IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

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

Jonas Frey[1], Hermann Blum[2], Francesco Milano[2], Roland Siegwart[2], Cesar Cadena[2]

***Abstract*—Semantic segmentation networks are usually pre-trained once 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 train the network on renderings of the accumulated semantic map, effectively resolving ambiguities and enforcing multi-view consistency through the 3D representation. In contrast to scene adaptation methods, we aim to retain the previously-learned knowledge, and therefore employ a continual learning experience replay strategy to adapt the network. 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 accuracy on average by 9.9% compared to the fixed pre-trained neural network, while retaining knowledge from the pre-training dataset.**

***Index Terms*—Continual Learning, Semantic Scene Understanding, Deep Learning for Visual Perception.**

I. INTRODUCTION

ROBOTIC perception tasks such as semantic segmentation or object detection rely on large-scale neural networks, which require gathering and annotating large datasets to be trained. This tedious process is costly, time-intensive, and error-prone. Furthermore, a dataset captured at a fixed point in time cannot cover every possible data point in the future for complex tasks in unknown environments. This leads to the common problem of a distribution mismatch between the available labeled training data (source domain) and the actual data of interest (target domain) encountered in a robot's mission. In semantic segmentation, such domain gaps commonly cause misclassification of small semantic details in favor of the dominant neighboring semantic class [1] and make it harder to segment a scene correctly from an arbitrary camera viewpoint, in particular under challenging lighting conditions. While these limitations call for the need of performing network adaptation in new environments, existing methods either require prior knowledge of the environment [2] or ground-truth supervision during deployment [3]. However, to be performed on autonomous robots, adaptation to a new environment should not rely on external supervision, since this is in general not available in a deployment scenario [4]. This therefore restricts adaptation methods for robotics to operate in an unsupervised manner.

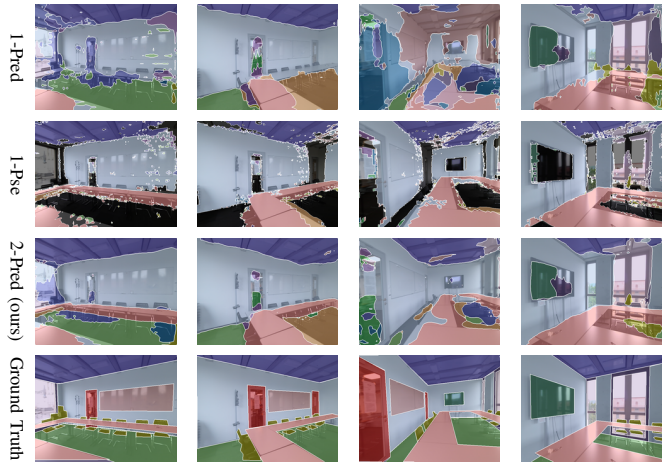

Fig. 1: Comparison of semantic segmentation performance on a recording from an indoor scene. From top to bottom: Pre-trained network predictions (1-Pred), rendered pseudo-labels leveraging multi-view consistency (1-Pse), our adapted network predictions after training on 1-Pse with a continual learning strategy (2-Pred), ground truth annotated by us. All semantic classes are color-coded. The continually-learned network produces more view-consistent predictions than the pre-trained network, as visible for instance for the desk. In the second row black indicates that no semantic class can be determined, due to the missing depth information.

Fortunately, mobile robots can usually observe the same area of their deployment environment from different viewpoints, thus potentially providing the means to resolve semantic ambiguities. This is however not exploited by standard approaches used in robotics for semantic segmentation, which operate on a per-frame basis, producing a prediction for each image separately. Small, occluded or partially observed objects may therefore be misclassified due to the lack of sufficient context, but may be correctly classified in consecutive frames with different viewpoints. While a number of works integrate semantic predictions into a global 3D representation to achieve more consistent labeling of the scene [5], [6], [7], none of

Manuscript received: April, 4, 2022; Revised Juni, 30, 2022; Accepted August, 5, 2022.

This paper was recommended for publication by Editor Jens Kober upon evaluation of the Associate Editor and Reviewers' comments.

This research is supported by the HILTI group and the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017008.

[1] First Author is with the Robotic Systems Lab, ETH Zurich, Switzerland jonfrey@ethz.ch;

[2] Authors are with the Autonomous Systems Lab, ETH Zurich, Switzerland;

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

these approaches leverages the view consistency to adapt the semantic segmentation network, which in their experiments is only trained once before deployment and not updated during the mission. As illustrated in Figure 1, this work instead proposes to explicitly leverage multi-view consistency both to increase the robustness of the semantic labels and to adapt the network to a new environment. To this extent, we accumulate 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 the best of our knowledge, we are the first to propose adapting a neural network 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 during deployment.

Naïve adaptation of the network to the generated supervision signal can however cause forgetting of previously-acquired knowledge [8]. This is undesirable for mobile robots, which have to frequently change operating environments, but can possibly return to previously encountered scenes. This problem relates to the field of *continual learning*, which studies how previous knowledge can be preserved while new one is integrated into a neural network. To achieve the adaptation of the network to the current environment while counteracting forgetting, we adopt an experience replay continual learning strategy [9], regularizing the adaptation to the pseudo-labels with stored samples from the pre-training dataset. Consequently, we refer to our continual learning based approach of network adaptation as continual adaptation. 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, real-world ScanNet [10] dataset, showing a remarkable increase of the semantic segmentation accuracy in average by 9.9% relative to the static network. Additionally, we provide qualitative deployment experiments with a handheld RGB-D sensor.

To summarize, our main contributions are the following:

1) We propose to render pseudo labels from a semantic map as a self-supervision technique for segmentation;
2) We employ the generated pseudo-labels to adapt a semantic segmentation network using continual learning;
3) We present quantitative evaluation on both a real-world dataset and in real-world experiments using an RGB-D sensor, showing consistent improvement of the prediction accuracy of the adapted network.

II. RELATED WORK

Fusing semantic information from different viewpoints to achieve better semantic understanding of a scene is proposed in a number of previous works. SemanticFusion [11] leverages a semantic segmentation network and a SLAM system to create a surfel-based representation of the scene, which probabilistically accumulates semantic information. Kimera [6] uses a similar approach but tracks semantics using a regular voxel grid instead of surfels. Voxblox++ [5] identifies individual object instances and organizes them in a volumetric object-centric map. More recently, [12] proposed to augment a 3D representation of the environment with hierarchical scene graphs, which encode relationships between elements. All the above methods, however, rely on a fixed network which is assumed to be accurate and not updated with the semantic information collected in the generated map. On the contrary, we propose a method that accumulates semantic information similarly to [5] and [6], but continually trains the neural network using the accumulated information about the current environment in an unsupervised manner.

Representation learning methods aim to create meaningful embeddings of input data points, that can be used for further downstream tasks. The majority of the methods in this field leverage unlabeled data, achieving state-of-the-art performance in generating visual representations [13]. Most methods are studied as an offline pre-training mechanisms [14]. Here, particularly relevant is the work by [15], which highlights the benefit of leveraging the view consistency enforced in 3D as a prior for 2D tasks involving semantics. However, while this work focuses on offline pre-training of network features, we explicitly integrate semantic predictions in 3D and tackle an online setting with adaptation of the network to new data.

Domain adaptation methods allow to transfer previously learned information from a source domain to a desired target domain. In the context of semantic segmentation the domain gap is often studied from training on simulated data and adapting to real-world data [16]. Methods can be categorized based on their mode of operation. Input-level approaches modify the input to the neural network to mimic the source domain, also referred to as input style transfer [17], [18]. Feature-level approaches align the latent feature representation within the neural network between samples from the source and target domain. This can be achieved by using a domain classifier network [19] or adversarial domain adaptation [16], resulting in a high degree of similarity between the features obtained from the target domain and those from the source domain. At the output level, label statistics are used to adapt the prediction of the network. URMA [20] performs unsupervised domain adaptation (UDA) for semantic segmentation by reducing the uncertainty in the target domain through auxiliary decoders, which predict multiple segmentation estimates from dropout-corrupted latent feature maps. The multiple predictions are used to compute an uncertainty loss, which is minimized together with the cross-entropy loss of the network predictions to achieve adaptation to the new domain. In general, all domain adaptation methods focus on adapting a network to a specific domain, while discarding the performance achieved on the source domain. In our robotic mission scenario, instead, domains (i.e., scenes) may be revisited and knowledge should thus be retained, rendering most domain adaptation strategies unsuitable for this task.

Continual learning [4] focuses on the problem of integrating new knowledge into a network while preventing the accuracy on previously seen data from decreasing significantly when training on a new data distribution, a commonly-

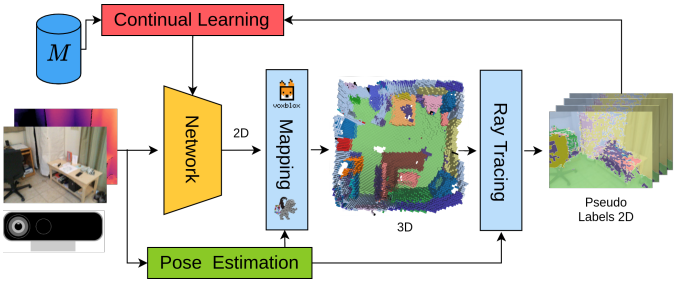

Fig. 2: Overview: An RGB-D camera provides inputs to a segmentation network (yellow) and pose estimation (green). 2D semantic estimates are accumulated in a 3D voxel map. Using ray tracing, we render 2D pseudo-labels from the map. These are used to adapt the network using a continual learning strategy (red), which can access previously stored samples in a memory buffer (dark blue).

observed phenomenon referred to as catastrophic forgetting [8]. Among the different strategies proposed to address this issue, experience replay has proven to be particularly effective [21], [9], [22], [23], often outperforming approaches based on more complex designs [24], [25]. In experience replay, a subset of previously learned samples is stored and used to regularize the training procedure. Despite its relevance for real-world settings, where non-stationary data distributions are common, little research is conducted to address robotic settings within the continual learning field. The sparse existing literature evaluates on a small scale or unrealistic problem sets [22], [23], [26], and mainly focuses on classification tasks [27], [28], [26]. A limited number of works has explored semantic segmentation in continual learning [29], [30], [31], [32], [3]. However, these approaches are not designed for application in an online scenario, tackle segmentation purely in 2D on a per-frame basis, and rely on the availability of ground-truth supervision. On the contrary, we focus on the setting of online deployment where no ground-truth labels are available, and exploit multi-view consistency across frames, both to increase the robustness of the segmentation and to generate a learning signal for adaptation. Recently, the authors of [2] proposed to adopt a continual learning strategy to improve the segmentation and localization capability of a construction robot in a self-supervised manner. However, their approach relies on known building meshes and a LiDAR sensor to generate pseudo-labels. Additionally, it is only evaluated on the task of binary classification into fore- and background. Our method is more versatile, as it performs multi-class semantic segmentation and does not rely on any external knowledge such as precise CAD maps or expensive sensors.

III. APPROACH

Our proposed method consists of two main components: 1.) The *Pseudo-Label Generation* (Sec. III-A) probabilistically accumulates semantic information in a 3D voxel-based map and generates pseudo-labels using ray tracing. 2.) The *Continual Learning* (Sec. III-B) component adapts the parameters of the neural network and receives as input the generated pseudo-labels with the corresponding camera images. It is implemented using an experience replay continual-learning strategy and minimizing the cross-entropy between the generated pseudo-labels and the network predictions.

A. 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 use Kimera Semantics [6] to create a dense semantic map of the robot's environment. The geometry of the scene is represented by 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. A SLAM module [33], [7] estimates the camera extrinsics H_n, which are used together with D_n to integrate the predicted semantics Y_n^{pred} into both volumes. The TSDF is calculated following [34]. For each voxel close to the TSDF surface, the semantic label probability is updated following recursive Bayesian estimation [6]. The predicted semantic labels Y_n^{pred} are one-hot encoded and used together with the depth image D_n to generate a 3D point cloud, where each point contains the information about its predicted semantic label. All the points in the generated cloud that fall into the boundaries of a given voxel are used as new measurements (Y_n^{pred}) to update the stored probability distribution of the voxel belonging to a semantic class.

This mapping procedure is performed for each camera trajectory within a scene individually. After integration of all N measurements, Marching Cubes [35] 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 *pseudo-label* Y_n^{pseudo}. The pseudo-labels aggregate information from multiple viewpoints and enforce 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. We exploit the learning signal to adapt the pre-trained network's parameters θ (Sec. III-B). Exact implementation details are discussed in Section IV.

B. Continual Adaptation

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 neural 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 pseudo-labels themselves require sufficient prediction accuracy of the network, while the network training has the potential to filter out undesirable artifacts from the voxel rendering.

For training, we one-hot encode the pseudo-labels according to the most likely class per pixel. This experimentally outperformed the probabilistic pseudo-labels and reduces storage and computation needed. To update the model parameters θ we use an experience replay strategy. 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 function is used for both the new samples annotated with the pseudo-labels and the replayed samples with ground-truth annotations. We use stochastic gradient descent (SGD) to optimize the cross-entropy loss function. We can explicitly distinguish between the loss induced by samples stored in the memory buffer and the pseudo-labels in the SGD update:

$$l_{\text{total}} = \sum_{i=1}^{n_{\text{pseudo}}} l_{\text{CE}}(f_\theta(x_i), y_i) \; + \; \sum_{j=1}^{n_{\text{rep}}} l_{\text{CE}}(f_\theta(x_j), y_j) \tag{1}$$

$$\theta_{t+1} = \theta_t - \; \frac{\mu}{n_{\text{pseudo}} + n_{\text{rep}}} \, \frac{d}{d\theta} \, l_{\text{total}}, \tag{2}$$

where n_{rep} and n_{pseudo} denote the number of replayed and pseudo-labels samples respectively. The learning rate is denoted by μ and l_{CE} is the cross-entropy loss function. 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}. The network training routine is elaborated in Section IV.

IV. IMPLEMENTATION DETAILS

A. Network and Dataset

We use Fast-SCNN [36] as a semantic segmentation network. During inference it runs at over $250\,\text{fps}$ using a resolution of 320×640 pixels with $1.1\,\text{M}$ parameters. For quantitative evaluation, we use the ScanNet [10] dataset. It consists of 1513 Microsoft Kinect camera trajectories recorded at $30\,\text{fps}$ within 707 distinct indoor spaces. For each scan, the dataset provides a dense 3D map, which is manually annotated with NYU40 classes [37] and per-frame labels generated by 2D re-projection. The pre-training dataset consists of every 100th frame of scene 11-707 resulting in $\sim 25\,\text{k}$ frames. From this we use $20\,\text{k}$ frames for the actual pre-training and $5\,\text{k}$ 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. From each sequence, 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 on novel views of the same scene. 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.

B. Network Pre-Training

For pre-training, we use Adam [38] with a batch size of 8. The learning rate starts at 10^{-3} and polynomially decays 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.

C. Pseudo-Label Generation

To construct the semantic map, we use the provided implementation by Kimera Semantics [6], which builds on Voxblox [34], a mapping framework based on voxel grids. We set the voxel resolution to $3\,\text{cm}$, 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 $330\,\text{ms}$ at a resolution of 320×640. For typical room-sized scenes $10 - 20\,\text{m}^2$ Marching Cubes produces a mesh of around $5\,\text{MB}$. The voxel volume storing the full semantic posterior has a size of $\sim 500\,\text{MB}$ for 4.1x3.6x1.5 m volume (Fig. 4). The high-performance CPU-based ray tracing implementation infers pseudo-labels at a rate of $30\,\text{fps}$.

D. 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 [39] and a batch size of 8 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 [9].

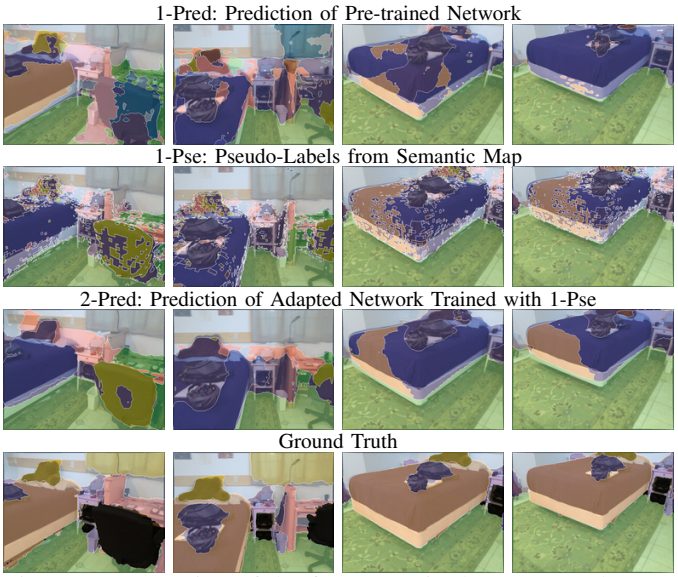

Fig. 3: Segmentation of the first scene in the ScanNet dataset, using ScanNet color coding. First row: Pre-trained network predictions (1-Pred). Second row: Generated pseudo-labels (1-Pse). Third row: Adapted neural network (2-Pred). Fourth row: Ground-truth labels.

V. EXPERIMENTS

In the following, we evaluate each component in the pipeline individually to measure its performance. We then test the fully operating pipeline and show experimental results for the ScanNet dataset and data recorded in a real-world indoor scene using a handheld RGB-D sensor.

A. Pseudo-Label Performance

To evaluate the pseudo-label generation procedure, we compare the segmentation accuracy achieved by the pre-trained neural network to that of the resulting pseudo-labels. We illustrate 4 frames of the first scene in Figure 3. The pre-trained network predictions disagree for the same location over consecutive frames. The pseudo-labels, on the other hand, include minor artifacts induced by the voxel discretization and the ray tracing process, but are consistent over multiple iterations. Moreover, as we show in Section V-B, these artifacts are not reflected in the adapted network predictions and do not prevent the signal from being beneficial for improving the prediction accuracy. To verify the correctness of the pseudo-label generation and set an upper bound for the pseudo-label performance, we additionally use the ground-truth segmentation to generate pseudo-labels.

As metrics for our evaluations we use primarily the total accuracy (Acc), and additionally report mean Intersection of Union (mIoU). We report each metric on the test dataset both per scene and averaged over all scenes. As shown in Table I, the pseudo-labels (1-Pse) generated based on the pre-trained network predictions (1-Pred) improve the accuracy, on average from 51.5% to 54.9%, a relative increase of 6.6%. While for scene 2 and 5 the pseudo-labels accuracy is lower (-2.3% in scene 2) or not significantly higher (+0.8% in scene 5) compared to that of the pre-trained (1-Pred) model,

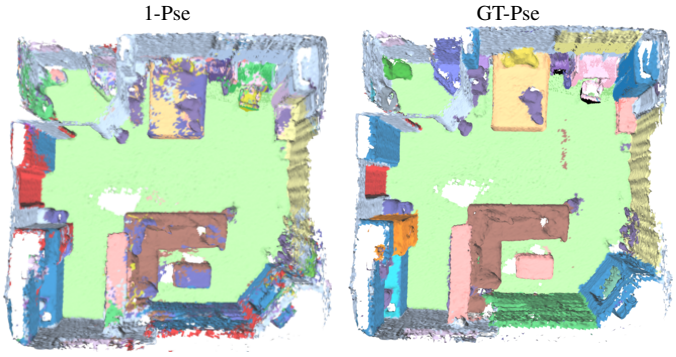

Fig. 4: Meshes used for pseudo-label generation of the first ScanNet scene. Left: Pseudo-label map (1-Pse) generated using the pre-trained neural network. Right: Pseudo-label map (GT-Pse) generated using the ground-truth labels.

on the training dataset the accuracy is 60.6% and 64.1% for scene 2 and scene 5 respectively, making it a viable training signal. Since the artifacts induced by voxelization and ray tracing limit the accuracy of the pseudo-labels, we also report as an upper bound the accuracy of the pseudo-labels generated by integrating into the map the ground-truth data (GT) instead of the network predictions. We show the reconstructed mesh used to generate 1-Pse and GT-Pse for an example scene in Figure 4. As visible from the right image, all the objects can be clearly identified in the GT map. Despite a small number of misclassifications in 1-Pse (*desk*, top right; *toilet* top left) and artifacts (*sofa* bottom middle, *bed* top middle) that cannot be resolved, the overall scene is segmented correctly. Furthermore, the quantitative results reported in the next Sections demonstrate that enforcing view-consistency using a 3D representation and reprojecting the aggregated information to 2D provides a suitable learning signal for network adaptation.

We observed that our pseudo label generation method favours the reconstruction of bigger objects or classes within the scene. This may be attributed to the fact that small objects are commonly observed from fewer viewpoints, thus providing less data to enforce multi-view consistency.

B. Continual Learning

We now analyse how the generated pseudo-labels can be used to integrate the gained domain-specific knowledge into the segmentation network. Specifically, we are interested in how much the segmentation network can improve its performance on the current scene, and how much it loses generalisation to other scenes, essentially posing the adaptation task as a continual learning problem. In a simpler binary segmentation setting, [2] evaluated a range of continual learning methods, and found that the strongest competitor to experience replay is naïve fine-tuning, which we therefore also compare here.

In addition to naïve fine-tuning we compare our approach to URMA [20], an UDA method for semantic segmentation. We highlight that UDA methods in general do not tackle the problem of integrating additional knowledge into the network, but focus instead on transferring knowledge from one domain to the other. Additionally, they are mostly studied in an offline

		Frames		Adaptation									Generalization				
	Scene	Train	Test	1-Pred	1-Pse	URMA	2-FT	2-Pred	2-Pse	3-Pred	3-Pse	GT-Pse	1-Pred	URMA	2-FT	2-Pred	3-Pred
accuracy	1	1415	352	61.7	66.1	62.8	66.9	66.9	68.6	**67.1**	68.0	94.8	60.7	54.5	47.3	**59.5**	58.3
	2	227	57	52.4	50.1	53.9	55.8	**59.4**	52.4	52.1	51.2	94.8	60.7	51.9	44.9	**59.4**	58.5
	3	997	249	38.3	40.5	37.0	39.8	**41.1**	41.9	41.0	40.8	93.6	60.7	55.6	42.4	**59.9**	58.0
	4	381	96	66.1	78.1	62.9	74.9	**75.2**	76.5	74.8	75.9	97.2	60.7	51.5	40.2	**58.6**	58.1
	5	75	18	38.8	39.6	36.6	40.2	**40.4**	49.0	39.6	49.6	84.5	60.7	40.9	36.9	**53.5**	48.4
	AVG	-	-	51.5	54.9	50.6	55.5	**56.6**	57.7	54.9	57.1	93.0	60.7	50.9	42.3	**58.2**	56.3
mIoU	1	1415	352	21.9	19.8	**24.6**	21.4	21.5	22.9	22.1	22.6	81.5	28.2	24.1	11.0	**27.6**	26.8
	2	227	57	23.0	19.8	**23.8**	21.2	21.4	20.4	19.8	19.9	77.1	28.2	21.3	10.6	**27.5**	26.7
	3	997	249	13.2	13.6	12.0	11.9	**13.5**	13.7	12.5	13.6	83.3	28.2	24.0	7.9	**27.4**	25.8
	4	381	96	33.7	49.5	30.1	42.7	**45.7**	48.3	45.6	47.6	90.9	28.2	30.9	9.3	**26.2**	26.3
	5	75	18	24.1	27.9	28.3	**31.0**	30.8	34.4	29.0	34.9	67.2	28.2	17.4	7.4	**23.0**	20.0
	AVG	-	-	23.2	26.1	23.8	25.6	**26.6**	27.9	25.8	27.7	80.0	28.2	23.5	9.2	**26.3**	25.1

TABLE I: Segmentation results: Top rows (Acc), bottom rows (mIoU); Methods are evaluated on the adaptation to novel scenes 1-5 of the ScanNet test dataset and Generalization performance on the pre-training test dataset; Methods: pre-trained network (1-Pred), URMA-baseline (URMA [20]), fine-tuned network (2-FT), continually-learned networks (2-Pred and 3-Pred), pseudo-labels generated from the pre-trained network (1-Pse), or from the continually-learned networks (2-Pse and 3-Pse), and pseudo labels based on ground truth as an upper bound (GT-Pse). In bold the best performing network is indicated and the best generated pseudo label is underlined.

scenario. We adapt the original authors' implementation by replacing the segmentation network with FastSCNN [36] for a fair comparison with our method. The authors choose a different number of training epochs depending on the dataset. We found that the performance of URMA strongly varies across training epochs and we could not determine a single training epoch that leads to good results across scenes. For this reason, we provide an additional benefit to URMA [20] by reporting the best performing epoch during 10 training epochs. In general, providing a fair benchmark that takes into account all the design parameters (e.g., number of update steps, number of samples, memory, time, compute) is extremely challenging in the field of continual learning, making results often highly dependent on the chosen method or benchmark [4]. The naïve fine-tuning is implemented using the same learning procedure as for the continual learning approach, described in Section IV-D, but without replaying any samples from the memory buffer. During continual learning and fine-tuning the same total number of new samples are provided to the network.

We evaluate the broader generalization performance (Gen) by measuring the accuracy on the test split of the pre-training dataset and we calculate the adaptation performance using the test dataset of the respectively adapted scene for scene 1-5 of the ScanNet dataset. As shown in Table I, the network adapted using our method (2-Pred) outperforms the pre-trained network in terms of adaptation (Acc $+5.1\%$). Moreover, the average accuracy of our method slightly increases over naïve fine-tuning (by $+1.1\%$, from 2-FT 55.5% to 2-Pred 56.6% in Acc). We hypothesize that for some scenes the samples stored within the memory buffer induce a positive forward transfer by regularizing the adaptation procedure. For instance, by providing the ground labels for the replayed samples the network may transfer properties from the ground-truth labels (e.g., their smoothness) to the current scene, where only pseudo labels with artifacts are available. However, given that this improvement is not significant, we conclude that continual learning does not hinder adaptation compared to fine-tuning. In addition, the continually-trained network (2-Pred) outperforms the generated pseudo labels in adaptation performance. To understand this effect, consider that we measure performance on the validation split of the scene, whereas 2-Pred is trained on the pseudo labels from the training split, potentially having a higher accuracy given the longer sequence (Sec. V-A). This result shows that the adapted network can transfer knowledge gained from the training dataset to the test dataset, where semantic mapping and reprojection cannot improve the performance. The predicted semantic segmentation of the continually-learned network for the selected key-frames of the first scene is illustrated in Figure 3. As evident in the third row, our methods predictions align with the pseudo-labels but filter out noise and artifacts, resulting in smooth boundary regions.

As expected, the continual learning strategy (Gen. Acc 58.2%) is capable of mitigating catastrophic forgetting compared to fine-tuning (Gen. Acc 42.3%) and URMA (Gen. Acc 50.9%), but minor forgetting still occurs compared to the pre-trained network (Gen. Acc 60.7%). Our evaluation does not reveal a correlation between the number of frames in a scene and the achieved adaptation performance. Overall, the mIoU metric varies more across scenes than the accuracy metric, which we suppose is related to the pixel-wise class imbalance within each scene, to the fact that we use the unbalanced cross-entropy loss function, as well as to the possibly weaker supervision signal for small objects (Sec. V-A). Therefore, while the averaged mIoU metric over all scenes increases for our method, the mIoU decreases slightly for 2 out of 5 scenes. In general the mIoU is sensitive to classes that are only presented within a few frames.

UMRA [20] outperforms in 2 out of 5 scenes the pre-trained

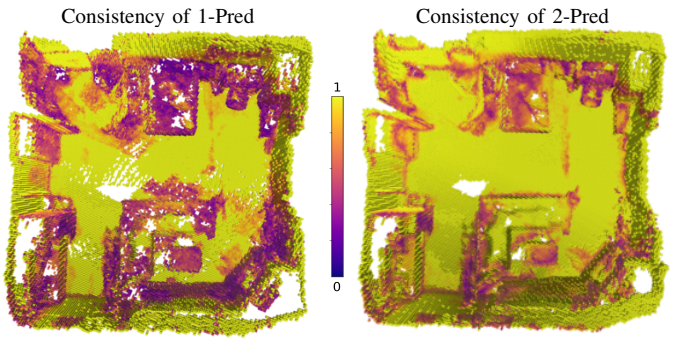

Fig. 5: Multi-view consistency measured as per-voxel confidence in the first ScanNet scene. Left: Confidence of the voxel volume when mapping the pre-trained neural network predictions (1-Pred). Right: Confidence of the voxel volume when mapping the adapted neural network predictions in the second iteration (2-Pred).

network, but overall performs worse (Adap Acc 50.6%) than the pre-trained network (Adap Acc 51.5%) and significantly worse compared to our continually-learned network (Adap Acc 56.6%) in accuracy. In terms of mIoU, URMA [20] outperforms our method on scene 1 and 2, when provided with the added benefit of selecting from 10 possible models per scene. We believe that the overall performance increase of our method with respect to [20] can be traced back to the generation process of the supervision signal used to update the network and to the availability of a small replay dataset. Our method exploits the fact that in a robotic scenario information is captured from different viewpoints and we therefore create a supervision signal based on the fused network predictions. On the other hand, [20] creates a supervision signal on a per-image basis, which makes it applicable to non-sequential data, but does not exploit the constraint of temporal or multi-view consistency.

We can evaluate this 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 5 shows these voxel uncertainties after integrating predictions from the pre-training network (1-Pred) and the adapted continually-learned network (2-Pred). We note that the presented uncertainty only depends on the one-hot encoded label predictions of the network and no information about uncertainty for a single-image prediction can be inferred. Clearly, our adaptation procedure increases the overall certainty and therefore multi-view consistency.

C. Iterative Operation

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 results similarly to Section V-B in Table I. The pseudo-label generation (2-Pse, 3-Pse) and network training (3-Pred) is performed strictly following the procedure elaborated in Section IV for all iterations.

As shown in Table I, 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 while mitigating forgetting.

D. 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 [33] to retrieve the camera poses after bundle adjustment and loop closures. We then build the volumetric map using these poses. 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 1 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 was recorded with a different sensor. The generated pseudo-labels (1-Pse) correctly classify the *desk* in all frames. Given the light-absorbing carpet and reflective television, no depth measurements 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-Pse. 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.

VI. CONCLUSION

We showed that leveraging complementary 2D-3D data representations creates a useful learning signal for semantic segmentation without any external supervision. To the best of our knowledge, we are the first to apply a continual learning strategy to adapt a multi-class semantic segmentation network in a robotic mission scenario. Our experiments show that a 3D data representation that intrinsically enforces multi-view consistency can be effectively used to retrain a network to comply with this consistency constraint already within one iteration. When evaluated on a real-world indoor dataset, our

method increases the semantic segmentation performance on average by 9.9% relative to the pre-trained network. Further experiments with hand-held sensor recordings show how our proposed approach can be applied in a real-world scenario. In conclusion, we show a ready-to-deploy continual learning approach for semantic segmentation that does not require any prior knowledge of the scene or any external supervision and can simultaneously retain knowledge of previously seen environments while integrating new knowledge.

Our approach requires volumetric mapping, which is so far limited by the assumption of a static world and by the operating characteristics of the existing RGB-D sensors. Furthermore, consistent misclassifications within a scene cannot be resolved by multi-view consistency. Future work may extend our approach by including prior knowledge, e.g., identification of object instances or application of scene understanding methods to potentially resolve more misclassifications. Investigation of the effects of incorrect feedback loops caused by a wrong self-supervision signal can provide insights on how to prevent consistent misclassifications. Additionally, a promising future direction may be to investigate using the prior of multi-view consistency enforced by the global 3D map as a self-supervision method for representation learning. Based on our work, further research can be conducted to reduce the gap between fundamental continual learning research and real-world robotic applications.

REFERENCES

[1] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. García Rodríguez, "A Review on Deep Learning Techniques Applied to Semantic Segmentation," *arXiv*, vol. 1704.06857, 2017.

[2] H. Blum, F. Milano, R. Zurbrügg, R. Siegwart, C. Cadena, and A. Gawel, "Self-Improving Semantic Perception on a Construction Robot," *CoRL*, 2021.

[3] A. Douillard, Y. Chen, A. Dapogny, and M. Cord, "PLOP: learning without forgetting for continual semantic segmentation," in *CVPR*, 2021.

[4] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. Díaz-Rodríguez, "Continual Learning for Robotics: Definition, Framework, Learning Strategies, Opportunities and Challenges," *Information Fusion*, vol. 58, 2020.

[5] 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*, 2019.

[6] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an Open-Source Library for Real-Time Metric-Semantic Localization and Mapping," in *ICRA*, 2020.

[7] A. Dai, M. Nießner, M. Zollöfer, S. Izadi, and C. Theobalt, "BundleFusion: Real-time Globally Consistent 3D Reconstruction using On-the-fly Surface Re-integration," *ACM Transactions on Graphics*, 2017.

[8] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," ser. Psychology of Learning and Motivation, G. H. Bower, Ed. Academic Press, 1989, vol. 24, pp. 109–165.

[9] P. Buzzega, M. Boschini, A. Porrello, and S. Calderara, "Rethinking experience replay: a bag of tricks for continual learning," in *International Conference on Pattern Recognition (ICPR)*, 2021.

[10] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes," in *CVPR*, 2017.

[11] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks," in *ICRA*, 2017.

[12] I. Armeni, Z.-Y. He, A. Zamir, J. Gwak, J. Malik, M. Fischer, and S. Savarese, "3d scene graph: A structure for unified semantics, 3d space, and camera," in *ICCV*, 2019.

[13] A. Kolesnikov, X. Zhai, and L. Beyer, "Revisiting Self-Supervised Visual Representation Learning," in *CVPR*, 2019.

[14] M. Caron, P. Bojanowski, J. Mairal, and A. Joulin, "Unsupervised Pre-Training of Image Features on Non-Curated Data," in *ICCV*, 2019.

[15] J. Hou, S. Xie, B. Graham, A. Dai, and M. Nießner, "Pri3d: Can 3d priors help 2d representation learning?" in *ICCV*, 2021.

[16] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, "Cycada: Cycle consistent adversarial domain adaptation," in *ICML*, 2018.

[17] Y. Chen, W. Li, and L. V. Gool, "Road: Reality oriented adaptation for semantic segmentation of urban scenes," *CVPR*, 2018.

[18] Y.-C. Chen, Y.-Y. Lin, M.-H. Yang, and J.-B. Huang, "Crdoco: Pixel-level domain transfer with cross-domain consistency," in *CVPR*, 2019.

[19] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," 2016.

[20] P. T. S and F. Fleuret, "Uncertainty reduction for model adaptation in semantic segmentation," in *CVPR*, 2021, pp. 9608–9618.

[21] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," in *NeurIPS*, 2019.

[22] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with a-GEM," in *ICLR*, 2019.

[23] M. Farajtabar, N. Azizan, A. Mott, and A. Li, "Orthogonal gradient descent for continual learning," in *International Conference on Artificial Intelligence and Statistics*, 2020.

[24] A. Prabhu, P. Torr, and P. Dokania, "Gdumb: A simple approach that questions our progress in continual learning," in *ECCV*, 2020.

[25] J. Knoblauch, H. Husain, and T. Diethe, "Optimal Continual Learning has Perfect Memory and is NP-hard," 2020.

[26] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio, "An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks," 2015.

[27] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," *CVPR*, 2017.

[28] V. Lomonaco and D. Maltoni, "CORe50: a New Dataset and Benchmark for Continuous Object Recognition," in *CoRL*, 2017.

[29] U. Michieli and P. Zanuttigh, "Continual semantic segmentation via repulsion-attraction of sparse and disentangled latent representations," in *CVPR*, 2021.

[30] M. Klingner, A. Bär, P. Donn, and T. Fingscheidt, "Class-incremental learning for semantic segmentation re-using neither old data nor old labels," in *ITSC*, 2020.

[31] F. Cermelli, M. Mancini, S. Rota Bulò, E. Ricci, and B. Caputo, "Modeling the background for incremental learning in semantic segmentation," in *CVPR*, 2020.

[32] A. Douillard, Y. Chen, A. Dapogny, and M. Cord, "Tackling Catastrophic Forgetting and Background Shift in Continual Semantic Segmentation," *CoRR*, vol. abs/2106.15287, 2021.

[33] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE T-RO*, 2017.

[34] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3D Euclidean Signed Distance Fields for On-Board MAV Planning," in *IROS*, 2017.

[35] W. E. Lorensen and H. E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *SIGGRAPH*, 1987.

[36] R. P. K. Poudel, S. Liwicki, and R. Cipolla, "Fast-scnn: Fast semantic segmentation network," in *BMVC*, 2019.

[37] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor Segmentation and Support Inference from RGBD Images," in *ECCV*, 2012.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, Y. Bengio and Y. LeCun, Eds., 2015.

[39] L. N. Smith and N. Topin, "Super-Convergence: Very Fast Training of Residual Networks Using Large Learning Rates," *ICLR*, 2018.