Unsupervised Domain Adaptation in Panoptic Segmentation

Unsupervised Adaptive Panoptic Segmentation with Language Guidance and Instance-Aware Domain Mixing

Master Thesis

Author(s):
Mansour, Elham Amin

Publication date:
2024

Permanent link:
https://doi.org/10.3929/ethz-b-000668905

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

CVL Computer Vision Lab

Unsupervised Domain Adaptation in Panoptic Segmentation

Unsupervised Adaptive Panoptic Segmentation with Language Guidance and Instance-Aware Domain Mixing

Master's Thesis

Elham Amin Mansour
Department of Computer Science

Advisors: Ozan Unal and Suman Saha
Supervisor: Prof. Dr. Luc Van Gool

March 15, 2024

Abstract

The increasing relevance of panoptic segmentation is tied to the advancements in autonomous driving and AR/VR applications. However, the deployment of such models has been limited due to the expensive nature of dense data annotation, giving rise to unsupervised domain adaptation (UDA). A key challenge in panoptic UDA is reducing the domain gap between a labeled source and an unlabeled target domain while harmonizing the subtasks of semantic and instance segmentation to limit catastrophic interference. While considerable progress has been achieved, existing approaches mainly focus on the adaptation of semantic segmentation. In this work, we focus on incorporating instance-level adaptation via a novel instance-aware cross-domain mixing strategy IMix. IMix significantly enhances the panoptic quality by improving instance segmentation performance. Specifically, we propose inserting high-confidence predicted instances from the target domain onto source images, retaining the exhaustiveness of the resulting pseudo-labels while reducing the injected confirmation bias. Nevertheless, such an enhancement comes at the cost of degraded semantic performance, attributed to catastrophic forgetting. To mitigate this issue, we regularize our semantic branch by employing CLIP-based domain alignment (CDA), exploiting the domain-robustness of natural language prompts. Finally, we present an end-to-end model incorporating these two mechanisms called LIDAPS, achieving state-of-the-art results on all popular panoptic UDA benchmarks.

Acknowledgements

I thank firstly my family for supporting me unconditionally. I also thank my supervisors whose guidance helped me develop novel methods that I can be proud of. Last but not least, I thank my friends who encouraged me to be the best version of myself.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Focus of this Work

Panoptic segmentation [38] unifies semantic and instance segmentation by not only assigning a class label to each pixel but also segmenting each object into its own instance. The common approach when tackling panoptic segmentation is to deconstruct it into two subtasks and later fuse the resulting dense predictions [33, 63]. The challenge in such an approach lies in the contradictory nature of the individual task objectives [33]. While semantic segmentation seeks to map the embeddings of semantically similar object instances into a class-specific representation, instance segmentation aims to learn discriminative features to separate instances from one another, resulting in conflicting gradients from two different objectives.

Despite the apparent challenges, the rich semantic information with instance-level discrimination is crucial for downstream applications such as autonomous driving or AR/VR. Yet, the complexity and cost of acquiring such panoptic annotations heavily hinder the real-world deployability of such models. Furthermore, given the variance in data distribution between different domains caused by geographical changes, object selection, weather conditions, or sensor setups, models trained on previously acquired annotated data often perform poorly in new domains. This phenomenon, known as the "domain gap", remains a further limiting factor. To this end, recent works have focused on incorporating data-efficiency into panoptic segmentation through the task of unsupervised domain adaptation (UDA) [33, 93, 63]. In contrast to the aforementioned supervised setting, in panoptic UDA a model is trained on labeled source domain images and unlabeled target domain images with supervision only available on the source domain. This allows (i) available labeled data to be used to tackle further domains (real-to-real adaptation) or (ii) to reduce annotation requirements altogether (synthetic-to-real adaptation).

However, under a panoptic UDA setting, balancing both tasks and limiting the effects that arise from the contradictory objectives becomes more challenging due to the lack of a supervisory signal on the target domain. In Tab. 1.1, we provide an overview of SOTA panoptic UDA methods [33, 93, 63] based on different criteria. Apart from CVRN [33] that avoids the problem by completely decoupling the two tasks and training individual networks, i.e., fully rely on task-specific representations (TR), previously proposed methods that utilize more memory-efficient unified network architecture (e.g. exploiting both shared (SR) and task-specific representations have tackled panoptic UDA by only adapting the semantic segmentation branch to improve panoptic quality. Specifically, EDAPS [63] utilizes ClassMix [57] to generate semantically cross-domain mixed inputs that align the target domain to the source, and, UniDAformer [93] hierarchically calibrates the semantic masks across generated regions, superpixels, and pixels. Such SOTA methods for panoptic UDA are thus able to learn good semantic segmentation masks in the target domain, however, are prone to predict inaccurate instance segmentation masks due to the conflicting objectives. This problem is more prominent when multiple overlapping or occluded object instances are present in a scene. An example

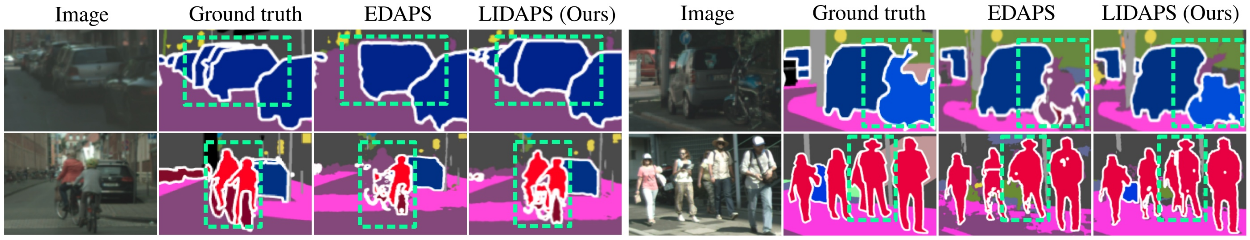


Figure 1.1: While previous SOTA methods for panoptic UDA such as EDAPS [63] achieve good semantic segmentation performance, they struggle to predict correct object boundaries and thus instance segmentation masks.

Table 1.1: Comparison of LIDAPS with SOTA on different aspects such as self-training (ST) type; ST feature space: semantic (Sem) vs. instance (Inst); shared (SR) vs. task-specific (TR) representations; sampling strategies: ClassMix [57] vs. proposed IMix (Sec. 3.3); and proposed CLIP-based domain alignment (CDA).

Method	ST	SemST	InstST	SR	TR	ClassMix	IMix	CDA
CVRN [33]	Offline	✓	✓		✓			
UniDAformer [93]	Online	✓	✓	✓				
EDAPS [63]	Online	✓		✓	✓	✓		
LIDAPS (Ours)	Online	✓	✓	✓	✓	✓	✓	✓

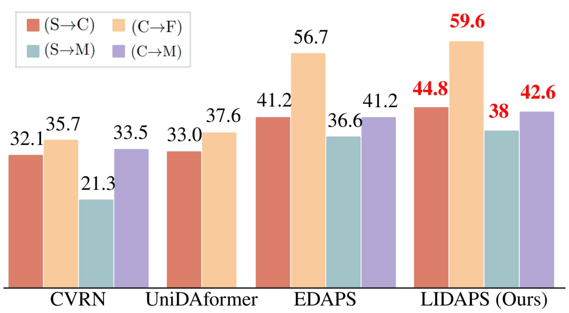


Figure 1.2: The two main contributions, IMix and CDA help improving the UDA panoptic (mPQ) over the SOTA on four UDA benchmarks S→C, C→F, S→M and C→M (Sec. 4).

is shown in Fig. 1.1 where it can be seen that while EDAPS correctly predicts the semantic segmentation masks for the "car" (top-left) and "person" (bottom-right) classes, it fails to identify individual instance boundaries resulting in the merger of objects. This limitation is expected, given the lack of adaptation for instance segmentation. In fact, in the current literature, the adaptation on an instance-level for panoptic UDA remains heavily underexplored [33, 93], with no work in instance-level cross-domain mixing.

In this work, we propose a novel instance-aware mixing strategy IMix (Sec. 3.3), to improve the recognition quality of a panoptic UDA model directly. With IMix, we leverage the panoptic predictions of a model to generate a cross-domain input image consisting of high-confidence instances from the target domain pasted onto a source image and finetune itself through self-supervision (Sec. 3.2). By employing target-to-source mixing, we retain the exhaustiveness of the generated panoptic pseudo-label, i.e. each object within the scene always has an associated instance label. This allows us to reduce the *confirmation bias* while directly learning target instance segmentation on a simpler source background.

While IMix enhances panoptic quality via improved instance segmentation performance, the enhancement is limited due to a drop in semantic segmentation performance (Table 4.2). The model finetuned with IMix becomes subject to catastrophic interference, yielding the ability to map semantically similar objects into a joint embedding in favor of increased instance separability [26]. To remedy this, we propose employing CLIP-based domain alignment (CDA) to act as a regularizer on the semantic branch (Sec. 3.4). In essence, CDA continually aligns both the target and source domains with a pre-trained frozen CLIP [59] model. Specifically, we leverage the rich feature space of CLIP to construct class-wise mean embeddings from a set of static text prompts. We then compute their inner product with the semantic decoder features to generate per-pixel-text similarity maps following DenseCLIP [60] that can be directly supervised via ground truth or pseudo-target labels.

Finally, we combine our two proposed modules with a unified transformer backbone and individual task decoders to construct LIDAPS, a language-guided instance-aware domain-adapted panoptic segmentation model. Our proposed LIDAPS, while improving instance segmentation, is also able to enhance the semantic quality through CDA. For example, LIDAPS predicts correct semantic and instance segmentation masks for the motor-bike (top-right) and the rider (botom-left), while EDAPS fails to do so (Fig. 1.1).
In summary, our contributions are as follows:

1. We introduce IMix, a novel target-to-source instance-aware cross-domain mixing strategy that generates exhaustively labeled source images with target instances for improved recognition quality (i.e. reduced false positives and negatives).

2. We reduce the catastrophic forgetting that arises when training with IMix by introducing CLIP-based domain alignment (CDA) as a regularizer for semantic segmentation.

3. We combine both proposed modules to form LIDAPS, a language-guided instance-aware domain-adapted panoptic segmentation model that achieves state-of-the-art results across multiple panoptic UDA benchmarks (Fig. 1.2).

While we propose an end-to-end model with LIDAPS, our individual contributions remain orthogonal to the development of better panoptic UDA frameworks and are model-agnostic. Furthermore, both contributions can be detached during inference and thus do not induce any memory or computational constraints on the final method.

1.2 Thesis Organization

This thesis contains an Introduction chapter 1 where we introduce the problem that we tackled. Then, there is the Related works chapter 2 where the different groups of work and the employed approaches in the field of segmentation are introduced, including UDA panoptic segmentation. Moreover, we discuss prior literature that use language and augmentation to improve performance. Additionally, we address how different previous works compare to ours. In the Methods chapter 3, after some explanations on preliminary knowledge that needs to be first understood. our two attempts that yielded two mechanisms that complement each other are introduced. Furthermore, in this chapter, we introduce a pipeline that incorporates the two mechanisms, resulting in a new set of state-of-the-arts(SOTA) scores across diverse panoptic UDA benchmarks. In the Experiments chapter 4, we first, provide details on the datasets, implementation details and evaluation metrics that we use. Secondly, we present the results of our work including the new SOTA scores in comparison to prior works, a Table consisting of an ablation study on the proposed mechanisms, an additional Table comparing a semantic mixing strategy as well as the direction of mixing, another Table finding the optimal threshold hyperparameter for IMix(our novel mechanism), and finally, some qualitative results showcasing our impressive instance segmentation improvement in comparison to EDAPS [63], a SOTA work addressing the same problem. In the Discussion chapter 5, we discuss the numbers reported in the previous chapter and analyze their significance. Furthermore, in the Conclusion chapter 6, we summarise the main contributions of this thesis. Lastly, in the Appendix, there are several sections discussing additional results and the different attempts that did not result in performance enhancement, yet were apart of the journey to finding the correct methods. In section A.4.1, we discuss the attempts that were made to enhance the EDAPS model by changing the backbone. In section A.4.4, the other methods we attempted for self-training on target instances are presented. In Sec. A.4.2, the alignment of the instance decoder embedding with CLIP is discussed. Lastly, in section A.4.3, a failed mixing strategy that we tried out is presented.

Chapter 2

Related Work

Numerous studies have investigated semantic, instance, and panoptic segmentation in different settings, encompassing unsupervised domain adaptation (UDA), domain generalization (DG), unsupervised, self-supervised, vanilla, open-vocabulary, and multi-dataset paradigms unified under one taxonomy. In the following section, we provide an overview of works within these domains. Our work specifically focuses on addressing unsupervised domain adaptation (UDA) for panoptic segmentation. Panoptic segmentation [9, 38, 81, 46, 76, 31, 82, 3] is becoming increasingly important with the rise of autonomous driving and AR/VR. Furthermore, we delve into the works using the guidance of language in segmentation and the spectrum of augmentation techniques employed to enhance segmentation for domain adaptation and how these works differ from our proposed mechanisms for enhancing UDA panoptic segmentation.

2.1 Segmentation Fields

2.1.1 Unsupervised domain adaptation(UDA)

Approaches in the domain of Unsupervised Domain Adaptation (UDA) for panoptic segmentation take input images from both source and target domains, along with the source ground truth label. The methodology involves training the model in a supervised manner on the source images. To achieve good performance on the unlabeled target images, these methods incorporate UDA techniques applied to the unlabeled target images such as, adversarial strategies [25, 50, 74, 24, 18, 72, 75], multiple resolution [27, 28], pseudo-label-based self-training on the target images [99, 5, 4, 37, 35, 19], contrastive learning [5, 43, 10], regularizors [7, 34, 70, 69, 95], domain adaptive architecture design [93, 52], large language-vision model knowledge distillation [4], style augmentation[33, 43] and ClassMixing [97, 71, 8, 2, 26, 63, 34].

The approach of pseudo-label-based self-training trains the model on the pseudo-labels of the targets images generated by a teacher network [5, 4, 37, 35, 19, 93, 33, 86] or the model itself [99, 39]. The approach presented in [35] employs two sets of teacher/student networks. One set is utilized for training on the source, while the other adapts to the target. These sets engage in a mutual supervision mechanism, generating pseudo-labels for each other. In contrast, the work introduced in [19] addresses the refinement of pseudo-labels through implicit neural representations, while [87] focuses on denoising pseudo-labels, particularly around boundaries, before engaging in self-training. The methodology in [37] unfolds in two phases: initially, the student is trained on the pseudo-labels of the teacher, and subsequently, in the second phase, the teacher leverages the student's region proposals to formulate pseudo-label predictions. The idea of the contrastive learning approach[5, 43, 10] or regularizors[7] in this setting is to pull together the embeddings of the same classes within the same domain or across domain. Additionally, there are few works such as [4] that align with large language-vision models to try to enhance the performance in a domain adaptive setting. Another UDA technique is augmentation where they try to stylize images in order to robustify to domain invariance [33, 43], or translate the style of the source images to target using Diffusion models, GANS, etc [41, 10]. Some augmentations include semantic mixing strategies [57] where pixels are pasted from source images to target images [71, 8, 2, 26, 97, 34] to synthesize new images to train on. The work [8], does copy-pasting in a scheduled and dynamic way based on the pixel class count and individual class performances. [2] does the mixing based on pixel count hierarchy. These works belong to panoptic[93, 33], instance[10, 7, 4, 37] and semantic[52, 8, 5, 2, 43, 19, 35, 71, 41] segmentation. However, unlike existing work, we explore instance-aware cross-domain mixing to adapt the instance branch while simplifying the learning of difficult target objects by pasting them onto easy-to-segment source backgrounds. Furthermore, unlike previous work, we are the first to exploit the domain-robustness of language-vision models to further align the source and target domains for panoptic UDA.

2.1.2 Weakly supervised domain adaptation

Works in the WDASS setting such as [13], relax UDA by allowing some annotated pixels in the target domain. [13] does semantic segmentation by using contrastive learning to align embeddings of the same class within and across domains and additionally uses self-training on pseudo labels.

2.1.3 Open vocabulary

Open vocabulary seeks to extend its generalizability beyond the constrained set of classes that are seen during the training phase. In this category, different works focus on panoptic, instance, and semantic segmentation. For panoptic segmentation, works like [82, 6, 90, 79] contribute, while others, such as [20, 78, 96, 36], specialize in instance segmentation. Semantic segmentation is addressed by works like [84, 47, 51, 85]. Noteworthy is [67], which tackles instance and semantic segmentation without merging the results into a panoptic annotation. Most works in this field use distillation or integrate large pre-trained language-vision models [47, 82, 90, 84, 67, 20, 85, 79]. Architectures in [82, 90, 51, 67] leverage substantial language-vision backbones. For example, [82] adopts a frozen text-to-image diffusion UNet as a backbone, while [90] uses a frozen convolutional CLIP backbone with the Mask2Former base architecture as a mask generator. Incorporation methods are diverse, with [6] utilizing both image and text CLIP encoders, building upon the Mask2Former model while [20] distills knowledge from a pre-trained open-vocabulary image classification model (teacher) into a two-stage detector (student) via embedding alignment. Some works like [85, 47] use two-stage methods to predict masks, subsequently using CLIP for mask classification. [47] extends this concept by incorporating mask prompt tuning and CLIP tuning. Additionally, [36] leverages large language models for image segmentation based on text inquiries. It's noteworthy that certain open-vocabulary works, like [78, 96], forego the utilization of language. In the absence of language integration, [78] uses stop-gradients to prevent the model from classifying unannotated objects as background, transformers, and contrastive learning. On the other hand, [96] employs augmentation techniques by copy-pasting pseudo masks onto new backgrounds, synthesized from stable diffusion images and real images. Evaluation in open-vocabulary segmentation often involves zero-shot and few-shot settings, where zero-shot testing entails testing a model on classes not encountered during training, and few-shot testing involves scenarios where only a limited number of examples have been seen during training. Unlike UDA, these works do not use unsupervised training.

2.1.4 Domain Generalization

In Domain Generalization (DG) segmentation, the goal is to generalize a model trained on source images to target images without updating the parameters of the model. One mainstream tactic in these works is to augment the features of source images to resemble those of target images. For instance, the works [73, 15] use CLIP-encoded text prompts to augment the source images. Another tactic is to directly augment the source image [58, 42] using stylization techniques or image-conditioned diffusion models[58, 56]. For example, the work [42] stylizes the source images, and [58] uses a conditioned diffusion model to translate the style of the image.

Some works, such as [73, 17], incorporate large vision-language models into their architecture to align their embeddings with more generic embeddings. [73] utilizes a CLIP vision encoder for image encoding and a CLIP text encoder for the classification of their generated masks. Meanwhile, [17] demonstrates how a frozen diffusion backbone is robust to domain invariance. They additionally learns scene prompts for test-time domain adaptations on the target domain. However, while they claim to outperform Unsupervised Domain Adaptation (UDA) semantic segmentation as well as DG semantic segmentation, they do not explore panoptic segmentation.

The works explained above fall under semantic [42, 15, 58, 17], and instance segmentation [73].

2.1.5 Text supervised

A new category of work has emerged in semantic segmentation where no mask annotations are available and these models solely exploit large vision-language models and image caption pairs[83, 88, 91]. [88] aligns vision and text CLIP embeddings in a sparse way to remove the bias towards contextual pixels. Moreover, the work [91] addresses this problem by generating its own artificial segmentation pair data to train on using word tokens. In particular, [23] does semantic segmentation by aligning the pixel embeddings with CLIP embeddings and also forcing the model to make the same pixel class predictions as CLIP. These works are typically evaluated on zero-shot benchmarks.

2.2 Language and Augmentation for Segmentation

2.2.1 Language in Segmentation

Several segmentation studies incorporate language to enhance their performance. This trend originated with DenseCLIP [60], an extension of CLIP [59] designed for dense downstream applications. While we also leverage dense per-pixel text similarity maps similar to DenseCLIP, as opposed to applying alignment on supervised images, we utilize the maps to align both the source and target domains via ground truth and generated pseudo-labels with domain-invariant CLIP text embeddings. Importantly, unlike DenseCLIP which applies this knowledge distillation to the encoder features, we apply deep in the semantic decoder to prevent losing class-agnostic features in the shared encoder that are key for the task of instance segmentation. Open-vocabulary segmentation works also largely integrate language into their architecture [47, 82, 90, 84, 67, 20, 85, 79]. These works do not perform unsupervised domain alignment. In previous works [83, 88, 91, 32], mask annotations are unavailable, and large vision-language models are solely relied on for knowledge distillation. In contrast, we leverage the direct supervision available from a source domain. Some domain generalization segmentation works [73, 17] also incorporate language to align their source embeddings with large language-vision embeddings to generalize to the target domain. However, these works [17, 73] do not address instance segmentation which is specifically challenging given that CLIP mainly consists of semantic knowledge. Moreover, while some works [40, 92, 68] investigate the incorporation of CLIP in UDA, only a few explore its effects in UDA segmentation. For instance, Chapman *et. al* [4] uses CLIP for UDA instance segmentation on an image level. In contrast, our work utilizes CLIP in a panoptic setting and calculates the text similarity on a pixel level.

2.2.2 Augmented Data for Domain Adaptation

A key strategy in UDA segmentation involves training on augmented images. A common approach is the stylization and augmentation of images [33, 43, 42, 54, 29, 1] or the features of source images [48, 73, 15]. Another approach is to leverage diffusion models and GANs to translate the style of source images or to synthesize training images [41, 10, 66, 58, 56, 11, 45, 77]. An alternative mainstream tactic is cross domain mix sampling (CDMS) [98, 71]. ClassMix [57], a CDMS technique, pastes pixels from half of the source image semantic classes onto the target image [71, 8, 2, 26, 97, 34]. However, instance-aware mixing for the domain invariance enhancement of the instance decoder remains largely unexplored. Lu *et. al* [49] explores instance mixing from source-to-target for UDA in action detection but neglects to refine the pseudo-masks. In contrast, in our work, we employ confidence-based thresholding to refine the pseudo-instance-masks which we find is key to reduce the confirmation bias. Furthermore, we apply the mixing in the opposite direction which yields a considerable performance gain by avoiding further bias injected due to an incomplete set of pseudo-labels arising from false negative predictions.

Chapter 3

Materials and Methods

In this section, we start by introducing the preliminaries for unsupervised domain adaptation (UDA) for panoptic segmentation (Sec. 3.1). Having established the groundwork, we construct a baseline pipeline by utilizing a mean-teacher framework and adapt the semantic branch via cross-domain mixing following the literature (Sec. 3.2). We then identify and tackle the shortcomings of this baseline model by introducing a novel instance-aware cross-domain mixing strategy (IMix) (Sec. 3.3), and reduce the resulting catastrophic interference by regularizing the semantic branch via CLIP-based domain alignment (CDA) (Sec. 3.4). Combining all modules, we build our model LIDAPS which we illustrate in Fig. 3.1.

3.1 Preliminary

3.1.1 Panoptic Segmentation

This kind of segmentation is commonly tackled by breaking it down into its subtasks: semantic and instance segmentation. A panoptic segmentation model is thereby trained on a panoptic segmentation loss $\mathcal{L}_{\text{pan}}$ given by the sum of a semantic and an instance loss:

$$\mathcal{L}_{\text{pan}} = \mathcal{L}_{\text{sem}} + \mathcal{L}_{\text{inst}}. \tag{3.1}$$

In this work, for semantic segmentation, we use pixel-wise categorical cross-entropy loss, while for instance segmentation, we follow a top-down approach, and compute RPN and RoIAlign box regression and classification losses following MaskRCNN [22].

3.1.2 Panoptic UDA

This is the task of transferring knowledge from a learned source domain to a target domain. In this setup, a machine learning model ϕ is trained on both source $\mathcal{D}^s = \{d_i^s\}_{i=1}^{N^s}$ and target domain images $\mathcal{D}^t = \{d_i^t\}_{i=1}^{N^t}$, with direct human annotated supervision only available on the source domain via semantic $y^s_{\text{sem}_i} \in \mathbb{R}^{H \times W \times C}$ and instance labels $y^s_{\text{inst}_i} \in \mathbb{R}^{H \times W \times B}$. Here C denotes the number of semantic classes and B denotes the number of ground truth instances given images of size $H \times W$.

The naïve approach to tackling panoptic UDA is to treat the problem similar to standard supervised training, and thus only train with the supervision from the source labels via the source loss $\mathcal{L}^s$. However, the variance in data distribution between the source and target images, i.e. the "domain gap", severely limits the transferability of learned knowledge across domains. Such a naïve approach consisting of only a source loss thus remains inadequate for achieving a good performance on the target domain.

Self-training is a common technique used to reduce the domain gap between source and target by leveraging a model's own predictions to extend the supervision onto the target domain [100, 95, 53, 71, 94, 30,

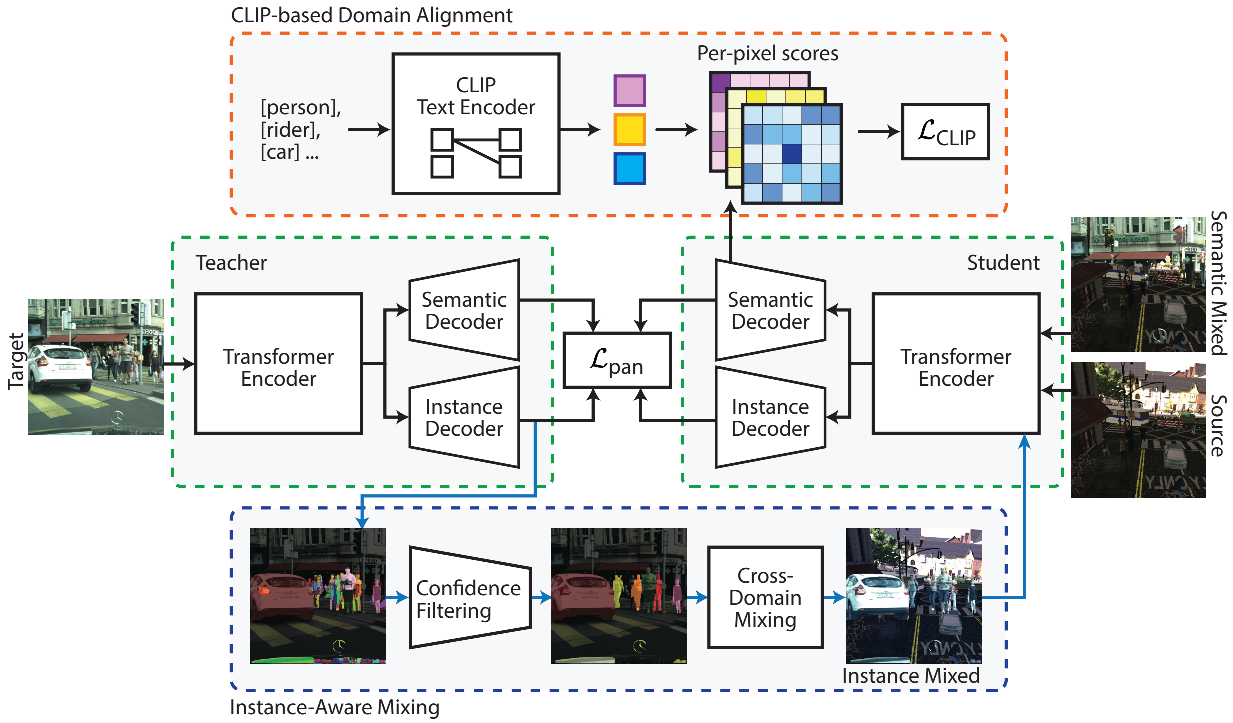


Figure 3.1: Illustration of the LIDAPS pipeline. (Green) The baseline panoptic UDA model is built on a mean-teacher framework and consists of a common transformer encoder and individual task decoders. The student model is supervised directly from source domain labels as well as semantically mixed inputs whose labels are generated by the teacher model. (Blue) We apply IMix to further adapt the instance segmentation branch of LIDAPS, mixing high-confidence predicted target instances with source images. Blue paths are only active during the fine-tuning phase. (Orange) We regularize the semantic branch via CLIP-based domain alignment that utilizes target DenseCLIP maps to reduce catastrophic forgetting.

26, 63]. In this work, we adopt a self-training approach that entails both the supervised loss on the source domain $\mathcal{L}^{s}_{pan}$, alongside a self-supervised loss $\mathcal{L}^{ss}_{pan}$, resulting in the final training objective:

$$\arg\min_{\phi} \mathcal{L}^{s}_{pan} + \mathcal{L}^{ss}_{pan} \tag{3.2}$$

3.2 Establishing a Baseline for Panoptic UDA

In a self-training framework, a model learns from its own predictions. This however can result in *confirmation bias* as the model trains on incorrect pseudo-labels, therefore commonly, predictions are refined prior to application [19, 87]. The mean-teacher framework [70] proposes a simple but effective way to generate stabilized on-the-fly pseudo-labels by leveraging the fact that the stochastic averaging of a model's weights yields a more accurate model than using the final training weights directly. A mean-teacher framework is therefore built with two models, namely the student that is trained (e.g. via gradient decent), and the teacher θ whose weights are updated based on the exponential moving average (EMA) of successive student weights:

$$\phi_{t+1} \leftarrow \alpha\phi_t + (1-\alpha)\theta_t \, [1] \tag{3.3}$$

[1] We use the notation for the model and its weights interchangeably for readability.

for time step t and α that denotes the smoothing coefficient.

While the mean-teacher extends the supervision for the target domain, the supervisory signal remains highly noisy and may still destabilize the training process. A common solution applied in UDA setups is to employ cross-domain mixing to generate images that contain both noisy target domain information alongside clean source domain ground truth annotations [71, 8, 2, 26, 97, 34]. Specifically, the teacher network θ predicts the pseudo-labels for the target image that forms the augmented input for the student via a cut-and-paste operation on the source image. Formally, given a binary mask of semantic labels to be cut $\mathbf{M}_{\text{sem}} \in \{0,1\}^{H\times W}$ and target semantic pseudo-labels generated by the teacher model y^t_{sem}, the semantic cross-domain mixed sampling (DACS) can be defined as:

$$\begin{aligned}\tilde{x} &= \mathbf{M}_{\text{sem}} \odot x^s + (1-\mathbf{M}_{\text{sem}}) \odot x^t \\ \tilde{y}_{\text{sem}} &= \mathbf{M}_{\text{sem}} \odot y^s_{\text{sem}} + (1-\mathbf{M}_{\text{sem}}) \odot y^t_{\text{sem}}\end{aligned} \tag{3.4}$$

with $\odot$ denoting a dot product, $\tilde{\cdot}$ indicating the mixed domain. Such DACS operations leveraging ClassMix [57] coupled with self-training have shown significant performance gains when tackling semantic UDA [26], with the core idea stemming from consistency regularization [57, 64, 70, 69] which states that predictions for unlabelled data should be invariant to perturbations or augmentation.

The semantic loss on the source domain is explained in Eq. 3.5 which defines a categorical cross-entropy loss on the predicted class probability for each pixel.

$$\mathcal{L}^s_{\text{sem}}(\hat{y}^s_{\text{sem}}, y^s_{\text{sem}}) = -\sum_{i,j,c} \left(y^s_{\text{sem}} \log(\hat{y}^s_{\text{sem}})\right)_{i,j,c} \tag{3.5}$$

Following [63], the self-supervised semantic loss applied to the semantic-aware mixed image [57] is shown in Eq. 3.8. The augmented or mixed image generated using the ClassMix [57] contains pixels from both the source and the target domain images. For the source pixels, we compute the categorical cross-entropy loss between the predicted and groundtruth semantic class labels. For the target pixels, we compute a weighted categorical cross-entropy loss as it takes into account the confidence of the pseudo-semantic class labels predicted by the teacher network.

Thus, $k^t_{(i,j)}$ defines the per-pixel confidence score for every pseudo-label predicted by the teacher network [71]. y^t_{sem} is the per-pixel pseudo-label as shown in Eq. 3.6 where θ_{sem}(the semantic decoder of the teacher) predicts per-pixel-class probabilities.

$$y^t_{\text{sem}} = \left[\arg\max_{c'}(\theta_{\text{sem}}(x^{(t)}))_{i,j}\right] \tag{3.6}$$

Formally, the self-supervised loss for the semantically adapted self-training baseline, built on a weighted cross-entropy, is given by:

$$\mathcal{L}^{ss}_{pan} = \mathcal{L}_{sem}(\hat{\tilde{y}}_{\text{sem}}, \tilde{y}_{\text{sem}}) \tag{3.7}$$

with $\hat{\cdot}$ denoting the prediction of the model and

$$\mathcal{L}_{sem}(\hat{\tilde{y}}_{\text{sem}}, \tilde{y}_{\text{sem}}) = \begin{cases} \mathcal{L}^s_{\text{sem}}(\hat{\tilde{y}}_{\text{sem}}, y^s_{\text{sem}}), & \text{if } \mathbf{M}^{(h,w,c)}_{\text{sem}} = 1, \\ -\sum k^t_{(h,w)} \left(y^t_{\text{sem}} \log(\hat{\tilde{y}}_{\text{sem}})\right)_{(h,w,c)}, & \text{otherwise} \end{cases} \tag{3.8}$$

Specifically, we apply a standard supervised loss on pixels coming from the source image ($\mathbf{M}^{(h,w,c)}_{\text{sem}} = 1$), and apply a weighted cross-entropy on the pixels coming from the target image, supervised via the teacher generated pseudo-label (y^t_{sem}). We illustrate this baseline in Fig. 3.1 - green.

Given that semantic segmentation forms one-half of panoptic segmentation, such a baseline approach that adapts the semantic maps between the source and target domains via DACS can contribute significantly

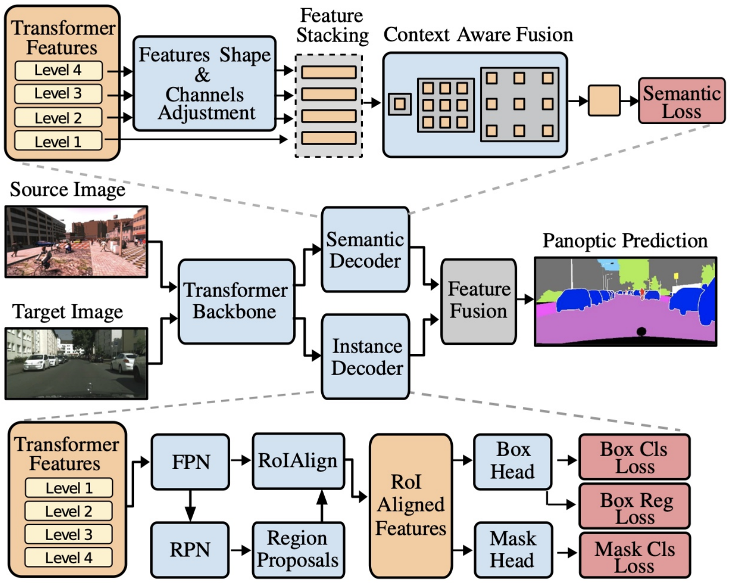


Figure 3.2: EDAPS [63] pipeline from the EDAPS paper.

to reducing the domain gap for panoptic segmentation. However, such an approach forgoes a crucial element of panoptic UDA altogether, adapting the instance segmentation task between two domains. In fact, DACS does not generate augmented images containing sufficient instance-specific information to adapt the instance branch. In the following section, we tackle the adaptation of instance segmentation between a source and target domain to improve panoptic segmentation performance.

3.2.1 EDAPS

$$\phi_{t+1} \leftarrow \alpha\phi_t + (1-\alpha)\theta_t$$

EDAPS[63], a state-of-the-art Unsupervised Domain Adaptation (UDA) panoptic segmentation model, forms the basis of our research. Utilizing a mixed transformer backbone (MiT-B5) with task-specific decoders, it integrates a semantic decoder inspired by DAFormer[26] and an instance decoder from MaskRCNN[22], featuring a proposal-based top-down instance decoding approach as illustrated by them[63] in Fig. 3.2 Additionally, EDAPS integrates rare class sampling, benefiting from more frequent sampling of under-represented classes and ClassMix[71] which does self-training on semantic mixed-domain images using pseudo-labels coming from a semantic decoder teacher network.

3.3 Instance-Aware Mixing (IMix)

We propose a novel mixing strategy called IMix, to reduce the domain gap when tackling instance segmentation. The goal of IMix is to apply cross-domain mixed sampling while not only retaining instance-level information but also simplifying the recognition of target objects by presenting them within source environments. A sample image utilizing IMix is compared to DACS in Fig. 3.7.

However, mixing source and target domain information on an instance level raises a crucial challenge, stemming from how the two tasks are supervised. Unlike semantic segmentation where losses are applied on a pixel level, instance segmentation is typically supervised by the injective function that maps the set of ground truth objects to the set of predicted instances. Therefore an instance segmentation model remains prone to confirmation bias if ground truth label exhaustiveness is not guaranteed, i.e. the model will learn to incorrectly identify objects as background if every visible object within the scene does not have an associated instance mask (see Fig. 3.8(d) for an example of false negative).

A mixing operation must account for such a challenge. We thus construct IMix such that the operation is handled from target-to-source, avoiding the incompleteness of instance labels that may emerge from false negative predictions. Formally we define our instance-aware mixing operation IMix as follows:

$$\begin{aligned}\tilde{x} &= \mathbf{M}_{\text{inst}} \odot x^t + (1 - \mathbf{M}_{\text{inst}}) \odot x^s \\ \tilde{y}_{\text{inst}} &= \mathbf{M}_{\text{inst}} \odot y^t_{\text{inst}} + (1 - \mathbf{M}_{\text{inst}}) \odot y^s_{\text{inst}}\end{aligned} \tag{3.9}$$

with y^t_{inst} and y^s_{inst} denoting the target pseudo-label and source ground truth label respectively, and $\mathbf{M}_{\text{inst}} \in \{0,1\}^{H \times W}$ the sum of binary instance masks based on the teacher's prediction.

Specifically, we cut the instances from the teacher model's output and paste them onto a source image, constructing the mixed pseudo-label by merging the ground truth instance labels with the teacher's predictions. Utilizing the source image as a background ensures that all visible objects have corresponding mask annotations. Furthermore, given that a model learns a source domain much more efficiently thanks to the available direct supervision, with IMix, we simplify the recognition task of target instances by presenting them on easy-to-separate source domain environments.

However, while retaining exhaustiveness limits confirmation bias caused by false negative predictions, a self-supervised model is still prone to such effects due to false positives as well. In other words, if incorrect instance masks are pasted on the mixed image, the model will learn to affirm its preexisting biases, causing an increased number of false positive predictions. To reduce the number of such cases, we propose a simple but effective confidence filtering step. We predict a confidence score alongside the instance masks of each object [22]. We apply filtering based on the predicted confidence values to redefine the joint mixing mask as:

$$\mathbf{M}_{\text{inst}} = \sum_{i \in I} \mathbb{1}[h^t_i > \tau]\, y^t_{\text{inst},i} \tag{3.10}$$

with I denoting the set of predicted instances $y^t_{\text{inst},i}$ the predicted i'th instance mask, h^t_i the corresponding confidence score and τ the threshold hyperparameter. Thus, our self-supervised panoptic loss from Eq. 3.7 can be updated as:

$$\mathcal{L}^{ss}_{pan} = \mathcal{L}_{sem}(\hat{\tilde{y}}_{\text{sem}}, \tilde{y}_{\text{sem}}) + \mathcal{L}_{\text{inst}}(\hat{\tilde{y}}_{\text{inst}}, \tilde{y}_{\text{inst}}) \tag{3.11}$$

In Fig. 3.8, we show an example where in (c) confidence-filtered target instances are pasted onto the source image while in (d) all source instances are pasted on to the target image. In Fig. 3.8(c), we can see that the target instances all have masks while in Fig. 3.8(d), the encircled instance (the truck) in red does not have a corresponding mask which is indicative of a false negative. When going from target to source, only the confidence-filtered instances are copies and thus inherently, all of the pasted instance objects have a corresponding mask. On the other hand, when remaining in the target image, target instances with absent pseudo-masks remain.

As commonly seen in multitask frameworks, the increase in supervisory signals from one task may cause catastrophic forgetting for another, i.e. the weights in the network that are important for one task may be changed to meet the objectives of another [40]. We observe similar behavior in our training when fine-tuning LIDAPS on IMix (please refer to Sec. 4). Specifically, the performance gains of our model for panoptic segmentation are hindered by the drop in semantic quality. In the following section, we address this problem by introducing a language-based regularization for semantic segmentation.

3.3.1 Losses

While the mechanisms we propose (i.e., IMix and CDA) are model agnostic, here we provide detailed mathematical notations of the all losses we used in our end-to-end trainable model, LIDAPS. These formulas have been introduced in prior works, nevertheless, we provide them for the sake of reproducibility. Moreover, we explain how the supervision changes in our novel proposed mechanism, IMix.

As explained in Eq. 3.1, a panoptic loss function consists of two terms; an instance segmentation and a semantic segmentation loss term. Our instance decoder [22] consists of an RPN network and a refinement (Ref) network. Each part has its own losses as shown in Eq. 3.12.

$$\mathcal{L}_{\text{inst}} = \mathcal{L}^{\text{RPN}} + \mathcal{L}^{\text{Ref}} \tag{3.12}$$

The RPN loss function [61] has two terms, one for the "objectness" ($\mathcal{L}^{\text{RPN}}_{\text{Cls}}$) and another one for the bounding-box (or region proposal) regression ($\mathcal{L}^{\text{RPN}}_{\text{Box}}$) loss as seen in Eq. 3.13. The RPN takes a predefined set of anchor boxes and the convolution feature map (encoding the input image) as inputs and it is optimized for correctly localizing objects present in the image. For each predicted bounding box, it predicts an "objectness" score indicating whether that box encompasses an object instance or not. The RPN box classification loss $\mathcal{L}^{\text{RPN}}_{\text{Cls}}$ is a binary cross-entropy loss which is computed between the predicted $\hat{l}$ and the ground truth l box class labels. For RPN, the boxes have binary class labels, i.e., a class label "1" denotes that the box region contains an object instance and a label "0" indicates that there is no object present within the box region. This loss encourages the RPN to predict region proposals with high "objectness" scores which are later used by the box refinement head for final object detection.

For the bounding-box regression loss $\mathcal{L}^{\text{RPN}}_{\text{Box}}$, an $L1$ loss is used. which is computed between the predicted ($\hat{q}$) and ground truth (q) bounding box coordinate offsets. Note, the regression loss is only computed for positive predicted boxes [61].

$$\mathcal{L}^{\text{RPN}} = \mathcal{L}^{\text{RPN}}_{\text{Cls}} + \mathcal{L}^{\text{RPN}}_{\text{Box}} \tag{3.13}$$

$$\mathcal{L}^{RPN}_{\text{Cls}} = L_{\text{BCE}}\left(\hat{l}, l\right) \tag{3.14}$$

$$\mathcal{L}^{\text{RPN}}_{\text{Box}} = \lambda_{RPN} \sum_{i \in x,y,w,r} L_1(\hat{q}_i, q_i) \tag{3.15}$$

LIDAPS is trained on both the source $\mathcal{D}^s = \{x^s_i\}^{N^s}_{i=1}$ and mixed $\mathcal{D}^t = \{x^t_i\}^{N^t}_{i=1}$ domain images containing target regions/pixels.

Thus, we use **Q** to denote the groundtruth bounding-boxes when training the student network on the source domain images. While training on the augmented images (output by the IMix), **Q** represents a union set of the groundtruth source and confidence-filtered pseudo-bounding-boxes from target as shown in Eq. 3.16. q^s denotes the groundtruth bounding-boxes of the source image, while q^t denotes pseudo-bounding-boxes (predicted by the teacher network) on the target image. Here, h_i is the confidence score predicted for the i-th box and the i-th mask by the teacher network.

$$\mathbf{Q} = \begin{cases} \mathbf{Q}^s = q^s_i & \text{if Source} \\ \mathbf{Q}^s \cup \bigcup_i \mathbb{1}[h_i > \tau]\, q^t_i & \text{if IMix} \end{cases} \tag{3.16}$$

The refinement network consists of a box-head and a mask-head following FastRCNN [16]. As seen in Eq. 3.17, the box-head is trained using a box classification loss $\mathcal{L}^{\text{Ref}}_{\text{Cls}}$ and a box regression loss $\mathcal{L}^{\text{Ref}}_{\text{Box}}$, while the mask-head has a mask segmentation loss $\mathcal{L}^{\text{Ref}}_{\text{Mask}}$.

$$\mathcal{L}^{\text{Ref}} = \mathcal{L}^{\text{Ref}}_{\text{Cls}} + \mathcal{L}^{\text{Ref}}_{\text{Box}} + \mathcal{L}^{\text{Ref}}_{\text{Mask}} \tag{3.17}$$

The box-head takes as inputs the RoIAlign [22] features and the region proposals output by the RPN network, and predicts refined bounding-boxes and their classification scores. The classification scores are the softmax probability scores for all the thing classes plus a background class(C_{things}+1).

Similar to the RPN, the box-head has a box classification loss $\mathcal{L}^{\text{Ref}}_{\text{Cls}}$ and a box regression loss $\mathcal{L}^{\text{Ref}}_{\text{Box}}$. The box classification loss is computed between the predicted per-class probabilities P_{cl} and the groundtruth $u \in \mathbf{U}$ class labels for each predicted box as in Eq. 3.18. Unlike RPN, where the box classification loss is a binary cross-entropy loss, $\mathcal{L}^{\text{Ref}}_{\text{Cls}}$ is a categorical cross-entropy loss for multi-class classification.

$$\mathcal{L}^{\text{Ref}}_{\text{Cls}} = L_{CE}(P_{cl}, u) \tag{3.18}$$

The box regression loss is computed between the predicted $\hat{v}_{u,i}$ and ground truth v_i bounding boxes as in Eq. 3.19. The predicted bounding box by the box-head $\hat{v}_{c,i}$ is for the class $c \in C$. Having predictions for all classes mitigates the competition between the classes.

$$\mathcal{L}^{\text{Ref}}_{\text{Box}} = \lambda_{Ref} \sum_{i \in x,y,w,r} L1(\hat{v}_{u,i}, v_i) \tag{3.19}$$

Similar to RPN training, the box-head is trained on both source and target domain bounding boxes $\mathbf{Q}$. While training on the source image, we use the groundtruth source bounding-boxes, and for training on augmented images (output by IMix), we use a union set of the groundtruth source and pseudo bounding-boxes as in Eq. 3.16.

$\mathbf{U}$ denotes the ground-truth source bounding box class labels $\mathbf{U}^s$ when training the student network on the source domain images. While training the student network on the augmented images generated by the IMix, $\mathbf{U}$ represents a union set of groundtruth source bounding boxes and confidence-filtered pseudo bounding-box class labels as shown in Eq. 3.20.

$$\mathbf{U} = \begin{cases} \mathbf{U}^s = u_i^s \quad \text{if Source} \\ \mathbf{U}^s \;\cup\; \bigcup_i \mathbb{1}[h_i > \tau]\; u_i^t \quad \text{if IMix} \end{cases} \tag{3.20}$$

The mask-head predicts C masks of dimension $w \times h$ for each of the RoIs. Each predicted mask , $\hat{m}_c$, is for an ROI and a specific class. This mitigates the competition in between the classes. Each predicted mask is associated to a groundtruth mask $m \in \mathbf{Masks}$. When training with IMix, $\mathbf{Masks}$ contains confidence-filtered pseudo-masks m^t from the target as well as groundtruth masks from the source m^s as shown in Eq. 3.21.

$$\mathbf{Masks} = \begin{cases} \mathbf{Masks}^s = m_i^s \quad \text{if Source} \\ \mathbf{Masks}^s \;\cup\; \bigcup_i \mathbb{1}[h_i > \tau]\; m_i^t \quad \text{if IMix} \end{cases} \tag{3.21}$$

Eq. 3.22 indicates the binary cross-entropy loss computed between the predicted $\hat{m}$ and groundtruth masks m, where $u \in C$ denotes the ground truth class label for the predicted mask.

$$\frac{1}{w \times h} \sum_{1 \le i,j \le h} m_{i,j} \log(\hat{m}_{u,i,j}) + (1 - m_{i,j}) \log(1 - \hat{m}_{u,i,j}). \tag{3.22}$$

Before training with IMix, we first pass the target images through the instance decoder of the teacher network θ_{inst} in order to gather the predictions which serve as pseudo-class labels, pseudo-masks, pseudo-bounding-boxes for the student network training. The instance decoder of the teacher network provides per-class probabilities for each of the regions of interest. We use the class with the highest probability as the pseudo-label for the i-th ROI which is shown below:

$$y^t_{\text{inst}_i} = \left[\arg\max_{c'} (\theta_{\text{inst}}(x^{(t)}))_i \right] \tag{3.23}$$

3.4 CLIP-based Domain Alignment (CDA)

A simple but effective solution to reducing catastrophic forgetting when multitask learning is to leverage the embedding space of a pre-trained model as an anchor, i.e. the intermediate features as continual auxiliary targets, which is also commonly employed in unsupervised domain adaptation frameworks to limit overfitting onto the source domain [26, 63]. In this work we exploit both use cases for weight anchoring by relying on CLIP [59] embeddings to regularize the semantic branch of our network. CLIP is trained on a very large-scale image-text pair dataset, providing a diversified, robust world model. We argue that by semantically aligning each domain to the CLIP embedding space, we can implicitly enforce domain invariance. In other words, we train our model such that the features of a source or target image both aim to generate high similarities to a joint CLIP embedding. An illustration of CLIP-based domain alignment (CDA) can be seen in Fig. 3.1 - orange.

However, to be able to exploit CLIP features to avoid the divergence of semantic features of source and target images, regularization needs to be applied deep within the network. This of course imposes limitations on the expressibility of the features or effectiveness of the regularization when directly using CLIP embeddings as targets. To this end, we construct a pixel-level representation from natural language prompts following DenseCLIP [60] and only supervise the similarity to the semantic decoder features. Specifically, our CLIP-based domain alignment strategy follows two steps as illustrated in Fig. 3.9. We first generate class-wise mean CLIP features by mean pooling over the CLIP embeddings generated from P text prompts for C semantic classes following set precedent [60], with P denoting the number of text prompts per class (Fig. 3.9a-c). Each row in the resulting matrix represents a CLIP embedding that encodes meaningful semantic information about a particular class. These embeddings act as anchors within our alignment module, with each generated semantic feature (Fig. 3.9d) aiming to achieve high similarity with a semantically corresponding vector. Finally, we compute the per-pixel text similarity maps σ^{sim} through the inner product of the decoder features and mean CLIP features (Fig. 3.9e).

Formally, the CLIP-based domain alignment loss can be stated as follows:

$$\mathcal{L}_{\text{CLIP}} = -\frac{1}{HWC} \sum_{h=1}^{H} \sum_{w=1}^{W} \sum_{c=1}^{C} \mathbb{1}[y_{(h,w)} = c] \log\left(\hat{y}^{\text{sim}}_{(h,w,c)}\right), \tag{3.24}$$

with $\mathbb{1}[\cdot]$ denoting the indicator function and $\hat{y}^{\text{sim}}$ denoting the embedding-text similarity probability given by:

$$\hat{y}^{\text{sim}}_{(h,w,c)} = \frac{\exp\left(\sigma^{\text{sim}}_{(h,w,c)}\right)}{\sum_{c'=1}^{C} \exp\left(\sigma^{\text{sim}}_{(h,w,c')}\right)}. \tag{3.25}$$

In our proposed LIDAPS model, the CLIP loss is incorporated in our $\mathcal{L}_{sem}$ loss.

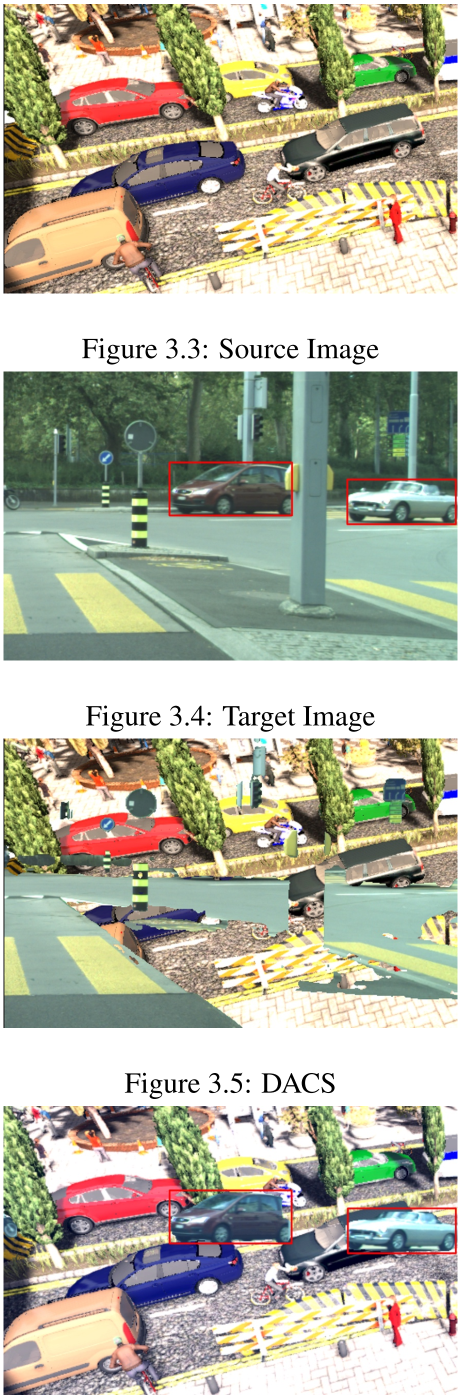


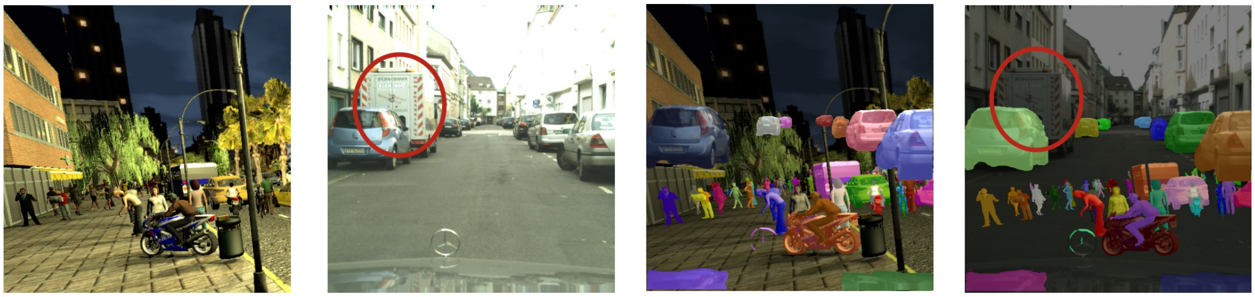
Figure 3.3: Source Image

Figure 3.4: Target Image

Figure 3.5: DACS

Figure 3.6: IMix (ours)

Figure 3.7: Comparison of instance-aware cross-domain mixing (IMix) to DACS that operates on semantics and thus does not preserve instance-level information.



(a) Source (b) Target (c) IMix target to source (d) IMix source to target

Figure 3.8: When using IMix to paste source instances from source to target (c), exhaustive pseudo-masks for the target instances are not guaranteed. For instance, in (d) the truck has no pseudo-mask. In (c), this exhaustiveness is guaranteed because only target instances with predicted pseudo-masks are pasted onto the source image. Thus, training on samples mixed from target to source allows the model to learn on pseudo-groundtruth sets with no false negative examples.

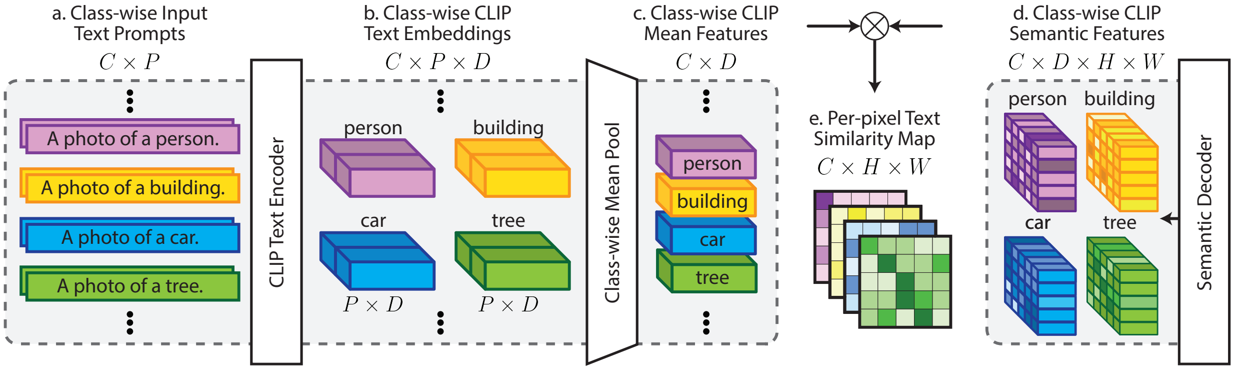


Figure 3.9: Illustrates the pipeline used to compute the pixel-text similarity map for CLIP-based domain alignment. We generate class-wise CLIP mean features from a series of fixed text prompts (a-c). The similarity maps can then be computed by taking their inner product with the semantic decoder features.

Chapter 4

Experiments and Results

4.1 Implementation Details

We train our method on a single NVIDIA GeForce RTX 3090. We use an AdamW optimizer with a learning rate of 6×10^{-5}, a weight decay of 0.01, starting with a linear learning rate warmup for 1.5k iterations, and afterward a polynomial decay. Furthermore, we train over 50k iterations with a batch size of two, consisting of cropped images of size 512x512. We apply a warmup training phase of 40k iterations and only enable IMix in the last 10k iterations. For the RPN and box refinement head losses, we set the loss weights λ_{RPN} and λ_{Ref} to 1.0. Following EDAPS[63], we use MiT-B5 [80] as our encoder backbone (shared by the instance and semantic decoders), MaskRCNN [22] as instance decoder and DAFormer [26] semantic head as the semantic decoder. For CLIP-based domain alignment, we use CLIP [59][1] as the pre-trained text encoder. We empirically set the IMix confidence threshold at 0.75 unless stated otherwise.

4.2 Datasets

We evaluate our method on the popular panoptic UDA benchmarks. For synthetic-to-real adaptation, we use SYNTHIA [62] as the source domain which contains 9,400 synthetic images. For the target domain, we use the Mapillary Vistas [55] dataset and Cityscapes [12]. Cityscapes contains 2,975 training images and 500 validation images, while Mapillary Vistas contains 18,000 training images and 2,000 validation images. For real-to-real adaptation, we use two different benchmarks. First, we train with Cityscapes as the source and Mapillary Vistas as the target domain, and second, we train with Cityscapes as the source and the adverse weather dataset Foggy Cityscapes [65] as the target domain.

4.3 Evaluation Metrics

We report the mean panoptic quality (mPQ) for panoptic segmentation, which measures both the semantic quality (SQ) and the recognition quality (RQ). To highlight the individual task performances, we further report the mIoU for semantic segmentation over 20 classes, and mAP for instance segmentation over 6 *thing* classes. All reported values are the averaged scores over three runs with three different seeds (1, 2, 3).

[1] https://huggingface.co/openai/clip-vit-large-patch14

4.4 Results

We compare our proposed LIDAPS with other state-of-the-art (SOTA) UDA panoptic segmentation methods on four different benchmarks including SYNTHIA → Cityscapes (S→C), SYNTHIA → Mapillary Vistas (S→M), Cityscapes → Mapillary Vistas (C→M) and Cityscapes → Foggy Cityscapes (C→F). As seen in Tab. 4.1, our model consistently outperforms existing works across the board, exceeding the performance of previous SOTA by up to +3.6 mPQ. In particular, for SYNTHIA → Cityscape, our method reaches 44.8 mPQ. Specifically, our work improves the mPQ through a significant gain in mAP as shown in Table 4.2. Similar trends are observed when considering benchmarks such as Cityscapes → Foggy Cityscapes, SYNTHIA → Mapillary Vistas, and Cityscapes to Mapillary Vistas, where our method outperforms the previous SOTA by $+2.9\%$, $+2.6\%$, and $+1.4\%$ mPQ respectively. Furthermore in Fig. 4.1, we provide qualitative results demonstrating the capabilities of LIDAPS. Compared to EDAPS [63], LIDAPS can better separate semantically similar neighboring instances by leveraging instance-aware adaptation via IMix and retain its semantic quality via CDA. Furthermore, in Tables 4.2, 4.3 and 4.4 we show results studying the different design choices such as the pasting direction for IMix, the effects of our different components and the optimal threshold for our model.

Table 4.1: Class-wise comparison to SOTA on four different benchmarks for UDA panoptic segmentation. Reported results are averaged over three runs with three different seeds.

Method	road	sidewalk	building	wall	fence	pole	light	sign	veg	sky	person	rider	car	bus	m.bike	bike	mSQ	mRQ	mPQ
SYNTHIA → Cityscapes																			
FDA[86]	79.0	22.0	61.8	1.1	0.0	5.6	5.5	9.5	51.6	70.7	23.4	16.3	34.1	31.0	5.2	8.8	65.0	35.5	26.6
CRST[99]	75.4	19.0	70.8	1.4	0.0	7.3	0.0	5.2	74.1	69.2	23.7	19.9	33.4	26.6	2.4	4.8	60.3	35.6	27.1
AdvEnt[75]	87.1	32.4	69.7	1.1	0.0	3.8	0.7	2.3	71.7	72.0	28.2	17.7	31.0	21.1	6.3	4.9	65.6	36.3	28.1
CVRN[33]	86.6	33.8	74.6	3.4	0.0	10.0	5.7	13.5	80.3	76.3	26.0	18.0	34.1	37.4	7.3	6.2	66.6	40.9	32.1
UniDAformer[93]	73.7	26.5	71.9	1.0	0.0	7.6	9.9	12.4	81.4	77.4	27.4	23.1	47.0	40.9	12.6	15.4	64.7	42.2	33.0
EDAPS[63]	77.5	36.9	80.1	17.2	1.8	29.2	33.5	40.9	82.6	80.4	43.5	33.8	45.6	35.6	18.0	2.8	72.7	53.6	41.2
LIDAPS(ours)	80.8	**48.8**	**80.8**	**17.6**	**2.5**	**29.9**	**34.6**	**42.9**	**82.8**	**82.9**	**44.4**	**40.5**	**51.7**	39.2	**27.4**	10.7	**74.4**	**57.6**	**44.8**
Cityscapes → Foggy Cityscapes																			
DAF[7]	94.0	54.5	57.7	6.7	10.0	7.0	6.6	25.5	44.6	59.1	26.7	16.7	42.2	36.6	4.5	16.9	70.6	41.7	31.8
FDA[86]	93.8	53.1	62.2	8.2	13.4	7.3	7.6	28.9	50.8	49.7	25.0	22.6	42.9	36.3	10.3	15.2	71.4	43.5	33.0
AdvEnt[75]	93.8	52.7	56.3	5.7	13.5	10.0	10.9	27.7	40.7	57.9	27.8	29.4	44.7	28.6	11.6	20.8	72.3	43.7	33.3
CRST[99]	91.8	49.7	66.1	6.4	14.5	5.2	8.6	21.5	56.3	50.7	30.5	30.7	46.3	34.2	11.7	22.1	72.2	44.9	34.1
SVMin[21]	93.4	53.4	62.2	12.3	15.5	7.0	8.5	18.0	54.3	57.1	31.2	29.6	45.2	35.6	11.5	22.7	72.4	45.5	34.8
CVRN[33]	93.6	52.3	65.3	7.5	15.9	5.2	7.4	22.3	57.8	48.7	32.9	30.9	49.6	38.9	18.0	25.2	72.7	46.7	35.7
UniDAformer[93]	93.9	53.1	63.9	8.7	14.0	3.8	10.0	26.0	53.5	49.6	38.0	35.4	57.5	44.2	28.9	29.8	72.9	49.5	37.6
EDAPS[63]	91.0	68.5	80.9	24.1	29.0	50.1	47.2	67.0	85.3	71.8	50.9	51.2	64.7	47.7	36.9	41.5	79.2	70.5	56.7
LIDAPS(ours)	92.3	**70.0**	**83.2**	23.8	**31.9**	**56.4**	**47.7**	**68.8**	**86.6**	**72.5**	**53.2**	**53.6**	**68.0**	**56.6**	**42.8**	**45.9**	**80.2**	**73.2**	**59.6**
SYNTHIA → Mapillary Vistas																			
FDA[86]	44.1	7.1	26.6	1.3	0.0	3.2	0.2	5.5	45.2	61.3	30.1	13.9	39.4	12.1	8.5	7.0	63.8	26.1	19.1
CRST[99]	36.0	6.4	29.1	0.2	0.0	2.8	0.5	4.6	47.7	68.9	28.3	13.0	42.4	13.6	5.1	2.0	63.9	25.2	18.8
AdvEnt[75]	27.7	6.1	28.1	0.3	0.0	3.4	1.6	5.2	48.1	66.5	28.4	13.4	40.5	14.6	5.2	3.3	63.6	24.7	18.3
CVRN[33]	33.4	7.4	32.9	1.6	0.0	4.3	0.4	6.5	50.8	76.8	30.6	15.2	44.8	18.8	7.9	9.5	65.3	28.1	21.3
EDAPS[63]	77.5	25.3	59.9	14.9	0.0	27.5	33.1	37.1	72.6	92.2	32.9	16.4	47.5	31.4	13.9	3.7	71.7	46.1	36.6
LIDAPS(ours)	76.5	25.2	**64.2**	14.0	**0.2**	**29.1**	**35.6**	35.3	72.1	**94.4**	**33.8**	**18.3**	**50.3**	**33.9**	**19.3**	5.9	**73.9**	**47.7**	**38.0**
Cityscapes → Mapillary Vistas																			
CRST[99]	77.0	22.6	40.2	7.8	10.5	5.5	11.3	21.8	56.5	77.6	29.4	18.4	56.0	27.7	11.9	18.4	72.4	39.9	30.8
FDA[86]	74.3	23.4	42.3	9.6	11.2	6.4	15.4	23.5	60.4	78.5	33.9	19.9	52.9	8.4	17.5	16.0	72.3	40.3	30.9
AdvEnt[75]	76.2	20.5	42.6	6.8	9.4	4.6	12.7	24.1	59.9	83.1	34.1	22.9	54.1	16.0	13.5	18.6	72.7	40.3	31.2
CVRN[33]	77.3	21.0	47.8	10.5	13.4	7.5	14.1	25.1	62.1	86.4	37.7	20.4	55.0	21.7	14.3	21.4	73.8	42.8	33.5
EDAPS[63]	58.8	43.4	57.1	25.6	29.1	34.3	35.5	41.2	77.8	59.1	35.0	23.8	56.7	36.0	24.3	25.5	75.9	53.4	41.2
LIDAPS(ours)	49.1	**44.3**	**70.1**	**26.5**	**29.9**	**37.4**	**37.2**	**43.2**	**80.0**	46.1	35.9	**25.0**	**57.1**	**41.6**	**29.6**	**28.4**	**76.6**	**54.9**	**42.6**

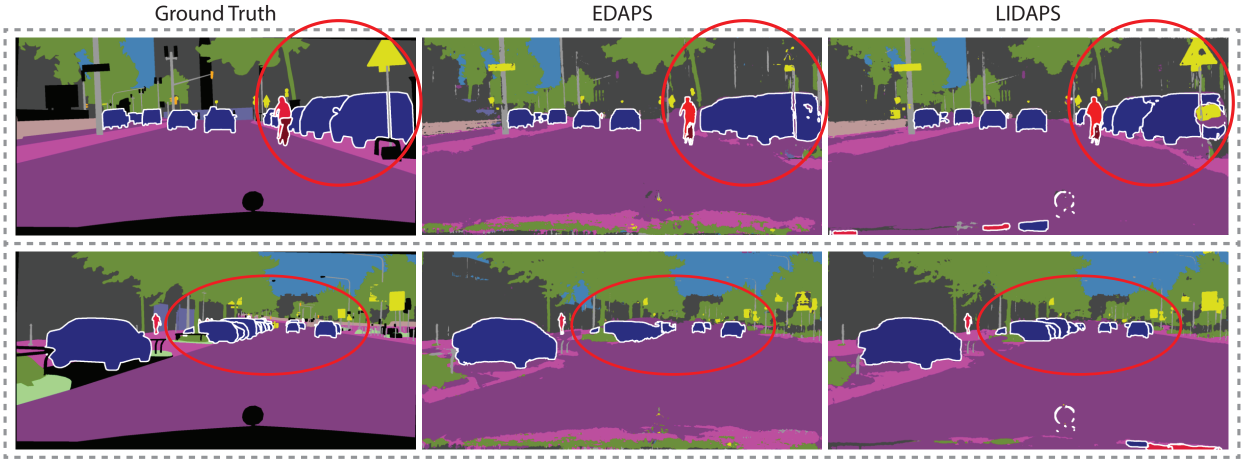


Figure 4.1: Qualitative results from SYNTHIA → Cityscape comparing EDAPS [63] to our proposed LIDAPS.

Table 4.2: Ablation study on proposed modules. Starting from a baseline EDAPS*, we individually introduce our instance-aware cross-domain mixing (IMix) and CLIP-based domain alignment (CDA).

EDAPS*	IMix	CDA	mSQ	mRQ	mPQ	mIoU	mAP
✓			72.3±0.2	53.3±0.8	41.0±0.4	58.0±0.2	34.1±1.0
✓	✓		73.0±0.0	54.7±0.8	42.3±0.6	57.7±0.3	39.5±2.3
✓		✓	73.9±0.3	55.0±0.6	42.9±0.6	59.6±0.6	34.4±0.6
✓	✓	✓	**74.4±0.28**	**57.6±0.294**	**44.8±0.2**	**59.6±0.6**	**42.6±0.7**

Table 4.3: Ablation study on mixing strategy for panoptic segmentation comparing (i) the mixing direction when applying IMix, (ii) the effects of ClassMix when applied from target-to-source as opposed to source-to-target. The baseline is EDAPS*+CDA.

	Method	Copy	Paste	mSQ	mRQ	mPQ	mIoU	mAP
	Baseline	-	-	73.9±0.3	55.0±0.6	42.9±0.6	59.6±0.6	34.4±0.6
(i)	+ IMix	Source	Target	62.0±3.3	37.6±0.63	29.3±0.5	56.2±0.7	1.9±1.9
		Target	Source	**74.4±0.2**	**57.6±0.2**	**44.8±0.2**	**59.6±0.6**	**42.6±1.7**
(ii)	+ ClassMix [57]	Target	Source	73.5±0.2	53.9±0.7	42.1±0.6	58.6±0.8	34.8±0.9

Table 4.4: Hyperparameter study on the confidence-filtering threshold applied to the pseudo-masks for IMix.

Filter	mSQ	mRQ	mPQ	mIoU	mAP
0	73.3±0.1	52.1±0.4	40.0±0.4	59.2±0.8	28.7±1.3
0.25	74.0±0.1	54.8±0.3	42.5±0.3	59.3±0.7	36.8±1.3
0.5	74.4±0.5	56.5±0.3	44.0±0.3	59.2±0.7	40.8±1.2
0.75	**74.4±0.2**	**57.6±0.2**	**44.8±0.2**	**59.6±0.6**	**42.6±0.7**
1	73.9±0.4	55.0±0.7	42.9±0.6	59.6±0.6	34.4±0.6

Table 4.5: Ablation study on EDAPS and LIDAPS in an equalized setting where EDAPS is trained for 50k iterations on three different benchmarks.

Method	mSQ	mRQ	mPQ	mIoU	mAP
SYNTHIA → Cityscapes					
EDAPS	72.4±0.4	53.2±1.0	40.8±0.9	57.5±0.7	33.7±0.6
LIDAPS	**74.4±0.28**	**57.6±0.294**	**44.8±0.2**	**59.6±0.6**	**42.6±0.7**
SYNTHIA → Mapillary Vistas					
EDAPS	72.9±0.4	46.1±0.2	36.6±0.2	55.4±4.1	32.8±0.3
LIDAPS	**73.9±1.9**	**47.7±0.2**	**38.0±0.2**	**58.8±0.5**	**38.7±0.2**
Cityscapes → Cityscapes foggy					
EDAPS	79.2±0.1	71.2±0.0	57.3±0.2	83.0±0.6	60.4±0.4
LIDAPS	**80.2±0.1**	**73.2±0.6**	**59.6±0.6**	**87.1±0.7**	**65.3±0.6**

Table 4.6: Ablation study on the FD component. We include feature distance (FD) in our proposed LIDPAS model (LIDAPS$_{\mathcal{FD}}$) and compare its performance to LIDAPS.

Method	mSQ	mRQ	mPQ	mIoU	mAP
LIDAPS$_{\mathcal{FD}}$	74.0±0.3	56.1±1.3	43.7±0.9	58.6±0.8	40.3±0.9
LIDAPS	**74.4±0.28**	**57.6±0.294**	**44.8±0.2**	**59.6±0.6**	**42.6±0.7**

Chapter 5

Discussion

In this chapter, we discuss the ablation studies on SYNTHIA → Cityscapes to demonstrate the effectiveness of our proposed components.

5.1 Effects of Network Components

In Tab. 4.2, we isolate the effects of the different modules of our proposed pipeline. Starting from the baseline EDAPS*, we introduce our cross-domain instance mixing (IMix) which significantly improves the panoptic segmentation performance ($+1.3\%$ mPQ) through instance segmentation ($+5.4\%$ mAP). Due to their contradictory goals, the improvement in instance segmentation comes in lieu of semantic performance (-0.3% mIoU) that becomes subject to catastrophic interference. To remedy this we propose our next contribution, CLIP-based domain alignment (CDA). First, we separately introduce CDA to understand its isolated effects. As observed, the module aids the panoptic segmentation performance with a $+1.9\%$ mPQ improvement, which solely stems from the gains in semantic segmentation ($+1.6\%$ mIoU increase as opposed to the relatively unchanged mAP). Next, we showcase the combined effects of the two components which improve both the semantic and instance segmentation performance of our baseline, allowing LIDAPS to achieve 44.8% mPQ. As seen, the final model demonstrates significant gains in instance segmentation ($+8.5$ mAP) thanks to IMix, while retaining its semantic segmentation gains from the CLIP-based domain alignment ($+1.6\%$ mIoU).

5.2 Cross-Domain Mixing direction

We investigate the impact of the mixing direction for IMix. Specifically, we compare the effects of source-to-target mixing, in which we cut ground truth instance masks from a source image and paste them onto a target image, to our proposed target-to-source mixing, where we rely on the filtered predicted instances from the target domain to augment onto a source image. As seen in Tab. 4.3 (i), cross-mixing from source-to-target substantially degrades the panoptic performance, specifically the instance segmentation quality. Comparing pasting masks from target-to-source versus pasting from source-to-target, we observe significant benefits in favor of the former approach. Copying from source-to-target((i) first row), we take groundtruth instance masks from the source domain and paste them onto the target domain, creating a new augmented image. The new augmented image contains a mixture of source instances and target instances. The masks of the target instances are predicted using the teacher network while the groundtruth masks of the source instances are available. Subsequently, the instance decoder (and the backbone encoder) are trained on all of the masks from this augmented image. Copying from target-to-source((i) second row), representing our LIDAPS model, pseudo masks predicted by the teacher on the target image with a confidence above the threshold 0.75, are pasted onto the source image to create a new augmented image. The new augmented image contains both source instances and target instances where the source instance have groundtruth masks. We find that cross-mixing from source-to-target substantially degrades the baseline mPQ from 42.9 to 29.33 and the baseline mAP performance from to 34.4 to 1.90. This degradation can be attributed to the absence of groundtruth masks for the target instance, where only teacher-predicted pseudo masks are available. Some of these pseudo-masks may have low confidence levels but are nevertheless trained on. Additionally, filtering out low-confidence pseudo masks in this scenario results in supervision with a set of incomplete masks, as instances with deleted pseudo masks would still persist in the image. This highlights the superiority of pasting from target-to-source for achieving better segmentation performance as it allows us to dispose of low-confidence masks and their instances.

Furthermore in Tab. 4.3 (i) and (ii), we isolate the effects of the mixing task by fixing the mixing direction. Specifically, we compare an inverted ClassMix [57] that cuts and pastes semantic masks from target-to-source, to our proposed IMix strategy that works on an instance level. As seen, the inverted ClassMix slightly underperforms compared to the baseline model and significantly underperforms compared to IMix (-2.7% mPQ). We speculate that this is because there is already a ClassMix in the base EDAPS model(pasting in the opposite direction).

5.3 Confidence-Filtering Threshold

In Tab. 4.4, we aim to identify the optimal threshold in our IMix strategy for filtering out pseudo-masks with confidence levels below that threshold on the SYNTHIA → Cityscapes setting. In this Table, the warmup of the experiments consists of EDAPS*+CDA and the refinement phase includes IMix with different confidence filtering thresholds. Setting the threshold at 1. indicates disabling IMix during the refinement phase. This can be considered as our baseline for this Table. When employing IMix with filters of 0 and 0.25, the performance experiences an mPQ decrease by 2.9% and 0.4%. This indicates that such thresholds are too low to filter out the low quality pseudo-masks. In contrast, for the 0.5 and 0.75 thresholds, the mPQ performance improves respectively by +1.1% and +1.9%. This finding suggests that filtering out pseudo masks with a confidence level around 0.75 enhances the model's segmentation performance over all evaluation metrics, making it the most effective threshold for refinement in this SYNTHIA→Cityscapes context. We find that this threshold needs to be finetuned for different domain shifts. For SYNTHIA → Mapillary and Cityscapes → Mapillary we find that the best threshold is 0.9 while for Cityscapes → Foggy Cityscapes, 0.75 remains the optimal threshold.

5.4 EDAPS*

In our experiments, the EDAPS* baseline follows the same setting as EDAPS [63] except that it does not include the features distance regularizer (FD) that EDAPS has. FD uses ImageNet features as an anchor in order to hinder the learned encoder from forgetting the knowledge it starts out with when initialized with a pre-trained ImageNet encoder. The regularizer is explained in Eq. 5.1. Noteworthy is that FD is applied only on source images in areas corresponding to thing classes. In Table 4.6 we show how the inclusion of FD hinders the performance of our method and thus explains why this component was removed from our experiments. We speculate that this is because the embedding spaces of ImageNet and CLIP are not aligned, therefore, aligning with both gives rise to a drop in performance. Additionally, EDAPS* is trained for 50k iterations instead of 40k which is the duration of training reported for EDAPS. In Table 4.5, we compare EDAPS with LIDPAS, both trained for 50k iterations. We can see that LIDAPS persists on beating EDAPS on three different benchmarks.

$$\mathcal{L}_{\mathrm{FD}} = \|\mathrm{Enc}_{\mathrm{ImgNet}}(x^s) - \mathrm{Enc}_{\theta}(x^s)\| \tag{5.1}$$

Chapter 6

Conclusion

In this work, we tackle the task of unsupervised domain adaptation for panoptic segmentation. To this end, we introduce a framework LIDAPS that reduces the domain gap between target and source images by leveraging instance-aware cross-domain mixing. Specifically, we propose a novel mixing strategy IMix, that cuts and pastes confidence-filtered instance predictions from the target to the source domain, and thus retains the exhaustiveness of the resulting pseudo-labels while reducing the injected confirmation bias. To limit the effects of emerging catastrophic forgetting, we then propose a CLIP-based domain alignment mechanism that employs CLIP embeddings as anchors for both the source and target domain. While these proposed mechansims can be combined with any off-the-shelf segment method, we put together an end-to-end model that incorporates our methods which we named LIDAPS. Our resulting LIDAPS model consistently outperforms existing SOTA models on popular UDA panoptic benchmarks.
AI tools Here I describe where I used AI tools for. After having written my thesis, I used ChatGPT[1](Online version 3.5) to check spelling mistakes and to explore options for making my own text more coherent by better phrasing the sentence connections and correcting grammar mistakes.

6.1 Limitations

Depending on the source and target domain, the threshold for pseudo-mask confidence filtering needs to be manually found with experiments. Moreover, we show that this threshold is different on different benchmarks. In future work, we will explore the prediction of the threshold using a jointly trained neural network. Furthermore, during the refinement phase where IMix is enabled (last 10k iterations), we are adding one forward pass and one backward pass to each iteration which increases the runtime.

[1] OpenAI. (2023). ChatGPT (3.5 version https://chat.openai.com) [Large language model].

Appendix A

Additional Results and Explored Directions

A.1 Overview

In this appendix, we provide additional results and an insight into the explored directions that did not result in performance improvement. In section A.2, some quantitative results of our method in comparison to the EDAPS[63] is provided. In section A.4.1, the different backbones that we tried to use for EDAPS are discussed. In section A.4.2, the alignment of the instance decoder embeddings with the CLIP model is reported and analyzed. In section A.4.3, another attempted strategy for mixing is explained and discussed. In section A.4.4, self-training on the target domain directly without any mixing is explored.

A.2 Additional Qualitative Results

In this section, we provide additional qualitative panoptic segmentation results in Fig. A.1.

A.3 Embedding Plotting

In this section, we provide plot embeddings of the semantic decoder to study how the embeddings look like with the incorporation of each component. Fig. A.2(a) illustrates the EDAPS* embeddings, Fig. A.2(b) illustrates the EDAPS*+CDA embeddings while Fig. A.2(c) illustrates EDAPS*+CDA+IMix or otherwise called LIDAPS embeddings. We can see that the visualizations do not show the improvement we see in qualitative and quantitative results. We speculate that this is because we are using PCA and TSNE to project embeddings of 768 and 256 dimentionality to 2 dimensions which destroys much of the properties that yield improvements.

A.4 Other Explored Attempts

A.4.1 Backbone

Within this thesis, different backbones were tested including a UNET diffusion pre-trained backbone, a clip pre-trained vision transformer, and a CLIP pre-trained ResNet(ResNet101 and ResNet50). We experimented with these different backbone architectures.

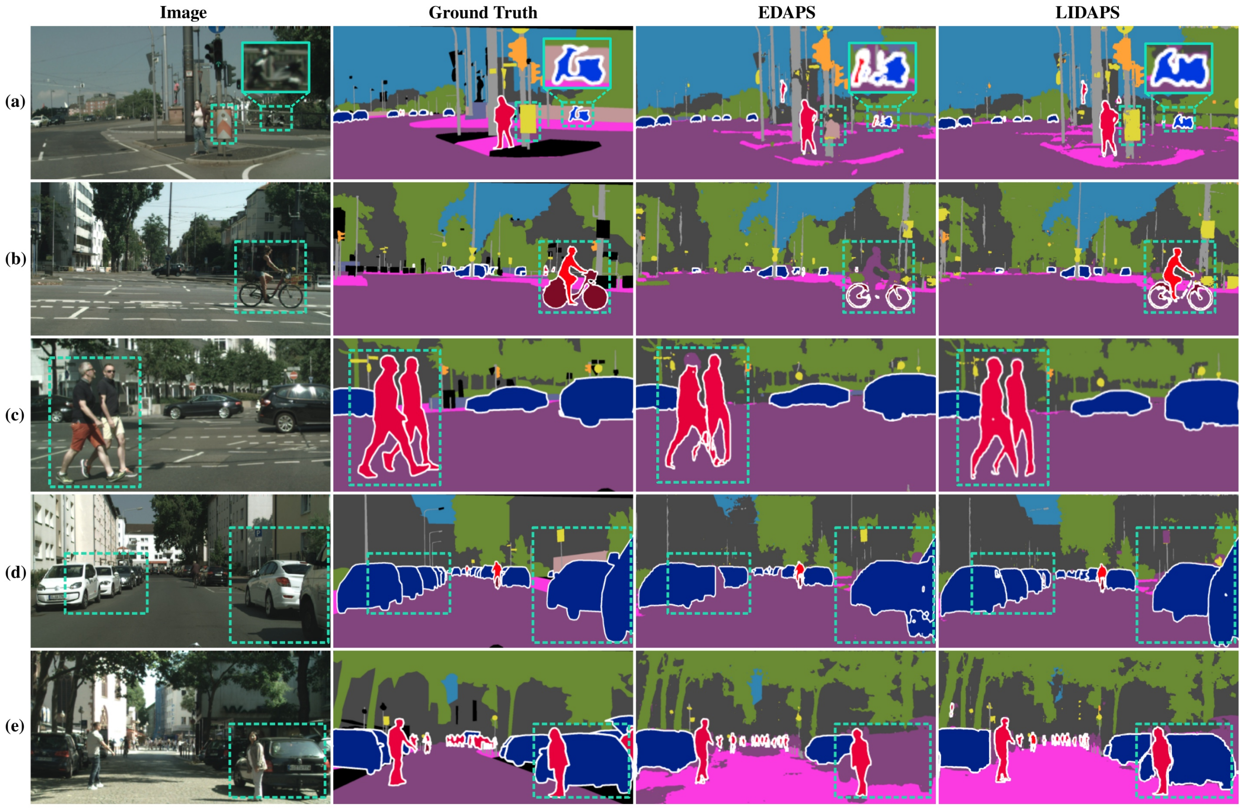


Figure A.1: Additional qualitative results on SYNTHIA → Cityscape UDA benchmark comparing EDAPS [63] to our proposed LIDAPS. Our proposed LIDAPS model predicts improved semantic and instance segmentation for several classes including "motor-bike" (a), "rider" (b), "person" (c) and "car" (d,e).

Pre-trained Diffusion UNET

According to [17], the UNET diffusion pretrained backbone, has a disentangled embedding space in terms of real and synthetic features. By using this frozen backbone as an encoder and fine-tuning their decoder on source domain data, they obtain impressive semantic segmentation results on target domain data. Unlike (unsupervised domain adaptive) UDA methods, they do not have access to the domain data during training. Following this, we studied how a UNET diffusion pre-trained model would affect EDAPS's panoptic quality, once it replaced EDAPS's current MiT-B5 backbone architecture. In a source-only setting, we noticed that the mPQ endured a significant drop due to a significant drop in the instance segmentation score, mAP. However, when training solely the semantic decoder, then we notice an improvement in the semantic segmentation score, mIoU. We conclude that the UNET diffusion pre-trained embedding space does not preserve features that are suitable for the instance segmentation task.

Vision Transformer

The mixed transformer(MiT-B5) [80] used in EDAPS is an optimized version of the vision transformer for semantic segmentation. However, the existing pre-trained weights for MiT-B5 are from training on the ImageNet dataset while there exist pre-trained weights for ViT [14] from a CLIP training setting. Since the CLIP training is done on a larger dataset than ImageNet, the CLIP pre-trained weights supposedly is more domain-robust and have richer semantic knowledge. Hence, through experimentation, we investigated whether a CLIP pre-trained ViT would do better than MiT-B5. We allow it to train for a longer number of iterations than the latter since it is less optimized [80]. Changing the architecture to ViT did not enhance

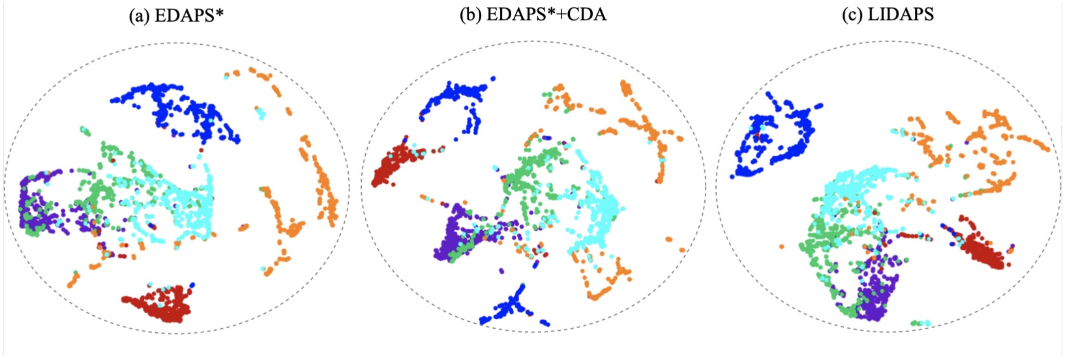


Figure A.2: Plotting of the semantic decoder embeddings from models with different components. The plots do not show a significant difference because the dimension projection does not preserve the information that yields significant improvement when going from EDAPS* to LIDAPS.

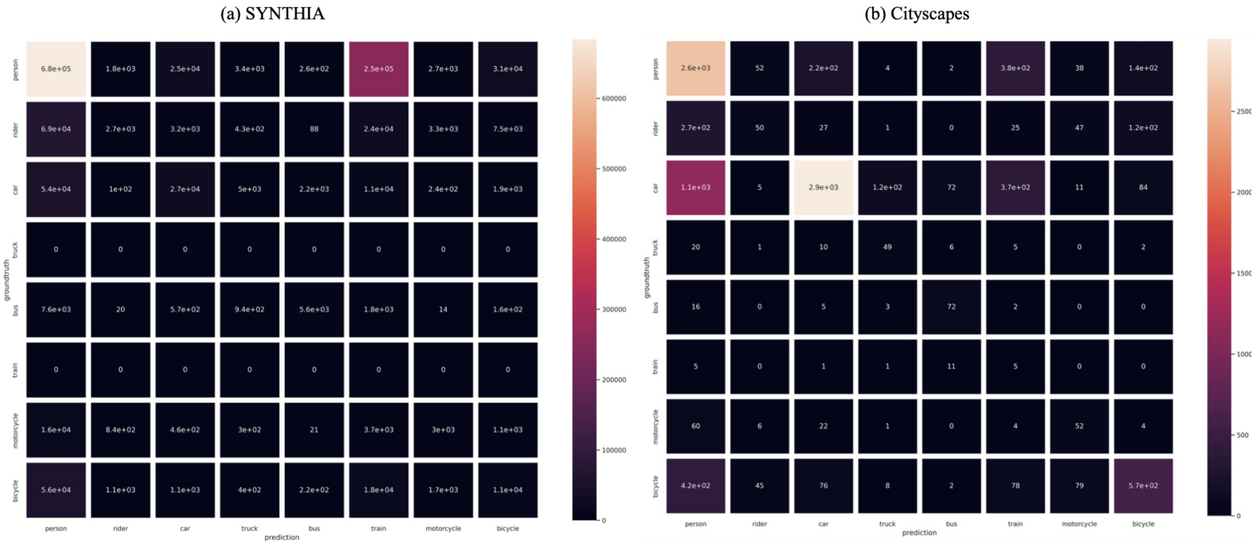
our results. This is not surprising given that the plain vision transformer is non-hierarchical meaning that it outputs features of the same scale which can be detrimental for segmentation. According to the work [44], this can be circumvented by adding a pyramid network. Therefore, similar to [44], we added a feature pyramid network at the output of ViT. This change increased the performance in comparison to a source-only EDAPS setting. However, in the UDA setting, EDAPS performed better during a shorter number of iterations.

ResNet

Replacing the backbone with the architectures ResNet101 and ResNet50 and training from scratch did not improve the panoptic quality of the model.

A.4.2 CLIP Alignment on the Instance Decoder

In addition to aligning the embeddings from the semantic decoder with clip embeddings, inspiring from [89], we align the instance embeddings which are the ROI aligned features of the groundtruth instance boxes with the CLIP embeddings of the resized cropped instance groundtruth boxes. Since the instance decoder aims for separability between instances of the same class, it is not suitable to use CLIP text embeddings which contain only the name of the class. However, CLIP vision embeddings are more suitable since they can encompass more information about an object instance than only its class. Before applying the CLIP vision encoder, the crop of the ground truth instance needs to be upsampled to 224x224 which is the required size for the CLIP vision encoder. Hence, crops that are below a certain threshold are not considered for alignment. This alignment is done via an L2 regularizer. Adding this regularisor to EDAPS*+CDA did not improve the results. In order to investigate this further, we attempted to predict the class of the object inside each crop using the CLIP vision and text encoders. The predicted class is the CLIP text encoding that has the most similarity with CLIP vision embedding. For this, we consider one mean text embedding for each of the 8 thing classes. We plot a grid illustrating the number of different groundtruth predictions in Fig. A.3. We speculate that some of the misinterpretations are due to crops (which are boxes, not masks) that can contain multiple instances. For instance, a crop of a bicycle will most likely contain its rider as well. One solution would be to put the pixels not within the instance object to zero. However, having many black pixels is outside of the distribution that CLIP has been trained on and would fail.



(a) SYNTHIA

	person	rider	car	truck	bus	train	motorcycle	bicycle
person	6.8e+05	1.8e+03	2.5e+04	3.4e+03	2.6e+02	2.5e+05	2.7e+03	3.1e+04
rider	6.9e+04	2.7e+03	3.2e+03	4.3e+02	88	2.4e+04	3.3e+03	7.5e+03
car	5.4e+04	1e+02	2.7e+04	5e+03	2.2e+03	1.1e+04	2.4e+02	1.9e+03
truck	0	0	0	0	0	0	0	0
bus	7.6e+03	20	5.7e+02	9.4e+02	5.6e+03	1.8e+03	14	1.6e+02
train	0	0	0	0	0	0	0	0
motorcycle	1.6e+04	8.4e+02	4.6e+02	3e+02	21	3.7e+03	3e+03	1.1e+03
bicycle	5.6e+04	1.1e+03	1.1e+03	4e+02	2.2e+02	1.8e+04	1.7e+03	1.1e+04

(b) Cityscapes

	person	rider	car	truck	bus	train	motorcycle	bicycle
person	2.6e+03	52	2.2e+02	4	2	3.8e+02	38	1.4e+02
rider	2.7e+02	50	27	1	0	25	47	1.2e+02
car	1.1e+03	5	2.9e+03	1.2e+02	72	3.7e+02	11	84
truck	20	1	10	49	6	5	0	2
bus	16	0	5	3	72	2	0	0
train	5	0	1	1	11	5	0	0
motorcycle	60	6	22	1	0	4	52	4
bicycle	4.2e+02	45	76	8	2	78	79	5.7e+02

Figure A.3: Using a CLIP [59] vision encoder and a CLIP text encoder, we calculate the similarity between the CLIP vision embedding of every groundtruth crop and the CLIP text embeddings of the eight thing classes. We do this for (a) SYNTHIA dataset and (b) the Cityscapes dataset. The predicted class consists of the text embedding that has the most similarity with the vision embedding. This analysis illustrates that using the CLIP vision encoder locally does not give good results as there are many false predictions.

A.4.3 Mixing Source to Source

Comparing the Cityscapes [12] and the SYNTHIA [62] datasets in the scope of domain adaptation from SYNTHIA to Cityscapes, we realized that in most Cityscapes images, there are many overlapping cars that create significant occlusions. In an attempt to bridge this gap, we decided to paste overlapping groundtruth SYNTHIA cars several times in each training image. An example is illustrated in Fig. A.4. However, training on these augmented samples did not improve our results.

A.4.4 Target Self-training for the Instance Decoder

In order to explore the instance self-training further, we experimented with different strategies:

- Only target: We explored training the instance branch (the instance decoder and the shared encoder) on the target images using as supervision the pseudo-instances predicted by the teacher. This method did not improve the scores.
- Only target with coefficients: In another attempt, similarly to the self-supervised semantic decoder training loss, within the cross entropy loss of the encoder, we took into account the confidence scores of the pseudo-masks. This experiment also did not improve the scores.
- In this experiment, we intended on self-training the instance decoder branch on the target images using only the confidence-filtered pseudo-masks generated by the teacher. However, deleting certain low-confidence pseudo-masks can result in false negatives as deleting a pseudo-mask can yield a target instance having no associated supervised mask. To evade this issue, we do not penalize via the objectness loss (binary classification loss) of the RPN when the groundtruth label is 0 and the predicted label is 1. We consider that this could be a case where the mask of an instance is not amongst the filtered or predicted pseudo-masks. This attempt also failed to improve the results.

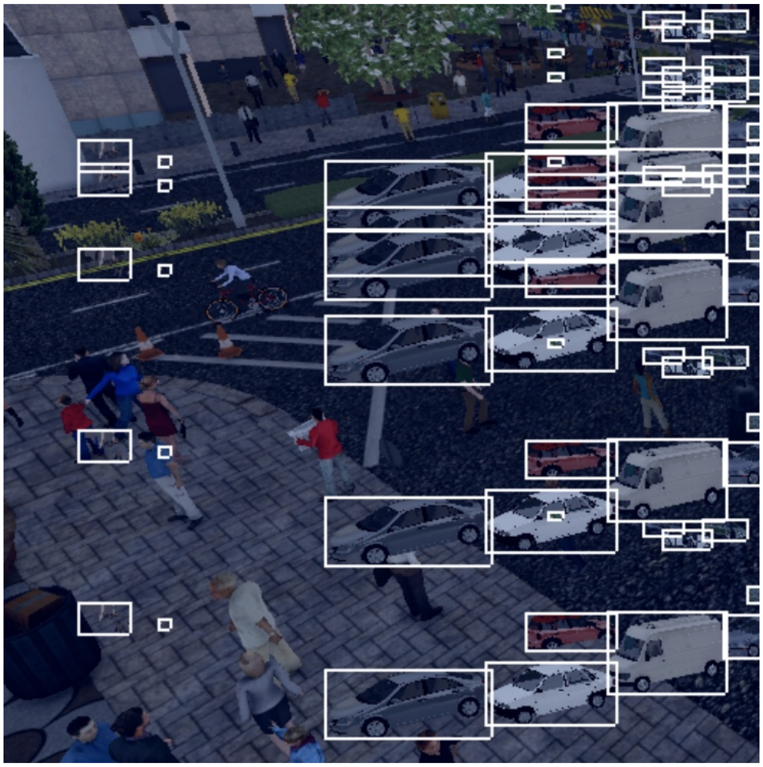


Figure A.4: Several groundtruth source cars are pasted onto the source image in order to mimic the overlaps, occlusions, and number of cars in the Cityscapes dataset

Bibliography

[1] Nikita Araslanov and Stefan Roth. Self-supervised augmentation consistency for adapting semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15384–15394, 2021.

[2] Edoardo Arnaudo, Antonio Tavera, Carlo Masone, Fabrizio Dominici, and Barbara Caputo. Hierarchical instance mixing across domains in aerial segmentation. *IEEE Access*, 11:13324–13333, 2023.

[3] Chia-Yuan Chang, Shuo-En Chang, Pei-Yung Hsiao, and Li-Chen Fu. Epsnet: Efficient panoptic segmentation network with cross-layer attention fusion. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, November 2020.

[4] Nicolas Harvey Chapman, Feras Dayoub, Will Browne, and Christopher Lehnert. Predicting class distribution shift for reliable domain adaptive object detection. *arXiv preprint arXiv:2302.06039*, 2023.

[5] Mu Chen, Zhedong Zheng, Yi Yang, and Tat-Seng Chua. Pipa: Pixel-and patch-wise self-supervised learning for domain adaptative semantic segmentation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 1905–1914, 2023.

[6] Xi Chen, Shuang Li, Ser-Nam Lim, Antonio Torralba, and Hengshuang Zhao. Open-vocabulary panoptic segmentation with embedding modulation. *arXiv preprint arXiv:2303.11324*, 2023.

[7] 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*, pages 3339–3348, 2018.

[8] Zheng Chen, Zhengming Ding, Jason M Gregory, and Lantao Liu. Ida: Informed domain adaptive semantic segmentation. *arXiv preprint arXiv:2303.02741*, 2023.

[9] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12475–12485, 2020.

[10] Xiao Cheng, Chunbo Zhu, Lijie Yuan, and Suhua Zhao. Cross-modal domain adaptive instance segmentation in sar images via instance-aware adaptation. In *Chinese Conference on Image and Graphics Technologies*, pages 413–424. Springer, 2023.

[11] Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6830–6840, 2019.

[12] 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*, pages 3213–3223, 2016.

[13] Anurag Das, Yongqin Xian, Dengxin Dai, and Bernt Schiele. Weakly-supervised domain adaptive semantic segmentation with prototypical contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15434–15443, 2023.

[14] 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.

[15] Mohammad Fahes, Tuan-Hung Vu, Andrei Bursuc, Patrick Pérez, and Raoul de Charette. Poda: Prompt-driven zero-shot domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18623–18633, 2023.

[16] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[17] Rui Gong, Martin Danelljan, Han Sun, Julio Delgado Mangas, and Luc Van Gool. Prompting diffusion representations for cross-domain semantic segmentation. *arXiv preprint arXiv:2307.02138*, 2023.

[18] Rui Gong, Wen Li, Yuhua Chen, Dengxin Dai, and Luc Van Gool. Dlow: Domain flow and applications. *International Journal of Computer Vision*, 129(10):2865–2888, 2021.

[19] Rui Gong, Qin Wang, Martin Danelljan, Dengxin Dai, and Luc Van Gool. Continuous pseudo-label rectified domain adaptive semantic segmentation with implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7225–7235, 2023.

[20] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021.

[21] Dayan Guan, Jiaxing Huang, Shijian Lu, and Aoran Xiao. Scale variance minimization for unsupervised domain adaptation in image segmentation. *Pattern Recognition*, 112:107764, 2021.

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

[23] Wenbin He, Suphanut Jamonnak, Liang Gou, and Liu Ren. Clip-s4: Language-guided self-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11207–11216, 2023.

[24] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. Pmlr, 2018.

[25] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016.

[26] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9924–9935, 2022.

[27] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Hrda: Context-aware high-resolution domain-adaptive semantic segmentation. In *European Conference on Computer Vision*, pages 372–391. Springer, 2022.

[28] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Domain adaptive and generalizable network architectures and training strategies for semantic image segmentation. *arXiv preprint arXiv:2304.13615*, 2023.

[29] 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*, pages 11721–11732, 2023.

[30] Lukas Hoyer, Dengxin Dai, Qin Wang, Yuhua Chen, and Luc Van Gool. Improving semi-supervised and domain-adaptive semantic segmentation with self-supervised depth estimation. *International Journal of Computer Vision*, 2021.

[31] Jie Hu, Linyan Huang, Tianhe Ren, Shengchuan Zhang, Rongrong Ji, and Liujuan Cao. You only segment once: Towards real-time panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17819–17829, June 2023.

[32] Xuefeng Hu, Ke Zhang, Lu Xia, Albert Chen, Jiajia Luo, Yuyin Sun, Ken Wang, Nan Qiao, Xiao Zeng, Min Sun, et al. Reclip: Refine contrastive language image pre-training with source free domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2994–3003, 2024.

[33] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Cross-view regularization for domain adaptive panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10133–10144, 2021.

[34] Xinyue Huo, Lingxi Xie, Hengtong Hu, Wengang Zhou, Houqiang Li, and Qi Tian. Domain-agnostic prior for transfer semantic segmentation. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 7075–7085, 2022.

[35] Xinyue Huo, Lingxi Xie, Wengang Zhou, Houqiang Li, and Qi Tian. Focus on your target: A dual teacher-student framework for domain-adaptive semantic segmentation. *arXiv preprint arXiv:2303.09083*, 2023.

[36] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1780–1790, 2021.

[37] Mikhail Kennerley, Jian-Gang Wang, Bharadwaj Veeravalli, and Robby T Tan. 2pcnet: Two-phase consistency training for day-to-night unsupervised domain adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11484–11493, 2023.

[38] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollar. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[39] Divya Kothandaraman, Rohan Chandra, and Dinesh Manocha. Ss-sfda: Self-supervised source-free domain adaptation for road segmentation in hazardous environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3049–3059, 2021.

[40] Zhengfeng Lai, Noranart Vesdapunt, Ning Zhou, Jun Wu, Cong Phuoc Huynh, Xuelu Li, Kah Kuen Fu, and Chen-Nee Chuah. Padclip: Pseudo-labeling with adaptive debiasing in clip for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16155–16165, October 2023.

[41] Kuan-Hui Lee, German Ros, Jie Li, and Adrien Gaidon. Spigan: Privileged adversarial learning from simulation. *arXiv preprint arXiv:1810.03756*, 2018.

[42] Suhyeon Lee, Hongje Seong, Seongwon Lee, and Euntai Kim. Wildnet: Learning domain generalized semantic segmentation from the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9936–9946, 2022.

[43] Tianyu Li, Subhankar Roy, Huayi Zhou, Hongtao Lu, and Stéphane Lathuilière. Contrast, stylize and adapt: Unsupervised contrastive learning framework for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4868–4878, 2023.

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

[45] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6936–6945, 2019.

[46] Zhiqi Li, Wenhai Wang, Enze Xie, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, Ping Luo, and Tong Lu. Panoptic segformer: Delving deeper into panoptic segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1280–1289, June 2022.

[47] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7061–7070, 2023.

[48] Yunan Liu, Shanshan Zhang, Yang Li, and Jian Yang. Learning to adapt via latent domains for adaptive semantic segmentation. *Advances in Neural Information Processing Systems*, 34:1167–1178, 2021.

[49] Yifan Lu, Gurkirt Singh, Suman Saha, and Luc Van Gool. Exploiting instance-based mixed sampling via auxiliary source domain supervision for domain-adaptive action detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4145–4156, 2023.

[50] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2507–2516, 2019.

[51] Chaofan Ma, Yuhuan Yang, Yanfeng Wang, Ya Zhang, and Weidi Xie. Open-vocabulary semantic segmentation with frozen vision-language models. *arXiv preprint arXiv:2210.15138*, 2022.

[52] Ruben Mascaro, Lucas Teixeira, and Margarita Chli. Domain-adaptive semantic segmentation with memory-efficient cross-domain transformers. In *The 34th British Machine Vision Conference (BMVC)*, 2023.

[53] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. In *Proceedings of the European Conference on Computer Vision*, pages 415–430. Springer, 2020.

[54] Luke Melas-Kyriazi and Arjun K Manrai. Pixmatch: Unsupervised domain adaptation via pixelwise consistency training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12435–12445, 2021.

[55] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 4990–4999, 2017.

[56] Joshua Niemeijer, Manuel Schwonberg, Jan-Aike Termöhlen, Nico M. Schmidt, and Tim Fingscheidt. Generalization by adaptation: Diffusion-based domain extension for domain-generalized semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2830–2840, January 2024.

[57] Viktor Olsson, Wilhelm Tranheden, Juliano Pinto, and Lennart Svensson. Classmix: Segmentation-based data augmentation for semi-supervised learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1369–1378, 2021.

[58] Duo Peng, Ping Hu, Qiuhong Ke, and Jun Liu. Diffusion-based image translation with label guidance for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 808–820, 2023.

[59] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[60] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18082–18091, 2022.

[61] 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.

[62] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016.

[63] Suman Saha, Lukas Hoyer, Anton Obukhov, Dengxin Dai, and Luc Van Gool. Edaps: Enhanced domain-adaptive panoptic segmentation. *arXiv preprint arXiv:2304.14291*, 2023.

[64] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in Neural Information Processing Systems*, 29, 2016.

[65] Christos Sakaridis, Dengxin Dai, Simon Hecker, and Luc Van Gool. Model adaptation with synthetic and real data for semantic dense foggy scene understanding. In *Proceedings of the european conference on computer vision (ECCV)*, pages 687–704, 2018.

[66] Fengyi Shen, Li Zhou, Kagan Kucukaytekin, Ziyuan Liu, He Wang, and Alois Knoll. Controluda: Controllable diffusion-assisted unsupervised domain adaptation for cross-weather semantic segmentation. *arXiv preprint arXiv:2402.06446*, 2024.

[67] Gyungin Shin, Samuel Albanie, and Weidi Xie. Zero-shot unsupervised transfer instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4847–4857, 2023.

[68] Mainak Singha, Harsh Pal, Ankit Jha, and Biplab Banerjee. Ad-clip: Adapting domains in prompt space using clip. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4355–4364, 2023.

[69] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

[70] 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.

[71] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. Dacs: Domain adaptation via cross-domain mixed sampling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1379–1389, 2021.

[72] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7472–7481, 2018.

[73] Vidit Vidit, Martin Engilberge, and Mathieu Salzmann. Clip the gap: A single domain generalization approach for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3219–3229, 2023.

[74] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2517–2526, 2019.

[75] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2517–2526, 2019.

[76] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5463–5474, June 2021.

[77] Zhonghao Wang, Mo Yu, Yunchao Wei, Rogerio Feris, Jinjun Xiong, Wen-mei Hwu, Thomas S Huang, and Honghui Shi. Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12635–12644, 2020.

[78] Jiannan Wu, Yi Jiang, Bin Yan, Huchuan Lu, Zehuan Yuan, and Ping Luo. Exploring transformers for open-world instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6611–6621, 2023.

[79] Pengfei Xian, Lai-Man Po, Yuzhi Zhao, Wing-Yin Yu, and Kwok-Wai Cheung. Clip driven few-shot panoptic segmentation. *IEEE Access*, 2023.

[80] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

[81] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[82] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2955–2966, 2023.

[83] Jilan Xu, Junlin Hou, Yuejie Zhang, Rui Feng, Yi Wang, Yu Qiao, and Weidi Xie. Learning open-vocabulary semantic segmentation models from natural language supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2935–2944, 2023.

[84] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2945–2954, 2023.

[85] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In *European Conference on Computer Vision*, pages 736–753. Springer, 2022.

[86] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4085–4095, 2020.

[87] Dongyu Yao and Boheng Li. Dual-level interaction for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4527–4536, 2023.

[88] Muyang Yi, Quan Cui, Hao Wu, Cheng Yang, Osamu Yoshie, and Hongtao Lu. A simple framework for text-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7071–7080, 2023.

[89] Wei Yin, Yifan Liu, Chunhua Shen, Anton van den Hengel, and Baichuan Sun. The devil is in the labels: Semantic segmentation from sentences. *arXiv preprint arXiv:2202.02002*, 2022.

[90] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. *arXiv preprint arXiv:2308.02487*, 2023.

[91] Sukmin Yun, Seong Hyeon Park, Paul Hongsuck Seo, and Jinwoo Shin. Ifseg: Image-free semantic segmentation via vision-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2967–2977, 2023.

[92] Giacomo Zara, Subhankar Roy, Paolo Rota, and Elisa Ricci. Autolabel: Clip-based framework for open-set video domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11504–11513, 2023.

[93] Jingyi Zhang, Jiaxing Huang, Xiaoqin Zhang, and Shijian Lu. Unidaformer: Unified domain adaptive panoptic segmentation transformer via hierarchical mask calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11227–11237, 2023.

[94] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12414–12424, 2021.

[95] Qiming Zhang, Jing Zhang, Wei Liu, and Dacheng Tao. Category anchor-guided unsupervised domain adaptation for semantic segmentation. *Advances in neural information processing systems*, 32, 2019.

[96] Hanqing Zhao, Dianmo Sheng, Jianmin Bao, Dongdong Chen, Dong Chen, Fang Wen, Lu Yuan, Ce Liu, Wenbo Zhou, Qi Chu, et al. X-paste: revisiting scalable copy-paste for instance segmentation using clip and stablediffusion. In *International Conference on Machine Learning*, pages 42098–42109. PMLR, 2023.

[97] Xingchen Zhao, Niluthpol Chowdhury Mithun, Abhinav Rajvanshi, Han-Pang Chiu, and Supun Samarasekera. Unsupervised domain adaptation for semantic segmentation with pseudo label self-refinement. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2399–2409, January 2024.

[98] Qianyu Zhou, Zhengyang Feng, Qiqi Gu, Jiangmiao Pang, Guangliang Cheng, Xuequan Lu, Jianping Shi, and Lizhuang Ma. Context-aware mixup for domain adaptive semantic segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(2):804–817, 2022.

[99] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305, 2018.

[100] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European Conference on Computer Vision*, pages 289–305, 2018.