Continual Neural Mapping: Learning An Implicit Scene Representation from Sequential Observations

Zike Yan[1] Yuxin Tian[2] Xuesong Shi[3] Ping Guo[3] Peng Wang[3] Hongbin Zha[1]
[1] Key Laboratory of Machine Perception (MOE), School of EECS, Peking University
PKU-SenseTime Machine Vision Joint Lab
[2] School of Automation Sceience and Electrical Engineering, Beihang University
[3] Intel Labs China

zike.yan@pku.edu.cn, tianyuxin@buaa.edu.cn,
{xuesong.shi, ping.guo, patricia.p.wang}@intel.com, zha@cis.pku.edu.cn

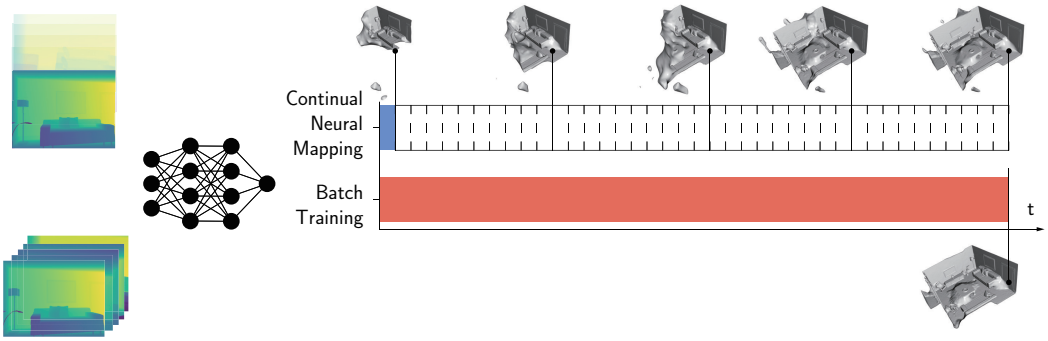


Figure 1: Continual neural mapping learns scene properties from sequential data. By memorizing past experience within shared weights of a single network, we eliminate the need to store the entire sets of data or learn from scratch at each time.

Abstract

Recent advances have enabled a single neural network to serve as an implicit scene representation, establishing the mapping function between spatial coordinates and scene properties. In this paper, we make a further step towards continual learning of the implicit scene representation directly from sequential observations, namely Continual Neural Mapping. The proposed problem setting bridges the gap between batch-trained implicit neural representations and commonly used streaming data in robotics and vision communities. We introduce an experience replay approach to tackle an exemplary task of continual neural mapping: approximating a continuous signed distance function (SDF) from sequential depth images as a scene geometry representation. We show for the first time that a single network can represent scene geometry over time continually without catastrophic forgetting, while achieving promising trade-offs between accuracy and efficiency.

1. Introduction

Scene representations convert visual sensory data into compact forms. Recent trends [58, 34] show that the mapping function $\boldsymbol{y} = f(\boldsymbol{x}; \theta)$ between the spatial coordinate $\boldsymbol{x}$ and the scene property $\boldsymbol{y}$ can serve as an implicit scene representation parameterized by a single neural network θ. Such a new paradigm has drawn significant attention: the neural network defined in a continuous and differentiable function space can be trained to recover fine-grained details at scene scale with efficient memory consumption, which offers great benefits over alternatives.

However, batch training of the implicit neural representation is impractical and inefficient when dealing with possibly unending streams of data. To handle the sequential observations and obtain a globally consistent representation over time, conventional approaches turn to a data fusion paradigm. A discretized scene representation is predefined in memory-inefficient parameter space and updated according to perceived observations at each time. The gap between the emerging neural representation paradigm and

the conventional data fusion paradigm addresses a critical issue: *how we can learn an implicit neural representation continually from sequential observations*?

In this paper, we introduce a novel problem setting of *continual neural mapping*. The central idea is to maintain a continually updated neural network at each time to approximate the mapping function $f(\cdot)$ within the environment. Past observations $(\boldsymbol{x}^{1:t}, \boldsymbol{y}^{1:t})$ are marginalized out and summarized into compact neural network parameters θ^t during training. The neural network not only serves as a memory of sequential data, but also makes predictions of scene properties within the entire environment. The prediction-updating fashion leads to a self-improved mapping function when constantly exploring the environment, which resembles human-like learning scenarios from a continual learning perspective.

We instantiate the proposed continual neural mapping problem by tackling the SDF approximation from sequential depth images. We propose an experience replay approach that distills past experience to guide the prediction without catastrophic forgetting. Experimental results demonstrate that the proposed method outperforms batch re-training/fine-tuning baselines and obtains comparable results against state-of-the-art approaches. The key contributions of our work are summarized as follows:

- We are the first to address the problem of learning an implicit neural scene representation continually from sequential data, namely *continual neural mapping*;

- We deal with the problem of SDF approximation from sequential data under the proposed continual neural mapping setting, outperforming competitive approaches;

- We propose an experience replay method to learn scene geometry continually without catastrophic forgetting. The memory consumption and training time are orders of magnitude less than the batch re-training baseline.

2. Related Work

The proposed continual neural mapping setting lies in the intersection of implicit neural representation, 3D data fusion, and continual learning. In this section, we review the most related work in each area and highlight the major differences over the proposed problem setting.

2.1. Implicit Neural Representation

Implicit neural representation takes a neural network as the continuous mapping function between the spatial coordinates and the scene properties. Shape-conditioned representations concatenate the coordinate $\boldsymbol{x}$ and a latent shape embedding $\boldsymbol{z}$ to represent multiple shape instances as $\boldsymbol{y} = f(\boldsymbol{x}, \boldsymbol{z}; \theta)$. The shape embedding $\boldsymbol{z}$ is latter conducted in a local fashion to recover fine-grained details at scene scale [21, 45, 5]. The output properties $\boldsymbol{y}$ infered from the shape-conditioned representations vary across shape [43, 33, 9], appearance [39, 41, 56], and motion [38]. Another line uses neural networks to regress the parameters of decomposed primitives directly from the input point set as $\{\boldsymbol{m}_j\} = f(\{\boldsymbol{x}_i\}; \theta)$. The regressed primitive parameters are then grouped together as the entire shape parameter space, representing the scene geometry as $\boldsymbol{y} = \phi(\{\boldsymbol{m}_j\}, \boldsymbol{x})$. Commonly used primitives include hyperplanes [14], Gaussian mixtures [19, 15, 16], volumes [61], and local planes [8].

Recent work has investigated the mapping from spatial coordinates to scene properties directly through MLPs as $\boldsymbol{y} = f(\boldsymbol{x}; \theta)$. The high-frequency details can be preserved well with the help of positional encoding [34], Fourier feature mapping [60], or the periodic activation [58]. We extend the implicit neural representation to a continual learning fashion, where an implicit representation can be directly learned from sequential data without computationally expensive re-training or catastrophic forgetting.

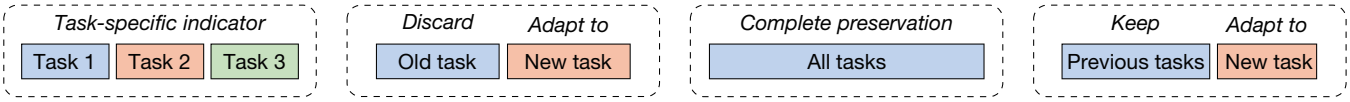
2.2. Incremental Depth Fusion

Conventional depth fusion paradigm aims to maintain a pre-defined output representation instead of the implicit network parameters. The mapping between coordinates and the output representation is accomplished through a deterministic data assignment, where the parameters of the output representation are incrementally updated through weighted averaging according to streaming observations. Most commonly used representations for incremental depth fusion include volumetric TSDF [10, 37, 40, 66, 12] and surfel [46, 67, 22, 54, 55]. On account that the representation is defined in the continuous output range, discretization is inevitable. Postprocessing is usually conducted to transform the discrete representation into a watertight mesh [28] or render the scene as a view-dependent dense image.

Recent advances have fostered a learning-based depth fusion fashion. [4, 72, 11] seek to find a compact and optimisable feature for estimated monocular depth, where the entire scene is represented by a set of keyframes with low-dimensional depth codes. Further extensions utilize neural networks to learn aggregation of learned image/depth features in a latent space [50, 44, 20, 69, 36, 65] as the global scene representation. RoutedFusion [64] follows the conventional TSDF fusion pipeline and learns how volumetric TSDF is updated. However, all previous approaches view depth fusion as a deterministic learnable operation. We, on the other hand, address the problem of knowledge fusion as a continual learning (training) procedure, where the neural network serves as a self-improving scene representation with parameters updated continually.

2.3. Continual Learning

The proposed *continual neural mapping* problem targets the updating of network parameters at each time when new



(a) Multi-task learning (b) Fine-tuning (c) Batch re-training (d) Continual learning

Figure 2: Relevant learning paradigms.

observations arrive that lead to a self-improved mapping function. This problem setting falls into a continual learning [42] category: the streaming data are no longer iid-sampled but highly correlated to adjacent ones. We want to update the mapping function in the newly observed areas while preserving an accurate mapping in previously visited regions without forgetting.

Following [13], existing methods can be generally categorized into three kinds: regularization, parameter isolation, and replay. Regularization-based methods aim to enforce parameter consistency during the training process by penalizing the changes of important parameters [23, 71, 6, 1] or through knowledge distallation [26, 47]. On the other hand, parameter-isolation methods assume that different subsets of the network parameters are attributed to different tasks, thus leading to a flexible gating mechanism. Isolated parameters for each sub-network can be achieved through a dynamically expanded network [53, 24, 2, 70, 48] or through task-specific masking [30, 29, 57]. Finally, replay-based methods store samples or generate pseudo samples as the memory of old knowledge. Special attention is devoted to different sample selection strategies [49, 52], sample generation strategies [3, 51, 68], and optimization constraints [27, 7]. In this paper, we propose an experience replay approach to tackle the proposed continual neural mapping problem by leveraging past experience to guide the continual learning of new observations.

3. Continual Neural Mapping

In this section, we formalize the proposed continual neural mapping problem setting. The connections to relevant learning paradigms are clarified afterwards.

3.1. Problem Statement

We consider a general setting within a 3D environment $\mathcal{W}$, where sequential data $\mathcal{D}^t$ are constantly captured. The data $\mathcal{D}^t = \{(\boldsymbol{x}_i^t, \boldsymbol{y}_i^t)\}_{n^t}$ consist of n^t tuples of spatial coordinates $\boldsymbol{x}_i^t \in \Omega^t$ and the corresponding scene properties $\boldsymbol{y}_i^t$ with observed areas $\Omega^t \subset \mathcal{W}$ specified. The objective of the continual neural mapping is to learn a mapping function $f(\cdot)$ parameterized by a neural network θ^t continually from the observed data $\mathcal{D}^t$ to depict the connections between the spatial coordinates and the scene properties as:

$$\boldsymbol{y} = f(\boldsymbol{x}; \theta^t), \forall \boldsymbol{x} \in \mathcal{W}. \tag{1}$$

Knowledge transfer. The mapping function $f(\cdot)$ serves as an implicit neural representation for the 3D environment, which can be queried at any time to predict the scene property $\boldsymbol{y}$ given the spatial coordinate $\boldsymbol{x}$. For previously visited areas $\boldsymbol{x} \in \Omega^{1:t}$, the mapping function serves as a compact memory of past observations $\mathcal{D}^{1:t}$. This is related to *backward transfer* [25, 27], where the neural network not only memorizes existing data, but also leads to better performance on previously visited areas when learning from new observations. On the other hand, for unseen areas $\boldsymbol{x} \in \mathcal{W} \cap \overline{\Omega^{1:t}}$, the mapping function serves as a predictor. *Forward transfer* may be facilitated that distills knowledge and skills for future exploration. Consequently, continual neural mapping alleviates the need for storing the entire dataset $\mathcal{D}^{1:t}$ while preserving the complete mapping function within the environment, guaranteeing a quick convergence to new observations.

Challenges. The objectiveness of the proposed continual neural mapping problem is to find an optimal neural network that shares parameters θ^t across all previous tasks[1] as:

$$\arg\min_{\theta_t} \frac{1}{\|\mathcal{D}^{1:t}\|} \sum_{\mathcal{D}^{1:t}} \mathcal{L}(f(\boldsymbol{x}; \theta^t), \boldsymbol{y}) \tag{2}$$

The major challenge lies in the gap between the proposed problem setting and the conventional Empirical Risk Minimization (ERM) principle [63]: the streaming data $\mathcal{D}^t$ lead to constant distribution shift. As non-stationary data distribution breaks the iid-sampled assumption, a learning solution is required to model the overall distribution of past observations without the need to store the entire dataset $\mathcal{D}^{1:t}$.

3.2. Connections to Relevant Learning Paradigms

Though the objectiveness in (2) is a joint optimization of temporally seperated tasks $\mathcal{L}(f(\boldsymbol{x}; \theta), \boldsymbol{y}), (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}^t$, continual neural mapping can be understood from the perspectives of four relevant learning paradigms as illustrated in Fig. 2. Clarifying our problem setting from the most relevant continual learning perspective, continual neural mapping falls into a *domain-incremental* [62] continual learning scenario: we aim to maintain a globally consistent representation with a single network from sequential data, where data distribution shifts and the objective remains the same.

[1] *Task* refers to a particular period of time where the data distribution is stationary and the objective function is constant [25].

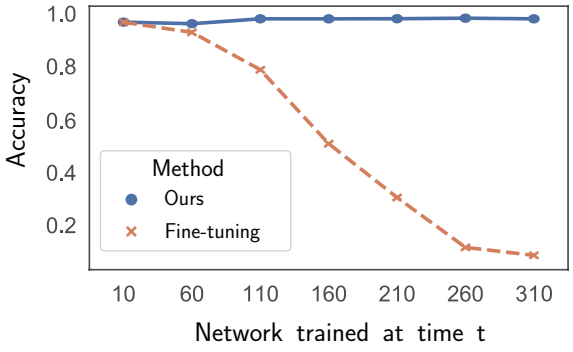


Figure 3: We validate the issue of forgetting by estimating per-point SDF value for the first frame with current network parameters. The percentage of $|f(\boldsymbol{x};\theta^t)| < 0.01$ declines drastically for the fine-tuning baseline, while the proposed method maintains a consistent accuracy level across time.

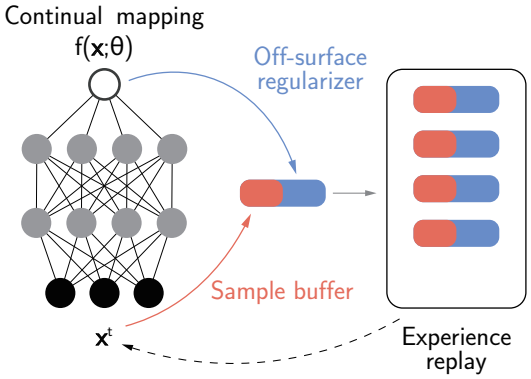


Figure 4: During training, newly observed data are preserved within a fixed size of buffer to constrain zero level-set. Off-surface samples are guided by last network parameters to penalize sign deviation.

On the other hand, multi-task learning splits the training process into a set of dependent tasks and optimizes all tasks jointly. For our continual neural mapping setting, the task boundary is unknown, thus requiring a continual task identifier that assigns training data to specific tasks consistently over time. Meanwhile, backward transfer addresses the problem of constant network adaptation for all tasks, which is opposed to conventional multi-task learning that fixes the network once the model is deployed [13]. The fine-tuning strategy maintains a single network consecutively, where network parameters of a new task are initialized with that of the last task. However, as neural networks tend to be overly plastic [35] from the "plasticity-stability dilemma" perspective [32], the performance of early tasks will degrade on current network parameters (Fig. 3), namely *catastrophic forgetting* [31]. Finally, batch re-training preserves all previously observed data $\boldsymbol{x}^{1:t}$ to satisfy the iid-sampled assumption. However, batch re-training learns a new model at each time from scratch without exploiting past experience. The linearly-growing number of training data results in expensive memory consumption and computational cost.

4. Example: SDF Regression

In this section, we instantiate the proposed continual neural mapping on the task of scene geometry approximation. The objective is a special case of (2) that defines the mapping function $f(\cdot)$ as the SDF parameterized by a single multilayer perceptron (MLP), representing the 3D surface as a zero level-set $\mathcal{M}$:

$$\mathcal{M} = \{\boldsymbol{x} \in \mathbb{R}^3 | f(\boldsymbol{x};\theta^t) = 0\}, f(\cdot) : \mathbb{R}^3 \mapsto \mathbb{R}. \quad (3)$$

In a batch training setting, the problem is studied by [17, 58] and solved as an Eikonal boundary value problem. The continuous SDF can be fit from oriented point cloud data that are iid-sampled from closed surfaces. We take a step further to tackle a more realistic and challenging case that continually learns SDFs parameterized by a single MLP from streaming posed depth images.

4.1. Solution

In practice, we split the energy function (2) into two terms with equal weights as:

$$\sum_{\mathcal{D}^{1:t-1}} \mathcal{L}(f(\boldsymbol{x};\theta^t), \boldsymbol{y}) + \sum_{\mathcal{D}^t} \mathcal{L}(f(\boldsymbol{x};\theta^t), \boldsymbol{y}), \quad (4)$$

where the loss function $\mathcal{L}(f(\boldsymbol{x};\theta^t), \boldsymbol{y})$ consists of a data term $|f(\boldsymbol{x};\theta)|$, an Eikonal term $|\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x};\theta)\| - 1|$, a normal constraint $|\nabla_{\boldsymbol{x}} f(\boldsymbol{x};\theta) - \boldsymbol{n}|$, and an off-surface constraint $\psi(f(\boldsymbol{x};\theta)) = \exp(-\alpha \cdot |f(\boldsymbol{x};\theta)|), \alpha \gg 1$ following [58].

The split terms in (4) can be understood as a combinatory constraint of the current observation $\mathcal{D}^t$ and the past experience $\mathcal{D}^{1:t-1}$. Notice that in the proposed continual neural mapping setting, the entire sequence of raw data $\mathcal{D}^{1:t-1}$ should not be preserved, we resort to an experience replay method to model the past experience as illustrated in Fig. 4.

4.2. Experience Replay

As mentioned in Sec. 1, the maintained neural network serves as a compact memory of previous observations. Hence, random coordinates with SDF values approximated by the last neural network $\{\boldsymbol{x}_i, f(\boldsymbol{x}_i;\theta^{t-1})\}_{n^t}$ can be viewed as iid samples replayed from past experience to regularize the current network. The replayed samples are twofold: 1) off-surface samples to regularize the distance sign; 2) zero level-set samples to regularize the data term and the normal constraint.

For off-surface samples $\{\boldsymbol{x}_o, f(\boldsymbol{x}_o, \theta^{t-1})\}$, we back-project the points to the camera coordinate at time t and

	1	2	3
Camera a	+	-	-
Camera b	-	+	-
Actual	+	+	+

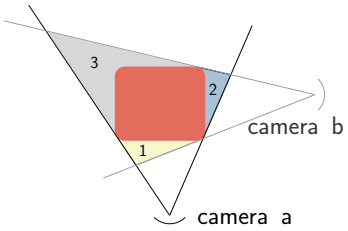


Figure 5: It is obvious that sign reasoning for areas behind the red surface may be false negative. We need to incorporate previous knowledge to better regularize the sign of off-surface samples.

compare them against the depth map. As illustrated in Fig. 5, points within the frustum that are in front of the surface should be true positive, while points within the frunstum that are behind the surface may be false negatives. Hence, we label samples with positive SDF approximation $f(\boldsymbol{x}_o;\theta^{t-1}) > 0$ or the samples that fall in front of the surface within the frustum as positive, and label samples that fall behind the surface within the frustum with negative SDF approximation $f(\boldsymbol{x}_o;\theta^{t-1}) < 0$ as negative. The sign regularizer for off-surface samples is then defined as:

$$\psi_s(f(\boldsymbol{x};\theta)) = \begin{cases} \exp(-\alpha \cdot f(\boldsymbol{x};\theta)) & \text{if positive} \\ \exp(\alpha \cdot f(\boldsymbol{x};\theta)) & \text{if negative} \end{cases} \tag{5}$$

For zero level-set samples $\boldsymbol{x}_z$, one intuitive solution is to construct voxel grids and estimate the SDF value for each vertex. The zero level-set samples can then be easily obtained on the extracted triangle mesh. However, due to incomplete observations, spurious zero level-set samples may be generated in unseen areas. Additional maintenance of occupancy status for each voxel grid is required to eliminate erroneous samples, which is memory inefficient. Here, we take a simple solution to randomly down-sample previous observations and maintain a fixed size of buffer data [24] to regularize the data term $|f(\boldsymbol{x}_z;\theta)|$. Experimental results show that this simple solution is effective enough to store previous knowledge without catastrophic forgetting.

5. Experiments

In this section, we demonstrate that the proposed continual neural mapping setting succeeds in representing constantly observed scene geometry with a single neural network from scratch. The recovered accuracy is comparable against competitive methods with orders of magnitude less memory consumption.

5.1. Experimental Setup

The experiments were conducted on a desktop PC with an Intel Core i7-8700 (12 cores @ 3.2 GHz), 32GB of RAM, and a single NVIDIA GeForce RTX 2080Ti.

Model. We use a single 5-layer SIREN MLP [58] with 256 units in each layer as our base network model.

Baselines. Following [13, 24], the *fine-tuning* baseline is initialized with the last model parameters at each time and is naively trained for each frame; the *re-training* baseline is trained with the entire sequence of data following the signed distance function setup of SIREN [58]. To further study the effect of the replay buffer, we also provide a *re-initialization* baseline that learns from scratch with the buffer data for each frame. Adam optimizer is adopted with a learning rate of 0.0001. The data for each method are trained for 1500 epochs if not specified, while the first frame is trained for 10000 epochs to ensure nice initialization.

Dataset. We mainly evaluate our results quantitatively and qualitatively on the synthetic ICL-NUIM livingroom dataset [18]. Additional qualitative evaluation is conducted on the real TUM dataset [59]. The entire sequence is down-sampled due to the extremely high cost of the batch re-training baseline. The normal is estimated and oriented towards the camera location using Open3D [73].

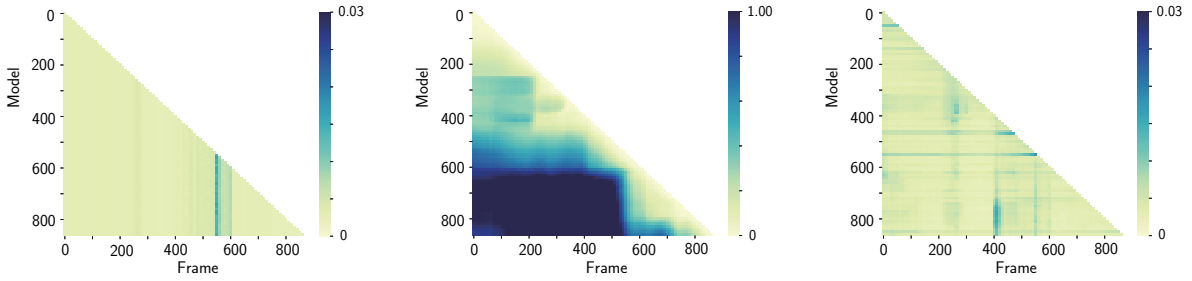
5.2. Model Analysis

We provide an in-depth analysis of the proposed experience replay method by comparing it with the aforementioned baselines and state-of-the-art methods. We refer readers to our supplementary video for better visualizing the continual changes over time.

Continual neural mapping without forgetting. We first assess the forgetting issue under the proposed continual neural mapping setting. The objective is to establish an accurate mapping between spatial coordinates and the corresponding SDF value in previously visited areas. Notice that the depth data sampled from the synthetic ICL dataset are exactly surface samples with zero distance. We calculate the mean distance of each frame $\boldsymbol{x}^t$ using the learned network parameters θ^t at each time. As illustrated in Fig. 6, the proposed method achieves comparable accuracy against the batch re-training baseline while eliminating the catastrophic forgetting issue compared to the fine-tuning baseline.

A similar conclusion can be drawn from the 2D visualization of the SDF approximation. As illustrated in Fig. 7, the fine-tuning baseline quickly forgets the geometry of previously visited areas, while the proposed method maintains a gradually improved SDF approximation during the exploration of the mobile sensor.

Effective solution of experience replay. We further study the role of experience replay in our continual neural mapping setup. Revisiting Sec. 4.2, past experience is used to initialize the network weight, regularize the sign of free space, and constrain the zero level-set. We find that these three issues are essential to the problem of SDF regression from sequential data, guaranteeing past knowledge transfer for accurate SDF approximation.



(a) Batch re-training (0.002/0.003) (b) Fine-tuning (0.496/0.627) (c) Ours (0.003/0.004)

Figure 6: The accuracy heatmap with overall mean/std. (m) of each method on the ICL dataset. The heatmap value at (m,n) is the mean SDF approximation of all points from frame m using nth network parameters. Noticeably, the proposed method maintains consistent accuracy for all frames, while the fine-tuning baseline suffers from catastrophic forgetting severely.

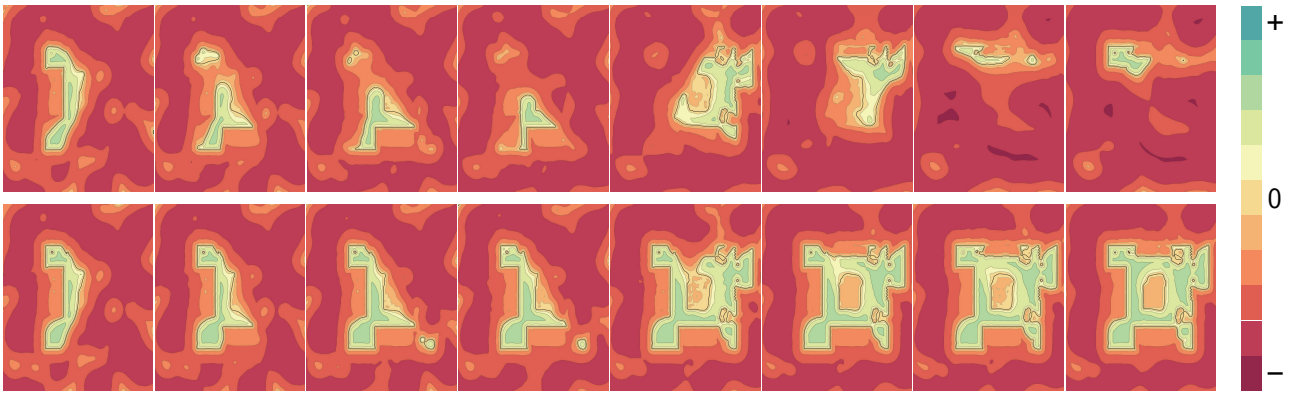


Figure 7: Top view visualization on the ICL dataset of SDF approximation from frame 100 to 800. Fine-tuning baseline (top) suffers from catastrophic forgetting, while ours (bottom) recover the scene geometry continually from sequential data.

We first study the effect of network initialization by comparing it against the re-initialization baseline. As illustrated in Fig. 8, the knowledge distillation through weight initialization leads to faster convergence and better results when a new frame arrives. Fig. 9 displays that the performance of re-initialization baseline deteriorates significantly due to the lack of parameter initialization from previous network parameters. It is noteworthy that the re-initialization baseline cannot recover comparable high-frequency details even after 10000 epochs of training.

On the other hand, the guided sign regularization is crucial to eliminate the false negative distance field arising from occlusion (see Fig. 5). As illustrated in Fig. 10, off-surface samples guided by past experience serve as a reliable regularization to constrain the sign of the distance function.

We can also experimentally find that the simple solution of storing a fixed number of buffer with the same size of each frame [24] is effective enough to serve as a replayed experience of zero level-set observations. Fig. 6, 7 and the supplementary material demonstrate that training a network without replayed buffer leads to catastrophic forgetting.

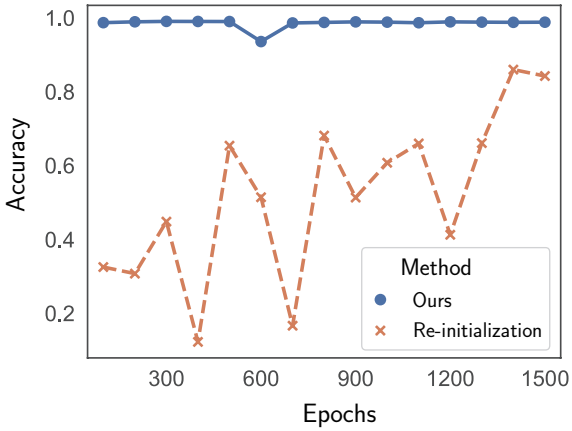
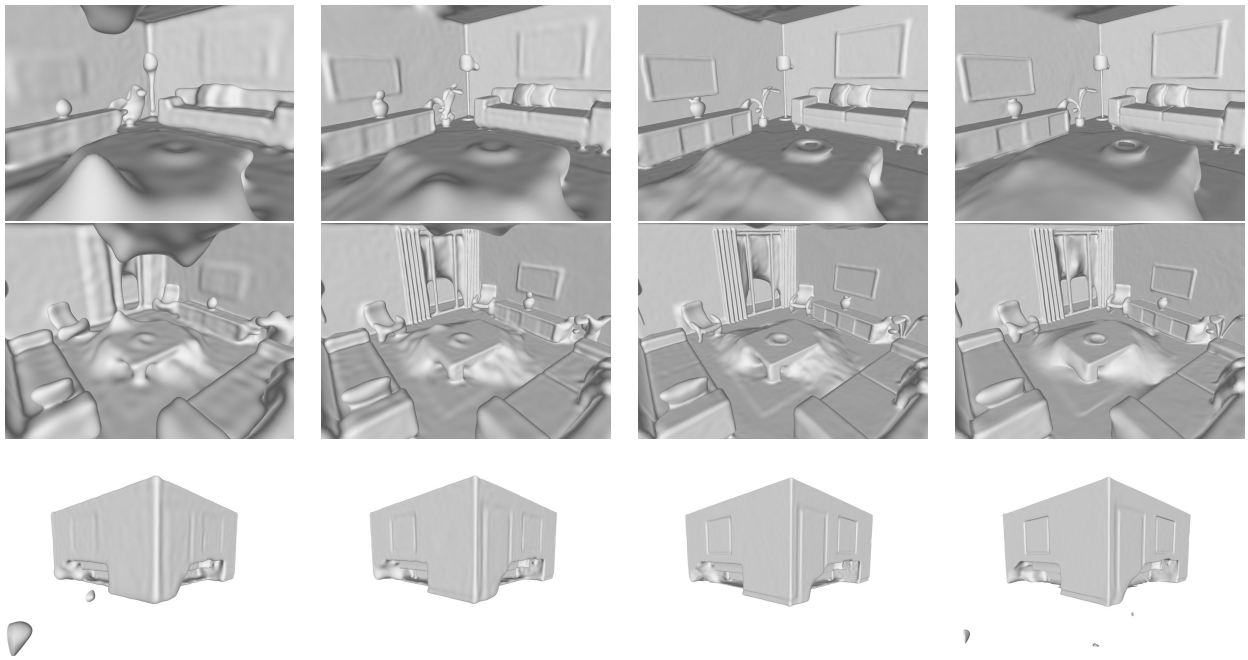


Figure 8: Compared to re-initialization, parameter sharing between frames is beneficial for knowledge distallation, resulting in faster convergence with better performance.



(a) Re-initialization (1500 epochs) (b) Re-initialization (10000 epochs) (c) Re-training (d) Ours

Figure 9: The extracted mesh using approximated SDF values. The proposed approach achieves comparable results against the computationally expensive re-training baseline and outperforms the re-initialization baseline. High-frequency details of the scene geometry cannot be well-recovered by re-initialization baseline even if it is trained for 10000 epochs.

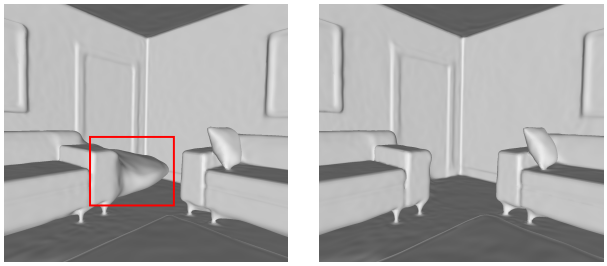


Figure 10: Without the guidance of the last network, false negative SDF approximation arising from partial occlusion may generate spurious zero level-set surface (left). We exploit past experience to alleviate this issue (right).

Tradeoffs between accuracy and efficiency. Our method guarantees constant training time for each frame (approximately 6 minutes for 1500 epochs) due to the fixed size of the replayed buffer. Though an additional memory is required to store the buffer data, the training time will not sacrifice as the batch size are equally divided and attributed to the current data and the buffer data at each iteration. On the contrary, batch re-training baseline will take data from the first frame to the last frame as the entire batch dataset, leading to linearly scaled training time arising from the constantly augmented training data. As illustrated in Fig. 1, when the #87 frame arrives, the batch re-training baseline will take about 13 hours to train the entire dataset for 1500 epochs. Hence, we obtain orders of magnitude smaller training time with comparable accuracy when compared with the batch re-training baseline. The storage of past observations is also orders of magnitude smaller. On the other hand, when compared to the fine-tuning and re-initialization baselines, we achieve better accuracy by exploiting the guidance of past experience. The proposed continual neural mapping ensures the incremental network parameter updating in a globally consistent way, achieving a nice trade-off between efficiency and accuracy when compared with alternative baselines.

Comparisons against state-of-the-art. We also compare against the state-of-the-art methods in terms of the maintained model size and the extracted mesh accuracy. For RoutedFusion [64], we use a voxel size of 2cm, corresponding to a grid resolution of 512^3 allocated voxels. For LIG [21], we use a part size of 25cm to meet the point density. As shown in Fig. 12, the continuous characteristic of our signed distance function leads to spurious zero level-set surfaces in unseen areas, hence resulting in low reconstruction accuracy. However, if we follow the competitive methods to reason the occupancy status of each voxel grid according to previous observations and only extract triangle

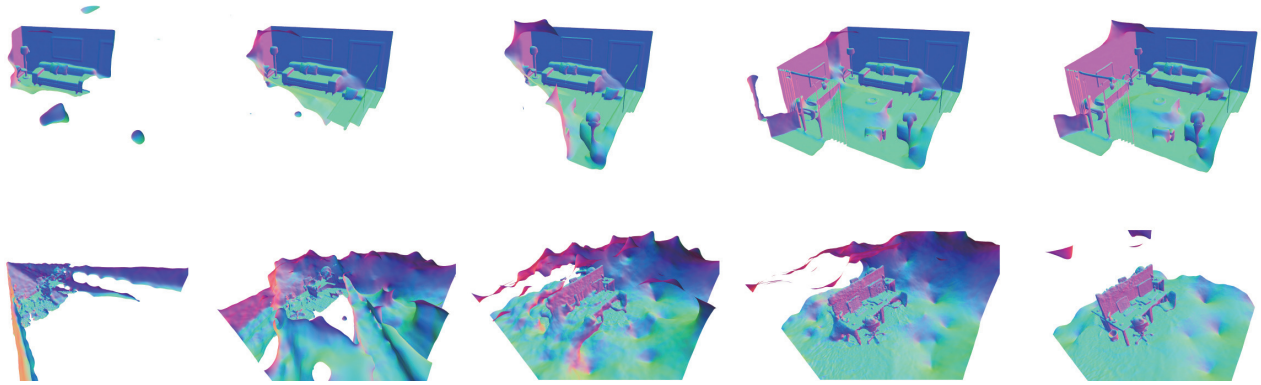


Figure 11: Incrementally updated geometry on the synthetic ICL (top) and real TUM (bottom) datasets. The mesh is visualized with the vertex normal. We refer readers to the supplementary materials for more details.

Table 1: The parameter size of the representations and the cloud/mesh distance error (m) of generated mesh models.

Method	Mean	Std.	Parameters
RoutedFusion [64]	0.0403	0.0687	512^3
LIG [21]	0.0106	0.0146	69,795×32
Ours	0.0584	0.2115	198,657
Ours (masked)	**0.0044**	**0.0010**	198,657

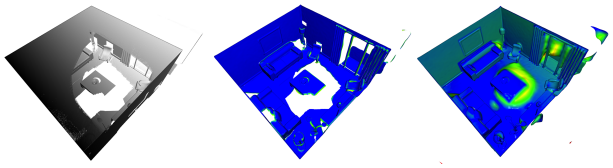


Figure 12: Error map of the extracted mesh model. The majority of erroneous surface lie in the unseen areas (right). By specifying voxel indices for mesh extraction according to the observations as masked model, the accuracy of the mesh outperforms competitives (middle).

meshes around occupied voxels, the overall accuracy outperforms the state-of-the-art methods (Tab. 1). It should be noted that we only maintain a single network with a size of less than 800 KB to achieve mm level of accuracy through continual learning. This is consistent with our average accuracy of SDF approximation in Fig. 6.

The continuous nature of the implicit representation discards the necessity of voxelization, thus guaranteeing a much compact and expressive representation. The contribution of the proposed approach can also be understood from the fusion perspective: Instead of maintaining the discretized value of SDF $\boldsymbol{y}$, we resort to the parameter space of a continuous signed distance function. The volumetric fusion of SDF value is replaced by the incremental updating of the network parameters that are learned continually from sequential observations. As illustrated in Fig. 11, accurate and smooth surfaces can be extracted from the incrementally updated network on both synthetic and real datasets.

6. Conclusion

In this paper, we introduce a novel *continual neural mapping* problem, aiming to bridge the gap between the prevalent batch-trained implicit neural representation and the commonly used streaming data for robotics and vision applications. We primarily aim to discern if a continual learning solution can eliminate the need for batch data preservation and re-training fashion without catastrophic forgetting for coordinate-based MLP. The answer is positive. Dealing with the SDF regression problem, continual neural mapping benefits from the guidance of past experience and enables a single network to model the scene geometry incrementally from sequential observations. This brings great potentials to tasks with online requirements. Besides, the general problem setting turns scene understanding into an incremental map-centric fashion.

Exploiting the expressiveness of the neural network as a memory for past sequential observations or a predictor for future exploration may be the route to exciting future work. Potential directions based on the continual neural mapping paradigm include how to achieve faster convergence for real-time applications, how to encode multiple scene properties within a single network continually, and how to enhance the expressiveness and the prediction quality with different network architectures and learning techniques.

Acknowledgements We thank anonymous reviewers for their fruitful comments and suggestions. This work is supported by the National Key Research and Development Program of China (2017YFB1002601) and National Natural Science Foundation of China (61632003, 61771026).

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *European Conf. on Computer Vision (ECCV)*, 2018.

[2] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[3] Craig Atkinson, Brendan McCane, Lech Szymanski, and Anthony Robins. Pseudo-rehearsal: Achieving deep reinforcement learning without catastrophic forgetting. *Neurocomputing*, 428:291–307, 2021.

[4] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J Davison. Codeslam—learning a compact, optimisable representation for dense visual slam. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[5] Rohan Chabra, Jan E. Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *European Conf. on Computer Vision (ECCV)*, 2020.

[6] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *European Conf. on Computer Vision (ECCV)*, 2018.

[7] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-GEM. In *Intl. Conf. on Learning Representations (ICLR)*, 2019.

[8] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. Bsp-net: Generating compact meshes via binary space partitioning. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[9] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[10] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, 1996.

[11] Jan Czarnowski, Tristan Laidlow, Ronald Clark, and Andrew J Davison. Deepfactors: Real-time probabilistic dense monocular slam. *IEEE Robotics and Automation Letters*, 5(2):721–728, 2020.

[12] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Trans. Graphics*, 36(3):24, 2017.

[13] Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Trans. Pattern Anal. Machine Intell.*, 2021.

[14] Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. Cvxnet: Learnable convex decomposition. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[15] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[16] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *Intl. Conf. on Computer Vision (ICCV)*, 2019.

[17] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Intl. Conf. on Machine Learning (ICML)*. 2020.

[18] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J Davison. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2014.

[19] Amir Hertz, Rana Hanocka, Raja Giryes, and Daniel Cohen-Or. Pointgmm: A neural gmm network for point clouds. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[20] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *Intl. Conf. on Computer Vision (ICCV)*, 2017.

[21] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3d scenes. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[22] Maik Keller, Damien Lefloch, Martin Lambers, Shahram Izadi, Tim Weyrich, and Andreas Kolb. Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *Intl. Conf. on 3D Vision (3DV)*, 2013.

[23] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proc. National Academy of Sciences*, 114(13):3521–3526, 2017.

[24] Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural dirichlet process mixture model for task-free continual learning. In *Intl. Conf. on Learning Representations (ICLR)*, 2020.

[25] Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information Fusion*, 58:52–68, 2020.

[26] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Trans. Pattern Anal. Machine Intell.*, 40(12):2935–2947, 2017.

[27] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[28] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH*, 1987.

[29] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *European Conf. on Computer Vision (ECCV)*.

[30] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[31] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165, 1989.

[32] Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in Psychology*, 4:504, 2013.

[33] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[34] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conf. on Computer Vision (ECCV)*, 2020.

[35] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of training regimes in continual learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2020.

[36] Zak Murez, Tarrence van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *European Conf. on Computer Vision (ECCV)*, 2020.

[37] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR)*, 2011.

[38] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Intl. Conf. on Computer Vision (ICCV)*, 2019.

[39] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[40] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Trans. Graphics*, 32(6):1–11, 2013.

[41] Michael Oechsle, Michael Niemeyer, Christian Reiser, Lars Mescheder, Thilo Strauss, and Andreas Geiger. Learning implicit surface light fields. In *Intl. Conf. on 3D Vision (3DV)*, 2020.

[42] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

[43] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[44] Despoina Paschalidou, Osman Ulusoy, Carolin Schmitt, Luc Van Gool, and Andreas Geiger. Raynet: Learning volumetric 3d reconstruction with ray potentials. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[45] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conf. on Computer Vision (ECCV)*, 2020.

[46] Hanspeter Pfister, Matthias Zwicker, Jeroen Van Baar, and Markus Gross. Surfels: Surface elements as rendering primitives. In *SIGGRAPH*, 2000.

[47] Amal Rannen, Rahaf Aljundi, Matthew B Blaschko, and Tinne Tuytelaars. Encoder based lifelong learning. In *Intl. Conf. on Computer Vision (ICCV)*, 2017.

[48] Dushyant Rao, Francesco Visin, Andrei A Rusu, Razvan Pascanu, Yee Whye Teh, and Raia Hadsell. Continual unsupervised representation learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2019.

[49] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[50] Gernot Riegler, Ali Osman Ulusoy, Horst Bischof, and Andreas Geiger. Octnetfusion: Learning depth fusion from data. In *Intl. Conf. on 3D Vision (3DV)*, 2017.

[51] Amanda Rios and Laurent Itti. Closed-loop memory gan for continual learning. In *Intl. Joint Conf. on AI (IJCAI)*, 2019.

[52] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NIPS)*, 2019.

[53] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[54] Thomas Schöps, Torsten Sattler, and Marc Pollefeys. Bad slam: Bundle adjusted direct rgb-d slam. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[55] Thomas Schöps, Torsten Sattler, and Marc Pollefeys. Surfelmeshing: Online surfel-based mesh reconstruction. *IEEE Trans. Pattern Anal. Machine Intell.*, 42(10):2494–2507, 2019.

[56] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *Advances in Neural Information Processing Systems (NIPS)*, 2020.

[57] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *Intl. Conf. on Machine Learning (ICML)*, 2018.

[58] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems (NIPS)*, 2020.

[59] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2012.

[60] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems (NIPS)*, 2020.

[61] Shubham Tulsiani, Hao Su, Leonidas J Guibas, Alexei A Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[62] Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

[63] Vladimir Vapnik. Principles of risk minimization for learning theory. In *Advances in Neural Information Processing Systems (NIPS)*, 1992.

[64] Silvan Weder, Johannes Schonberger, Marc Pollefeys, and Martin R Oswald. Routedfusion: Learning real-time depth map fusion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[65] Silvan Weder, Marc Schönberger, Johannes Lutz Pollefeys, and Martin Ralf Oswald. NeuralFusion: Online depth fusion in latent space. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[66] Thomas Whelan, Michael Kaess, Hordur Johannsson, Maurice Fallon, John J Leonard, and John McDonald. Real-time large-scale dense rgb-d slam with volumetric fusion. *Intl. J. of Robotics Research*, 34(4-5):598–626, 2015.

[67] Thomas Whelan, Stefan Leutenegger, R Salas-Moreno, Ben Glocker, and Andrew Davison. Elasticfusion: Dense slam without a pose graph. In *Robotics: Science and Systems (RSS)*, 2015.

[68] Ye Xiang, Ying Fu, Pan Ji, and Hua Huang. Incremental learning using conditional adversarial networks. In *Intl. Conf. on Computer Vision (ICCV)*, 2019.

[69] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *Intl. Conf. on Computer Vision (ICCV)*, 2019.

[70] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *Intl. Conf. on Learning Representations (ICLR)*, 2018.

[71] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Intl. Conf. on Machine Learning (ICML)*, 2017.

[72] Shuaifeng Zhi, Michael Bloesch, Stefan Leutenegger, and Andrew J Davison. Scenecode: Monocular dense semantic reconstruction using learned encoded scene representations. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[73] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv preprint arXiv:1801.09847*, 2018.