Challenges Towards Active Sampling for Online AI Model Refinement

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 refined for better modelling of the changing real-world environments and various online refinement 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 refinement, there is a section rarely discussed: sampling of training input from the real world. In this paper, we show from the perspective of online refinement of AI models deployed on edge devices (e.g., robots) that several challenges 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 refinement process. Notably, the online refinement 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 refinement of AI models, so that they can achieve resiliently and sustainably high performance in real-world tasks.**

I. Introduction

Artificial Intelligence (AI), with its huge representation power, is being more prevalently deployed and becoming the game-changer in various real-world applications such as transportation, image processing, human language processing, health care, etc. AI models are typically trained on well-prepared human-labelled datasets, but it is costly and often impossible for the datasets to cover all possible real-world situations and environments and potentially limits their performance in real-world tasks. Thus, online refinement is crucial for AI models deployed in real-world tasks and many online refinement methods (e.g., domain adaptation, few shot learning, long term learning) are proposed so that the AI model can be refined and adapted to changing real-world environments after training on real-time sampled training inputs without the need for further human-labelling. However, in the whole loop of online refinement, there is an important section rarely discussed: the online sampling strategy of the training inputs for better online AI model refinement. Traditionally, TODO [*traditional sampling strategy: routine collection / human select*] TODO [*introduce active sampling*]

The active sampling and online refinement can have positive reaction between each other to achieve high effectiveness (e.g., final training accuracy) and efficiency (e.g., online training accuracy gain per epoch). For example, a robot is monitoring the biodiversity of an area by taking pictures and feeding the pictures into object detection and recognition models to recognize the creatures, and it needs to refine the models to adapt to changing weather and plant shading. Given the statistics of training accuracy on different sampled objects and their related areas, we can guide the robot to further sample the objects that are inaccurately modelled (e.g., have low accuracy) and the related areas for further refinement, so as to achieve higher final training accuracy. The robot can also avoid areas related to objects that are well modelled so that the recognition models are always trained on inaccurately modelled objects to achieve high training accuracy gain per epoch. These together help the robot to adapt the deployed AI models to changing environments as soon as possible to retain high performance.

The ideas of sampled input and the related data processing interacts with the sampling strategy can be traced back to the problem of active SLAM. TODO [*How the traditional active SLAM works.*]

Although traditional active SLAM methods shed light on the design of online sampling strategy for better online refinement, the characteristics of online refinement of AI models brings about various challenges towards active sampling for online refinement. First, we observe that the consecutive sampling of a robot suffers the problem of sampling locality: take image processing AI models as an example, while the robot is moving in local state space (i.e., nearby position and orientation), the consecutive samples from a camera are often too similar, limiting training accuracy gain. But simply avoiding local state space and sampling another area is yet incorrect: when the robot state is shifted for a distance, the training accuracy gain of sampling the same area would raise again, implying that sampling in multiple distances and multiple angles is important for final training accuracy. As shown in Fig.1 which is about the training accuracy gain of an implicit SLAM task about an area of interest, we can find that there is a lowland of training accuracy gain around the starting state of the robot. As a result, to complete sampling of an area of interest, the robot has to move in and out the lowland of training accuracy gain during sampling, lowering the average training accuracy gain per sample.

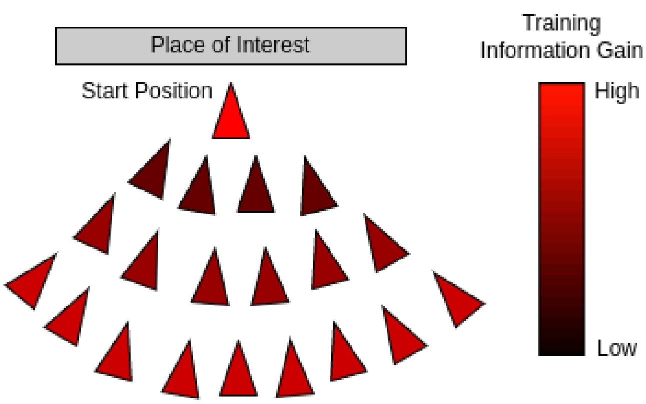

Fig. 1. Locality problem causes a lowland of training accuracy gain around the starting state

Second, different from traditional SLAM tasks that have an explicit representation of a map to guide decision making, it is difficult to obtain statistics of training accuracy of the training AI model. Typically, quantifying the training accuracy of the training model relies on statistical methods such as validating, which is computation-intensive for robots and infects the ongoing online training task. Without the real-time statistics of training quality, it is difficult to decide the destination for the robot to mobile in active sampling.

Third, TODO [*use the distributed computation power of edge devices.*]

In this paper, we take a simple by yet representative online refinement task, implicit SLAM, as a study case and reveal how the above challenges influence the active sampling for its online refinement in both quality and quantity. TODO [*data presentation*] TODO [*aim to raise research effort*]

II. Background

Online AI Model Refinement. 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.

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. [2], was the first fully direct method in the literature.

Neural Implicit-based SLAM. Neural implicit representations [3] have shown great performance in many different tasks, including 3D reconstruction [4]–[7], scene completion [8]–[10], novel view synthesis [11]–[15], etc. In terms of SLAM-related applications, some works [16], [17] 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 [18], [19] 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 [20] and iMAP [21] 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 [22]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.

Active Mapping/SLAM. 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 [23] came up with the phrase "Active SLAM," mostly as known as "exploration problems" [24], [25].

Specifically, iRotate [26] 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.

III. Challenges towards Active Sampling

This section explores the challenges towards active sampling for online AI model refinement in both quality and quantity. TODO [*methodology, figs*]

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] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

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

[26] 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.