Supplementary

1. Overview

In this section we give an overview on the addressed materials of the appendix. In section 2 we give implementation details such as the architecture details, hyper-parameter settings, CLIP pre-trained model setting, and the losses we used for training. In section 3, we present an example of a false negative pseudo-mask that occurs in source-to-target instance-aware mixing while target-to-source IMix eliminates it. Furthermore, in section 4, we discuss the datasets that we used in our different benchmark reports in the main paper. Moreover, in section 5, we explain the different evaluation metrics that we report numbers for in the tables of the paper. In section 6, we present additional qualitative results for the LIDAPS model for different instance classes and how they compare to the EDAPS results. In section 7, we present a plot bar that illustrates the quantative results for different methods on different benchmarks. In section 8, we provide the tables from the Ablation Section of the paper with standard deviation included. Lastly, in section 9, we address the limitations of our work.

2. Implementation Details

Hyper-parameter Settings: 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 afterwards a polynomial decay. Furthermore, we train for 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 (fine-tuning phase). Generally, we follow the hyper-parameter settings from EDAPS except the IMix confidence threshold and the CLIP loss weight.

Architecture: We use MiT-B5 [12] as our encoder backbone (shared by the instance and semantic decoders), MaskRCNN [3] as instance decoder and DAFormer [4] semantic head as the semantic decoder. For CLIP-based domain alignment, we use CLIP [7][1] as the pre-trained text encoder. We calculate the CLIP encodings of the categories only once before the start of training. During test time, these components are not needed and thus don't add any computational overhead.

Table 1. Hyperparameter study on the confidence-filtering threshold applied to the pseudo-masks for IMix.

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

IMix Threshold In Tab. 1, we conduct an hyperparameter study on the confidence-filtering threshold used for cross-domain instance mix-sampling. We evaluate a wide range of thresholds from 0 (no filtering) to 1 (disabling IMix). While we observe a local maxima at 0.75, we note that the method is relatively robust against this selection as it continues to outperform the baseline at a threshold of 0.5 with +1.9% mPQ and +6.4% mAP improvements.

Thus, we empirically set the IMix confidence threshold at 0.75 for the settings SYNTHIA → Cityscapes and Cityscapes → Cityscapes foggy while for SYNTHIA → Mapillary and Cityscapes → Mapillary we find that the best threshold is 0.9.

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 [2, 3, 8], nevertheless, we provide them for the sake of reproducibility and in order to explain the changes that occur to the ground truth supervision of some of these losses when training on IMix augmented images.

$$\mathcal{L}_{\text{pan}} = \mathcal{L}_{\text{sem}} + \mathcal{L}_{\text{inst}}. \tag{1}$$

As explained in the paper, Eq. 1, a panoptic loss function consists of two terms; an instance segmentation and a semantic segmentation loss term. Our instance decoder [3] consists of an RPN network and a refinement (Ref) network. Each part has its own losses as shown in Eq. 2.

$$\mathcal{L}_{\text{inst}} = \mathcal{L}^{\text{RPN}} + \mathcal{L}^{\text{Ref}} \tag{2}$$

The RPN loss function [8] has two terms, one for the "ob-

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

jectness" ($\mathcal{L}_{\text{Cls}}^{\text{RPN}}$) and another one for the bounding-box (or region proposal) regression ($\mathcal{L}_{\text{Box}}^{\text{RPN}}$) loss as seen in Eq. 3. The RPN takes a predefined set of anchor boxes and the convolution feature map (encoding the input image) as inputs and learns to correctly localize 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 $\mathcal{L}_{\text{Cls}}^{\text{RPN}}$ is a binary cross-entropy loss which is computed between the predicted objectness score $\hat{l}$ and the ground truth objectness label l. 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 $\mathcal{L}_{\text{Box}}^{\text{RPN}}$, an $L1$ loss is used which is computed between the predicted $(\hat{q})$ and ground truth (q) bounding-box coordinate offsets. Importantly, the regression loss is only computed for positive predicted boxes [8].

$$\mathcal{L}^{\text{RPN}} = \mathcal{L}_{\text{Cls}}^{\text{RPN}} + \mathcal{L}_{\text{Box}}^{\text{RPN}} \tag{3}$$

$$\mathcal{L}_{\text{Cls}}^{RPN} = L_{\text{BCE}}\left(\hat{l}, l\right) \tag{4}$$

$$\mathcal{L}_{\text{Box}}^{\text{RPN}} = \lambda_{RPN} \sum_{i \in x,y,w,r} L_1(\hat{q}_i, q_i) \tag{5}$$

We use $\mathbf{Q}$ to denote the set of ground truth bounding-box offset coordinates when training the student network. As explained in the main paper, LIDAPS is trained on both the source and mixed domain images containing target pseudo-instances. While training on the augmented images (output by IMix), $\mathbf{Q}$ represents a union set of the ground truth source bounding-boxes and confidence-filtered pseudo-bounding-boxes from a target image as shown in Eq. 6. Ground truth bounding-boxes of the source image are denoted by q^s, 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.

$$\mathbf{Q} = \begin{cases} \mathbf{Q}^s = \bigcup_i q_i^s & \text{if Source} \\ \mathbf{Q}^s \,\cup\, \bigcup_i \mathbb{1}[h_i > \tau]\; q_i^t & \text{if IMix} \end{cases} \tag{6}$$

The refinement network consists of a box-head and a mask-head following FastRCNN [2]. As seen in Eq. 7, the box-head is trained using a box classification loss $\mathcal{L}_{\text{Cls}}^{\text{Ref}}$ and a box regression loss $\mathcal{L}_{\text{Box}}^{\text{Ref}}$, while the mask-head has a mask segmentation loss $\mathcal{L}_{\text{Mask}}^{\text{Ref}}$.

$$\mathcal{L}^{\text{Ref}} = \mathcal{L}_{\text{Cls}}^{\text{Ref}} + \mathcal{L}_{\text{Box}}^{\text{Ref}} + \mathcal{L}_{\text{Mask}}^{\text{Ref}} \tag{7}$$

The box-head takes as inputs the RoIAlign [3] 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 $\mathcal{L}_{\text{Cls}}^{\text{Ref}}$ and a box regression loss $\mathcal{L}_{\text{Box}}^{\text{Ref}}$. The box classification loss is computed between predicted per-class probabilities P_{cl} and the ground truth class label $u \in \mathbf{U}$ for the predicted box as shown in Eq. 8. Unlike RPN, where the box classification loss is a binary cross-entropy loss, $\mathcal{L}_{\text{Cls}}^{\text{Ref}}$ is a categorical cross-entropy loss for multi-class classification.

$$\mathcal{L}_{\text{Cls}}^{\text{Ref}} = L_{CE}(P_{cl}, u) \tag{8}$$

The box regression loss is computed between the predicted bounding-box $\hat{v}_{u,i}$ and the ground truth bounding-box v_i as shown in Eq. 9. The predicted bounding-box $\hat{v}_{c,i}$ by the box-head is for the class $c \in C_{things}$. Having predictions for all classes mitigates the competition between the classes.

$$\mathcal{L}_{\text{Box}}^{\text{Ref}} = \lambda_{Ref} \sum_{i \in x,y,w,r} L1(\hat{v}_{u,i}, v_i) \tag{9}$$

Similar to RPN training, the box-head is trained on both source and target domain bounding-boxes $\mathbf{Q}$. While training on source images, we use the ground truth source bounding-boxes, and for training on augmented images (output by IMix), we use a union set of the ground truth source and pseudo bounding-boxes as in Eq. 6.

$\mathbf{U}$ denotes the ground-truth bounding-box class labels. When training the student network on the source domain images, we use the source ground-truth labels $\mathbf{U}^s$ and while training on the augmented images generated by IMix, $\mathbf{U}$ represents a union set of ground truth source bounding boxes and confidence-filtered pseudo bounding-box class labels as shown in Eq. 10.

$$\mathbf{U} = \begin{cases} \mathbf{U}^s = \bigcup_i u_i^s & \text{if Source} \\ \mathbf{U}^s \,\cup\, \bigcup_i \mathbb{1}[h_i > \tau]\; u_i^t & \text{if IMix} \end{cases} \tag{10}$$

For the RPN and box refinement head losses, we set the loss weights λ_{RPN} and λ_{Ref} to 1.0.

The mask-head predicts C_{things} 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 ground truth mask $m \in$ **Masks** according to maximum IoU. When training with IMix, **Masks** contains confidence-filtered pseudo-masks m^t from the target as well as ground truth masks from the source m^s as shown in Eq. 11.

$$\textbf{Masks} = \begin{cases} \textbf{Masks}^s = \bigcup_i m_i^s & \text{if Source} \\ \textbf{Masks}^s \, \cup \, \bigcup_i \mathbb{1}[h_i > \tau] \, m_i^t & \text{if IMix} \end{cases} \tag{11}$$

Eq. 12 indicates the binary cross-entropy loss computed between the predicted $\hat{m}$ and ground truth masks m, where $u \in C_{things}$ denotes the ground truth class label for the predicted mask.

$$\frac{1}{w \times h} \sum_{1 \le i,j \le h} m_{i,j} \log(\hat{m}_{u,i,j}) + (1 - m_{i,j}) \log(1 - \hat{m}_{u,i,j}). \tag{12}$$

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:

$$y^t_{\text{inst}_i} = \left[\arg\max_{c'} (\theta_{\text{inst}}(x^{(t)}))_i \right] \tag{13}$$

The semantic loss on the source domain is explained in Eq. 14 which defines a categorical cross-entropy loss on the predicted class probability for each pixel.

$$\mathcal{L}^s_{\text{sem}}(\hat{y}^s_{\text{sem}}, y^s_{\text{sem}}) = -\sum_{i,j,c} (y^s_{\text{sem}} \log(\hat{y}^s_{\text{sem}}))_{i,j,c} \tag{14}$$

Following [10], the self-supervised semantic loss applied to the semantic-aware mixed image [6] is shown in Eq. 15. The augmented or mixed image generated using the ClassMix [6] 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 ground truth 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. y^t_{sem} is the per-pixel pseudo-label as shown in Eq. 16 where θ_{sem}(the semantic decoder of the teacher) predicts per-pixel-class probabilities.

$$\mathcal{L}^{ss}_{sem}(\hat{\tilde{y}}_{\text{sem}}, \tilde{y}_{\text{sem}}) = \begin{cases} \mathcal{L}^s_{\text{sem}}(\hat{\tilde{y}}_{\text{sem}}, y^s_{\text{sem}}), \\ \qquad \text{if } \mathbf{M}^{(i,j,c)}_{\text{sem}} = 1, \\ -\sum k^t_{(i,j)} \left(y^t_{\text{sem}} \log(\hat{\tilde{y}}_{\text{sem}}) \right)_{(i,j,c)}, \\ \qquad \text{otherwise} \end{cases} \tag{15}$$

$$y^t_{\text{sem}} = \left[\arg\max_{c'} (\theta_{\text{sem}}(x^{(t)}))_{i,j} \right] \tag{16}$$

EDAPS*: This baseline follows the same setting as EDAPS [10] except that it does not include the features distance regularizor (FD) that EDAPS applies during training. 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 regularizor is explained in Eq. 17. Noteworthy is that FD is applied only on source images in areas corresponding to thing classes. In Table 3 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 2, we compare EDAPS with LIDPAS, both trained for 50k iterations. We can see that LIDAPS persists on beating EDAPS on three different benchmarks.

$$\mathcal{L}_{\text{FD}} = \|\text{Enc}_{\text{ImgNet}}(x^s) - \text{Enc}_\theta(x^s)\| \tag{17}$$

Table 2. 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 3. 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

3. False Negatives

As explained in the paper, when instance-aware mixing is done from source to target, exhaustive pseudo-masks for the target instances are not guaranteed. In Fig. 1, we show an example where in (c) confidence-filtered target instances are pasted onto the source image while in (d) all ground truth source instances are pasted on to the target image. In Fig. 1(c), we can see that the target instances all have a pseudo-mask while in Fig. 1(d), the encircled instance (the truck) in red does not have a corresponding pseudo-mask which is indicative of a false negative. When going from target to source, only the instances with a corresponding pseudo-mask are copy and pasted. Thus, inherently, all of the pasted target instances have a corresponding pseudo-mask. On the other hand, when remaining in the target image, target instances with absent pseudo-masks remain.

4. Datasets

We evaluate our method on the popular panoptic UDA benchmarks. For synthetic-to-real adaptation, we use SYNTHIA [9] as the source domain which contains 9,400 synthetic images. For the target domain, we use the Mapillary Vistas [5] dataset and Cityscapes [1]. 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 [11] as the target domain.

5. 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).

6. Additional Qualitative Results

In this section, we provide additional qualitative panoptic segmentation results in Fig. 2.

7. Additional Qualitative Results

In Fig. 3, we display the quantative results of LIDAPS and other UDA panoptic methods for different benchmarks on a bar plot.

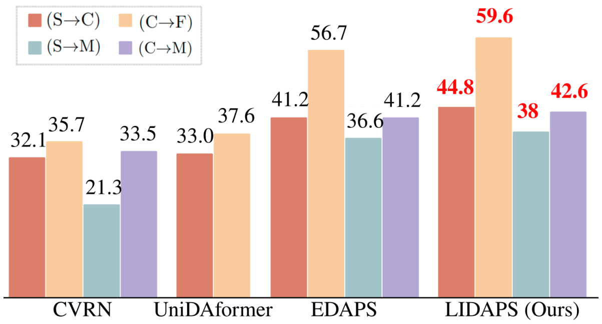


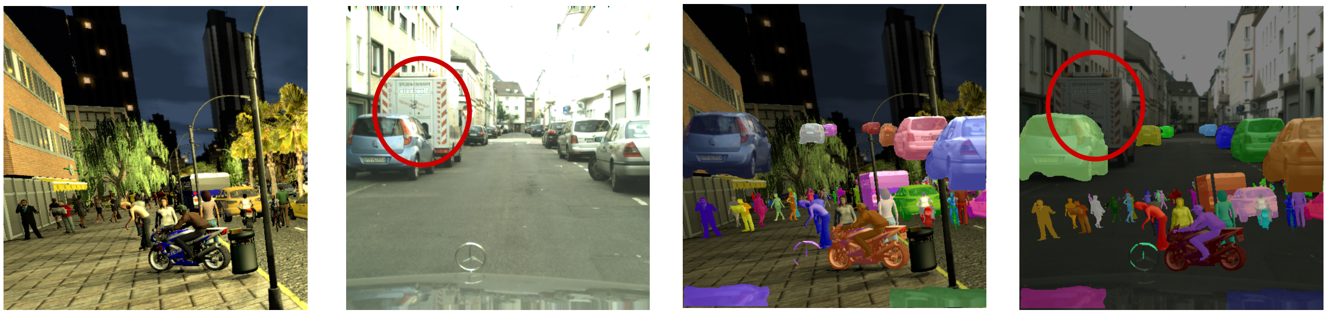
Figure 3. The two main contributions, IMix and CDA help in improving the UDA panoptic (mPQ) over the SOTA on four UDA panoptic segmentation benchmarks S→C: SYNTHIA to Cityscapes, C→F: Cityscapes to Foggy Cityscapes, S→M: SYNTHIA to Mapillary and C→M: Cityscapes to Mapillary.

8. Ablation studies including Standard Deviations

In this section, we provide Table 4 and Table 5 for the ablation studies of the main paper where we additionally include the standard deviation of the results each conducted for three rounds.

9. Limitations

Depending on the source and target domain, the threshold for pseudo-mask confidence filtering needs to be manually found with experiments. Thus, 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.



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

Figure 1. When using IMix to paste source instances from source to target (c), exhaustive pseudo-masks for the target instances is 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 supervised sets with no false negative examples.

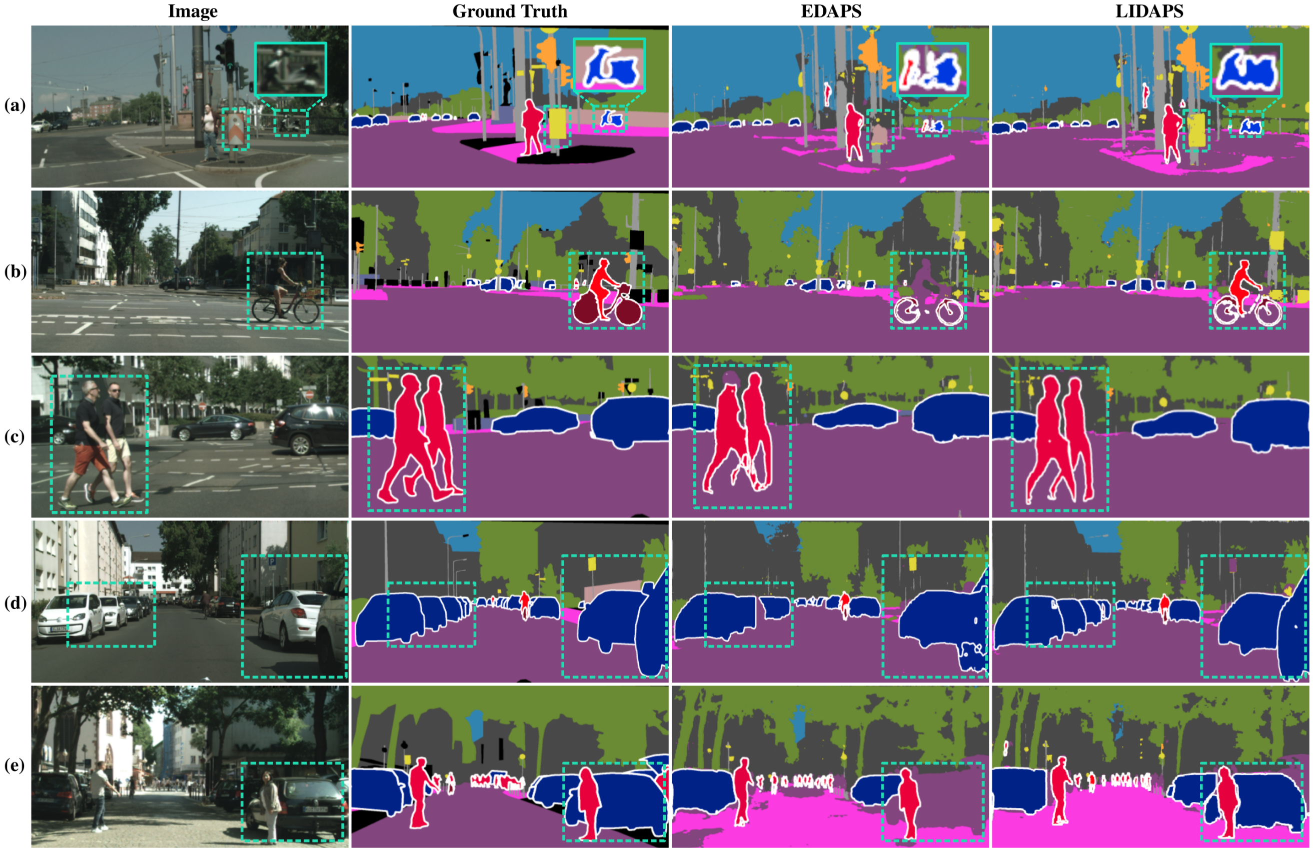


Figure 2. Additional qualitative results on SYNTHIA → Cityscape UDA benchmark comparing EDAPS [10] to our proposed LIDAPS. Our proposed LIDAPS model predicts improved semantic and instance segmentation for several classes including "motor-bike" (a), "rider" (b), "person" (c) and "car" (d,e).

Table 4. 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). Each experiment is run three times.

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**±$_{\mathbf{0.2}}$	**57.6**±$_{\mathbf{0.2}}$	**44.8**±$_{\mathbf{0.2}}$	**59.6**±$_{\mathbf{0.6}}$	**42.6**±$_{\mathbf{0.7}}$

Table 5. 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. Each experiment is run three times. S stands for source while T stands for target.

	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	S	T	62.0±$_{3.3}$	37.6±$_{0.6}$	29.3±$_{0.5}$	56.2±$_{0.7}$	1.9±$_{1.9}$
		T	S	**74.4±$_{0.2}$**	**57.6±$_{0.2}$**	**44.8±$_{0.2}$**	**59.6±$_{0.6}$**	**42.6±$_{1.7}$**
(ii)	+ ClassMix [6]	T	S	73.5±$_{0.2}$	53.9±$_{0.7}$	42.1±$_{0.6}$	58.6±$_{0.8}$	34.8±$_{0.9}$

References

[1] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016) 4

[2] Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015) 1, 2

[3] He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017) 1, 2

[4] Hoyer, L., Dai, D., Van Gool, L.: 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. pp. 9924–9935 (2022) 1

[5] Neuhold, G., Ollmann, T., Rota Bulo, S., Kontschieder, P.: The mapillary vistas dataset for semantic understanding of street scenes. In: Proceedings of the IEEE international conference on computer vision. pp. 4990–4999 (2017) 4

[6] Olsson, V., Tranheden, W., Pinto, J., Svensson, L.: Classmix: Segmentation-based data augmentation for semi-supervised learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1369–1378 (2021) 3, 6

[7] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021) 1

[8] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015) 1, 2

[9] Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: 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. pp. 3234–3243 (2016) 4

[10] Saha, S., Hoyer, L., Obukhov, A., Dai, D., Van Gool, L.: Edaps: Enhanced domain-adaptive panoptic segmentation. arXiv preprint arXiv:2304.14291 (2023) 3, 5

[11] Sakaridis, C., Dai, D., Hecker, S., Van Gool, L.: Model adaptation with synthetic and real data for semantic dense foggy scene understanding. In: Proceedings of the european conference on computer vision (ECCV). pp. 687–704 (2018) 4

[12] Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. Advances in Neural Information Processing Systems **34**, 12077–12090 (2021) 1