Supplementary material for
One-shot Unsupervised Domain Adaptation with Personalized Diffusion Models

Yasser Benigmim[1,2] Subhankar Roy[1] Slim Essid[1] Vicky Kalogeiton[2] Stéphane Lathuilière[1]
[1] LTCI, Télécom-Paris, Institut Polytechnique de Paris
[2] LIX, Ecole Polytechnique, CNRS, Institut Polytechnique de Paris
yasser.benigmim@telecom-paris.fr

The supplementary material is organized as follows: Sec. A reports additional experiments and ablation analysis of our proposed method. Sec. B provides additional implementation details. Sec. C presents the segmentations maps and then we conclude with a discussion about the broader impact of our work.

A. Additional experiments

Impact of number of shots on FID. We also explore the connection between the number of shots (#TS) and the photorealism of the generated target images using the Fréchet Inception Distance (FID) [3] score. The FID score measures how close are the generated images to the real target data distribution. Lower the FID score, closer are the two distributions. We plot the FID scores in Fig. A1, and we observe that Stable Diffusion (SD) has very high FID score, showing that the generated images have very little resemblance to the target domain Cityscapes. Low similarity with the target domain is also reflected in poorer performance, as shown in Fig. 4 of the main paper.

When compared with SD, the generations from DATUM are much closer to the real target domain, which is evident from the lower FID scores. We notice that when we fine-tune SD with fewer real target images, the FID score shows an upward trend as the number of training iterations increases. Whereas, as the #TS increases from 1 to 5, longer training leads to decreased FID score, up until the 800th interations. Finally, for the 10-shot setting, the FID score plateaus for a while and then starts going down after the 600th interations. All these observations are as per expectations, since having more real images necessitates longer training to fit to that data distribution.

Impact of prompting on class-wise IoU. Next we examine the impact of using *things* and *stuff* classes on the class-wise IoU scores. We report the results computed using DAFormer [5] on the GTA → Cityscapes benchmark in Tab. A1. We consider the DAFormer trained on a single real target image as the baseline, and the gain/loss attained by all the other methods are color coded. Warmer colors indicate gain, while cooler ones signify drops in performance. We compare the following methods: SD (using things class names during inference), DATUM (without things and stuff class names at inference), DATUM (using things and stuff class names at inference), DATUM (using things class names

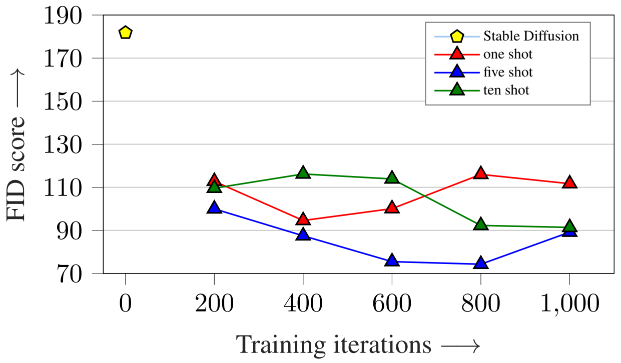

Figure A1. Impact of number of shots (#TS) on the FID score

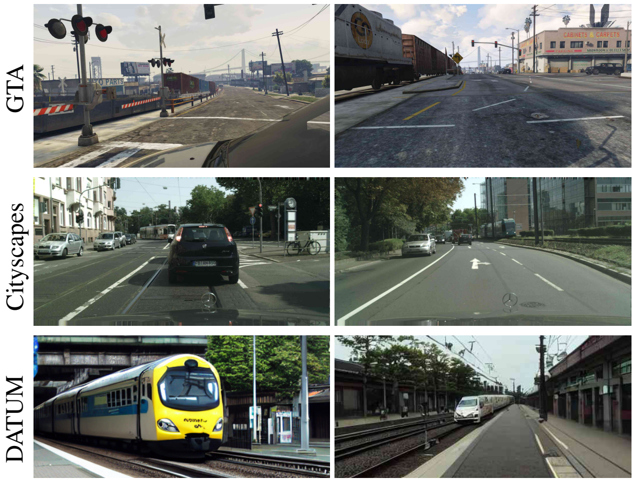

Figure A2. Real and synthetic images from the things class *train*

	Tr.Light	Sign	Person	Rider	Car	Truck	Bus	Train	M.bike	Bike	Road	S.walk	Build.	Wall	Fence	Pole	Veget.	Terrain	Sky
Real target	41.2	36.4	68.0	35.3	84.0	33.8	36.9	34.6	30.7	25.7	82.7	14.7	83.8	34.1	19.8	31.8	86.0	30.9	83.5
SD (things)	45.7	27.8	68.0	36.4	88.5	48.8	54.1	20.4	44.3	41.8	78.4	24.0	85.2	44.9	34.0	40.5	88.2	39.1	86.4
DATUM	43.8	47.4	67.8	36.2	87.7	47.0	46.2	42.2	37.4	31.1	86.6	28.3	85.0	38.7	22.2	44.7	87.6	40.3	85.1
DATUM (things & stuff)	48.3	44.0	68.6	38.4	90.2	55.0	63.8	23.3	46.7	55.0	85.9	29.7	87.1	38.2	40.0	44.4	88.7	42.5	86.9
DATUM (things) (w/ prior-loss)	48.1	46.4	67.9	37.6	87.2	52.3	50.4	27.4	48.3	48.8	86.4	22.0	86.1	42.5	25.6	45.9	88.4	41.9	87.6
DATUM (things) (w/o prior loss)	47.6	42.8	69.3	36.2	90.0	53.7	59.8	26.5	50.8	55.9	87.4	34.0	87.2	43.3	38.5	44.9	88.6	43.6	87.0

Table A1. Class-wise mIoU comparison for GTA → Cityscapes using MiT-B5 encoder. The left part of the table indicates th *things* classes, whereas the right part of the table indicates *stuff* classes. The color visualizes the IoU difference with respect to the first row, which is trained with the single target image.

at inference, and w/ prior-preservation loss [7]), and DATUM (using things class names at inference, and w/o prior-preservation loss), which is our final method.

We observe from Tab. A1 that using synthetic data, either with SD or our method brings improvements in a majority of the classes. Big improvements are noteworthy in the *things* classes (shown in the left half of Tab. A1). Interestingly, for some things classes, such as *person*, *rider* and *car*, the improvement with synthetic data is meagre. It could be potentially due to the fact that the source domain already encodes a strong prior about these objects, and additional data do not provide useful information.

Careful scrutiny of the table also reveals that there is a drop in the performance of the things class *train*. In an attempt to investigate this drop, we visualize in Fig. A2 the images annotated as *train* in GTA and Cityscapes, as well as synthetic images of *train* generated by DATUM. We observe an ambiguity in annotations for the *train* class in GTA and Cityscapes. While in GTA, the train image really corresponds to the vehicle of type "train", in Cityscapes one can reasonably recognize that the vehicle is actually a *tram*. Since, we utilize the class names of the source domain, our DATUM generates images with an object, *i.e.*, *train*, which is irrelevant to the target domain, despite both the vehicles exhibiting similar appearance.

B. Other Implementation details :

Data Augmentation. To enhance the robustness of the learned features and allow fair comparison, we adopt the identical set of data augmentation techniques as those employed in DAFormer [5]. The augmentation process entails applying a Random Crop of size 512 × 512 to both source and target images, followed by Random Flip with a 0.5 probability. Next, we employ the photometric distortion utilized in DACS [8], which comprises of a Gaussian Blur, Color Jittering, and ClassMix [6].

Personalization and generation. In the personalization stage, we employ the default DDPM [4] noise scheduler as in Dreambooth [7]. In the data generation stage, we also use the default parameters of Dreambooth [7]: 50 inference steps and a guidance scale of 7.5.

C. Qualitative visualizations

Finally, we show the qualitative results of the segmentation maps generated by our method and the comparison with other state-of-the-art methods in Fig. A3. Despite being trained on synthetic data, our DATUM is still able to capture several fine-grained details, especially the objects that appear far away from the camera. Note that, we do *not* make efforts to cherry pick the segmentation maps, and simply report our results for the same RGB input maps, as reported in CACDA [2].

Broader Impact

Although SD is adept at generating high-fidelity images of geometrically coherent scenes, sometimes the generations are gibberish and defy commonsense reasoning. As shown in Fig.3(d) of the main paper, the fine-tuned SD generates a very convincing-looking yet unintelligible "traffic sign", which has no meaning in a driving manual. Thus, to avoid model poisoning [1], the practitioners should exercise utmost caution when deploying segmentation models, for autonomous driving, that are trained using such synthetic datasets.

References

[1] Xiaoyu Cao and Neil Zhenqiang Gong. Mpaf: Model poisoning attacks to federated learning based on fake clients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3396–3404, 2022. 2

[2] Rui Gong, Qin Wang, Dengxin Dai, and Luc Van Gool. One-shot domain adaptive and generalizable semantic segmentation with class-aware cross-domain transformers. *arXiv preprint arXiv:2212.07292*, 2022. 2

[3] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium.

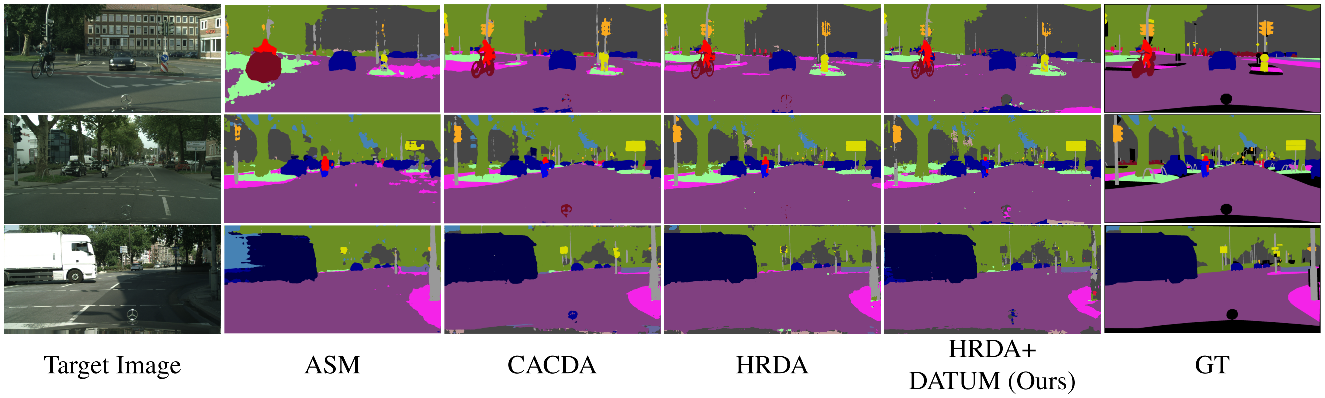

Figure A3. **Qualitative results of segmentation maps.** We compare the segmentation maps from different UDA methods on the GTA → Cityscapes benchmark.

Advances in neural information processing systems, 30, 2017. 1

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2

[5] 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. 1, 2

[6] 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. 2

[7] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[8] 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. 2