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

Zike Yan Yuxin Tian Xuesong Shi Ping Guo Peng Wang Hongbin Zha

1. Introduction

We refer readers to the video material for better visualizing the dynamical changes of the network, the corresponding SDF approximation, and the extracted mesh models. Here, we provide additional experimental results to complement the experiment section of our main paper. We also outline revenues for interesting future work (highlighted in bold font) through the supplementary experiments.

2. Baseline Settings

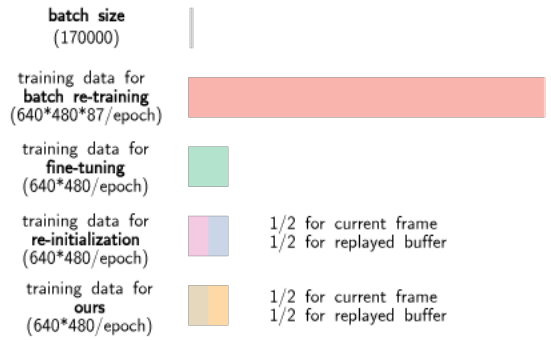
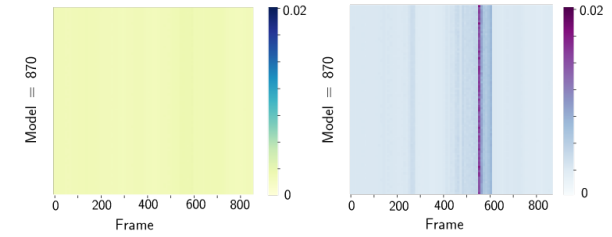
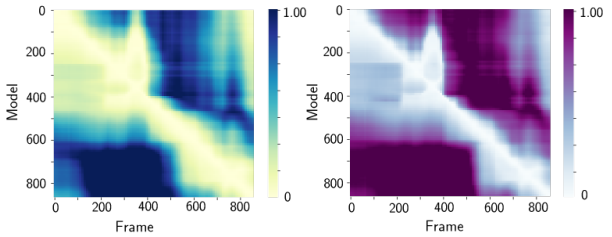


Figure 1: Training data size for different baselines when the #870 frame arrives. Note that we downsample the data every ten frames. Hence, there are 86 frames that have been seen before the #870 frame.

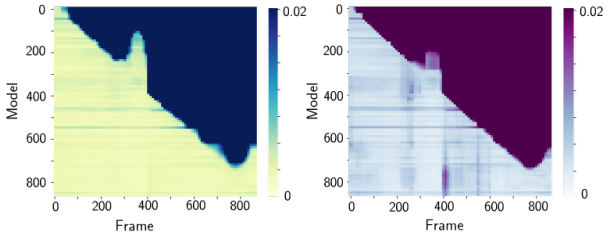
A more illustrative explanation of the proposed baseline setting is presented in this section. As presented in Fig. 1, the data size for batch re-training is 87 times as large as that of ours, thus leading to much more iterations within each epoch (and more training time accordingly). As is presented in Sec. 5.2 of the main paper, we achieve nice trade-offs between accuracy and efficiency with much less training time and data storage over the batch re-training baseline (comparable accuracy) and much better accuracy compared to other alternatives (same training time).



(a) Batch training (mean: 2.06mm, std.: 3.38mm)



(b) Fine-tuning (mean: 496.34mm, std.: 626.92mm)



(c) Ours (mean: 2.58mm, std.: 4.22mm)

Figure 2: The heatmap indexed by (m, n) presents the mean (left) and standard deviation (right) of the approximated SDF accuracy $f(\boldsymbol{x}^m; \theta^n)$. Notice that the range of fine-tuning baseline differs from others for better visualization. The mean and std. are calculated with values in the lower triangle to measure the accuracy of memory.

It can also be understood from Fig. 1 that the training time of batch re-training will be linearly growing with the increasing number of frames. Ours, on the other hand, maintain a constant training time and memory consumption without forgetting.

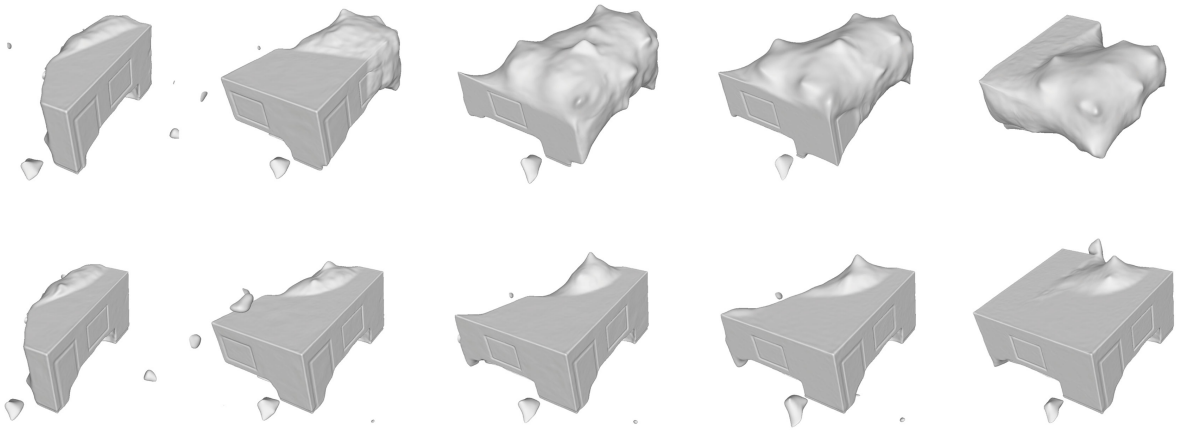


Figure 3: Recovered scene geometry from the implicit mapping function at frame 100-500. Top row: without the replayed buffer to regularize previously visited areas, the network tends to forget the geometry gradually; Bottom row: a simple solution of maintaining a fixed size of buffer can effectively preserve pre-visited scene geometry with high-frequency details.

3. Memory and Predictor

We here provide a more thorough analysis of Fig. 6 of our main paper. As illustrated in Fig. 2, the heat map indexed by (m, n) presents the mean and the standard deviation in meter of the SDF $f(\boldsymbol{x}_i^m; \theta^n)$ approximated using the network θ^n for the observation $\boldsymbol{x}^m$ (batch training baseline is trained with the entire sequence of data). *The upper triangle denotes the prediction performance for unseen areas as $m > n$ (not applicable for batch training baseline), while the lower triangle denotes the memory performance for previously seen areas as $m < n$.* We can see that the proposed method achieves comparable results of memory against the batch re-training baseline, while the fine-tuning baseline suffers from catastrophic forgetting.

The heatmap also demonstrates the forward transfer ($\rightarrow$) and backward transfer ($\downarrow$) performance [1] of each method. It is clear that at this stage, a simple MLP does not perform well for predicting unseen areas. **The incorporation of geometry prior for better prediction** may be an interesting follow-up suggestion.

4. Analysis of the Replayer Buffer

We provide additional 3D visualization of the scene geometry changes over time as a complement to Fig. 7 of our main paper, depicting the role of the replayed buffer. As illustrated in Fig. 3. the replayed buffer of zero level-set samples properly regularizes the previously visited surface information and alleviates the catastrophic forgetting issue. Further experiments can be conducted to **analyze the relationship between the convergence rate and the sample selection strategy**. It would be the key to achieving real-time performance based on the continual neural mapping paradigm.

5. Experiments on the TUM Dataset

We provide additional qualitative results on the real-world TUM dataset [3] from different viewpoints. We refer readers to the supplementary video for better visualizing the recovered model. As illustrated in Fig. 4 and 5, we can see that the proposed continual neural mapping setting can mitigate the side effect of noisy observations. However, it is also noteworthy that high-frequency signal recovery and denoising are controversial. **How to use a single network to recover fine-grained geometry details with sensor noise reduced continually** is one interesting extended case for continual neural mapping.

6. Error Map of Extracted Mesh Model

As illustrated in Fig. 6, we here present the visualization of the extracted mesh accuracy[1] of [4, 2]. The parameter settings for the two methods are specified as follows: for RoutedFusion [4][2], we use a voxel size of 2cm with 512^3 voxels to best fit the entire scene within the volumetric field using a NVIDIA GeForce RTX 2080Ti; for LIG [2][3], we set a part size of 0.25 as suggested by the paper. Though the proposed implicit representation does not rely on the discretized volume, we can maintain active volume indices to extract clean mesh triangles with state-of-the-art accuracy. **The incorporation of priors to eliminate spurious zero level-set due to incomplete observations** is worth studying (similar to Sec. 3).

[1] CloudCompare: `http://www.danielgm.net/cc/`
[2] RoutedFusion. https://github.com/weders/RoutedFusion
[3] LIG. https://github.com/tensorflow/graphics

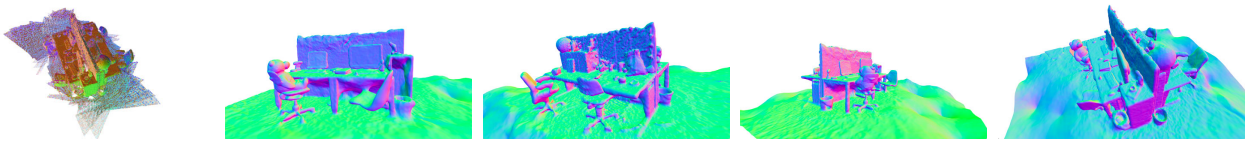


Figure 4: Extracted mesh (right) with the network that continually learned till the last frame. The mesh is visualized with the vertex normal to present the smooth surface even if the network is learned from noisy sequential data (left).

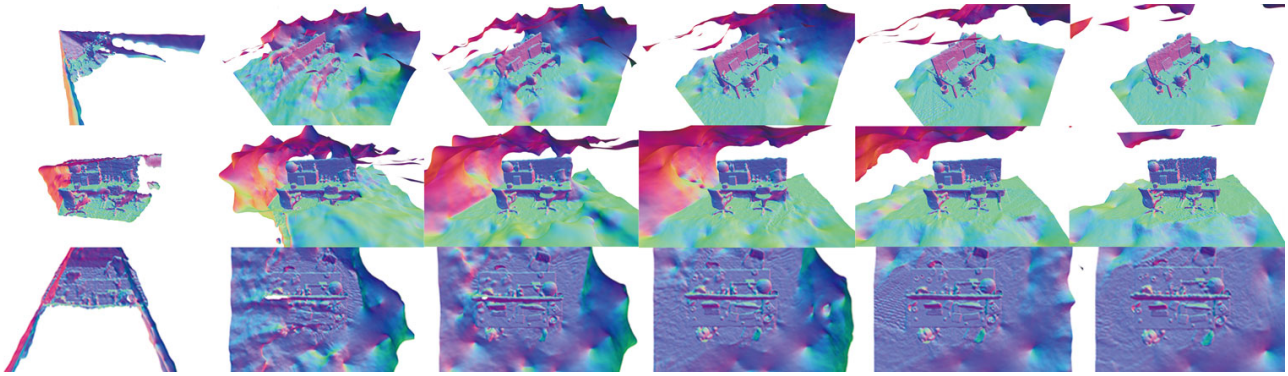
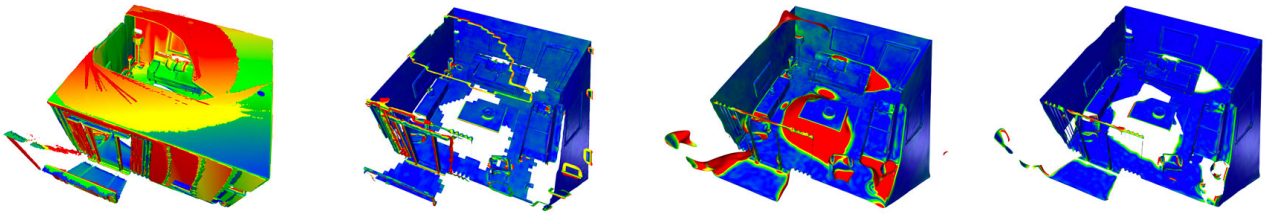


Figure 5: Self-improved SDF approximation on TUM *long office* dataset. Each row is recorded at a specific view point. Each column denotes the mesh extracted from the network trained with the frame t ($t = 30, 480, 930, 1380, 1830$ respectively). We refer readers to the supplementary video for better visualization.



(a) RoutedFusion (b) LIG (c) Ours (d) Ours (masked)

Figure 6: The visualization of mesh accuracy for each method (scaled up to 5cm, red: low accuracy, blue: high accuracy). We here provide the back face visualization to see the internal error distribution. RoutedFusion generates thick faces. Therefore, the internal view of the room cannot be seen. It is clear that our *continuous SDF approximation* leads to low accuracy in *previously unseen* areas, while the geometry recovered in already seen areas (masked) outperforms the state-of-the-art methods.

References

[1] 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.

[2] 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.

[3] 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)*, pages 573–580, 2012.

[4] 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.