Continual Learning of Semantic Segmentation using Complementary 2D-3D Data Representations

Jonas Frey
ETH Zurich, Switzerland
jonfrey@ethz.ch

Hermann Blum
ETH Zurich, Switzerland
blumh@ethz.ch

Francesco Milano
ETH Zurich, Switzerland
fmilano@ethz.ch

Roland Siegwart
ETH Zurich, Switzerland
rsiegwart@ethz.ch

Cesar Cadena
ETH Zurich, Switzerland
cesarc@ethz.ch

Abstract

Semantic segmentation networks are usually pre-trained and not updated during deployment. As a consequence, misclassifications commonly occur if the distribution of the training data deviates from the one encountered during the robot's operation. We propose to mitigate this problem by adapting the neural network to the robot's environment during deployment, without any need for external supervision. Leveraging complementary data representations, we generate a supervision signal, by probabilistically accumulating consecutive 2D semantic predictions in a volumetric 3D map. We then retrain the network on renderings of the accumulated semantic map, effectively resolving ambiguities and enforcing multi-view consistency through the 3D representation. To preserve the previously-learned knowledge while performing network adaptation, we employ a continual learning strategy based on experience replay. Through extensive experimental evaluation, we show successful adaptation to real-world indoor scenes both on the ScanNet dataset and on in-house data recorded with an RGB-D sensor. Our method increases the segmentation performance on average by 11.8% compared to the fixed pre-trained neural network, while effectively retaining knowledge from the pre-training dataset.

1 INTRODUCTION

Acquisition of large datasets for training neural networks is costly, time-intensive, and error-prone. Furthermore, a dataset captured at a fixed point in time cannot fully cover every possible data point in the future for complex tasks in unknown environments, leading to a distribution mismatch between the available labeled dataset and the actual data of interest. While this calls for the need of performing network adaptation in new environments, a straightforward adaptation is hindered by the lack of a ground-truth supervision signal during deployment.

Fortunately, mobile robots can usually observe the same area of their deployment environment from different viewpoints, which thus potentially provides the means to resolve ambiguities in the task of semantic segmentation. Misclassifications of small, occluded, or partially observed objects may be corrected in consecutive frames with different viewpoints. We explicitly leverage multi-view consistency by fusing individual 2D semantic predictions into a 3D map and generate a new training signal for the network by reprojecting the fused semantic information back into 2D pseudo-labels.

To mitigate catastrophic forgetting [17] we use an experience replay continual learning strategy, which regularizes the training procedure, to adapt the neural network according to the generated pseudo-labels. To the best of our knowledge, we are the first to propose adapting a neural network

NeurIPS 2021 Workshop on Robot Learning: Self-Supervised and Lifelong Learning, Virtual, Virtual

according to a supervision signal generated by explicitly transforming network predictions between 2D and 3D. Through extensive experimental evaluation, we show that the multi-view consistency enforced by this supervision signal allows the network to reliably increase its accuracy with respect to a fixed network. Contrary to the other methods that explore semantic segmentation in a continual learning setting, our approach is suited for online deployment, exploits a 3D representation to enforce multi-view consistency, and only requires an RGB-D sensor and the associated camera poses. We evaluate our method on the indoor, ScanNet [8] dataset, showing a remarkable increase of the semantic segmentation performance on average by 11.8% relative to the static network. Additionally, we provide qualitative deployment experiments with a handheld RGB-D sensor.

2 RELATED WORK

Prior work focuses on accumulation of 2D semantic network predictions in a global 3D representation to enhance the semantic predictions [18, 12, 25, 3]. However, they rely on a fixed network which is not updated with the semantic information collected in the generated map. A limited number of works has explored semantic segmentation in continual learning [20, 30, 6, 10, 9, 19]. In contrast to our work, these approaches are not designed for online scenarios, tackle segmentation purely in 2D on a per-frame basis, and rely on the availability of ground-truth supervision. Recently, multiple works showed successful online adaptation on robot systems. [4] use continual learning strategies to adapt the binary segmentation and localization capability of a construction robot. [28] use reinforcement learning to fully autonomously learn navigation and manipulation on a robotic system. Additionally, [13] show the effectiveness of enforcing view consistency in 3D as a prior for 2D representations.

3 APPROACH

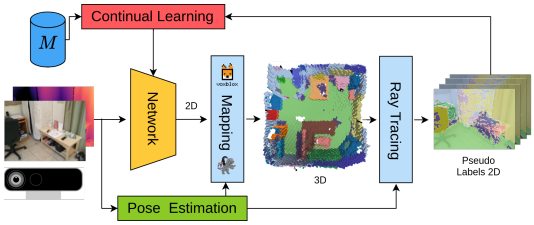

Figure 1: Overview of the architecture.

The architecture of our approach is illustrated in Figure 1. Images from an RGB-D camera are provided to a segmentation network (yellow). The pose estimation (green) additionally utilizes the depth information to estimate camera poses. Individual 2D semantic estimates are accumulated in a 3D voxel map, which is ray traced to create 2D pseudo-labels, as explained in Section 3.1. These pseudo-labels are used to adapt the network using an experience replay continual learning strategy (red), which can access previously stored samples in a memory buffer (dark blue) to regularize the training, as explained in Section 3.2. The continual learning strategy trades-off adaptation to the new data and preserving previously learned (stability-plasticity dilemma [1]).

3.1 Pseudo-Label Generation

A pre-trained semantic segmentation network f_θ predicts initial semantic estimates Y_n^{pred} from a provided video sequence consisting of individual key frames I_n, where n denotes the index in the sequence of length N. We create a dense semantic map of the robot's environment using a voxel-based truncated signed distance function (TSDF). In addition to the TSDF volume, a semantic voxel volume stores the probability of each voxel belonging to a semantic class. The camera extrinsics H_n, and depth map D_n are used to integrate the predicted semantics Y_n^{pred} into both volumes. The TSDF is calculated following [22]. For each voxel close to the TSDF surface, the semantic label probability is updated following recursive Bayesian estimation [25]. The mapping procedure is performed for each camera trajectory within a scene individually. After integration of all N measurements, Marching Cubes [16] is used to estimate a high-resolution mesh. We ray trace the mesh for each camera pose H_n to determine for each pixel in the camera plane the first intersection of the corresponding ray with the mesh. Each of the resulting 3D locations is then used to index the semantic voxel volume in $O(1)$ time to retrieve the semantic label probabilities for the associated pixel. We refer to the resulting re-projected semantic segmentation label as Y_n^{pseudo} as *pseudo-label*. The pseudo-labels aggregate information from multiple viewpoints and enforces multi-view consistency. At the same time, gathering information from multiple viewpoints allows filtering out semantic segmentation errors induced by bad lighting, motion blur, and outlier predictions in individual frames. This allows

us to generate a learning signal of higher accuracy that can be used to adapt the network in the absence of ground-truth supervision. In the Appendix, we provide further details for the network and dataset (Appx. A.1), pre-training (Appx. A.2), and pseudo-label generation (Appx. A.3).

3.2 Continual Learning

While it would be possible to directly replace the single-frame prediction of the segmentation network with the pseudo-labels, we instead use the pseudo-labels to train the network and adapt it to the scene. This has two reasons. First, the accuracy gain of the pseudo-labels cannot be transferred to a different environment, since the map is bound to the geometry of the scene. The (adapted) network on the other hand has the ability to transfer the gained knowledge to any future frame from any environment. Second, the network training has the potential to filter out undesirable artifacts from the voxel rendering (Figure 4). Finally, as the pseudo-labels themselves require sufficient prediction accuracy of the network, improving the prediction accuracy is beneficial for all similar environments. For training, we one-hot encode the pseudo-labels according to the most likely class per pixel. This experimentally outperformed probabilistic pseudo-labels and reduces storage and computation needed. To update the model parameters θ we use an experience replay strategy, which has shown promising results in prior works [5, 24, 4]. For this, a small subset of N_M randomly selected samples of pre-training dataset is stored in a memory buffer M, which can be accessed to *replay* samples (i.e., feed them again to the network) when adapting the parameters θ to the current scene. The standard cross-entropy loss l_{CE} is used for both the new samples annotated with the pseudo-labels and the replayed samples with ground-truth annotations. Stochastic gradient descent (SGD) is used to optimize the objective. We can explicitly distinguish between the loss induced by samples stored in the memory buffer and the pseudo-labels in the SGD update:

$$\theta_{t+1} = \theta_t - \frac{\mu}{n_{\text{pseudo}} + n_{\text{rep}}} \frac{d}{d\theta} \left(\sum_{i=1}^{n_{\text{pseudo}}} l_{\text{CE}}(f_\theta(x_i), y_i) + \sum_{j=1}^{n_{\text{rep}}} l_{\text{CE}}(f_\theta(x_j), y_j) \right), \tag{1}$$

where μ, n_{rep} and n_{pseudo} denote the learning rate, number of replayed and pseudo-labels samples respectively. On one side, minimizing the loss of the replayed samples motivates preservation of previously learned information: the diversity of the samples in the memory buffer favors generalization and mitigates overfitting to the small pseudo-label dataset. On the other side, the loss of the pseudo-labels encourages the learning of new knowledge. Additionally, since the samples in the memory buffer are annotated with ground-truth labels, common patterns of the ground-truth annotations may be transferred to the pseudo-labeled samples and act as a regularization mechanism. Finally, storing a subset of N_M samples significantly reduces the memory needed with respect to the full pre-training dataset size N_{pre} and proven successful even for small N_M [7]. The chosen strategy similar to [5] performs well, is simple, and needs less compute compared to more complex experience replay strategies, which e.g. explicitly utilize the gradient induced by stored samples [7, 2, 11] or are designed for online learning [15]. The detailed training routine is elaborated in Appendix A.4.

4 EXPERIMENTS

The **pseudo-label generation procedure** is evaluated by comparing the segmentation accuracy achieved by the pre-trained network to the resulting pseudo-labels. An example segmentation for the pre-trained network, pseudo-label generation, and continually trained network is provided in Figure 4. The pseudo-labels include minor artifacts induced by the voxel discretization and the ray tracing process, but are consistent over multiple viewpoints. Moreover, these artifacts are not reflected in the adapted network predictions and do not prevent the signal from being beneficial for improving the prediction accuracy. In the Appendix B.1 (Fig. 5) we provide four additional example frames for the ScanNet dataset which illustrate disagreeing pre-trained network predictions for the same location over consecutive frames. This highlights the need for multi-view consistent adaptation. We compute an upper bound for the pseudo-label generation procedure by using the ground-truth segmentation to generate pseudo-labels. As expected, given the artifacts contained in the pseudo-labels the accuracy slightly decrease for all scenes as illustrated in Table 1, but still results in a nearly perfect estimate of 92.5% accuracy. As shown in Table 1, the pseudo-labels (1-Pseudo, AVG 57.5%) generated based on the pre-trained network predictions (1-Pred, AVG 50.9%) improve the accuracy for all scenes with an average gain of 6.6% (relative 11.8%). We illustrate the reconstructed

Scene	1-Pred		1-Pseudo		2-Finetune		2-Pred (CL)	
	Gen	Adap	Adap	GT	Gen	Adap	Gen	Adap
1	43.0	60.5	70.2	94.3	36.1	71.7	37.6	71.2
2	43.0	48.5	53.7	94.9	35.7	51.6	37.2	52.9
3	43.0	30.5	38.0	88.7	34.7	36.7	36.3	36.8
4	43.0	74.6	81.6	96.0	27.3	81.3	31.8	80.6
5	43.0	40.3	44.0	88.7	24.5	41.0	32.4	42.7
AVG	43.0	50.9	57.5	92.5	31.7	56.7	35.1	56.9

Table 1: Segmentation accuracy of pre-trained (1-Pred), fine-tuned (2-Finetune) and continually-learned (2-Pred (CL)) network, as well as pseudo-labels (1-Pseudo) and their upper bound (GT). We measure Adap*tation* performance on the novel scenes 1-5 of ScanNet and Gen*eralization* performance on the pre-training test dataset.

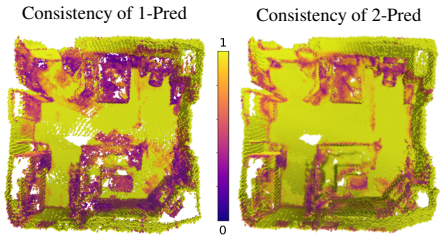

Figure 2: Multi-view consistency per-voxel confidence for the first ScanNet scene. Left: Confidence mapping the pre-trained network predictions (1-Pred). Right: Confidence mapping the adapted network prediction in the second iteration (2-Pred).

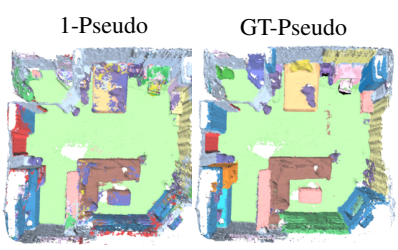

Figure 3: Mapping results for the first ScanNet scene. Color-coded semantic classes using the ScanNet color schema.

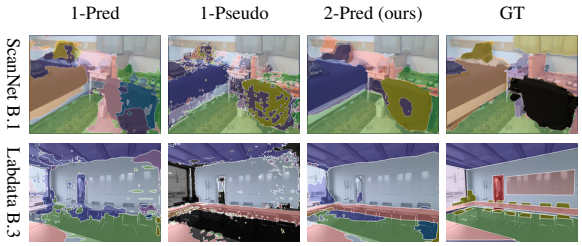

Figure 4: Comparison of pre-trained network (1-Pred), generated pseudo-label (1-Pseudo), adapted trained network (2-Pred) and ground-truth label (GT).

mesh used to generate 1-Pseudo and GT-Pseudo in Figure 3. All objects can be clearly identified in the GT-Pseudo map. In 1-Pseudo only few misclassifications (*desk*, top right; *toilet* top left) and artifacts (*sofa* bottom middle, *bed* top middle) cannot be resolved. We conclude that the pseudo-label generation process significantly improves the accuracy of the semantic segmentation and therefore is suitable as a learning signal to adapt the network.

The **continual learning strategy** is compared with a naïve fine-tuning approach. To evaluate the generalization performance we measure the accuracy on the test set of the pre-training dataset and calculate the adaption performance using the accuracy on scene 1-5 of the ScanNet dataset as explained in the Appendix A. As shown in Table 1, the experience replay strategy outperforms fine-tuning for all scenes on the pre-training dataset ($+3.4\%$). Moreover, the average accuracy on the adaptation scene is slightly higher ($+0.2\%$) than finetuning. This strongly suggests that the replay of highly accurate ground-truth data does not prohibit adaptation to the pseudo-labels. The predicted semantic segmentation of the continually-learned network for the selected key-frames of the first scene are illustrated in Figure 4 and Appendix B.1. As evident in the column (2-Pred), the predictions align with the pseudo-labels but filter out noise and artifacts, resulting in smooth boundary regions. We evaluate the multi-view consistency of the network predictions before and after adaptation. When integrating disagreeing network predictions in the same voxel, the uncertainty of the specific voxel increases. Figure 2 shows these voxel uncertainties after integrating predictions from the pre-training network (1-Pred) and from the adapted, continually-learned network (2-Pred). Clearly, our adaptation procedure increases the certainty and therefore multi-view consistency of 2-Pred over 1-Pred.

5 CONCLUSION

We showed that leveraging complementary 2D-3D data representations creates a useful learning signal for semantic segmentation without any external supervision. We further show the benefits of applying a continual learning strategy to adapt the multi-class semantic segmentation network in a robotic mission scenario. To the best of our knowledge, this is the first ready-to-deploy domain adaptation approach that does not require prior knowledge of the scene or any external supervision and can simultaneously retain knowledge of previously seen environments.

References

[1] Wickliffe C. Abraham and Anthony Robins. Memory retention – the synaptic stability versus plasticity dilemma. *Trends in Neurosciences*, 28(2):73–78, 2005. ISSN 0166-2236. doi: https://doi.org/10.1016/j.tins.2004.12.003. URL https://www.sciencedirect.com/science/article/pii/S0166223604003704.

[2] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *arXiv preprint arXiv:1903.08671*, 2019.

[3] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir Roshan Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3D Scene Graph: A Structure for Unified Semantics, 3D Space, and Camera. *CoRR*, abs/1910.02527, 2019. URL http://arxiv.org/abs/1910.02527.

[4] Hermann Blum, Francesco Milano, René Zurbrügg, Roland Siegwart, Cesar Cadena, and Abel Gawel. Self-Improving Semantic Perception on a Construction Robot. *CoRR*, abs/2105.01595, 2021. URL https://arxiv.org/abs/2105.01595.

[5] Pietro Buzzega, Matteo Boschini, Angelo Porrello, and Simone Calderara. Rethinking Experience Replay: a Bag of Tricks for Continual Learning. *CoRR*, abs/2010.05595, 2020. URL https://arxiv.org/abs/2010.05595.

[6] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulò, Elisa Ricci, and Barbara Caputo. Modeling the Background for Incremental Learning in Semantic Segmentation. *CoRR*, abs/2002.00718, 2020. URL https://arxiv.org/abs/2002.00718.

[7] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and M Ranzato. Continual learning with tiny episodic memories. 2019.

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[9] Arthur Douillard, Yifu Chen, Arnaud Dapogny, and Matthieu Cord. PLOP: Learning without Forgetting for Continual Semantic Segmentation. *CoRR*, abs/2011.11390, 2020. URL https://arxiv.org/abs/2011.11390.

[10] Arthur Douillard, Yifu Chen, Arnaud Dapogny, and Matthieu Cord. Tackling Catastrophic Forgetting and Background Shift in Continual Semantic Segmentation. *CoRR*, abs/2106.15287, 2021. URL https://arxiv.org/abs/2106.15287.

[11] Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal Gradient Descent for Continual Learning. *CoRR*, abs/1910.07104, 2019. URL http://arxiv.org/abs/1910.07104.

[12] M. Grinvald, F. Furrer, T. Novkovic, J. J. Chung, C. Cadena, R. Siegwart, and J. Nieto. Volumetric Instance-Aware Semantic Mapping and 3D Object Discovery. *IEEE Robotics and Automation Letters*, 4(3):3037–3044, July 2019. ISSN 2377-3766. doi: 10.1109/LRA.2019.2923960.

[13] Ji Hou, Saining Xie, Benjamin Graham, Angela Dai, and Matthias Nießner. Pri3D: Can 3D Priors Help 2D Representation Learning? *CoRR*, abs/2104.11225, 2021. URL https://arxiv.org/abs/2104.11225.

[14] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, 2014. URL http://arxiv.org/abs/1412.6980. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[15] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *arXiv preprint arXiv:1706.08840*, 2017.

[16] William E. Lorensen and Harvey E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, August 1987. ISSN 0097-8930. doi: 10.1145/37402.37422. URL https://doi.org/10.1145/37402.37422.

[17] Michael McCloskey and Neal J. Cohen. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press, 1989. doi: https://doi.org/10.1016/S0079-7421(08)60536-8. URL https://www.sciencedirect.com/science/article/pii/S0079742108605368.

[18] John McCormac, Ankur Handa, Andrew J. Davison, and Stefan Leutenegger. SemanticFusion: Dense 3D Semantic Mapping with Convolutional Neural Networks. *CoRR*, abs/1609.05130, 2016. URL http://arxiv.org/abs/1609.05130.

[19] Umberto Michieli and Pietro Zanuttigh. Incremental learning techniques for semantic segmentation. *arXiv preprint arXiv:1907.13372*, 2019.

[20] Umberto Michieli and Pietro Zanuttigh. Continual Semantic Segmentation via Repulsion-Attraction of Sparse and Disentangled Latent Representations. *CoRR*, abs/2103.06342, 2021. URL https://arxiv.org/abs/2103.06342.

[21] Raul Mur-Artal and Juan D. Tardós. ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. *CoRR*, abs/1610.06475, 2016. URL http://arxiv.org/abs/1610.06475.

[22] Helen Oleynikova, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan Nieto. Voxblox: Incremental 3D Euclidean Signed Distance Fields for On-Board MAV Planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[23] Rudra P. K. Poudel, Stephan Liwicki, and Roberto Cipolla. Fast-SCNN: Fast Semantic Segmentation Network. *CoRR*, abs/1902.04502, 2019. URL http://arxiv.org/abs/1902.04502.

[24] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P Lillicrap, and Greg Wayne. Experience replay for continual learning. *arXiv preprint arXiv:1811.11682*, 2018.

[25] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an Open-Source Library for Real-Time Metric-Semantic Localization and Mapping. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2020. URL https://github.com/MIT-SPARK/Kimera.

[26] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor Segmentation and Support Inference from RGBD Images. In *European Conference on Computer Vision (ECCV)*, 2012.

[27] Leslie N. Smith and Nicholay Topin. Super-Convergence: Very Fast Training of Residual Networks Using Large Learning Rates. *CoRR*, abs/1708.07120, 2017. URL http://arxiv.org/abs/1708.07120.

[28] Charles Sun, Jędrzej Orbik, Coline Devin, Brian Yang, Abhishek Gupta, Glen Berseth, and Sergey Levine. Fully autonomous real-world reinforcement learning for mobile manipulation. *arXiv preprint arXiv:2107.13545*, 2021.

[29] Sven Woop, Louis Feng, Ingo Wald, and Carsten Benthin. Embree Ray Tracing Kernels for CPUs and the Xeon Phi Architecture. In *ACM SIGGRAPH 2013 Talks*, SIGGRAPH '13, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450323444. doi: 10.1145/2504459.2504515. URL https://doi.org/10.1145/2504459.2504515.

[30] Lu Yu, Xialei Liu, and Joost van de Weijer. Self-Training for Class-Incremental Semantic Segmentation. *CoRR*, abs/2012.03362, 2020. URL https://arxiv.org/abs/2012.03362.

A IMPLEMENTATION DETAILS

A.1 Network and Dataset

We use Fast-SCNN [23] as a semantic segmentation network. During inference it runs at over $250\,\mathrm{fps}$ using a resolution of 320×640 pixels with $1.1\,\mathrm{M}$ parameters. For quantitative evaluation, we use the ScanNet [8] dataset. It consists of 1513 Microsoft Kinect camera trajectories recorded at $30\,\mathrm{fps}$ within 707 distinct indoor spaces. For each scan, the dataset provides a dense 3D map, which is manually annotated with NYU40 classes [26] and per-frame labels generated by 2D re-projection. The pre-training dataset consists of every 100th frame of scene 11-707 resulting in $\sim 25\,\mathrm{k}$ frames. From this we use $20\,\mathrm{k}$ frames for the actual pre-training and $5\,\mathrm{k}$ for testing the performance on the pre-training dataset. All scans recorded in scenes 1-5 are used to evaluate the adaptation performance of our proposed method. For each scene, up to 3 separate video sequences are provided in the dataset. We chain these sequences into a single long sequence for each scene, from which the first 80% of the frames are used for pseudo-label generation and continually training the network. The final 20% of the frames are only used for testing the adaptation performance. Despite the strict training-test split, the same objects may be observed within the same scene in both datasets. We stress that the created benchmark mimics a real robotic scenario, in which a large dataset of annotated data is commonly available, but adaptation during a mission has to be performed in an unsupervised manner.

A.2 Network Pre-Training

We train the network on the pre-training dataset using Adam [14] with a batch size of 8. The starting learning rate is set to 10^{-3} and polynomially decayed over 150 epochs to 10^{-6} with a rate of 0.9. We stop the training procedure early after 65 epochs (~ 200.000 optimization steps) given convergence on a test set. During training, we apply standard data augmentation, including color jitter, horizontal flipping, and random cropping.

A.3 Pseudo-Label Generation

To construct the semantic map, we use Kimera Semantics [25], which builds on VoxBlox [22], a mapping framework based on voxel grids. We set the voxel resolution to $3\,\mathrm{cm}$, which we found to provide a good balance between level of semantic detail captured and computational efficiency. Kimera Semantics tracks the full posterior probability for all 40 NYU40 labels per voxel. Integration of a single measurement into the TSDF and semantic volume on a Ryzen 5900X CPU takes $330\,\mathrm{ms}$ at a resolution of 320×640. For typical room-sized scenes $10 - 20\,\mathrm{m}^2$ the mesh produced by Marching Cubes has a size of $5\,\mathrm{MB}$. The voxel volume storing the full uncompressed semantic posterior has a size of $\sim 500\,\mathrm{MB}$ for $4.1\,\mathrm{m} \times 3.6\,\mathrm{m} \times 1.5\,\mathrm{m}$ volume (Fig. 3). The high-performance CPU-based ray tracing implementation [29] infers pseudo-labels at a rate of $30\,\mathrm{fps}$.

A.4 Continual Learning

We retrain the network on the generated pseudo-labels for a total of 50 epochs. SGD is used with a 1cycle learning rate schedule [27] and a batch size of 16 to adapt the parameters. The scheduler linearly increases the learning rate from 10^{-6} to 0.05 over the initial 5 epochs and successively decays it to 10^{-3} over the remaining 45 epochs. Starting with a slow learning rate is important to avoid strongly perturbing the model parameters within the first iterations of training. We empirically found that storing 10% of the pre-training dataset in the memory buffer (resulting in a memory size N_M of 2000) is capable of representing the training dataset adequately. During training the samples of each mini-batch are randomly chosen with a ratio of 4:1 from the pseudo-labels and memory buffer. We experimentally found this ratio to provide a good trade-off between integrating new knowledge and preserving the performance on the pre-training dataset. During training the same data augmentation used for pre-training is applied to the replayed and pseudo-labeled samples. We found data augmentation to be particularly beneficial for small buffer sizes, which aligns with the findings reported in [5].

B EXPERIMENT DETAILS

B.1 Examples ScanNet

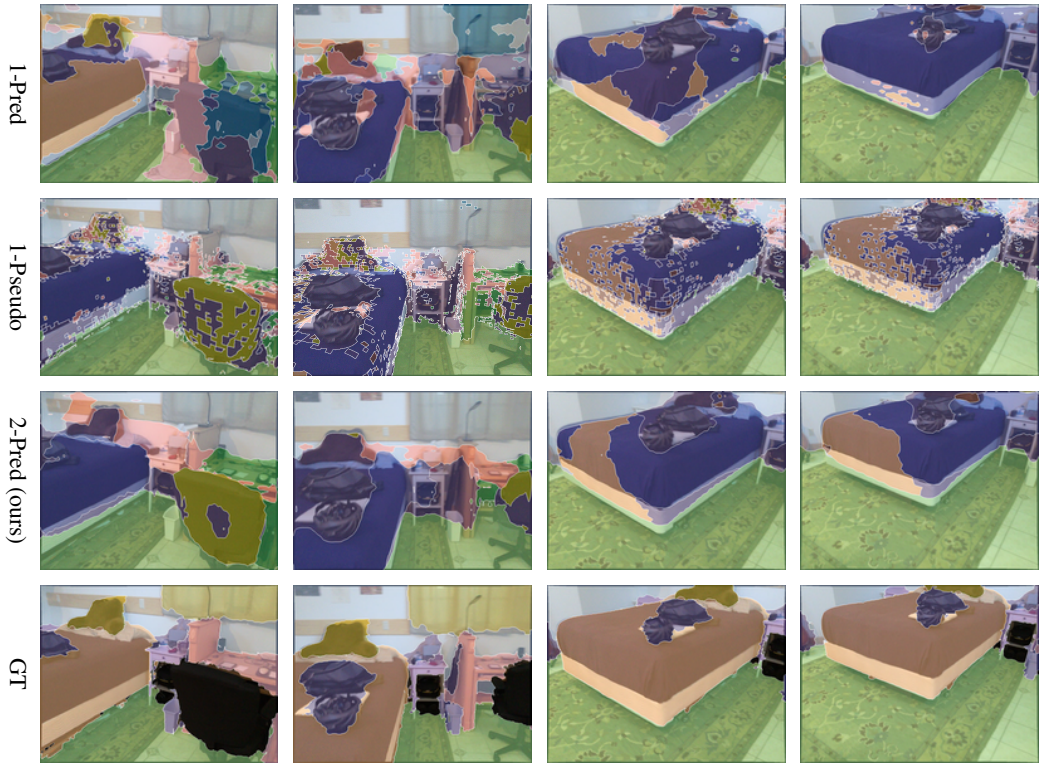

Figure 5: Segmentation of the first scene in the ScanNet dataset, using ScanNet color coding. First row: Pre-trained network predictions (1-Pred). Second row: Generated pseudo-labels leveraging multi-view consistency (1-Pseudo, black areas indicate no semantic estimate available given missing depth data). Third row: Our adapted network predictions after training on 1-Pseudo with a continual learning strategy (2-Pred). Fourth row: Ground-truth labels

B.2 Iterative Operation

Scene	**1-Pred**	**1-Pseudo**	**2-Pred**	**2-Pseudo**	**3-Pred**	**3-Pseudo**
1	60.5	70.2	71.6	73.6	**72.1**	73.2
2	48.5	53.7	**52.9**	49.4	47.5	40.7
3	30.5	38.0	**36.8**	39.4	36.7	37.9
4	74.6	81.6	**80.6**	80.9	79.7	80.2
5	40.3	44.0	**42.7**	42.3	40.1	39.1
AVG	50.9	57.5	**56.9**	57.1	55.2	54.2

Table 2: Network prediction and pseudo-label accuracy for scene 1-5 of the ScanNet dataset. **Pred** denotes network predictions (1: pre-trained network, 2: first iteration adapted network, 3: second iteration adapted network). **Pseudo** denotes pseudo-label. Underlined numbers indicate the best performing pseudo-labels; bold numbers the best performing network.

The process of generating pseudo-labels and adapting the neural network using continual learning can also be performed for multiple steps within the same scene, by iteratively generating pseudo-labels and retraining the network on these. We hypothesized that iterative adaptation by consecutively

transforming between data representations from 2D to 3D could increase the performance further given that the labels used to generate the 3D map are more accurate. We report the accuracy achieved for each of the five adaption scenes. The pseudo-label generation and network training is performed strictly following the procedure elaborated in Appendix A.4 for all iterations.

As shown in Table 2, after the first adaptation step the network accuracy does not improve for four of the five scenes tested. We reason that after the first iteration the adapted network already aligns with the multi-view consistency constraint. Therefore, remapping the multi-view consistent labels into 3D cannot resolve disagreeing semantic estimations and potentially introduces discretization artifacts. This leads us to the conclusion that a single iteration of mapping and continual learning is the most effective for achieving a positive network adaptation.

B.3 Deployment on Handheld Device

To test our proposed method in the wild, we capture data of multiple scenes with a hand-held Azure Kinect RGB-D sensor in different office spaces. The network pre-trained on ScanNet is used to estimate an initial semantic segmentation of the captured data. We use the open-source RGB-D SLAM system ORB-SLAM2 [21] to retrieve the camera poses after bundle-adjustment and loop closures. We then build the volumetric map using these poses.

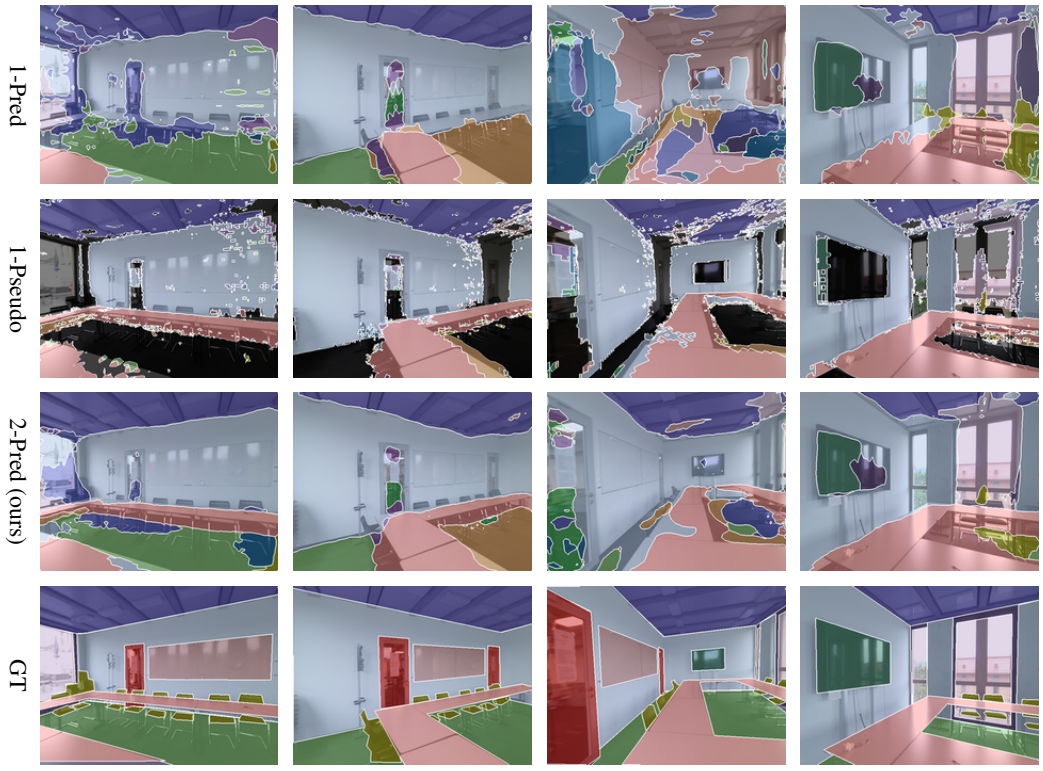

Figure 6: Comparison of semantic segmentation performance of in-house recorded conference room scene. First row: Pre-trained network predictions (1-Pred). Second row: Generated pseudo-labels leveraging multi-view consistency (1-Pseudo). Third row: Our adapted network predictions after training on 1-Pseudo with a continual learning strategy (2-Pred). Fourth row: Ground-truth labels annotated by us. In 1-Pseudo black regions indicate that no depth measurements are present for the corresponding regions.

The Azure Kinect sensor cannot measure depth for reflective and light-absorbing surfaces and is limited to a maximum distance of $5.45\,\mathrm{m}$. Figure 6 shows examples of the resulting pre-training, pseudo-label and adapted network predictions. As clear from the top row, the pre-trained network misclassifies multiple objects (*table*, *floor*). This evidence is in line with the significant distribution mismatch between the recorded data and the pre-training dataset, which does not include scenes

with conference rooms and was recorded with a different sensor given the good supervision signal available. The generated pseudo-labels (1-Pseudo) correctly classify the *desk* in all frames.

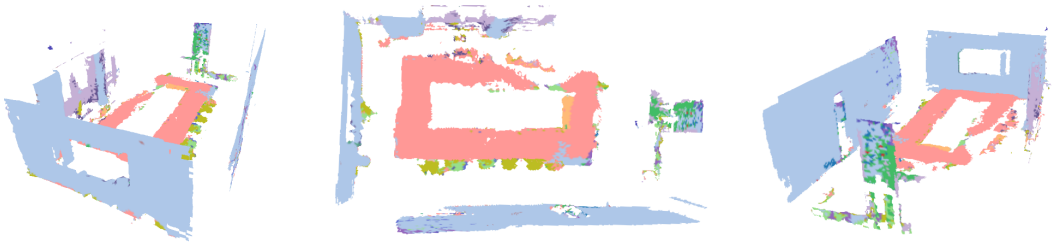

Figure 7: Resulting mesh in the pseudo-label generation of the lab data conference room scene.

The estimated segmented map of the conference room is shown in Figure 7. Given the light-absorbing carpet and reflective television, no depth measurement can be integrated into the volumetric map, leading to a semi-dense mapping, which induces undefined semantics when ray tracing the pseudo-labels. When training the network, we default to the 1-Pred predictions for these pixels with undefined semantics in the pseudo-labels 1-Pseudo. This allows effectively avoiding training on a sparse supervision signal, which would lead the network to wrongly classify not mapped regions (*floor*, *television*) with the label of the closest mapped pixels. We conclude that our method generalizes well to this less controlled deployment scenario, showing the suitability of our approach for real-world robotic applications.

B.4 Failure Cases

Our method is strongly limited by the available supervision signal. As elaborate previously, we only make use of the multi-view consistency constraint and therefore encode the prior knowledge that the same objects observed from various viewpoints belong to the same semantic class. Our method cannot resolve failure cases where an object is misclassified over all viewpoints by the pre-trained neural network. We can observe this failure for the data recorded within the lab as well as the ScanNet dataset. E.g in Figure 6 we can observe that the whiteboard is associated with the incorrect class (wall) over all frames. The pseudo labels (1-Pseudo) and therefore the adapted network (2-Pred) cannot resolve these failure cases. To overcome this we suggest applying higher level semantic understanding methods or providing further supervision by a human annotator.