Reliable Spatial-Temporal Voxels For Multi-Modal Test-Time Adaptation

Haozhi Cao[1], Yuecong Xu[1,2], Jianfei Yang[1,3](✉), Pengyu Yin[1], Xingyu Ji[1], Shenghai Yuan[1], and Lihua Xie[1]

[1] School of Electrical and Electronic Engineering, Nanyang Technological University
{haozhi002,jianfei.yang,pengyu001,xinyu001,shyuan,elhxie}@ntu.edu.sg
[2] Department of Electrical and Computer Engineering, National University of Singapore
yc.xu@nus.edu.sg
[3] School of Mechanical and Aerospace Engineering, Nanyang Technological University

Abstract. In this appendix, we present more details about our experimental benchmarks, class-wise performance, and efficiency comparison, and more qualitative comparison with previous methods. **Firstly**, *benchmarks details* including class mapping and data preparation procedures are thoroughly illustrated. **Secondly**, we provide the detailed description of *implementation details and baseline settings* for all previous methods we compare Latte with. **Thirdly**, we illustrated the *class-wise performance and online efficiency* for further analysis. **Fourthly**, more *qualitative comparisons* with previous SOTA methods are illustrated to justify the effectiveness of Latte.

1 Benchmark Details

As mentioned in Sec. 4.1, we conduct our experiments on three different MM-TTA benchmarks, including USA-to-Singapore (U-to-S), A2D2-to-SemanticKITTI (A-to-S), and Synthia-to-SemanticKITTI (S-to-S). Here we provide more details about the included benchmarks.

1.1 USA-to-Singapore

For U-to-S, we adopt a similar setting of most previous MM-UDA methods [3, 8, 11], while the major difference lies in two points. Firstly, considering both source and target domains are derived from NuScenes [2] with the same class map, we discard the commonly used class mapping [3, 8] that alleviate the segmentation challenges and directly utilize the original semantic categories from NuScenes. Secondly, we release the restriction that only utilizes the points located in the front camera view and leverage points of full range for 3D networks pre-training and online updating instead. We modify this setting since SPVCNN [14] usually performs better when receiving the full range of point clouds. Furthermore, different from other MM-TTA methods, Latte can sometimes leverage the out-of-view points through spatial-temporal revisiting (*e.g.*, Latte can refer to points

Table 1: Class mapping of the benchmark A2D2-to-SemanticKITTI.

A-to-S Class	A2D2 classes	SemanticKITTI classes
car	Car 1-4, Ego car	car, moving-car
truck	Truck 1-3	truck, moving-truck
bike	Bicycle 1-4, Small vehicles 1-3	bicycle, motorcycle, bicyclist, motorcyclist, moving-bicyclist, moving-motorcyclist
person	Pedestrian 1-3	person, moving-person
road	RD normal street, Zebra crossing, Solid line, RD restricted area, Slow drive area, Drivable cobblestone, Dashed line, Painted driv. instr.	road, lane-marking
parking	Parking area	parking
sidewalk	Sidewalk, Curbstone	sidewalk
building	Buildings	building
nature	Nature object	vegetation, trunk, terrain
other-objects	Traffic signal 1-3, Traffic sign 1-3, Sidebars, Speed bumper, Irrelevant signs, Road blocks, Obstacles/trash, Animals, Signal corpus, Electronic traffic, Traffic guide obj, Grid structure, Poles	fence, traffic-sign, other-object

Table 2: Class mapping of the benchmark Synthia-to-SemanticKITTI. Class names in red contain zero pixels in Synthia [12]

S-to-S Class	Synthia classes	SemanticKITTI classes
car	car, bus, truck	car, moving-car, truck, moving-truck
bike	bicycle, motorcycle, rider	bicycle, motorcycle, bicyclist, motorcyclist, moving-bicyclist, moving-motorcyclist
person	pedestrian	person, moving-person
road	road, lanemarking, parking-slot	road, lane-marking, parking
sidewalk	sidewalk	sidewalk
building	building, wall	building
nature	vegetation, terrian	vegetation, trunk, terrain
pole	pole	pole
other-objects	fence, traffic-sign, traffic-light	fence, traffic-sign, other-object

out-of-view during a sharp turn). For MMTTA [13], we maintain their original methodology to utilize the points within the camera field-of-view, while other pseudo-label-based MM-TTA methods (including xMUDA+PL [8], PsLabel and Latte) utilize the 3D teacher predictions after class-wise median filtering as in [8] as the pseudo-labels for out-of-view points. Different from previous works based on SCN [6] which discards the point-wise features, we preserve such meaningful point-wise intensity as partial 3D input.

1.2 A2D2-to-SemanticKITTI

In terms of A-to-S, since the original class mapping is not identical across the source and target domain, we utilize the same class mapping as in [3, 8, 11], which is also detailed in Tab. 1. Considering A2D2 [5] only contains a limited

range of point clouds with ground-truth labels, we follow the same protocol of MMTTA [13], which utilizes the images from the front camera and points within the camera FOV for pre-training and online adapting. Similar to U-to-S, we preserve point-wise features (*i.e.*, reflectivity for A2D2 [5] and intensity for SemanticKITTI [1]) for 3D input, where different properties of these input features cause domain shift between the source and target domain in 3D predictions, causing the inferior performance of 3D source only predictions in Tab. 1.

1.3 Synthia-to-SemanticKITTI

The S-to-S benchmark is initially proposed in [13] to study a more challenging MM-TTA scenario with significant domain shifts in both 2D and 3D inputs. We attempt to follow the same setting as in [13], while some important details (*e.g.*, class mapping and downsampling strategies) are missing from the official benchmark description in their paper. To better facilitate the future exploration of challenging MM-TTA scenarios, we re-construct the S-to-S benchmark in this work, following the design details in [13] as much as we can. For class-mapping, as some semantic classes shared across S-to-S contain zero pixels in Synthia [12] (*i.e.*, those highlighted in red in Tab. 2), we merge them into other classes to avoid a redundant class mapping, resulting in a 9-class segmentation benchmark. On the other hand, instead of randomly downsampling the dense depth images of Synthia to sparse point clouds as in [13], we adopt a more vivid sampling strategy same as [4] to imitate the point cloud pattern recorded by sweeping LiDAR sensors. Since raw 3D input from Synthia does not contain point-wise features as the other two benchmarks, we empirically replace point-wise features with all-zero vectors for both source and target domain datasets.

2 Implementation Details and Baseline Settings

As mentioned in Sec. 4.1, we perform a parameter search for all baseline methods to obtain their best performance for a fair comparison. Specifically, the hyper-parameter search is conducted on U-to-S for all methods including Latte, starting from the initial parameter settings during the pre-training stage and the official settings of each baseline method. The optimal setting is subsequently applied to the other two benchmarks.

2.1 Implementation details

Universal settings. During the pre-training stage, most settings are identical to the default settings of backbones from both 2D and 3D modality. Specifically for 3D SPVCNN [14], we follow the pre-training settings of previous works [4, 7] to set the initial learning rate to 0.001 with an Adam optimizer. The learning rate is divided by 10 at 80k and 90k steps, while the total training iteration is set to 100k. The voxel size of SPVCNN is set to 0.05m for all benchmarks. All methods utilize a batch size of 6 by default.

In terms of 2D SegFormer [18], most settings are universal across three benchmarks, where the initial learning rate is set to 6E-5 with an AdamW optimizer using the same configuration as in [18]. A Poly scheduler is additionally applied to SegFormer to gradually decrease its learning rate and the total pre-training iteration is also set to 100k. For all pre-training, SegFormer is initialized with the parameters pre-trained on ADE20K [19] with an input resolution of 512×512. Considering different datasets have different image resolutions, we slightly alter the size augmentations for each benchmark to fully leverage the potential of all methods. Specifically, for U-to-S, images from the source domain dataset are firstly resized to 800×450 and then randomly cropped from the bottom to 450×450, while the bottom crop is discarded during adaptation and the whole image is inferred in a slide-window manner with a window size of 450×450 and a stride of 300×300. For A-to-S, the image augmentation on the source domain dataset follows the same setting as in [7] and the inference protocols are identical to U-to-S. For S-to-S, we change to resize the image into a shape of 640×380 followed by a bottom crop of 350×350 on the source domain dataset. The slide window size and stride during inference are changed to 350×350 and 230×230, respectively. During the online adaptation of all methods, we first reset the parameters of all normalization layers before adaptation. For methods based on pseudo-label training, the class weights from the labeled source domain are introduced to alleviate the class-imbalanced issue. Note that all Dropout layers of both student and teacher models are disabled during the adaptation process. For pose generation, we utilize the default settings of KISS-ICP [15] for each dataset.

2.2 Baseline Specific Settings

TENT [16]. Since TENT does not contain any hyper-parameters for tunning, we mainly adjust the learning rate of 2D and 3D backbones. Specifically, the performance of TENT on U-to-S peaks when 2D and 3D learning rates are set to 6E-8 and 1E-4, respectively. We therefore utilize the same learning rate settings for the other two benchmarks. In fact, most methods minimizing prediction entropy favor smaller learning rates compared to pseudo-label-based methods.
ETA [9]. There exist two hyper-parameters in ETA, including entropy threshold E_0 and similarity threshold ϵ. Specifically, we found a combination of the default $E_0 = 0.4 \times \ln 10^3$ and $\epsilon = 0.005$ performs the best, while the optimal 2D and 3D learning rates are 6E-7 and 1E-3, respectively. It is worth mentioning that ETA requires more GPU memory compared to other methods and we alter its batch size from 6 to 4 so that it can fit in a single RTX 3090.
SAR and SAR-rs [10]. Both SAR and SAR-rs perform similarly across different parameter settings, except for the 2D and 3D learning rates, where we found the 2D and 3D learning rates of 6E-7 and 1E-3, respectively, achieve the best performance.
MMTTA [13]. We mainly tune the 2D and 3D learning rate for MMTTA to achieve its best performance for a fair comparison, where we found a 2D and 3D learning rate of 6E-6 and 1E-3 as its optimal settings, respectively.

Table 3: Class-wise performance and efficiency comparison on S-to-S. Note that CoTTA [17] refers to its original version while CoTTA* is the variant that updates only the parameters of normalization layers. "MM" and "MF" indicate whether the method contains multi-modal or multi-frame learning, respectively. Here we report the cross-modal prediction score for all methods. All inference time is evaluated with an i7-12700 and an RTX 3090.

Methods	Publication	MM	MF	Inf. Time (ms)	*Car*	*Bike*	*Person*	*Road*	*Sidewalk*	*Building*	*Nature*	*Pole*	*O. Object*	mIoU
Source only	-	-	-	71.5	73.6	27.1	7.3	73.1	36.4	46.4	55.6	23.6	0.6	38.2
Oracle TTA	-	-	-	71.5	79.1	32.0	18.4	86.0	57.3	69.4	77.4	33.0	39.2	54.6
TENT [16]	ICML-21	✗	✗	147.5	**73.4**	**31.5**	10.1	**78.0**	**38.7**	37.2	47.9	20.1	0.4	37.5
ETA [9]	ICML-22	✗	✗	103.8	65.0	13.9	3.1	67.7	33.4	40.6	45.6	25.2	**1.6**	32.9
SAR [10]	ICLR-23	✗	✗	180.3	70.2	17.8	5.1	71.8	35.9	39.2	45.2	10.2	0.5	32.9
SAR-rs [10]	ICLR-23	✗	✗	240.6	65.4	14.1	3.1	68.8	33.8	39.2	43.2	25.3	1.5	32.7
xMUDA [8]	PAMI-22	✓	✗	215.6	20.4	2.7	0.4	22.7	15.5	21.5	36.2	6.8	0.8	14.1
xMUDA+PL [8]	PAMI-22	✓	✗	215.9	67.0	26.5	0.3	52.4	37.0	56.0	**61.9**	25.5	0.2	36.3
MMTTA [13]	CVPR-22	✓	✗	197.3	65.7	20.1	8.4	51.5	26.6	**58.4**	61.4	27.0	0.2	35.5
PsLabel	-	✓	✗	193.1	71.0	27.0	0.6	67.4	41.2	40.4	47.2	22.8	0.5	35.3
CoTTA [17]	CVPR-22	✗	✓	406.6	68.0	14.6	3.4	69.5	34.2	40.4	45.2	26.6	**1.6**	33.7
CoTTA* [17]	CVPR-22	✗	✓	327.3	67.9	14.8	3.5	69.2	34.1	40.5	45.5	27.0	1.5	33.8
Latte (ours)	-	✓	✓	270.9	72.6	27.7	**11.6**	74.5	37.9	56.4	**61.9**	**31.4**	0.2	**41.6**

CoTTA [17]. CoTTA is a multi-frame baseline included in this appendix as an additional comparison for the multi-frame efficiency of Latte. Here we include two versions of CoTTA, including the original CoTTA (CoTTA) that updates all parameters and its variant (CoTTA*) which only updates the parameters of normalization layers as other methods. Both versions utilize a confidence threshold of 0.9 and the same 2D and 3D learning rate of 6E-5 and 1E-3. CoTTA* utilizes a batch size of 6, while the batch size of CoTTA is decreased from 6 to 4 due to its high occupancy of GPU memory.

xMUDA [8], xMUDA+PL [8], PsLabel, and Latte. For the remaining methods, the 2D and 3D learning rates are set to 6E-5 and 1E-3, respectively, while the cross-modal coefficient for xMUDA and xMUDA+PL is set to 0.1.

3 Class-wise Performance and Efficiency Comparison

To provide a more detailed performance and efficiency comparison between Latte and previous methods, we include additional results of class-wise performance and per-frame processing time on the challenging S-to-S benchmark. As shown in Tab. 3, Latte can achieve consistent improvement on most semantic classes compared with previous state-of-the-art MM-TTA methods (*i.e.*, "✓" in the "MM" column), where Latte significantly outperforms MMTTA [13] by relatively 17.2%. Compared with TTA methods, Latte achieves a more balanced class-wise performance compared to TENT [16], achieving a relative improvement of 10.9%.

In terms of computational cost, we additionally compare our Latte with the recent CoTTA [17] which stabilizes the noisy single-frame predictions by averaging predictions from various augmented frames. Compared to the original

Table 4: Supplementary comparison between Latte and combining existing TTA methods with cross-modal interaction. Here † denotes the Latte variant with window size set to 1 while "xM" stands for the cross-modal scheme from xMUDA [8].

Method	Publication	U-to-S			A-to-S			S-to-S			
		2D	3D	xM	2D	3D	xM	2D	3D	xM	Avg
TENT+xM	ICML-21	37.1	39.7	45.2	43.3	47.1	50.3	23.9	36.5	37.7	44.4
ETA+xM	ICML-22	36.9	35.2	44.8	43.1	44.2	50.1	23.7	30.8	33.1	42.7
SAR+xM	ICLR-23	36.9	39.4	45.5	43.1	46.3	50.5	22.8	32.8	34.5	43.5
SAR-rs+xM	ICLR-23	36.9	36.8	45.0	43.1	45.6	50.5	23.8	30.7	33.1	42.9
Latte	-	**37.4**	**41.0**	**46.0**	**46.1**	**52.6**	**54.3**	**33.2**	**39.3**	**41.6**	**47.3**

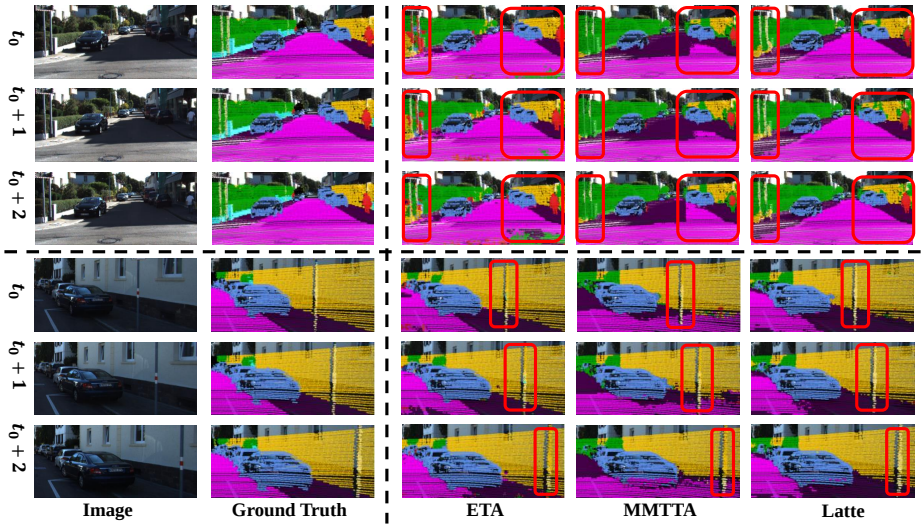

Fig. 1: Qualitative comparison with ETA [9] and MMTTA [13] on S-to-S. Red boxes highlight the area where Latte produces more accurate predictions compared to ETA and MMTTA. Figure best viewed in color and zoomed in.

CoTTA and its efficient variant CoTTA*, Latte established a much more computationally efficient and effective strategy to mitigate the single-frame instability, reducing the inference time relatively by 33.4% and 17.4% while achieving significant improvements by 23.4% and 23.1%, respectively. This justifies both the efficiency and effectiveness of Latte compared to previous multi-frame methods.

4 Comparision with TTA Methods with Multi-Modal Interactions

Due to the fact that the pure single-modal learning scheme may severely hinder the performance of existing TTA methods, we thus supplement the results with existing TTA methods combined with a widely used cross-modal learning scheme

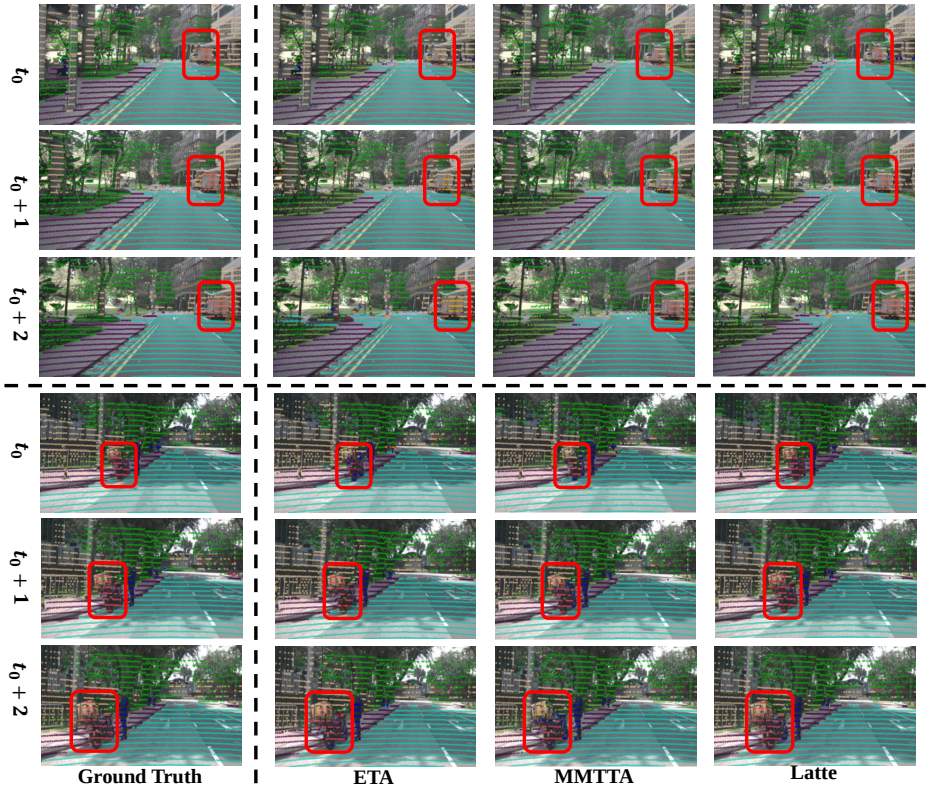

Fig. 2: Qualitative comparison with ETA [9] and MMTTA [13] on U-to-S. Red boxes highlight the area where Latte produces more accurate predictions compared to ETA and MMTTA. Figure best viewed in color and zoomed in.

xMUDA [8] as in Tab. 4. Specifically, the cross-modal prediction consistency loss from [16] is included for each TTA method as their additional optimization objective with a coefficient of 0.1. Although the cross-modal learning scheme brings an average relative improvement of 1.0-4.0%, the performance gap between existing TTA methods and Latte is still non-trivial, which justifies the superiority of our cross-modal learning scheme in Latte. We have included this discussion in the updated version.

5 Qualitative Comparision with Previous Methods

To demonstrate the improvement brought by Latte, we provide some additional qualitative comparison between Latte and previous SOTA TTA (ETA [9]) and MM-TTA (MMTTA [13]) methods on U-to-S and S-to-S. As shown in Fig. 1 and Fig. 2, the cross-modal predictions from Latte are more accurate compared to ETA and MMTTA (*e.g.*, more accurate pole recognition on the lower set

of consecutive frames in Fig. 1). Furthermore, the predictions from Latte are more consistent across time. For instance, the pedestrians and poles in the red rectangles in Fig. 1 as well as the motorcycle and the truck in Fig. 2 can be consistently recognized by Latte, while both ETA and MMTTA suffer from the instability of single-frame predictions. This justifies the effectiveness of Latte and the improvement brought by our multi-frame aggregation strategy.

References

1. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: Semantickitti: A dataset for semantic scene understanding of lidar sequences. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9297–9307 (2019)
2. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11621–11631 (2020)
3. Cao, H., Xu, Y., Yang, J., Yin, P., Yuan, S., Xie, L.: Mopa: Multi-modal prior aided domain adaptation for 3d semantic segmentation. arXiv preprint arXiv:2309.11839 (2023)
4. Cao, H., Xu, Y., Yang, J., Yin, P., Yuan, S., Xie, L.: Multi-modal continual test-time adaptation for 3d semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 18809–18819 (October 2023)
5. Geyer, J., Kassahun, Y., Mahmudi, M., Ricou, X., Durgesh, R., Chung, A.S., Hauswald, L., Pham, V.H., Mühlegg, M., Dorn, S., et al.: A2d2: Audi autonomous driving dataset. arXiv preprint arXiv:2004.06320 (2020)
6. Graham, B.: Sparse 3d convolutional neural networks. arXiv preprint arXiv:1505.02890 (2015)
7. Jaritz, M., Vu, T.H., Charette, R.d., Wirbel, E., Perez, P.: xmuda: Cross-modal unsupervised domain adaptation for 3d semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (June 2020)
8. Jaritz, M., Vu, T.H., De Charette, R., Wirbel, É., Pérez, P.: Cross-modal learning for domain adaptation in 3d semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(2), 1533–1544 (2022)
9. Niu, S., Wu, J., Zhang, Y., Chen, Y., Zheng, S., Zhao, P., Tan, M.: Efficient test-time model adaptation without forgetting. In: International Conference on Machine Learning. pp. 16888–16905. PMLR (2022)
10. Niu, S., Wu, J., Zhang, Y., Wen, Z., Chen, Y., Zhao, P., Tan, M.: Towards stable test-time adaptation in dynamic wild world. arXiv preprint arXiv:2302.12400 (2023)
11. Peng, D., Lei, Y., Li, W., Zhang, P., Guo, Y.: Sparse-to-dense feature matching: Intra and inter domain cross-modal learning in domain adaptation for 3d semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7108–7117 (2021)
12. 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)

13. Shin, I., Tsai, Y.H., Zhuang, B., Schulter, S., Liu, B., Garg, S., Kweon, I.S., Yoon, K.J.: Mm-tta: multi-modal test-time adaptation for 3d semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16928–16937 (2022)
14. Tang, H., Liu, Z., Zhao, S., Lin, Y., Lin, J., Wang, H., Han, S.: Searching efficient 3d architectures with sparse point-voxel convolution. In: European Conference on Computer Vision. pp. 685–702. Springer (2020)
15. Vizzo, I., Guadagnino, T., Mersch, B., Wiesmann, L., Behley, J., Stachniss, C.: Kiss-icp: In defense of point-to-point icp–simple, accurate, and robust registration if done the right way. IEEE Robotics and Automation Letters **8**(2), 1029–1036 (2023)
16. Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: Fully test-time adaptation by entropy minimization. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=uXl3bZLkr3c
17. Wang, Q., Fink, O., Van Gool, L., Dai, D.: Continual test-time domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7201–7211 (2022)
18. 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)
19. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. International Journal of Computer Vision **127**(3), 302–321 (2019)