New Problems in Active Sampling for Mobile Robotic Online Learning

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

4th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

5th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

6th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—AI models deployed in real-world tasks (e.g., surveillance, implicit mapping, health care) typically need to be online trained for better modelling of the changing real-world environments and various online training methods (e.g., domain adaptation, few shot learning) are proposed for refining the AI models based on sampled training input from the real world. However, in the whole loop of AI model online training, there is a section rarely discussed: sampling of training input from the real world. In this paper, we show from the perspective of online training of AI models deployed on edge devices (e.g., robots) that several problems in sampling of training input are hindering the effectiveness (e.g., final training accuracy) and efficiency (e.g., online training accuracy gain per epoch) for the online training process. Notably, the online training relies on training input consecutively sampled from the real world and suffers from locality problem: the consecutive samples from nearby states (e.g., position and orientation of a camera) are too similar and would limit the training efficiency; on the other hand, while we can choose to sample more about the inaccurate samples to better final training accuracy, it is costly to obtain the accuracy statistics of samples via traditional ways such as validating, especially for AI models deployed on edge devices. These findings aim to raise research effort for practical online training of AI models, so that they can achieve resiliently and sustainably high performance in real-world tasks.**

I. Introduction

Online learning refers to real-time training a pre-trained Artificial Intelligence (AI) model on training inputs consecutively sampled from the real world for various edge AI applications (e.g. transportation, human language processing, implicit SLAM) deployed on edge devices (e.g., robots), so that the model can adapt to changing real-world environments and retain high performance. Various online learning methods have been proposed (e.g., domain adaptation, few shot learning) in pursuit of high training accuracy on given samples. While their samples are traditionally collected by human labors or along a given routine that lacks interaction with training, automating the sampling process in reaction to real-time training statistics (i.e., active sampling) has the potential to further boost training accuracy, beside saving human labor.

Active sampling was first proposed in the problem of active simultaneous localization and mapping (SLAM): a robot automatically decides its sampling destination in real-time reaction to estimated localization quality and mapping quality: either sample areas of high mapping accuracy to optimize localization quality or samples areas of low mapping accuracy to optimize mapping quality. Instead of aimlessly circulating, active sampling in active SLAM boosts both mapping speed and accuracy. For online learning, enabling active sampling has the potential to achieve both high *training effectiveness* (e.g., final training accuracy) and high *training efficiency* (e.g., online training accuracy gain per epoch) by automatically sampling areas of high potential training accuracy gain and avoiding those low.

Although active SLAM methods shed light on the design of active sampling for online learning, in this paper, we show that several problems are hindering the training effectiveness and training efficiency in active sampling for online learning.

First, we observed low training efficiency during the consecutive sampling and named it the locality problem: while the edge device is moving around local state space (e.g., nearby position and orientation), the consecutive samples are often too similar, limiting training accuracy gain between consecutive samples. Fig.1 shows a typical lowland of training accuracy gain around the starting state of the robot around a place of interest in an implicit SLAM (building dense 3D map via online learning) task. To complete sampling of an area of interest, the robot has to sample around in the lowland of training accuracy gain, leading to low training efficiency. In our evaluation, compared with consecutive samples, 46.14% of the key samples selected from the consecutive samples achieved the same level of training accuracy, implying 53.86% of the samples were wasted due to the locality problem.

Second, real-time estimation of potential training accuracy

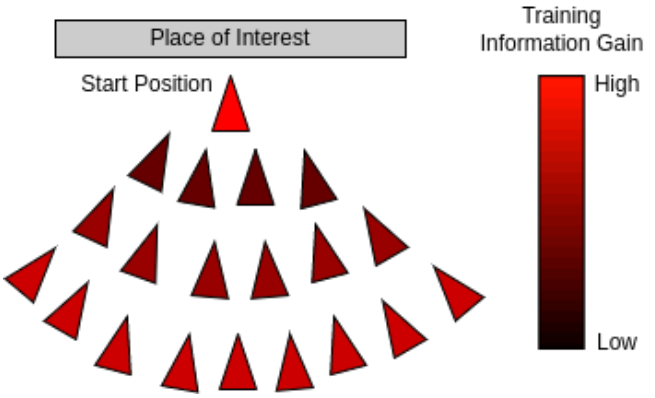


Fig. 1: Locality problem causes a lowland of training accuracy gain around the starting state

gain of the training AI model for real-time active sampling decision making is difficult. While the estimation of accurate potential training accuracy gain in AI training is yet an open problem, an approximate way to quantify potential training accuracy gain is to validate training samples and those with low training accuracy have high potential training accuracy gain. But it is too computation-intensive for edge devices and easily breaks the control loop. We evaluated that the validation of the training accuracy in an implicit SLAM task typically takes 30 to 200 seconds and the robot has to wait for such a long time before each decision making.

The key reason of these problems is that AI training process is probabilistic and implicit, different from the traditional deterministic and explicit SLAM process. [TODO*AI training requires sampling from multiple angles and distances, cannot be resolved by single sampling; such sampling requires real-time estimation of potential training accuracy gain to avoid locality...*] These problems together caused the implicit SLAM process in our evaluation be slowed down by xx% to reach a same high accuracy and reduced mapping accuracy by xx% after training for the same time, compared with the case with selected key frames and [TODO*no estimation time cost*].

In this paper, we take the first step to reveal and evaluate in both quality and quantity the problems hindering the active sampling for online learning from achieving both high training effectiveness and high training accuracy. These findings aim to raise research effort for practical active sampling for online learning of AI models, so that they can achieve resiliently and sustainably high performance in real-world tasks. As we are borrowing the idea of active sampling from active SLAM, we choose implicit SLAM as the main evaluation item for simplicity since implicit SLAM shares a similar task with active SLAM.

The rest of the paper is organized as follows: the second chapter provides background; the third chapter describes in detail about the problems and the estimation methodology; the final chapter concludes.

II. Background

Online Training. Machine learning (ML) approaches are generally trained for a specific task on a dedicated training set. However, in many real-world applications, Labeling datasets are very expensive, and the data distributions can differ or even change over time. Therefore, Some unsupervised methods are proposed to learn knowledge from unlabeled data and make the machine learning model to adapt the new dynamic environment. For example, dynamic unsupervised domain adaptation methods [1] is proposed to adapt a pretrained model to a new environment by training it with both unlabeled data from the dynamic environment.

With the rapid development of such methods, robots can adapt their pretrained models to new scenarios(e.g., domain shifts or changing data distributions) after training with online collected data to retain the high accuracy of the models. As another example, neural implicit representations have recently become popular in simultaneous localization and mapping (SLAM), especially in dense visual SLAM. This method enables high-fidelity and dense 3D scene reconstruction by collecting unlabeled image sequences with RGB-D sensors in real-time. We envision the prosperity of these multi-robot collaborations and unsupervised learning methods are making online training on real-time collected data on multi-robot realistic.

Traditional Active SLAM Methods. In the interest of exploring the environment by planning the path of mobile robots, active SLAM combines SLAM with path planning. This improves and speeds up the SLAM algorithm's ability to produce high-precision maps. The three active vision issues (localization, mapping, and planning) are combined by active SLAM. Robots can now autonomously carry out localization and mapping tasks, which helps to improve the accuracy of both those tasks and the representation of the environment. This topic has been studied before [2] came up with the phrase "Active SLAM," mostly as known as "exploration problems" [3], [4].

Specifically, iRotate [5] offers an active visual SLAM approach for omnidirectional robots because the static camera restricts the freedom of visual information acquisition. During the path execution, the robot can actively and continuously control its camera heading to maximize the environment coverage by taking advantage of its omnidirectional nature. The robot can significantly speed up the information-gathering process and quickly reduce the level of map uncertainty by actively performing coverage. In particular, these methods need to explicitly build maps before they can work, so they cannot be directly applied to the implicit SLAM framework. At the same time, the memory overhead of building explicit maps is large, and the lack of memory resources of robots often cannot support such active SLAM methods.

Dense Visual SLAM. Visual SLAM is an online approach that incrementally creates the map of an environment while localizing the robot within it. Meanwhile, it is an area that has received much attention in both industry and academia.

Specifically, sparse visual SLAM algorithms estimate accurate camera poses and only have sparse point clouds as the map representation, While sparse visual SLAM algorithms estimate accurate camera poses and only have sparse point clouds as the map representation, dense visual SLAM approaches focus on recovering a dense map of a scene, which makes the method very suitable for 3D reconstruction. Dense tracking and mapping (DTAM), proposed by Newcombe et al. [6], was the first fully direct method in the literature.

Neural Implicit-based SLAM. Neural implicit representations [7] have shown great performance in many different tasks, including 3D reconstruction [8]–[11], scene completion [12]–[14], novel view synthesis [15]–[19], etc. In terms of SLAM-related applications, some works [20], [21] try to jointly optimize a neural radiance field and camera poses, but they are not suitable for large objects or wide range of camera motion. In addition, some recent works [22], [23] can support large-scale mapping, but they mainly rely on state-of-the-art SLAM systems like ORB-SLAM to obtain accurate camera poses, and do not produce 3D dense reconstruction.

NICE-SLAM [24] and iMAP [25] are the most famous two SLAM pipelines using neural implicit representations for both mapping and camera tracking. Since iMAP uses a single MLP as the scene representation so they are only adapt to small scenes,whereas NICE-SLAM, which uses hierarchical feature grids and small MLPs as the scene representation, can scale up to considerably bigger interior spaces. Nevertheless, it calls for RGB-D inputs, which restricts their use in outdoor settings or when only RGB sensors are available. In order to solve this problem, a new work named NICER-SLAM [26]was proposed, which is the first dense RGB-only SLAM, optimizes mapping and tracking end-to-end and also allows the high-quality synthesis of new views.

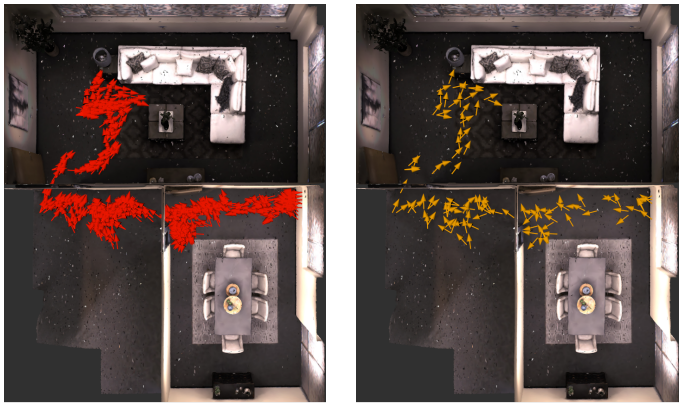
III. Problems in Active Sampling

This section explores how the two main problems in active sampling for online training, locality problem and high time consumption of estimation of potential training accuracy gain, affect online training in both quality and quantity.

Testbed. We use a four-wheel robot equipped with an NVIDIA Jetson Xavier NX [27] as the main testbed. We select a state-of-the-art implicit SLAM method, NICE SLAM as our main evaluation item and integrates it as a ROS package deployed over the robot. We connect the robot with RGBD input from a popular dense SLAM dataset, Habitat [28], [29], and port the real-time RGBD input to NICE SLAM backend to online train an implicit model of a dense map of the environments. The used implicit model has 7 million parameters. At best effort, roughly every 10 seconds the NICE SLAM backend takes an RGBD input as a key frame and trains the implicit model against the key frame together with previous sampled key frames for 30 iterations. We disable tracking in NICE SLA and use the ground-truth positions and orientations in online training, because we want to explore how the aforementioned problems affect mapping error and avoid the influence of tracking error.

Consecutive Sampling. If a sampling plan (plan of the following movements) is made, the robot executes the sampling plan by moving around its local state before the next sampling plan is made, which results in consecutive sampling that suffers locality problem (shown in Fig. 2a). To make sampling plans, every 10 key frames being sampled, we validate the implicit model of NICE SLAM to get the explicit dense map, and then input the dense map to iRotate [5] to decide the following sampling positions and orientations that optimize training accuracy gain.

Sparse Sampling. In contrast to the consecutive sampling, to reveal the locality problem, we select sampling positions and orientations from those of the aforementioned consecutive sampling, but maximizing the distances from the previous sampled positions and orientations (sparse sampling) (shown in Fig. 2b). We virtualize a robot that can teleport in the environment (move beyond its local state) to avoid the local low land of training accuracy gain and execute sparse sampling. This is achieved by spawning the RGBD camera at any given position and orientation in Habitat dataset. Note this is meant to contrast consecutive sampling and not meant to be optimal.



(a) Consecutive sampling: robot consecutively sampling around local states. (b) Sparse sampling: robot virtually teleporting to selected sparse poses.

Fig. 2: Consecutive / sparse sampling.

A. Locality Problem

[TODO] [TODO *Pausing sampling during consecutive sampling and then resuming at a distance is equivalent to inserting training steps with no training accuracy gain to sparse sampling from the view of wallclock time.*]

B. Estimation of Potential Training Accuracy Gain

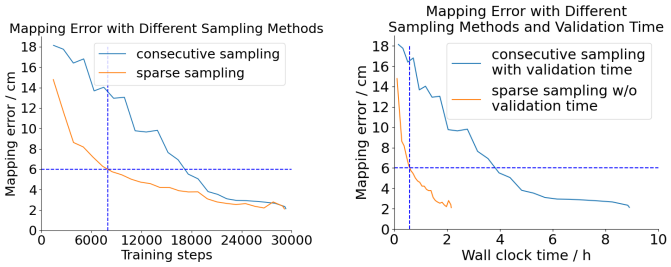
[TODO *Validation of implicit model helps traditional active SLAM methods to actively sample the environment for NICE SLAM, but time consuming; time consumption even increases as the sampling proceeds and with increased model sizes.*]

C. Others

[TODO*Cost of distributed training*]

IV. Conclusion

[TODO*conclusion*]



(a) Locality problem in consecutive sampling lowers training accuracy gain per training step

(b) Validation time cost further deteriorates training accuracy gain against wallclock time.

Fig. 3: Locality problem and validation time cost evidently lower training efficiency.

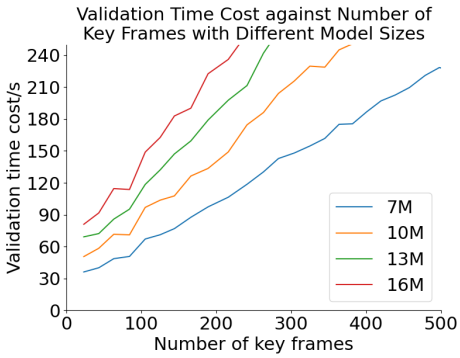


Fig. 4: Time consumption validating Implicit SLAM model with various model sizes (M: millions of parameters) and number of key frames

References

[1] Q. Tian, Y. Zhu, H. Sun, S. Chen, and H. Yin, "Unsupervised domain adaptation through dynamically aligning both the feature and label spaces," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 12, pp. 8562–8573, 2022.

[2] A. Davison and D. Murray, "Simultaneous localization and map-building using active vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 865–880, 2002.

[3] C. Stachniss, D. Hahnel, and W. Burgard, "Exploration with active loop-closing for fastslam," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 2. IEEE, 2004, pp. 1505–1510.

[4] J. Moody, S. Hanson, and R. Lippmann, "Active exploration in dynamic environments," in *Advances in Neural Information Processing Systems 4*. Citeseer, 1992.

Fig. 5: Time composition using distributed training with various model sizes

[5] E. Bonetto, P. Goldschmid, M. Pabst, M. J. Black, and A. Ahmad, "irotate: Active visual slam for omnidirectional robots," *Robotics and Autonomous Systems*, vol. 154, p. 104102, 2022.

[6] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *2011 international conference on computer vision*. IEEE, 2011, pp. 2320–2327.

[7] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[8] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4460–4470.

[9] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 165–174.

[10] S. Peng, C. Jiang, Y. Liao, M. Niemeyer, M. Pollefeys, and A. Geiger, "Shape as points: A differentiable poisson solver," *Advances in Neural Information Processing Systems*, vol. 34, pp. 13 032–13 044, 2021.

[11] S. Liu, Y. Zhang, S. Peng, B. Shi, M. Pollefeys, and Z. Cui, "Dist: Rendering deep implicit signed distance function with differentiable sphere tracing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2019–2028.

[12] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, "Convolutional occupancy networks," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer, 2020, pp. 523–540.

[13] S. Lionar, D. Emtsev, D. Svilarkovic, and S. Peng, "Dynamic plane convolutional occupancy networks," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1829–1838.

[14] C. Jiang, A. Sud, A. Makadia, J. Huang, M. Nießner, T. Funkhouser *et al.*, "Local implicit grid representations for 3d scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6001–6010.

[15] C. Reiser, S. Peng, Y. Liao, and A. Geiger, "Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 335–14 345.

[16] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7210–7219.

[17] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, "Block-nerf: Scalable large scene neural view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8248–8258.

[18] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 318–10 327.

[19] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. T. Barron, and P. P. Srinivasan, "Ref-nerf: Structured view-dependent appearance for neural radiance fields," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 5481–5490.

[20] S.-F. Chng, S. Ramasinghe, J. Sherrah, and S. Lucey, "Gaussian activated neural radiance fields for high fidelity reconstruction and pose estimation," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*. Springer, 2022, pp. 264–280.

[21] R. Clark, "Volumetric bundle adjustment for online photorealistic scene capture," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6124–6132.

[22] C.-M. Chung, Y.-C. Tseng, Y.-C. Hsu, X.-Q. Shi, Y.-H. Hua, J.-F. Yeh, W.-C. Chen, Y.-T. Chen, and W. H. Hsu, "Orbeez-slam: A real-time monocular visual slam with orb features and nerf-realized mapping," *arXiv preprint arXiv:2209.13274*, 2022.

[23] A. Rosinol, J. J. Leonard, and L. Carlone, "Nerf-slam: Real-time dense monocular slam with neural radiance fields," *arXiv preprint arXiv:2210.13641*, 2022.

[24] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "Nice-slam: Neural implicit scalable encoding for slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 786–12 796.

[25] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "imap: Implicit mapping and positioning in real-time," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6229–6238.

[26] Z. Zhu, S. Peng, V. Larsson, Z. Cui, M. R. Oswald, A. Geiger, and M. Pollefeys, "Nicer-slam: Neural implicit scene encoding for rgb slam," *arXiv preprint arXiv:2302.03594*, 2023.

[27] "The World's Smallest AI Supercomputer." [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-xavier-nx/

[28] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. Chaplot, O. Maksymets, A. Gokaslan, V. Vondrus, S. Dharur, F. Meier, W. Galuba, A. Chang, Z. Kira, V. Koltun, J. Malik, M. Savva, and D. Batra, "Habitat 2.0: Training home assistants to rearrange their habitat," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[29] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A Platform for Embodied AI Research," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.