDI++. UMT exhibits variable performance due to the differences in the difficulty of image generation across datasets (see Appendix C for examples). SADA underperforms other methods on CS → FCS and CFC Kenai → Channel, but closes this gap on the more difficult Sim10k → CS. These results demonstrate the utility of CFC-DAOD as another point of comparison for DAOD methods; we see that method performance on synthetic benchmarks like CS → FCS is not necessarily indicative of performance on real-world domain shifts.

Comparison with fair source-only and oracle models. Re-implementing methods in ALDI improves absolute performance of most methods due to upgraded training settings; however performance decreases

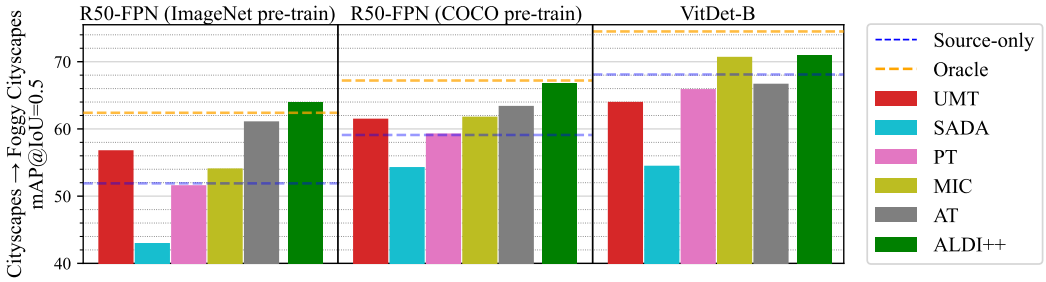


Figure 6: **Comparison across backbones and pre-training data.** Pre-training strategy does not significantly change the relative strength of methods compared to each other nor compared to source-only and oracle models. However, all models perform worse compared to source-only and oracle models when using VitDet backbones, with only ALDI++ and MIC outperforming a source-only model.

dramatically compared to source-only and oracle models. There are several instances where modernized DAOD methods are actually *worse* than a fair source-only model. Notably, a source-only model outperforms upgraded versions *all* previously-published work on Sim10 → CS. We also see that no state-of-the-art methods outperform a fair oracle on any dataset, in contrast to claims made by prior work (Li et al., 2022b; Chen et al., 2022; Cao et al., 2023).

Comparison across backbones and pre-training data. We compare all methods on CS → FCS using ImageNet vs. COCO pre-training, as well as Resnet-50 FPN backbones vs. VitDet-B backbones, in Fig. 6. See Appendix F for other datasets. We see that while COCO pre-training improves absolute performance of all methods, their ranking does not change significantly compared to ImageNet pre-training, nor does their performance in relation to source-only baselines. Interestingly, we find that ALDI++ outperforms an oracle on CS → FCS when using ImageNet pre-training, however we do not observe this trend on the other datasets (see Appendix F). We hypothesize that this may be due to noise in the Foggy Cityscapes target-train labels due to their programmatic generation, and that COCO pre-training helps prevent the oracle from overfitting to these erroneous boxes. We show that only ALDI++ and MIC continue to show improvements over an upgraded VitDet source-only model (see Appendix A.2 for other datasets), with ALDI++ performing slightly better than MIC (+0.4 AP50). We see there is a larger gap between the ViT ALDI++ and the ViT oracle compared to ResNet backbones, indicating the potential for future work to improve performance. Across all experiments in Fig. 6 we see that the source-only–oracle gap shrinks as the underlying model improves due to either stronger pre-training or backbone upgrades, indicating that DAOD may offer diminishing returns with stronger models.

6.3 Ablation Studies

In this section we ablate the performance of each component of ALDI on CS → FCS.

Base settings. For each ablation, unless otherwise specified we begin with the following training settings. We initialize $\theta_{stu}, \theta_{tch}$ with COCO pre-training followed by a burn-in phase on X_{src} with weak augmentations and early stopping based on validation performance. T_{src} includes random horizontal flip and random scaling. T_{tgt} includes random horizontal flip, random scaling, color jitter, and cutout. The $B_{\text{src}} : B_{\text{tgt}}$ batch ratio is 1:1. L_{distill} is hard pseudo-labeling with a confidence threshold of 0.8, and L_{align} is disabled. Note these base settings are not necessarily those of ALDI++ but rather the most commonly chosen values in prior work for each component. Additional training settings are reported in Appendix C.

θ_{stu}, θ_{tch} **Network initialization (burn-in).** In Fig. 7a we analyze the effects of our proposed burn-in strategy (see Section 4). We measure performance in terms of target-domain AP50 as well as convergence time, defined as the training time at which the model first exceeds 95% of its final target-domain performance. We compare our approach with: (1) No dataset-specific burn-in, *i.e.* starting with COCO weights, and (2) The approach used by past work—using a fixed burn-in duration, *e.g.* 10k iterations. We find that our method results in significant improvements in both training speed and accuracy, leading to upwards of 10% improvements in AP50 and reducing training time by a factor of 10 compared to training without burn-in.

Table 1: **Ablation studies.** **(a)** Effects of target-domain augmentation on self-training. Augmentations applied to student inputs (T_{tgt} in Fig. 2). Stronger augmentations improve performance considerably. **(b)** Effects of distillation objectives on self-training. We compare hard targets—used by most prior art, which thresholds teacher predictions to create pseudo-labels—with our proposed soft targets. Soft targets can improve overall performance. Results are the mean and standard deviation over 3 runs. **(c)** Feature alignment has diminishing returns. Compared to a source-only baseline AP50 of 59.1, feature alignment objectives L_{align} without self-training provides up to 2.6 AP50 of benefit (first row), but diminishes to 0.2 AP50 additional gain when used alongside self-training (last row).

(a)

T_{tgt}	$AP50_{FCS}$
Weak (scale & flip)	52.6
+ Color jitter	59.0
+ Color jitter + Erase	63.1
+ Color jitter + MIC	64.3

(b)

$L_{distill}$	$AP50_{FCS}$
Hard targets	63.7 ± 0.1
Soft targets	64.0 ± 0.4

(c)

L_{align}	$L_{distill}$	$AP50_{FCS}$
✓		61.7
	✓	63.7
✓	✓	63.9

T_{src} **Source augmentations.** In Fig. 7b we ablate the set of source-domain data augmentations. We compare using weak augmentations (random flipping and random scaling), strong augmentations (color jitter and cutout), and a combination of weak and strong, noting that prior works differ in this regard but do not typically report the settings used. We find that using strong source augmentations on the entire source-domain training batch outperforms weak augmentations and a combination of both.

T_{tgt} **Target augmentations.** In Table 1a we investigate the use of different augmentations for target-domain inputs to the *student* model. (We note that weak augmentations are always used for target-domain inputs to the teacher in accordance with prior work). We see that stronger augmentations consistently improve performance, with best performance coming from the recently-proposed MIC augmentation (Hoyer et al., 2023).

B_{tgt}/B **Batch composition.** In Fig. 7c we ablate the ratio of source and target data within a minibatch. We note that prior works differ in this setting but do not typically report what ratio is used. We see that using equal amounts of source and target data within each minibatch leads to the best performance. Notably, we also find that the inclusion of source-domain imagery is essential to see benefits from self-training—without any source imagery, $AP50_{FCS}$ drops from 64.5 to 59.3.

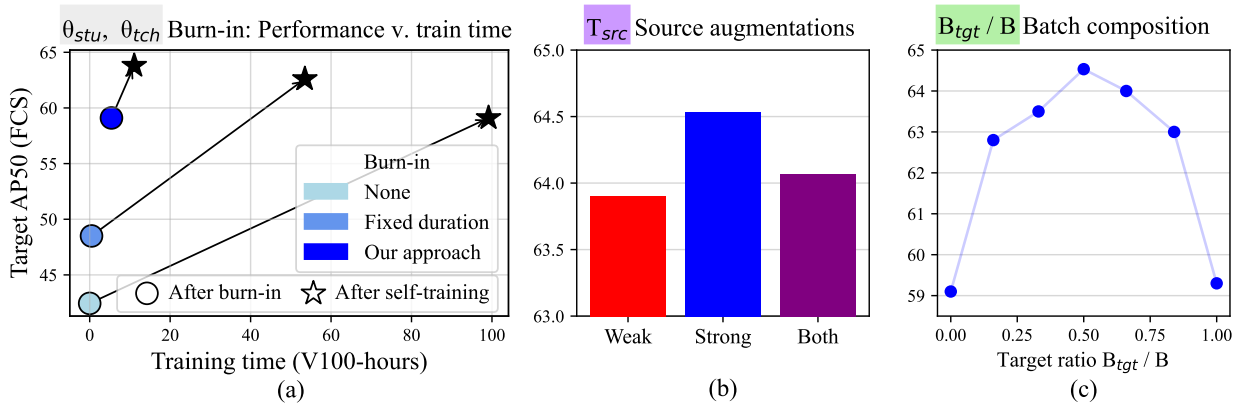


Figure 7: **Ablation studies.** **(a)** Our proposed burn-in strategy (Section 4) improves $AP50_{FCS}$ by +4.7 and reduces training time by 10x compared to no burn-in. **(b)** Strong source-data augmentations during self-training lead to better performance. **(c)** An equal ratio of source and target data during self-training leads to best performance.

$L_{distill}$ **Self-distillation.** In Table 1b we analyze the effects of our proposed multi-task soft distillation approach (see Section 4). We compare our approach with the "hard" pseudo-label approach used by prior work, where teacher predictions are post-processed with non-maximum suppression and a hard confidence threshold of 0.8 (Hoyer et al., 2023; Li et al., 2022b; Deng et al., 2021; Liu et al., 2021). For our proposed "soft" distillation method, we first sharpen teacher predictions at both detector stages using a sigmoid for objectness predictions and a softmax for classification predictions, both with a default temperature of 1. We see that our proposed soft targets improve performance compared to hard targets.

L_{align} **Feature alignment.** Finally we investigate the use of feature alignment. We implement an adversarial feature alignment approach consisting of an image-level and instance-level feature discriminator (our implementation performs on par with SADA while being simpler to train; see Appendix A.3). In Table 1c, we show that feature alignment used in isolation (*i.e.* without self-training) offers performance gains up to 2.6 AP50. However, these performance gains are smaller than those seen from self-training ($AP50_{FCS}$ of 61.7 vs. 63.7, respectively). When used in combination with self-training techniques, the additional benefit of feature alignment drops to $\leq$ 0.2 $AP50_{FCS}$. This suggests that self-training is currently the most promising avenue for progress and that more research is needed to develop complementary approaches. We also note that feature alignment approaches introduce training instability that may not be worth the small performance gain for practical use.

7 Discussion and Conclusions

In this work we proposed: the ALDI framework and an improved DAOD benchmarking methodology, providing a critical reset for the DAOD research community; a new dataset CFC-DAOD, increasing the diversity and real-world applicability of DAOD benchmarks; and a new method ALDI++ that advances the state-of-the-art. We conclude with key findings.

Network initialization has an outsized impact. We find that general advancements in computer vision eclipse progress in DAOD: a Resnet50-FPN source-only model outperforms all VGG-based DAOD methods, and a VitDet source-only model outperforms all Resnet50-FPN based DAOD methods. Similarly, simply adding stronger augmentations and EMA to source-only models leads to *better target-domain performance than some adaptation methods*, and including these upgrades during network initialization (burn-in) improves adaptation performance as well.

DAOD techniques are helpful, but do not consistently achieve oracle-level performance as previously claimed (Li et al., 2022b; Chen et al., 2022; Cao et al., 2023). Top-performing DAOD methods, including ALDI++, demonstrate improvements over source-only models (see Fig. 1 and Fig. 5). However, in contrast to previously-published results, no DAOD method consistently reaches oracle-level performance across datasets, architectures, and pre-training strategies, suggesting there is still room for improvement. The gap between DAOD methods and oracles is even larger for stronger architectures like VitDet. This is a promising area for future research.

Benchmarks sourced from real-world domain adaptation challenges can help the community develop generally useful methods. We find that DAOD methods do not necessarily perform equivalently across datasets (see Fig. 5). Diverse benchmarks are useful to make sure we are not overfitting to the challenges of one particular use case, while exposing and supporting progress in impactful applications. Our contributed codebase and benchmark dataset provide the necessary starting point to enable this effort.

A lack of transparent comparisons has incentivized incremental progress in DAOD. Most highly-performant prior works in DAOD are some combination of DANN (Ganin et al., 2016) and Mean Teacher (Tarvainen & Valpola, 2017) plus custom training techniques. Without fair comparisons it has been possible to propose near-duplicate methods that still achieve state-of-the-art performance due to hyperparameter tweaks. Our method ALDI++ establishes a strong point of comparison for *Align and Distill*-based approaches that will require algorithmic innovation to surpass.

Validation is the elephant in the room. All of our experiments, and all previously published work in DAOD, utilize a target-domain validation set to perform model and hyperparameter selection. This violates a key assumption in unsupervised domain adaptation: that no target-domain labels are available

to begin with. Prior work has shown that it may not be possible to achieve performance improvements in domain adaptation *at all* under realistic validation conditions (Musgrave et al., 2021; 2022; Kay et al., 2023). Therefore our results (as well as previously-published work) can really only be seen as an upper bound on DAOD performance. While this is valuable, further research is needed to develop effective unsupervised validation procedures for DAOD.

References

Devansh Arpit, Huan Wang, Yingbo Zhou, and Caiming Xiong. Ensemble of averages: Improving model selection and boosting performance in domain generalization. *Advances in Neural Information Processing Systems*, 35:8265–8277, 2022.

Elizabeth Bondi, Fei Fang, Mark Hamilton, Debarun Kar, Donnabell Dmello, Jongmoo Choi, Robert Hannaford, Arvind Iyer, Lucas Joppa, Milind Tambe, et al. Spot poachers in action: Augmenting conservation drones with automatic detection in near real time. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Qi Cai, Yingwei Pan, Chong-Wah Ngo, Xinmei Tian, Lingyu Duan, and Ting Yao. Exploring object relation in mean teacher for cross-domain detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11457–11466, 2019.

Shengcao Cao, Dhiraj Joshi, Liang-Yan Gui, and Yu-Xiong Wang. Contrastive mean teacher for domain adaptive object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23839–23848, 2023.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. *Advances in neural information processing systems*, 30, 2017.

Meilin Chen, Weijie Chen, Shicai Yang, Jie Song, Xinchao Wang, Lei Zhang, Yunfeng Yan, Donglian Qi, Yueting Zhuang, Di Xie, et al. Learning domain adaptive object detection with probabilistic teacher. *arXiv preprint arXiv:2206.06293*, 2022.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3339–3348, 2018.

Yuhua Chen, Haoran Wang, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Scale-aware domain adaptive faster r-cnn. *International Journal of Computer Vision*, 129(7):2223–2243, 2021.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

Jinhong Deng, Wen Li, Yuhua Chen, and Lixin Duan. Unbiased mean teacher for cross-domain object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4091–4101, 2021.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pp. 1180–1189. PMLR, 2015.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

Irena Gao, Shiori Sagawa, Pang Wei Koh, Tatsunori Hashimoto, and Percy Liang. Out-of-distribution robustness via targeted augmentations. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2022.

Hao Guan and Mingxia Liu. Domain adaptation for medical image analysis: a survey. *IEEE Transactions on Biomedical Engineering*, 69(3):1173–1185, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

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

Lukas Hoyer, Dengxin Dai, Haoran Wang, and Luc Van Gool. Mic: Masked image consistency for context-enhanced domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11721–11732, 2023.

Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5001–5009, 2018.

Peidong Jia, Jiaming Liu, Senqiao Yang, Jiarui Wu, Xiaodong Xie, and Shanghang Zhang. Pm-detr: Domain adaptive prompt memory for object detection with transformers. *arXiv preprint arXiv:2307.00313*, 2023.

Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023. URL `https://github.com/ultralytics/ultralytics`.

Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? *arXiv preprint arXiv:1610.01983*, 2016.

Tarun Kalluri, Wangdong Xu, and Manmohan Chandraker. Geonet: Benchmarking unsupervised adaptation across geographies. *CVPR*, 2023.

Justin Kay, Peter Kulits, Suzanne Stathatos, Siqi Deng, Erik Young, Sara Beery, Grant Van Horn, and Pietro Perona. The caltech fish counting dataset: A benchmark for multiple-object tracking and counting. In *European Conference on Computer Vision*, pp. 290–311. Springer, 2022.

Justin Kay, Suzanne Stathatos, Siqi Deng, Erik Young, Pietro Perona, Sara Beery, and Grant Van Horn. Unsupervised domain adaptation in the real world: A case study in sonar video. In *NeurIPS 2023 Computational Sustainability: Promises and Pitfalls from Theory to Deployment*, 2023.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664. PMLR, 2021.

Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *European Conference on Computer Vision*, pp. 280–296. Springer, 2022a.

Yu-Jhe Li, Xiaoliang Dai, Chih-Yao Ma, Yen-Cheng Liu, Kan Chen, Bichen Wu, Zijian He, Kris Kitani, and Peter Vajda. Cross-domain adaptive teacher for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7581–7590, 2022b.

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 *European conference on computer vision*, pp. 740–755. Springer, 2014.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. Unbiased teacher for semi-supervised object detection. *arXiv preprint arXiv:2102.09480*, 2021.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022.

Daniel Morales-Brotons, Thijs Vogels, and Hadrien Hendrikx. Exponential moving average of weights in deep learning: Dynamics and benefits. *Transactions on Machine Learning Research*, 2024.

Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. Unsupervised domain adaptation: A reality check. *arXiv preprint arXiv:2111.15672*, 2021.

Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. Benchmarking validation methods for unsupervised domain adaptation. *arXiv preprint arXiv:2208.07360*, 2022.

Dang-Khoa Nguyen, Wei-Lun Tseng, and Hong-Han Shuai. Domain-adaptive object detection via uncertainty-aware distribution alignment. In *Proceedings of the 28th ACM international conference on multimedia*, pp. 2499–2507, 2020.

Poojan Oza, Vishwanath A Sindagi, Vibashan Vishnukumar Sharmini, and Vishal M Patel. Unsupervised domain adaptation of object detectors: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11410–11420, 2022.

Minh Pham, Minsu Cho, Ameya Joshi, and Chinmay Hegde. Revisiting self-distillation. *arXiv preprint arXiv:2206.08491*, 2022.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016. URL `https://arxiv.org/abs/1506.02640`.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

Albert Reuther, Jeremy Kepner, Chansup Byun, Siddharth Samsi, William Arcand, David Bestor, Bill Bergeron, Vijay Gadepally, Michael Houle, Matthew Hubbell, Michael Jones, Anna Klein, Lauren Milechin, Julia Mullen, Andrew Prout, Antonio Rosa, Charles Yee, and Peter Michaleas. Interactive supercomputing on 40,000 cores for machine learning and data analysis. In *2018 IEEE High Performance extreme Computing Conference (HPEC)*, pp. 1–6. IEEE, 2018.

Mikel Rodriguez, Ivan Laptev, Josef Sivic, and Jean-Yves Audibert. Density-aware person detection and tracking in crowds. In *2011 International Conference on Computer Vision*, pp. 2423–2430. IEEE, 2011.

Aruni RoyChowdhury, Prithvijit Chakrabarty, Ashish Singh, SouYoung Jin, Huaizu Jiang, Liangliang Cao, and Erik Learned-Miller. Automatic adaptation of object detectors to new domains using self-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 780–790, 2019.

Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6956–6965, 2019.

Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126:973–992, 2018.

Stefan Schneider, Graham W Taylor, and Stefan Kremer. Deep learning object detection methods for ecological camera trap data. In *2018 15th Conference on computer and robot vision (CRV)*, pp. 321–328. IEEE, 2018.

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

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

Vibashan Vs, Vikram Gupta, Poojan Oza, Vishwanath A Sindagi, and Vishal M Patel. Mega-cda: Memory guided attention for category-aware unsupervised domain adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4516–4526, 2021.

Wen Wang, Yang Cao, Jing Zhang, Fengxiang He, Zheng-Jun Zha, Yonggang Wen, and Dacheng Tao. Exploring sequence feature alignment for domain adaptive detection transformers. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 1730–1738, 2021.

Ben G Weinstein, Lindsey Gardner, Vienna Saccomanno, Ashley Steinkraus, Andrew Ortega, Kristen Brush, Glenda Yenni, Ann E McKellar, Rowan Converse, Christopher Lippitt, et al. A general deep learning model for bird detection in high resolution airborne imagery. *bioRxiv*, 2021a.

Ben G Weinstein, Sarah J Graves, Sergio Marconi, Aditya Singh, Alina Zare, Dylan Stewart, Stephanie A Bohlman, and Ethan P White. A benchmark dataset for canopy crown detection and delineation in co-registered airborne rgb, lidar and hyperspectral imagery from the national ecological observation network. *PLoS computational biology*, 17(7):e1009180, 2021b.

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

Zhiyun Xue, Feng Yang, Sivaramakrishnan Rajaraman, Ghada Zamzmi, and Sameer Antani. Cross dataset analysis of domain shift in cxr lung region detection. *Diagnostics*, 13(6):1068, 2023.

Jinze Yu, Jiaming Liu, Xiaobao Wei, Haoyi Zhou, Yohei Nakata, Denis Gudovskiy, Tomoyuki Okuno, Jianxin Li, Kurt Keutzer, and Shanghang Zhang. Mttrans: Cross-domain object detection with mean teacher transformer. In *European Conference on Computer Vision*, pp. 629–645. Springer, 2022.

Huayi Zhou, Fei Jiang, and Hongtao Lu. Ssda-yolo: Semi-supervised domain adaptive yolo for cross-domain object detection. *Computer Vision and Image Understanding*, 229:103649, 2023.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

Xinge Zhu, Jiangmiao Pang, Ceyuan Yang, Jianping Shi, and Dahua Lin. Adapting object detectors via selective cross-domain alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 687–696, 2019.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

A Additional Experiments

A.1 YOLO and DETR Architectures

To demonstrate the architecture-agnosticism of our framework and enable further research, we implement ALDI for the one-stage detection architecture YOLOv5 Redmon et al. (2016); Jocher et al. (2023) and the transformer-based architecture Deformable DETR Carion et al. (2020); Zhu et al. (2020). Implementation details are further described in Appendix B.

Table 2: **ALDI-YOLO results.**

(a) CS → FCS

Method	AP50
Source-only (YOLOv5m)	58.8
SSDA-YOLO Zhou et al. (2023) (YOLOv5l)	55.9
ALDI-YOLO (ours, YOLOv5m)	**62.5**
Oracle (YOLOv5m)	66.3

(b) Sim10k → CS

Method	AP50
Source-only	75.0
ALDI-YOLO (ours)	75.0
Oracle	88.0

(c) CFC-DAOD

Method	AP50
Source-only	60.2
ALDI-YOLO (ours)	52.4
Oracle	76.7

Table 3: **DETR results on Cityscapes → Foggy Cityscapes.** We use 800px input size for consistency with prior work.

Method	AP50
Source-only	44.5
SFA Wang et al. (2021)	41.3
MTTrans Yu et al. (2022)	43.4
PM-DETR Jia et al. (2023)	44.3
ALDI-DETR (ours)	**44.8**
Oracle	50.0

A.2 ViT and ConvNeXt backbones

Table 4: Sim10k → Cityscapes

Method	$AP50_{CS}$
ViT-B baseline	81.7
ALDI++ + ViT-B	81.8
ViT-B oracle	89.8

Table 5: CFC Kenai → Channel

Method	$AP50_{Channel}$
ViT-B baseline	69.0
ALDI++ + ViT-B	71.1
ViT-B oracle	76.7

For completeness we show results using ALDI++ in combination with VitDet-B Li et al. (2022a) in Table 4 (Sim10k → Cityscapes) and Table 5 (CFC Kenai → Channel). We see that ALDI continues to demonstrate improvements over baselines even as overall architectures get stronger, though these improvements are smaller in magnitude than VitDet-B results on the CS → FCS dataset.

We also demonstrate the performance of ALDI++ with even larger backbones to examine how performance and domain gaps change. We show results from VitDet-L in Table 6, and with ConvNeXt-L in Table 7. Results are similar to our main results; we continue to see improvements of baselines, oracles, and ALDI++ in these settings.

A.3 Adversarial Feature Alignment

We report additional ablations for the adversarial feature alignment network(s) used, comparing our implementations of image-level alignment and instance-level alignment with a baseline and SADA. As we see in

Table 6: Cityscapes → Foggy Cityscapes

Method	$AP50_{FCS}$
ViT-L baseline	70.2
ALDI++ + ViT-L	76.1
ViT-L oracle	77.4

Table 7: Cityscapes → Foggy Cityscapes

Method	$AP50_{FCS}$
ConvNext-L baseline	58.9
ALDI++ + ConvNext-L	63.3
ConvNext-L oracle	64.1

Table 8a, Table 8b, and Table 8c, the best settings to use differ by dataset. By default our feature alignment experiments in Sec. 6.1 of the main paper use both instance and image level alignment. See Appendix B.6 below for further implementation details.

Table 8: **Comparison of adversarial alignment methods. (a)** Cityscapes → Foggy Cityscapes. We see that our implementations outperform SADA Chen et al. (2021) while being simpler. Image-level alignment is best, followed by Image + Instance. **(b)** Sim10k → Cityscapes. Instance-level alignment is best. **(c)** CFC Kenai → Channel. Image + Instance is best. We see there is no consistently-best strategy across datasets; however, we note that for all datasets, the benefit of using adversarial feature alignment is smaller than self-training (see Sec. 6.3 of the main paper).

(a)

Method	$AP50_{FCS}$
Source-only	51.9
SADA	54.2
Image-level (ours)	55.8
Instance-level (ours)	54.3
Image + Instance (ours)	54.9

(b)

Method	$AP50_{CS}$
Source-only	70.8
Image-level (ours)	71.8
Instance-level (ours)	73.3
Image + Instance (ours)	71.5

(c)

Method	$AP50_{Channel}$
Source-only	65.8
Image-level (ours)	65.2
Instance-level (ours)	66.0
Image + Instance (ours)	66.9

A.4 Visualizing Alignment

We investigate the overlap of source and target data in the feature space of different methods. For each method, we pool the highest-level feature maps of the backbone, either globally ("image-level") or per instance ("instance-level"). We then embed the pooled feature vectors in 2D space using PCA for visual inspection (see Fig 8). We also compute a dissimilarity score based on FID Heusel et al. (2017), by fitting Gaussians to the source and target features and then computing the Fréchet distance between them.

A.5 Teacher update

We compare other approaches to updating the teacher during self-training vs. using exponential moving average in Table 9. We see that EMA significantly outperforms using a fixed teacher (*i.e.* vanilla self-training, where pseudo-labels are generated once before training) as well as using the student as its own teacher without EMA.

Table 9: Comparison of teacher update approaches on Cityscapes → Foggy Cityscapes. Mean teacher greatly outperforms other options.

Method	$AP50_{FCS}$
Source-only baseline	51.9
No update (vanilla self-training)	52.9
Student is teacher	53.8
EMA (mean teacher)	63.5

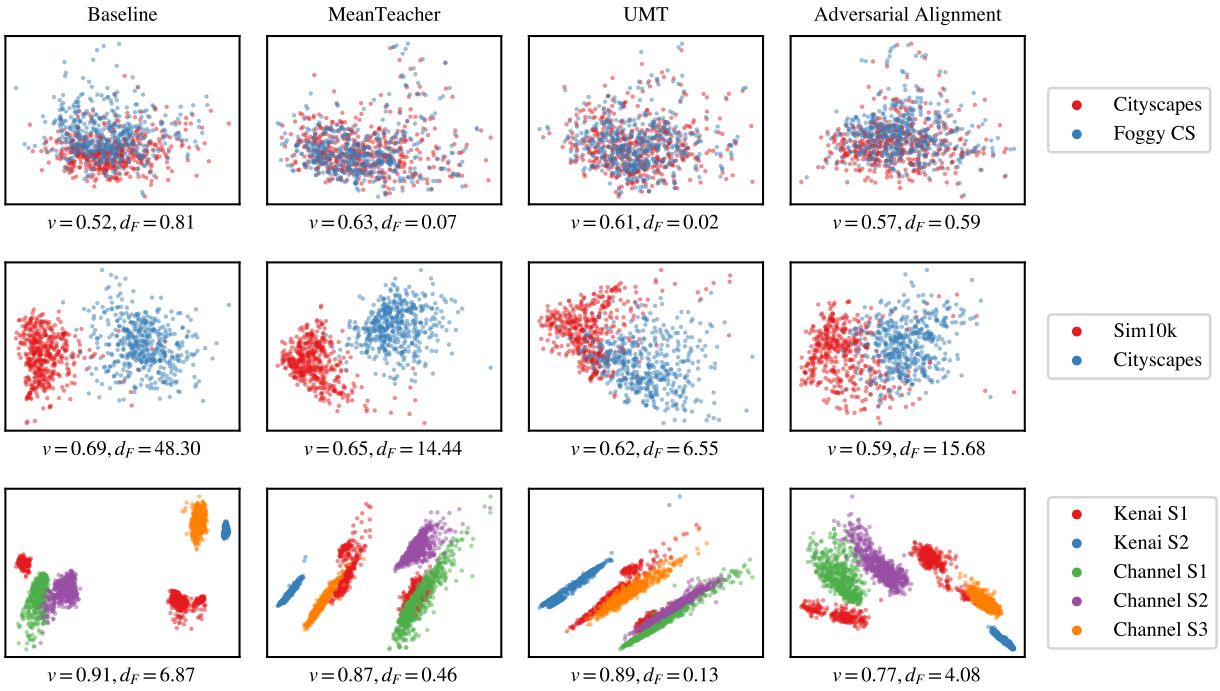


Figure 8: Embedding of pooled features from the final backbone layer in 2D space using PCA. We compare: (1) A source-only baseline, (2) The base settings from Section 6.3 ("MeanTeacher"), (3) UMT (which utilizes image-to-image translation), and (4) MeanTeacher + adversarial feature alignment using image-level features. The ratio of variance explained by the first two PCA components is given by v and a dissimilarity score between source and target features is given by d_F. d_F is lower than the baseline for all alignment methods and does roughly match the overall visual trend in feature overlap. In all cases, the simple MeanTeacher model significantly reduces the distance between source and target data even though there is no explicit alignment criterion, even resulting in a smaller d_F than adversarial alignment methods for CS → FCS & CFC Kenai → Channel.

A.6 Example of (Un)Fair Comparisons

In Fig. 9 we show a case study of why fair comparisons are impactful for DAOD research. We compare two similarly-performing prior works, AT and MIC, and see that implementation inconsistencies have led to nontransparent comparisons between the two methods. Notably, the originally reported results even used different ground truth test labels. When re-implemented on top of the same modern framework using ALDI, we are able to fairly compare the two methods for the first time.

A.7 Method-specific Source-Only Models

Our protocol for training fair source-only models introduced in Section 6.1 is to utilize all techniques from the methods being studied that do not need target data to run. For simplicity, in the main paper we have only displayed the source-only model that utilizes the same components as ALDI++, though these settings differ slightly from the other methods studied. In Fig. 10 we show an alternate view in which for each method we train a bespoke source-only model using the exact same training settings as the DAOD method. The main difference is the set of image augmentations used, except for SADA, which also does not use EMA. We see that there is only a small variation in the strength of the source-only models corresponding to each method, so our choice to only visualize one in the main paper for simplicity is reasonable. The exception is SADA, whose source-only model is significantly weakened by not using EMA.

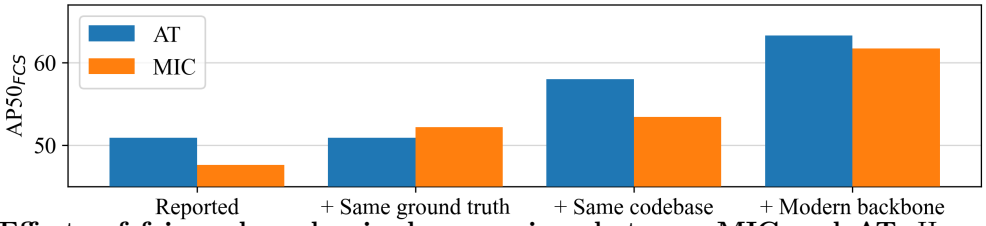


Figure 9: **Effects of fair and modernized comparison between MIC and AT.** Here we show an example of why fair and modern comparisons are necessary for making principled progress in DAOD. Moving left to right: **(1)** Published results report a difference of 3.3 AP50 on Cityscapes → Foggy Cityscapes between the two methods; **(2)** However the authors used *different truth test labels*, and when this is corrected we see that the originally-published MIC model actually outperforms the originally-published AT model; **(3)** The authors also used different object detection libraries (Detectron2 for AT and maskrcnn-benchmark for MIC); when we re-implement them on top of ALDI (still using the VGG-16 backbones proposed in the original papers), we see that AT significantly outperforms MIC, but **(4)** These performance differences are less pronounced when using a modern backbone, indicating that for practical use there is less difference between these two methods than previously reported.

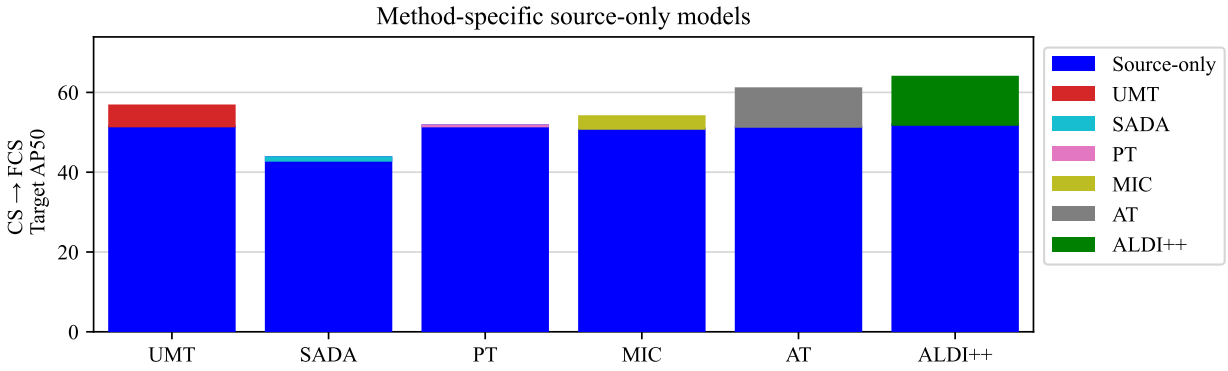


Figure 10: **Comparing methods to method-specific source-only models** that use the exact same training settings as the DAOD method in question. These results use ImageNet pre-training. Most of the bespoke source-only models perform very similarly, so in the main paper we only visualize one representative source-only model. The exception is SADA, whose corresponding source-only model performs worse due to the lack of EMA during training. See Appendix A.7.

B Implementation Details

B.1 ALDI-YOLO

We use an Detectron2 implementation of YOLOv5m as our starting point. All hyperparameter settings are identical to those of ALDI++.

We implement soft distillation for YOLOv5 as follows. We compute an "objectness" (foreground/background) loss for each proposal, and compute classification and localization losses for pseudo-foreground labels only. Given pre-softmax student logits l and teacher logits $\hat{l}$:

$$L_{obj,soft} = BCE(l_{obj}, \hat{l}_{obj}) \tag{14}$$

$$L_{cls,soft} = CE(l_{cls}, \alpha(\hat{l}_{cls}) \tag{15}$$

$$L_{loc,soft} = CIOU(l_{loc}, \alpha(\hat{l}_{loc})) \tag{16}$$

$$L_{distill,soft} = L_{obj,soft} + L_{cls,soft} + L_{loc,soft} \tag{17}$$

Where BCE is the binary cross-entropy loss, CE is the cross-entropy loss, CIOU is the Complete IoU loss **?**, and α is still a function of the post-softmax scores. See **??** for a visual depiction.

B.2 ALDI-DETR

We use an Detectron2 implementation of Deformable DETR as our starting point. There is not an established technique for using soft knowledge distillation in end-to-end transformer-based queries like those of DETR; thus, we use hard distillation with a pseudo label threshold of 0.8. Similar to prior work Wang et al. (2021); Yu et al. (2022); Jia et al. (2023), we disable the EMA update for object query parameters.

Table 10: **Settings to reproduce five prior works and our method ALDI++. Burn-in:** fixed duration (Fixed), our approach (Ours, Section 4). **Augs.** $\boldsymbol{T_{src}, T_{tgt}}$**:** Random flip (F), multi-scale (M), crop & pad (CP), color jitter (J), gaussian blur (B), cutout (DeVries & Taylor, 2017) (C), MIC (Hoyer et al., 2023). $\frac{1}{2}$: augs used on half the images in the batch. $\boldsymbol{\frac{B_{tgt}}{B}}$**:** Target-domain portion of minibatch of size B. **Postprocess:** Processing of teacher preds before distillation: sigmoid/softmax (Sharpen), sum class preds for pseudo-objectness (Sum), conf. thresholding (Thresh), NMS. $\boldsymbol{L_{distill}}$**:** Distillation losses: hard pseudo-labels (Hard), continuous targets (Soft). $\boldsymbol{L_{align}}$**:** Feature alignment losses: image-level adversarial (Img), instance-level adversarial (Inst), image-to-image translation (Img2Img). †: settings used in ALDI implementation (last column) but not in the original implementation (second-to-last column). $^{\boldsymbol{at}}$: source-only and oracle results sourced from Li et al. (2022b).

Method	$\theta_{stu}, \theta_{tch}$ Burn-in	T_{src}	T_{tgt}	$\frac{B_{tgt}}{B}$	Post-process	$L_{distill}$	L_{align}
Source-only	–	F, M†, C†, B†, E†	–	0.0	–	–	–
SADA (Chen et al., 2021)	–	F, M†	F, M†	0.5	–	–	Img, Inst
PT (Chen et al., 2022)	Fixed	F, M†	F, M†, J, C, B	0.3	Sharpen, Sum	Soft	–
UMT (Deng et al., 2021)	–	F†, M†	F†, M†, CP, J, B	0.5	Thresh, NMS	Hard	Img2Img
MIC (Hoyer et al., 2023)	–	F, M†	F, M†, J, B, MIC	0.5	Thresh, NMS	Hard	Img, Inst
AT (Li et al., 2022b)	Fixed	F, M†, J$^{\frac{1}{2}}$, C$^{\frac{1}{2}}$, B$^{\frac{1}{2}}$	F, M†, J, C, B	0.3	Thresh, NMS	Hard	Img
ALDI++	Ours	F, M, J, C, B	F, M, J, B, MIC	0.5	Sharpen	Soft	–
Oracle	–	–	F, M†, J†, C†, B†	1.0	–	–	–

B.3 Re-implementations of Other Methods

Here we include additional details regarding our re-implementations of prior work on top of the ALDI framework. All hyperparameters are reported in Table 10. We visualize our implementations in Fig. 11.

B.3.1 Adaptive Teacher (Li et al., 2022b)

Adaptive Teacher (AT) uses the default settings from the base configuration in Table 2 of the main paper, plus an image-level alignment network. For fair reproduction, we used the authors' alignment network implementation instead of our own for all AT experiments.

B.3.2 MIC (Hoyer et al., 2023)

We reimplemented the masked image consistency augmentation as a Detectron2 `Transform` in our framework for efficiency. We also implemented MIC's "quality weight" loss re-weighting procedure, though in our experiments we found that it makes performance slightly worse (AP50 on Foggy Cityscapes of 62.8 vs. 63.1 without).

B.3.3 Probabilistic Teacher (Chen et al., 2022)

Probabilistic Teacher (PT) utilizes: (1) a custom Faster R-CNN architecture that makes localization predictions probabilistic, called "Gaussian R-CNN", (2) a focal loss objective, (3) learnable anchors. We ported implementations of these three components to our framework. Note that we first had to burn in a Gaussian R-CNN, so PT was not able to use the exact same starting weights as other methods.

B.3.4 SADA (Chen et al., 2021)

We port the official implementation of SADA to Detectron2. Note that SADA does not include burn-in or self-training, so the base implementation is the Detectron2 baseline config.

B.3.5 Unbiased Mean Teacher (Deng et al., 2021)

Our implementation mirrors the UMT_{SCA} configuration from Deng et al. (2021).

B.4 Faster R-CNN Losses

Here we describe the standard Faster R-CNN losses before describing how we modify them into "soft" distillation losses. Faster R-CNN consists of two stages: a region proposal network and the region-of-interest heads.

B.4.1 Region proposal network (RPN):

Inputs. The RPN takes as input:

1. Features extracted by a backbone network (*e.g.* a Resnet-50 with feature pyramid network in most of our experiments).
2. A set of anchor boxes that represent the initial candidates for detection.

Outputs. For each anchor, the RPN predicts two things:

1. A binary classification called "objectness" representing whether the content inside the anchor box is foreground or background.
2. Regression targets for the anchor, representing adjustments to the box to more closely enclose any foreground objects.

Computing the loss. In order to evaluate these predicted proposals, each proposal is matched to either foreground or background based on its intersection-over-union with the nearest ground truth box. Based

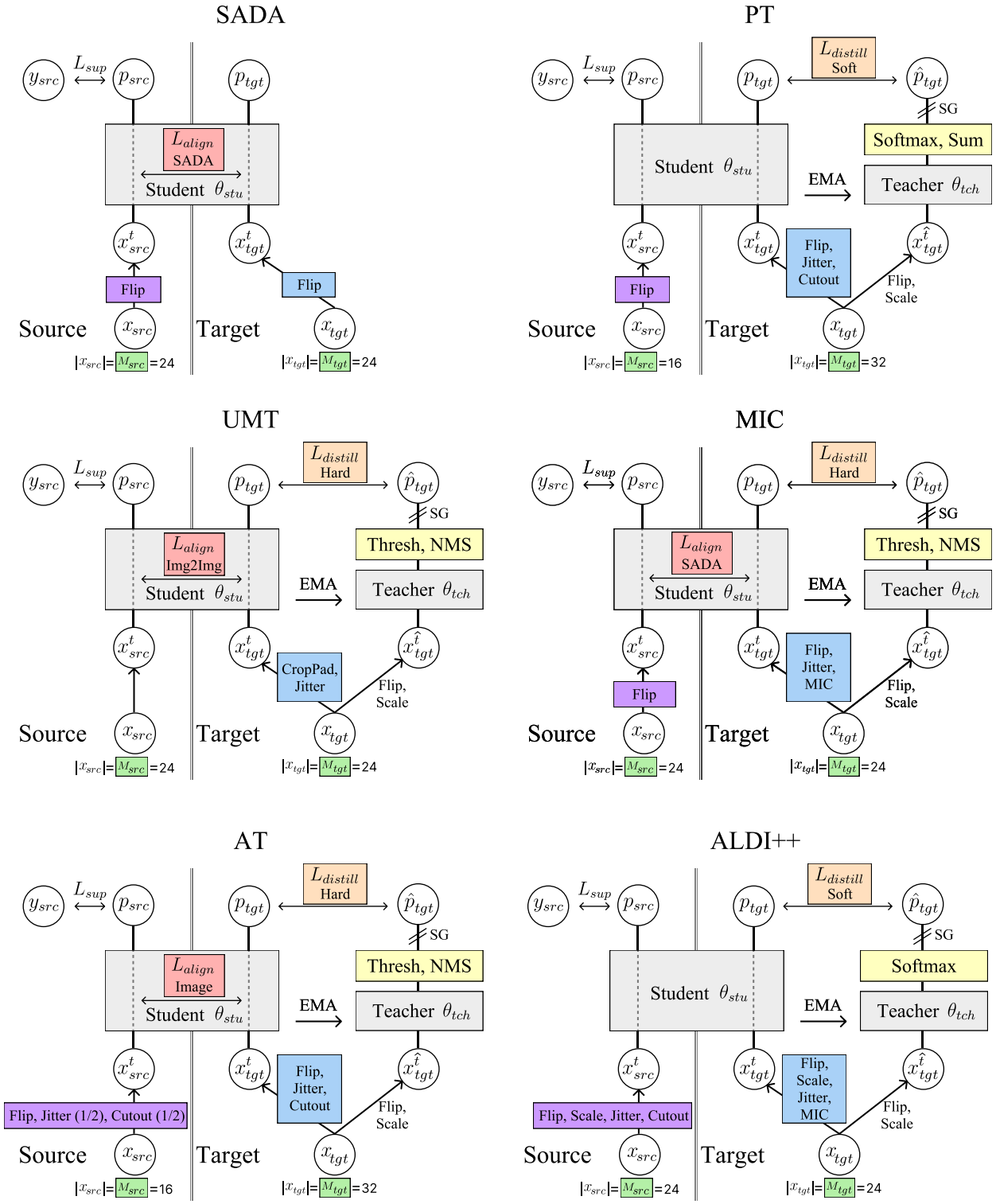


Figure 11: **Visual depiction of ALDI settings for reproducing prior work.**

on these matches, in the Detectron2 default settings a binary cross-entropy loss is computed for (1) and a smooth L1 loss is computed for (2).

A key challenge in Faster R-CNN is the severe imbalance between foreground and background anchors. To address this, a smaller number of proposals are sampled for computing the loss (256 in the default settings) with a specified foreground ratio (0.5 in the default setting). Objectness loss is computed for all proposals, while the box regression loss is computed only for foreground proposals (since it is undefined how the network should regress background proposals).

B.4.2 Region of interest (ROI) heads:

Inputs. The ROI heads take as input:

1. Proposals from the RPN. In training, these are sampled at a desired foreground/background ratio, similar to the procedure used for computing the loss in the RPN. Note, however, that these will be different proposals than those used to compute RPN loss. In the Detectron2 defaults, 512 RPN proposals are sampled as inputs to the ROI heads at a foreground ratio of 0.25.
2. Cropped backbone features, extracted using a procedure such as ROIAlign (He et al., 2017). These are the features in the backbone feature map that are "inside" each proposal.

Outputs. The ROI heads then predict for each proposal:

1. A multi-class classification.
2. Regression targets for the final bounding box, representing adjustments to the box to more closely enclose any foreground objects.

Computing the loss. Predicted boxes are matched with ground truth boxes again based on intersection-over-union in order to compute the loss. By default we compute a cross-entropy loss for (1) and a smooth L1 loss for (2). (2) is again only computed for foreground predictions.

B.5 Soft Distillation Losses for Faster R-CNN

Distillation losses are computed between teacher predictions and student predictions. One option is select the teacher's most confident predictions based on a confidence threshold parameter to be "pseudo-labels." These take the place of ground truth boxes in the standard Faster R-CNN losses for the student. We refer to this approach as using "hard targets."

In contrast, here we describe how we compute "soft" losses using intermediate outputs from the teacher to guide the student without thresholding.

RPN. The teacher and student RPNs start with the same anchors. We use the same sampling procedure described in B.4.1 for choosing proposals for loss computation. Importantly, we ensure the *same* proposals are sampled from the teacher and student so that they can be directly compared. We postprocess the teacher's objectness predictions with a sigmoid function to sharpen them. We then compute a binary cross-entropy loss between the teacher's post-sigmoid outputs and student's objectness predictions. We also compute a smooth L1 loss between the teacher's RPN regression predictions and the student's RPN regression predictions. Regression losses are only computed on proposals where the teacher's post-sigmoid objectness score is ≥ 0.8.

ROI heads. The second stage of Faster R-CNN predicts a classification and regression for each RPN proposal; therefore, we need the input proposals to the student and teacher to be the same in order to directly compare their outputs. To achieve this, during soft distillation we initialize the student and teacher's ROI heads with the student's RPN proposals—intuitively, we want the teacher to tell the student "what to do with" its proposals from the first stage.

We postprocess the teacher's classification predictions with a softmax to sharpen them, then compute a cross-entropy loss between the teacher's post-softmax predictions and the student's classification predictions. We also compute a smooth L1 loss between the teacher's regression predictions and the student's regression predictions. We only compute regression losses where the teacher's top-scoring class prediction is not the background class.

B.6 Adversarial Feature Alignment

We implement two networks to perform adversarial alignment at the image level and instance (bounding box) level. Our approach is inspired by Faster R-CNN in the Wild (Chen et al., 2018) and SADA (Chen et al., 2021).

Image-level alignment. We build an adversarial discriminator network that takes in backbone features at the image level. By default we use the "p2" layer of the feature pyramid network as described in Lin et al. (2017). We use a simple convolutional head consisting of one hidden layer. Our defaults result in this `torch` module:

```
ConvDiscriminator(
    (model): Sequential(
      (0): Conv2d(256, 256,
                  kernel_size=(3, 3),
                  stride=(1, 1))
      (1): ReLU()
      (2): AdaptiveAvgPool2d(output_size=1)
      (3): Flatten(start_dim=1, end_dim=-1)
      (4): Linear(in_features=256,
                  out_features=1,
                  bias=True)
    )
  )
```

Instance-level alignment. We also implement an instance-level adversarial alignment network that takes as input the penultimate layer of the ROI heads classification head. By default, our instance level discriminator consists of one hidden fully-connected layer. Our defaults result in this `torch` module:

```
FCDiscriminator(
    (model): Sequential(
      (0): Flatten(start_dim=1, end_dim=-1)
      (1): Linear(in_features=1024,
                  out_features=1024,
                  bias=True)
      (2): ReLU()
      (3): Linear(in_features=1024,
                  out_features=1,
                  bias=True)
    )
)
```

C Experiment Details

C.1 Backbone Pretraining

In our experiments, we evaluate two different backbones: a ResNet-50 (He et al., 2016) with Feature Pyramid Network (Lin et al., 2017), and a ViT-B (Dosovitskiy et al., 2020) with ViTDet (Li et al., 2022a). Both backbones are pre-trained on the ImageNet-1K classification and the COCO instance segmentation (Lin et al., 2014) tasks. In addition, the ViT-B backbone is also pre-trained using the masked autoencoder objective proposed in He et al. (2022).

Figure 12: **Exemplary results of our CycleGAN models.** Source and target are the original images. Target-like and source-like are images translated by CycleGAN. Since FCS is derived from CS, CS → FCS is the only case in which we have paired images and can therefore show the translation from source into target-like and from target into source-like for the same example.

C.2 Image-to-Image Translation

In contrast to the adversarial alignment in feature space as in SADA (Chen et al., 2021), UMT (Deng et al., 2021) aligns the domains in image (*i.e.* pixel) space. This is achieved by training and using an unpaired image-to-image translation model to try to transform images from the source dataset into images that look like images from the target dataset ("target-like") and vice-versa ("source-like"). We follow Deng et al. (2021) by using the CycleGAN (Zhu et al., 2017) image-to-image translation model. We train the CycleGAN for 200 epochs (Cityscapes ↔ Foggy Cityscapes, Sim10k ↔ CS) or 20 epochs (Kenai ↔ Channel) and respectively select the best model according to the average Fréchet inception distance (FID) (Heusel et al., 2017) between the source & source-like and the target & target-like images in the training dataset. For FID computation, we use the clean-fid implementation proposed in Parmar et al. (2022). We compute FID on the training datasets as UMT only uses translated images thereof, which is why we are only interested in the best fit on the training data. We follow Deng et al. (2021) by then generating source-like and target-like dataset using the selected model ahead of time, before the training of the main domain adaptation method. We note that tuning CycleGAN's hyperparameters or using other image-to-image translation methods could possibly improve UMT's performance however for the fair reproduction we use the defaults. We show some exemplary results of our CycleGAN models that are used to train UMT (Deng et al., 2021) in Fig 12.

C.3 Other Training Settings

We fix the total effective batch size at 48 samples for fair comparison across all experiments. For training, we perform each experiments on 8 Nvidia V100 (32GB) GPUs distributed over four nodes. We use the MIT Supercloud (Reuther et al., 2018).

D CFC-DAOD Dataset Details

Like other DAOD benchmarks, CFC-DAOD consists of data from two domains, source and target.

D.1 Source data

Train: In CFC-DAOD, the source-domain training set consists of training data from the original CFC data release, i.e., video frames from the "Kenai left bank" location. We have used the 3-channel "Baseline++" format introduced in the original CFC paper (Kay et al., 2022). For experiments in the ALDI paper, we subsampled empty frames to be around 10% of the total data, resulting in 76,619 training images. For reproducibility, we release the exact subsampled set. When publishing results on CFC-DAOD, however, researchers are allowed to use the orignial CFC training set however they see fit and are not required to use our subsampled "Baseline++" data.

Validation: The CFC-DAOD Kenai (source) validation set is the same as the original CFC validation set. We use the 3-channel "Baseline++" format from the original CFC paper. There are 30,454 validation images.

D.2 Target data

Train: In CFC-DAOD, the target-domain "training" set consists of new data from the "Kenai Channel" location in CFC. These frames should be treated as unlabeled for DAOD methods, but labeled for Oracle methods. We also use the "Baseline++" format, and use the authors' original code for generating the image files from the original video files for consistency. There are 29,089 target train images.

Test: The CFC-DAOD target-domain test set is the same as the "Kenai Channel" test set from CFC. We use the "Baseline++" format. There are 13,091 target test images. Researchers should publish final mAP@Iou=0.5 numbers on this data, and may use this data for model selection for fair comparison with prior methods.

E The ALDI Codebase

We release ALDI as an open-source codebase built on a modern detector implementation. The codebase is optimized for speed, accuracy, and extensibility, training up to 5x faster than existing DAOD codebases while requiring up to 13x fewer lines of code. These qualities make our framework valuable for practitioners developing detection models in real applications, as well as for researchers pushing the state-of-the-art in DAOD.

E.1 Detection Framework

We designed the ALDI codebase to be lightweight and extensible. For this reason, we build on top of a recent version of Detectron2 (Wu et al., 2019). The last tagged release of Detectron2 was `v0.6` in November 2021, however there have been a number upgrades since then leading to state-of-the-art performance. Thus, we use a fixed version that we call `v0.7ish` based off of an unofficial pull request for `v0.7`, commit `7755101` dated August 30 2023. We include this version of Detectron2 as a `pip`-installable submodule in the ALDI codebase for now, noting that once the official version is released it will no longer need to be a submodule (*i.e.* it will be able to be directly installed through `pip` without cloning any code).

Table 11: **Open-source codebases in domain adaptive object detection.** Existing methods use a variety of different detector implementations, including deprecated frameworks (maskrcnn-benchmark) and versions (Detectron2 < v0.6). In contrast, ALDI is built on top of a modern framework, optimized for training speed, and is able to reproduce all five of these methods while requiring fewer lines of code (LOC) than any individual existing implementation. Our codebase can serve as a strong starting point for future research.

Codebase	Faster R-CNN Implementation	Backbone	Input size	LOC
UMT (Deng et al., 2021)	faster-rcnn.pytorch	VGG-16	600px	19k
SADA (Chen et al., 2021)	maskrcnn-benchmark	Resnet50-FPN	800px	7k
PT (Chen et al., 2022)	Detectron2 v0.5	VGG-16	600px	3.4k
MIC (Hoyer et al., 2023)	maskrcnn-benchmark	Resnet50-FPN	800px	20k
AT (Li et al., 2022b)	Detectron2 v0.3	VGG-16	600px	4k
ALDI (Ours)	Detectron2 ∼v0.7	Resnet50-FPN	1024px	1.5k

Our codebase makes no modifications to the underlying Detectron2 code, making it a lightweight standalone framework. This is in contrast to existing DAOD codebases (see Table 11) that often duplicate and modify the underlying framework as part of their implementation. By building *on top of* Detectron2 rather than *within* it, our codebase is up to 13x smaller than other DAOD codebases while providing more functionality. We note that in Table 11, other codebases implement a single method while ours supports all methods studied.

E.2 Speedups

We found significant bottlenecks in training in other Detectron2-based codebases. Notably, we found that dataloaders and transform implementations were inefficient. These included, for instance:

- Converting tensors back and forth between torch, numpy, and PIL during augmentation. We addressed this, reimplementing transforms as needed so that everything stays in torch.
- Using the random hue transform from torchvision. We found minimal changes in performance from disabling this component of the ColorJitter transform.
- Using separate dataloaders for weakly and strongly augmented imagery. We instead use a single dataloader per domain, with a hook to retrieve weakly augmented imagery before strong augmentations are performed.

We reimplemented the dataloaders and augmentation strategies used by AT, MIC, and others to be more efficient, leading to a 5x speedup in training time per image compared to AT.

F Tabular Results

G Qualitative Results

We visualize predictions from all models on Cityscapes → Foggy Cityscapes in Fig. 13, Sim10k → Cityscapes in Fig. 14, and CFC-DAOD in Fig. 15.

Cityscapes → Foggy Cityscapes We choose 1 random frame from each city in the validation set (Munster, Lindau, and Frankfurt) and show the 0.005 and 0.02 fog levels. We see that, compared to ALDI++, UMT and PT consistently suffer from more false positive detections, while other methods display both false positive and false negative detections. AT and MIC perform qualitatively similary to ALDI++ in many cases. AT suffers from more false positives than ALDI++ in all locations but fewer false negatives in Lindau. MIC

Table 12: **Results with ResNet50-FPN backbones and ImageNet pre-training.** Previously-published results are shown in gray. Best results for each benchmark are in **bold** and second-best are underlined.

Method	CS → Foggy CS	Sim10k → CS	CFC Kenai → Channel
Source-only	(23.5) 51.9	(35.5) 61.4	67.3
UMT (Deng et al., 2021)	(41.7) 56.7	(43.1) 41.8	64.9
SADA (Chen et al., 2021)	(44.0) 42.9	55.7	61.6
PT (Chen et al., 2022)	(47.1) 51.5	(55.1) 64.3	67.0
MIC (Hoyer et al., 2023)	(47.6) 54.0	64.3	71.9
AT (Li et al., 2022b)	(50.9) 61.0	56.7	69.2
ALDI++ (Ours)	**63.9**	**69.1**	**72.6**
Oracle	(42.7) 62.4	(66.4) 83.3	77.1

Table 13: **Results with ResNet50-FPN backbones and COCO pre-training.** Best results for each benchmark are in **bold** and second-best are underlined.

Method	CS → Foggy CS	Sim10k → CS	CFC Kenai → Channel
Source-only	59.1	76.8	66.7
UMT (Deng et al., 2021)	61.4	58.7	61.2
SADA (Chen et al., 2021)	54.2	71.8	58.9
PT (Chen et al., 2022)	59.2	70.6	69.0
MIC (Hoyer et al., 2023)	61.7	73.1	75.5
AT (Li et al., 2022b)	63.3	72.0	69.1
ALDI++ (Ours)	**66.8**	**77.8**	**76.1**
Oracle	67.2	86.1	73.8

Table 14: **Results with VitDet-B backbones and ImageNet pre-training.** Best results are in **bold** and second-best are underlined.

Method	CS → Foggy CS
Source-only	68.1
UMT (Deng et al., 2021)	63.9
SADA (Chen et al., 2021)	54.4
PT (Chen et al., 2022)	65.8
MIC (Hoyer et al., 2023)	70.6
AT (Li et al., 2022b)	66.6
ALDI++ (Ours)	**71.0**
Oracle	74.5

performs better than ALDI++ in this randomly-selected Lindau frame. In the Frankfurt example, we see one downside of the Foggy Cityscapes benchmark: since detection annotations are generated programatically from segmentations, strange false positives can exist in the ground truth. Our new dataset CFC-DAOD addresses this problem by focusing directly on object detection.

Sim10k → Cityscapes In Fig. 14 we compare results of all methods on the Sim10k → Cityscapes benchmark on two random images from each location in the Cityscapes validation set. In Munster, we see that all methods, including ALDI++, struggle with differentiating overlapping cars in the first image. In the second

image, ALDI and MIC outperform other methods in differentiating cars in groups with less overlap than the first example. In Lindau, most prior work exhibits significantly more false negatives than ALDI++ and AT. AT, however, merges detections of multiple cars, leading to false negatives as well. In Frankfurt, most methods again exhibit significant false negatives, while ALDI++ and MIC show small false positive detections. Across all locations, MIC and SADA predict large false positives caused by foreground street lines.

CFC-DAOD In Fig. 15 we compare results of all methods on the CFC-DAOD Kenai → Channel benchmark. We select two random images from each camera view in the test set. These camera views represent different range windows of the sonar camera, with Stratum1 the nearest-range (leading to the highest resolution and largest fish), followed by Stratum2, and finally Stratum3 is the longest-range (lowest resolution and smallest fish). We can see that each Stratum exhibits its own challenges in differentiating fish from the background, dealing with low signal-to-noise ratios, and differences in target size. In Stratum1, all methods except SADA detect the easily identifiable large fish in the first frame, but all methods suffer from the same false negative in the second frame. ALDI++ and UMT predict the same false positive near the edge of the field of view. In Stratum2, all methods suffer from false positives caused by background texture, and UMT, SADA, and PT suffer false negatives. In Stratum3, all methods fail to detect the furthest-range fish which are highly occluded by background texture, but exhibit fewer false positives than the other strata.

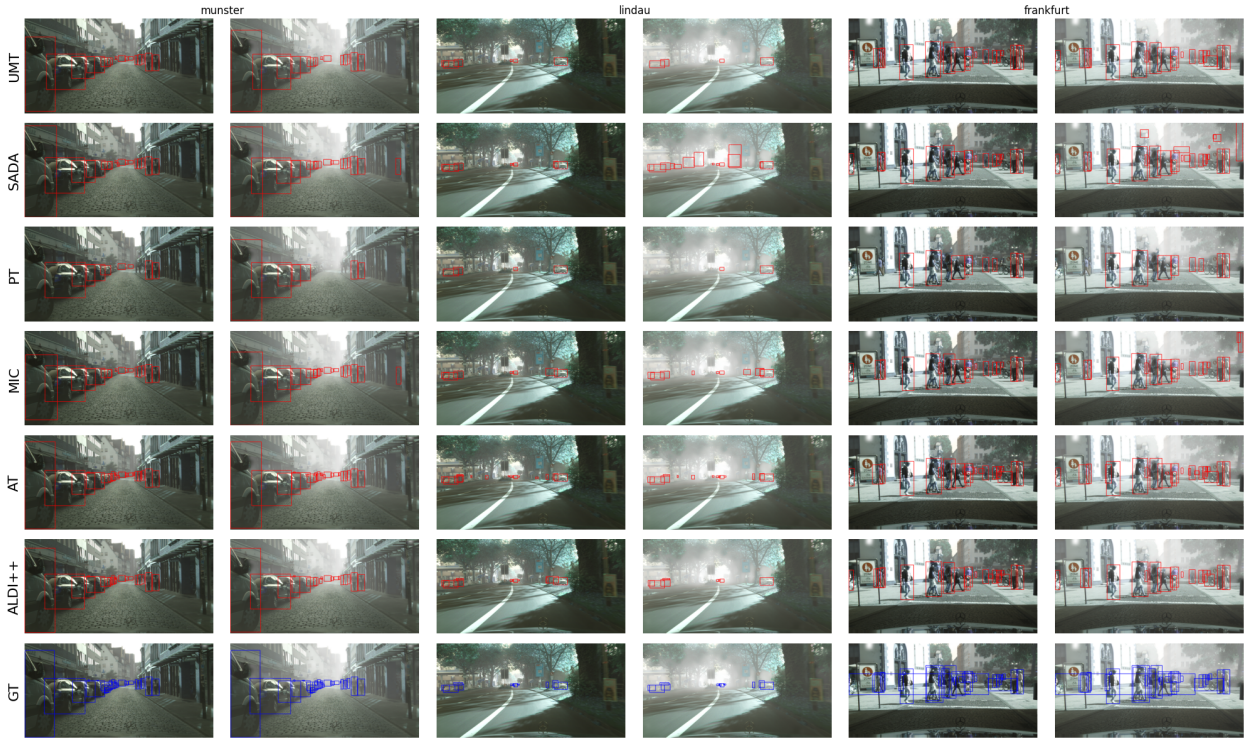


Figure 13: Qualitative results on Foggy Cityscapes. Best viewed maginified.

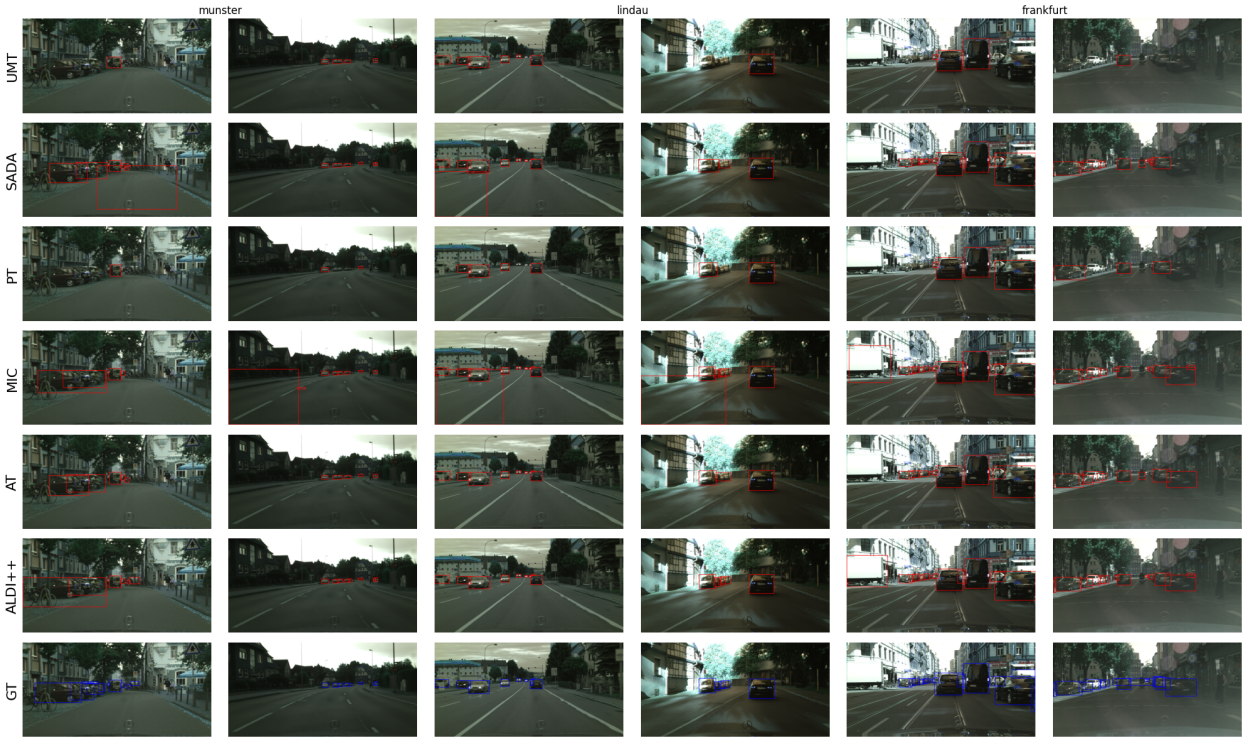


Figure 14: Qualitative results on Sim10k → Cityscapes. Best viewed maginified.

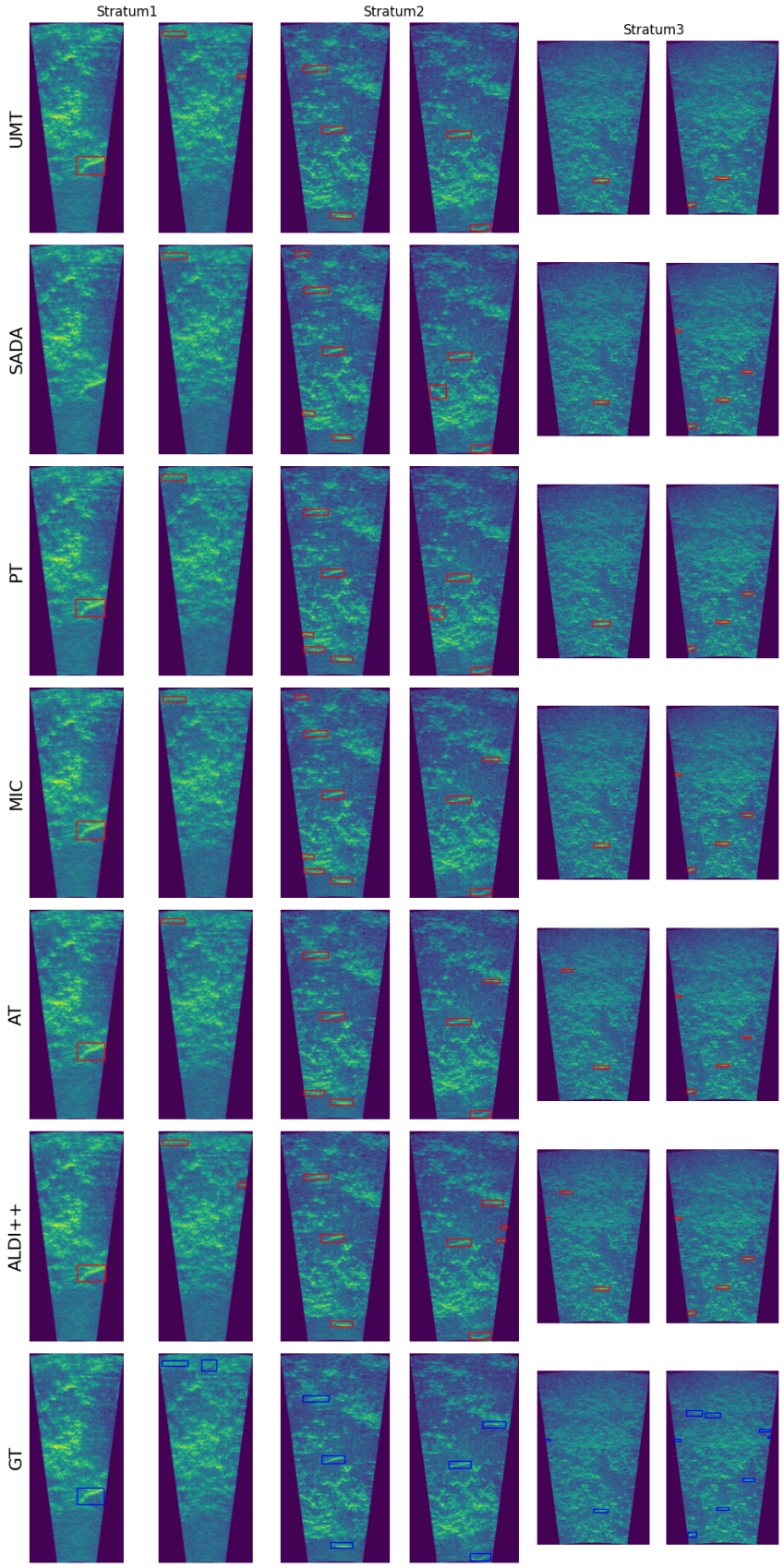


Figure 15: Qualitative results on the CFC-DAOD test set. Best viewed maginified.