Embodied Depth Prediction

Qingquan Bao*[1], Yilun Du*[2], Feng Chen[3], Joshua B. Tenenbaum[2],
Tomas Lozano-Perez[2], Leslie Kaelbling[2], Chuang Gan[4,5]

***Abstract*— We study the problem of embodied depth prediction, where an embodied agent in an environment must learn to accurately estimate the depth of its surroundings. Such a task can be useful for embodied AI where it is highly desirable to accurately predict 3D structure when deploying robots in novel environments. However, directly using existing pre-trained depth prediction models in this setting is difficult as images are often captured in out-of-distribution viewpoints. Instead, it is important to construct a system that may adapt and learn depth prediction by interacting and gathering information from the environment and which may utilize the rich information in past observations captured from ego-motion. Towards this problem, we propose a framework for actively interacting with the environment to learn depth prediction, leveraging both explorations of new areas of space and exploration of areas of space where depth prediction is inconsistent. To exploit the rich information captured from past observations in the embodied setting, we further jointly utilize current and past image observations and their corresponding egomotions to predict depth. We illustrate the efficacy of our approach in obtaining accurate depth predictions in both simulated and real household environments. More information is available at https://embodied-depth.github.io/.**

I. Introduction

Depth estimation is a long-standing problem of interest in computer vision and robotics. Recent approaches have enabled sharp depth prediction on complex scenes using large labeled datasets of images [46] or videos [18]. However, directly applying existing depth prediction approaches to embodied environments is difficult, as observations of the world are often captured at odd viewpoints not seen in training datasets (Fig. 1) and further consists of both novel objects and scene arrangements never seen before. Instead, to accurately predict depth in embodied settings, an agent must learn to refine its depth predictions through *active interaction* with the environment, and explore and gather images in the environment so that it may accurately estimate depth across different viewpoints. While previous work has explored self-supervised depth prediction [70], [17], [18], [1], it assumes access to offline datasets of monocular RGB videos. In contrast, an agent in an embodied environment must actively interact and explore to gather such data.

The ability to adapt depth prediction models is immensely helpful for robotics. Robots are often deployed in novel environments, where underlying visual viewpoints are substantially different than those depth prediction models are trained on, making transferring existing depth models difficult.

*The first two authors contributed equally to the paper.
[1] University of Pennsylvania [2] Massachusetts Institute of Technology, [3] Tsinghua University, [4] UMass Amherst, [5] MIT-IBM Watson AI Lab

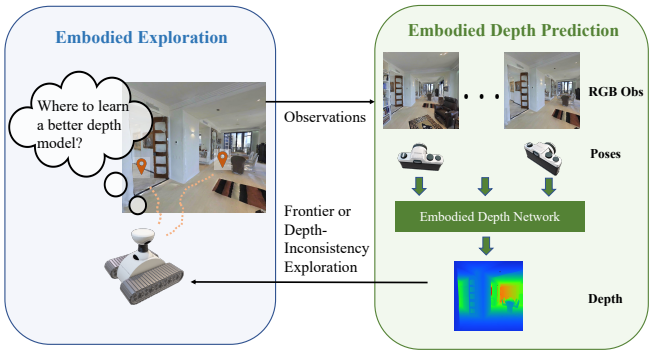


Fig. 1: **Embodied Depth Prediction.** By exploring an environment, our approach learns efficiently to predict depth accurately from a wide range of viewpoints.

Furthermore, using RGB-D sensors to gather depth data in the environment is both expensive and noisy, often missing reflective and metallic surfaces [19], [26]. Simultaneously, accurate depth prediction is extremely helpful in robotics, enabling accurate navigation [42], [2], and obstacle avoidance [56], [25], as well as robust object manipulation [36], [29]. Thus in this paper, we propose the problem of embodied depth prediction, where a robot, without access to a depth sensor, gathers new images in the world so that it can adapt and learn a depth prediction model. We study this problem in indoor scenes which are particularly relevant to robotics but also where depth prediction is especially complex due to occlusions and obstacles.

Given an initially poor depth estimation network, how can we accurately explore an environment to obtain better depth prediction? One possible solution is to leverage recent work in self-supervised monocular depth prediction [70], [17], [18], [1]. Concretely, we use the gathered RGB observations in combination with photometric loss to refine our depth prediction network. However, a challenge remains as to how we may collect effective data for accurate depth prediction. We wish to explore and construct a diverse dataset of RGB observations across all locations in the environment, but also prioritize locations where our depth prediction is likely to be incorrect *i.e.*when it is inconsistent with ego-motion. To prioritize both sets of objectives, we propose a joint exploration policy that uses both frontier exploration, based on current depth predictions, as well as inconsistency-driven exploration, which prioritizes 3D areas where depth predictions are inconsistent with robot ego-motion.

In the setting of embodied depth prediction, an agent sees both its current image observation, as well as a history of past observations and the associated ego-motions. Such past

observations can be greatly informative to depth prediction – limited optical flow in the presence of substantial ego-motion indicates a faraway object. Thus to further improve embodied depth prediction performance, we propose to leverage both past and current observations and their associated egomotion to predict depth. We use an off-the-shelf optical flow network to triangulate a coarse depth map across past image observations and use this fused depth information to accurately derive the final depth predictions.

In summary, our work makes the following contributions.

- We propose and formulate the embodied depth prediction problem.
- We present an active exploration policy that gathers data for effective depth prediction.
- We present a depth prediction architecture that can leverage both current and past observations collected with ego-motion, fused together through optical flow, to jointly predict depth in the embodied setting.
- We illustrate the efficacy of our approach to depth prediction in both simulated and real environments. In real environments, we find unique challenges present in real environments that are not in simulated ones, and we curate and will release a dataset/benchmark for further research in this area.

II. Related Work

Embodied Learning. Embodied learning, where a physical agent must actively interact, navigate and learn from its environment, has seen increased interest in vision in recent years with the advent of simulators such as Matterport3D [4], Habitat [53], ThreeDworld [15]. These simulators enable agents to actively explore and obtain realistic visual observations from the environment, enabling the construction of new tasks such as visual navigation [63], [62] and instruction following [50]. Furthermore, embodied learning provides unique challenges for traditional computer vision tasks, as an agent must now actively gather the data it learns from, such as object detection [13], [33], and object segmentation [65] and representation learning [8]. Similarly, in robotics, embodied learning has also been extensively studied for long term autonomy [22], [35], [37], [9], [34]. Across both domains, accurate depth maps are often crucial for accurate exploration and learning. In this work, we explore how we may learn to estimate such depth maps in an unsupervised manner, where an agent may freely move in its underlying environment.

Self-Supervised Monocular Depth. Self-supervised monocular depth learning has seen significant development in recent years. Zhou *et al.* [70] proposed using photometric loss to train depth and ego-motion networks, which was further improved upon by subsequent works [55], [3], [51], [21] such as the use of stereo image pairs [17] and occlusion solving [18], [20]. Additional constraints such as ICP regularization [38], two-view sparse triangulated depth supervision [69], and geometric constraints [1], [5] have also been introduced. Some methods have utilized recent frames to predict current depth with cost volumes [60] and recurrent neural networks [44], [58], [68], or for test-time training [5], [3]. Existing approaches operate without the assumption of ego-motion, which is easy to acquire in the embodied setting, and operate primarily on large outdoor scenes that differ significantly from the indoor environments studied in our work.

3D Reconstruction and Mapping. 3D Reconstruction and Mapping have been extensively researched in computer vision and robotics. Visual SLAM [32] estimates the camera's pose and reconstructs 3D environments with either pixel intensities [12], [11], [14] or feature-based matching [16], [43]. On the other hand, SFM and MVS reconstruct sparse or dense consistent 3D models from unordered image collections [48], [49], [66], [24], [57], [27]. Recently, Neural Radiance Field [41] approaches gain popularity, which synthesizes novel and realistic views of a scene from a set of images by continuously mapping 3D coordinates and view directions into color and densities [61], [39], [45], [10], [67]. In contrast to these 3D reconstruction approaches, which require dense multi-view images of a scene and focus on obtaining accurate ego-motion, consistent 3D models, or realistic images, we are interested in learning a neural network that can accurately estimate depth from the past few observed frames.

III. Embodied Depth Prediction

We begin in Sec. III-A by formulating the problem of embodied depth prediction as an optimization problem and propose a photometric loss function used to train the depth prediction model in Sec. III-B. We then discuss our method of modeling the indoor environment with predicted depth maps in Sec. III-C, and how an agent can explore an environment to collect observations to ensure accurate depth estimation in the embodied setting in Sec. III-D. Finally, in Sec. III-E, we discuss an embodied depth prediction architecture that exploits the multi-frame input and ego-motion of available in the embodied setting.

A. Problem Formulation

We consider an embodied environment in which an agent receives a continuous sequence of RGB images $\{I_1, \ldots, I_t\}$ and corresponding ego-motion $\{T_1, \ldots, T_t\}$, where $T_i = [R_i, u_i] \in \mathbb{R}^{3\times4}$ represents the relative transformation of ego-motion at time i. We assume that the relative transformation of ego-motion is reliable within a certain distance range R, such that the agent at time t can use as input $\{T_i \mid i \leq t, \|u(T_i^t)\|_2 \leq R\}$, where $u(T_i^t)$ denotes the translation component of the ego-motion T_i in the coordinate system of time t.

The goal of embodied depth prediction is to construct a policy π to actively explore the environment and a derived depth prediction model $f_{\theta^*(\pi)}$ learned from images gathered from π such that $f_{\theta^*(\pi)}$ accurately predicts depth of images drawn from a test distribution μ. We formulate this as the following optimization problem:

$$\begin{aligned} \max_{\pi} \quad & \mathbb{E}_{\mu}\left[M\left(f_{\theta^*(\pi)}(\mathbf{x}_\mu), d_\mu\right)\right] \\ \text{s.t. } & \theta^*(\pi) = \arg\min_{\theta} \mathbb{E}_{\pi}\left[L\left(f_\theta(\mathbf{x}_\pi)\right)\right], \end{aligned}$$

where $f_\theta(\cdot)$ is a depth prediction model parameterized by θ, M is a metric function that computes the accuracy of the

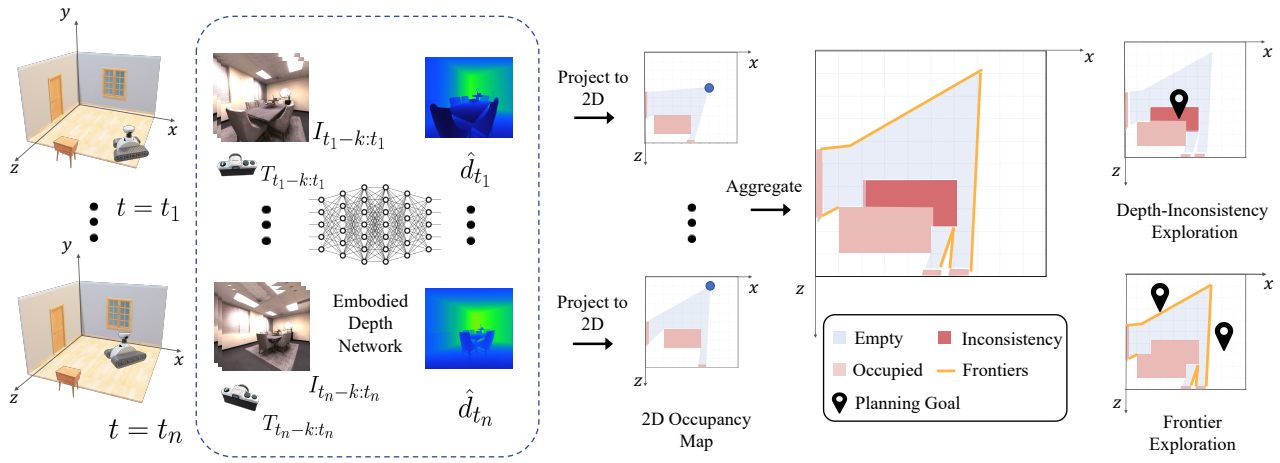


Fig. 2: **Method Overview.** Our approach to embodied depth prediction gathers and maintains a buffer of RGB images from the environment. A set of k contiguous data including images $I_{t_n-k:t_n}$ and their associated ego-motion $T_{t_n-k:t_n}$ are sampled from the replay buffer and used to generate the depth prediction $\hat{d}_{t_n}$ with Embodied Depth Network that is trained using photometric loss. More images are gathered with an active policy using a combination of frontier exploration and depth-inconsistency between predicted depth at nearby RGB observations.

depth prediction model, $\mathbf{x}_\mu$ and d_μ are a test input and its corresponding ground-truth depth from μ, $\mathbf{x}_\pi$ denotes the data collected by following the policy π, and $L(\cdot)$ is a photometric loss function used to train the model, which we introduce in Sec. III-B.

B. Photometric Depth Prediction Loss

To train our embodied depth network, we require a self-supervised loss function as the agent has no prior knowledge of the 3D structure of the environment. In line with previous work [17], [18], [70], we use a photometric loss L_{photo} and smoothness loss L_{smooth}, with our total training loss a weighted combination of both losses.

To construct a photometric loss, we render a warped image $I_{s\to t}$ using a differentiable bilinear interpolation [28], given the relative pose from time t to neighboring time s, and the current depth prediction $\hat{\mathbf{d}}_t$. We then compute the photometric loss with the SSIM function [59]:

$$L_P(I_t, I_{s\to t}) = \frac{\alpha}{2}(1-\text{SSIM}(I_t, I_{s\to t}))+(1-\alpha)\|I_t-I_{s\to t}\|_1.$$

To handle occlusions and disocclusions, we adopt the minimum photometric loss per pixel across all source images, following the approach in [18]:

$$L_{\text{photo}} = \min_{t'} L_P(I_t, I_{t'\to t}). \tag{1}$$

To encourage smoothness of depth, we use an $L1$ penalty with edge-aware smoothness over depth, following the approach in [17]. The loss function is given by:

$$L_{\text{smooth}} = |\partial_x d_t^*|e^{-|\partial_x I_t|} + |\partial_y d_t^*|e^{-|\partial_y I_t|},$$

where the mean-normalized inverse depth $d_t^* = d_t/\text{mean}(d_t)$ is used to prevent the shrinkage of the estimated depth.

C. Occupancy Map Construction and Navigation

To effectively navigate and explore, an agent must be able to accurately construct a 2D occupancy map of its environment as it goes. As both our odometry and our underlying depth prediction are unreliable, we construct a local 2D occupancy map of the nearby neighborhood which we will use downstream for exploration in Sec. III-D.

The occupancy grid map m is constructed by back-projecting depth maps $\hat{d}_k$ into 2D points z_k and we apply Bayesian filtering to update occupancy probabilities. Log-odds representation is used for efficiency:

$$l_{t,i} = l_{t-1,i} + P(m_i|z_t) - l_{0,i} \tag{2}$$

where $l_{t,i} = \log \frac{P(m_i|z_{1:t-1})}{1-P(m_i|z_{1:t-1})}$ is the log-odds of occupancy at each cell, $P(m_i|z_t)$ is the occupancy probability of each pixel, and $l_{0,i}$ is the prior log-odds of occupancy at pixel i.

The constructed occupancy grid map serves as a graph for the agent's planning and navigation. We use the A* algorithm to find the shortest path from the current position to the destination.

Algorithm 1 Embodied Exploration

Input: agent action to image/pose mapping $Env(\cdot)$, strategy switch threshold τ, training data distribution $\mu_{\mathbf{b}}$
Initialize: $f_\theta : \mathbb{R}^{T\times 3\times H\times W} \to \mathbb{R}^{1\times H\times W}$, data replay buffer $\mathbf{b}$ of length l, $t \leftarrow l-k$
while len($\mathbf{b}$) $\leq$ max data size **do**
  for batch $\sim \mu_{\mathbf{b}}$ **do** ▷ Model training
    $\theta \leftarrow \theta + \eta\nabla L_\theta$
  while new data $\leq$ r **do** ▷ New data collection
    for t **to** len($\mathbf{b}$) **do** ▷ Predict history depth maps
      $\hat{d}_t = f_\theta(I_t, ..., I_{t-N}, T_t^{t-N}, ..., T_t^{t-1})$
    if recent data size $\leq \tau$ **then**
      $\mathbf{a} \leftarrow$ Inconsistency$(\hat{d}_t, \hat{d}_{t-k})$
    else
      $\mathbf{a} \leftarrow$ Frontier$(\hat{d}_t, \hat{d}_{t-1}, ..., \hat{d}_{t-k})$
    $\mathbf{b}$.push$(Env(\mathbf{a}))$

D. Embodied Exploration

To gather data for effective depth learning, our embodied approach involves exploring as much of the area of the

environment as possible, as well as exploring regions where the depth estimation is uncertain, *i.e.*, frontier exploration and depth-inconsistency exploration respectively.

Embodied Training. After N frames of data collection, we collect and maintain a replay buffer of prior observations to train a depth prediction model. We provide an overview of our training pipeline in Algorithm 1.

Depth-inconsistency Exploration. We use inconsistencies between past depth predictions ($\hat{d}t$, $\hat{d}t-k$) and their relative camera poses (T_t^{t-k}) to guide data collection. First, We create local occupancy grid maps ($\mathbf{m}t$, $\mathbf{m}t-k$) from these depth predictions. After scanning each pixel i and identifying inconsistent ones satisfying $\mathbf{m}_t(i) \neq \mathbf{m}_{t-k}(i)$, connected inconsistent pixels form regions for exploration. We then sample an empty point from the center of the largest inconsistent region using a normal distribution as the planning goal. This strategy collects data where significant differences exist between current and past depth predictions, enhancing learning in uncertain regions.

Frontier Exploration. To ensure comprehensive data collection in diverse subareas (e.g., bedrooms and kitchens), we adopt the frontier exploration strategy. We identify frontier edge cells (empty points neighboring unknown cells) and group them into frontier groups. A minimum group size threshold determines significant groups. We randomly select a frontier group's center as the planning goal and explore the surrounding area for data collection. We alternate between frontier and depth-inconsistency exploration after a set number of iterations, balancing the exploration of novel regions with areas featuring inconsistent depth.

E. Embodied Depth Network

In the embodied setting, we have access to multiple prior RGB observations as well as agent ego-motion. We propose a new network, Embodied Depth Network (EDN) which exploits information uniquely available for embodied depth prediction.

To get an estimate of depth from ego-motion, we establish pixel-correspondences between past RGB observations and relative poses using optical flow. In particular, we employ RAFT [54] to compute a displacement $\Delta\mathbf{u}_i$ for every pixel position $\mathbf{u}_i^{(t)} = [x_i^{(t)}, y_i^{(t)}, 1]^T$ of the current frame I_t to its corresponding pixel in the past frame $I_{t-k}, k > 0$, such that $\mathbf{u}_i^{(t-k)} = \mathbf{u}_i^{(t)} + \Delta\mathbf{u}_i$. Given these correspondences and the relative poses T_t^{t-k} from t to $t-k$, we can compute a coarse depth map $\bar{d}_t$ by solving the following linear least-squares problem [30],

$$\bar{d}_t = \arg\min_{d_t} \sum_{k=1}^{N} \| \frac{\mathbf{u}_i^{(t-k)}}{\|\mathbf{u}_i^{(t-k)}\|} \times \left(T_t^{t-k} \mathbf{K}^{-1} \begin{bmatrix} \mathbf{u}_i^{(t)} d_t \\ 1 \end{bmatrix} \right) \|^2$$

where $T_t^{t-k} \in \mathbb{R}^{3\times4}$ denotes the relative pose, and $\mathbf{K} \in \mathbb{R}^{4\times4}$ is the camera intrinsic matrix.

We then compute the final output depth map $\hat{d}_t$ from a coarse depth map $\bar{d}_t$, by encoding it along with the current RGB image I_t into a refinement network which follows the DispNet architecture [18]. This overall architecture of EDN enables us to both exploit ego-motion to obtain coarse depth maps of the environment and to use visual observations to refine and correct depth.

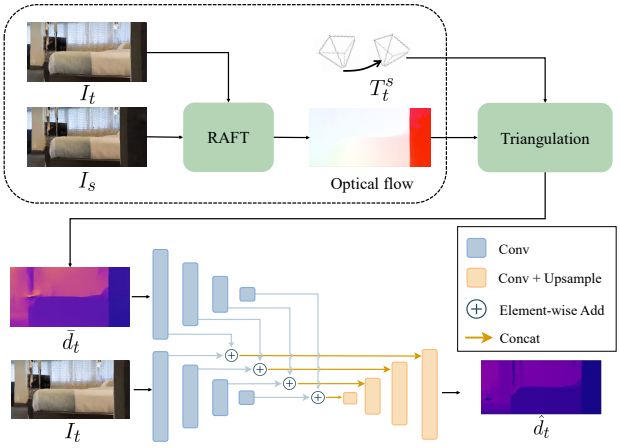


Fig. 3: **EDN Architecture.** EDN uses optical flow correspondences and ego-motion poses of past RGB observations to obtain coarse depth map of the scene. The coarse depth map is then refined with visual observations to obtain a final depth prediction.

IV. Experiments

A. Implementation Details

Experimental Environments. We conducted experiments in six different environments: three from Matterport3D [4] (JeFG25nYj2p, 17DRP5sb8fy, 2t7WUuJeko7) and three from Gibson [64] (Herricks, Eastville, Cordor). These environments are referred to as Large1, Medium1, Small1, Large2, Medium2, and Small2. We used Habitat-Sim [53] with the action space included forward movement (0.03 meters) and 2-degree left and right turns. This diverse set of environments allows us to assess our method's robustness.

Baselines. We compare EDN with three state-of-the-art depth-prediction networks: Monodepth2 [18], SC-depth [1], and Manydepth [60]. These networks are adapted to our embodied settings, with data collected through random or active exploration, totaling about 30,000 frames. In random exploration, the agent randomly selects forward movement, left turn, or right turn actions.

Evaluation Metrics. For depth evaluation, we use the accuracy under threshold ($\delta < 1.25$) following previous methods [70], [18]. To align predicted depth maps with the ground truth scale, we scale them by a factor that matches the median to the ground truth scale, as in [70]. The validation and test datasets consist of manually collected data throughout all visible regions, with approximately 2,000 to 3,000 frames per scene, ensuring diversity and representativeness.

B. Simulation Results

Embodied Depth Prediction. Our active exploration policy significantly improves depth prediction model performance as shown in Table I. Additionally, our embodied depth network combined with the active exploration policy outperformed all other approaches, achieving state-of-the-art performance on our dataset. Figure 4 visually demonstrates the superiority

Fig. 4: **Depth Prediction Visualization.** Our proposed approach demonstrates better accuracy and consistency in producing depth maps compared to the conventional random exploration strategy with Monodepth2. This can be attributed to the enhanced exploration capabilities of our active policy, which also leverages multi-frame inputs to effectively overcome limitations associated with single-frame inputs.

Method	Policy	Large1	Large2	Medium1	Medium2	Small1	Small2	Mean
Monodepth2	Random	0.436±0.014	0.404±0.004	0.464±0.031	0.507±0.025	0.504±0.017	0.525±0.004	0.473
SC-depth	Random	0.386±0.014	0.430±0.003	0.281±0.037	0.487±0.015	0.473±0.032	0.573±0.020	0.438
Manydepth	Random	0.179±0.040	0.386±0.009	0.216±0.052	0.419±0.011	0.289±0.012	0.572±0.008	0.344
Monodepth2	Active	**0.505**±0.030	0.526±0.004	**0.551**±0.012	0.563±0.017	0.635±0.048	0.699±0.018	0.580
SC-depth	Active	0.464±0.032	0.552±0.012	0.498±0.003	0.550±0.002	0.619±0.002	0.688±0.008	0.562
Manydepth	Active	0.484±0.030	0.462±0.010	0.531±0.006	0.557±0.011	0.585±0.027	0.724±0.017	0.557
EDN (Ours)	Active	**0.505**±0.014	**0.570**±0.023	0.548±0.006	**0.569**±0.013	**0.668**±0.025	**0.732**±0.006	**0.599**

TABLE I: **Quantitative Depth Results.** Accuracy of depth prediction $\delta < 1.25$ with standard error (across 3 experiment runs) with active or random exploration. Active exploration improves performance substantially

Method	Large1	Medium1	Medium2	Mean
Forward	0.420±0.006	0.433±0.017	0.525±0.002	0.459
Random	0.436±0.014	0.464±0.031	0.507±0.025	0.469
Frontier	0.476±0.008	0.547±0.005	0.539±0.018	0.521
Active	**0.505**±0.014	**0.551**±0.012	**0.563**±0.017	**0.540**

TABLE II: **Ablation of Exploration.** Effect of different active exploration policies on embodied depth prediction. Depth accuracy ($\delta < 1.25$) reported across 3 seeds.

of our approach over random exploration with Monodepth2, producing more accurate and consistent depth maps.

Our embodied depth prediction setting poses greater challenges than existing depth benchmarks on KITTI [40], where state-of-the-art models achieve accuracy above 90% [60]. In comparison to KITTI, indoor environments have substantial occlusions, challenging the learning depth maps with photometric loss. Furthermore, image distribution varies considerably across different positions in a large environment, with full navigation across the environment challenging due to narrow corridors connecting different regions. Finally, images collected in rotation motion, are more challenging to use for depth supervision compared to translation in KITTI.

Exploration Strategies. We conduct an ablation study to assess our exploration policy's effectiveness, comparing four strategies: forward exploration, random exploration, frontier-based exploration, and our active policy. Table II presents accuracy measurements for each strategy, and Fig. 5 illustrates their exploration trajectories.

Forward exploration, driven by continuous forward movement, limited data diversity due to trapping the agent. In contrast, random exploration covered a wider area however with more variance. Frontier-based exploration enabled passage through narrow spaces while incorporating depth-

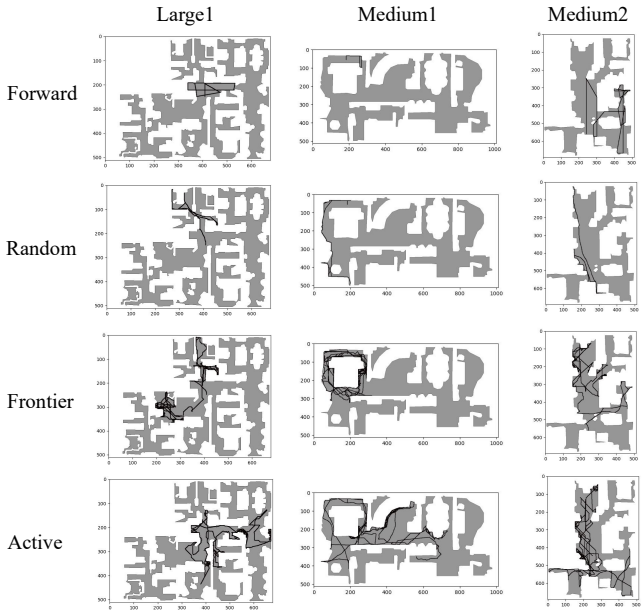


Fig. 5: **Exploration Trajectories.** Exploration trajectory maps displaying traversable areas (grey) and untraversable areas (white) with the agent's trajectory represented by a black line.

inconsistency exploration significantly improved embodied depth prediction.

Moreover, we examined data size effects on model performance in Fig. 6. Forward exploration, despite a strong start, restricts data distribution, leading to performance stagnation. In contrast, the active policy improved the most with increasing data, and frontier-based exploration outperformed pure randomness. Our findings emphasize the pivotal role of exploration strategy selection and combination in enhancing depth prediction performance.

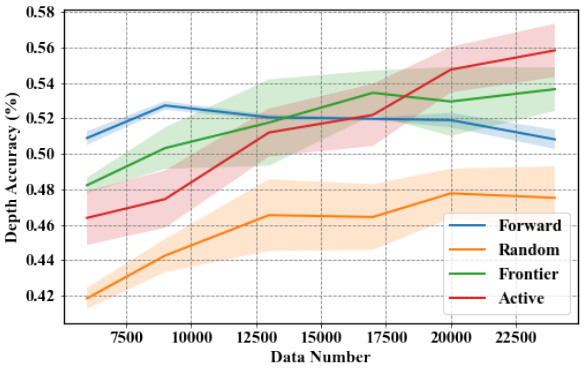


Fig. 6: **Performance vs Data Gathered.** Accuracy of depth prediction ($\delta < 1.25$) vs data gathered.

Fig. 7: **Real-world Exploration Trajectories.** Overhead exploration of our active policy. We also manually collect Trajectory 1 and 2 for testing which cover different areas and views in our environment.

C. Real Results

Experimental Settings. We conducted real-world experiments using a Jackal robot equipped with an Intel Realsense L515 camera for capturing RGB images and sparse depth maps during robot motion. LiDAR generated depth maps with occasional sparsity beyond its 9-meter range, marked as 0 meters. Robot Operating System (ROS) [52] controlled robot movement, and ORB-SLAM2 [43] provided recent camera trajectories using images and sparse depth maps as input. The indoor experiments, illustrated in Fig. 7, employed a pre-trained model, fine-tuned offline to optimize training time and accommodate limited GPU memory on the robot. We collected 10,000 frames for training our method with random and active policies, as shown in Fig. 7. For evaluation, we manually gathered two trajectories (4,000 frames each) in the indoor environment's primary area, using LiDAR readings as ground truth depth. These real-world experiments validate the proposed method's practical applicability.

Qualitative Visualization. In Fig. 8, we present depth visualizations comparing our model's predictions with ground truth. Real-world experiments introduced unique challenges compared to simulation. Firstly, lighting conditions varied due to factors like sunlight, artificial lighting, and camera exposure adjustments, impacting depth prediction accuracy and photometric loss. Additionally, the presence of moving objects like pedestrians posed challenges in distinguishing ego-motion from object motion. Lastly, furniture positions changed over time, causing images in our test distribution to differ from those in training.

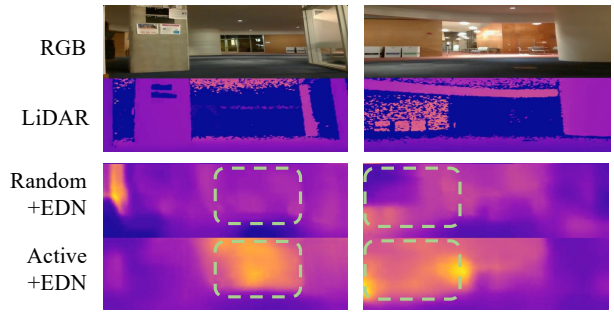


Fig. 8: **Real-World Depth Prediction.** Comparison of embodied depth prediction using active or random exploration. Active EDN obtains more accurate depth prediction in far away regions in images. The second row corresponds to LiDAR readings of depth – regions with depth greater than 9 meters have empty depth.

Method	Trajectory1	Trajectory2
Pretrained	0.233	0.301
Random	0.267	0.325
Active	**0.400**	**0.432**

TABLE III: **Real World Quantitative Depth Results.** Depth accuracy ($\delta < 1.25$) on real images gathered across two test trajectories. Active exploration improves performance substantially.

Quantitative Results. To quantify the impact of these challenges, we report the accuracy of our model's performance on a dataset collected from a real-world environment in Table III. We compute the accuracy by masking the empty pixels in LiDAR. Our results highlight the added difficulty of embodied depth prediction in real-world images. As many of these difficulties are not easily simulatable in existing simulators, yet crucial for effective depth prediction in robotics, we plan to release both our gathered images and evaluation set to enable a reproducible benchmark to support future research in embodied depth prediction.

V. Conclusion

This paper has introduced the problem of embodied depth prediction, where an agent in an environment must actively gather information to learn to estimate depth. Our proposed approach substantially surpasses prior approaches in effectiveness. We further carry out an in-depth analysis of our model on the real-world dataset and find the model falls short due to changing lighting conditions. This can be potentially solved by using more robust photometric losses. To facilitate the development of more robust embodied depth prediction approaches, we release a dataset for studying embodied depth prediction on real images. We hope our formulation and dataset stimulate future research toward solving this important problem.

REFERENCES

[1] Jia-Wang Bian, Huangying Zhan, Naiyan Wang, Zhichao Li, Le Zhang, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth learning from video. *Int. J. Comput. Vision*, 129(9):2548–2564, sep 2021.

[2] Michael Brock and Per Ola Kristensson. Supporting blind navigation using depth sensing and sonification. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pages 255–258, 2013.

[3] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019.

[4] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[5] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In *The International Conference on Computer Vision (ICCV)*, 2019.

[6] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In *International Conference on Learning Representations (ICLR)*, 2016.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[8] Yilun Du, Chuang Gan, and Phillip Isola. Curious representation learning for embodied intelligence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10408–10417, 2021.

[9] Yilun Du, Tomas Lozano-Perez, and Leslie Pack Kaelbling. Learning object-based state estimators for household robots. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12558–12565. IEEE, 2022.

[10] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B. Tenenbaum, and Jiajun Wu. Neural radiance flow for 4d view synthesis and video processing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[11] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):611–625, 2018.

[12] Jakob Engel, Thomas Schoeps, and Daniel Cremers. Lsd-slam: large-scale direct monocular slam. volume 8690, pages 1–16, 09 2014.

[13] Zhaoyuan Fang, Ayush Jain, Gabriel Sarch, Adam W. Harley, and Katerina Fragkiadaki. Move to see better: Towards self-supervised amodal object detection, 2020.

[14] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: fast semi-direct monocular visual odometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22, 2014.

[15] Chuang Gan, Jeremy Schwartz, Seth Alter, Damian Mrowca, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, et al. Threedworld: A platform for interactive multi-modal physical simulation. *NeurIPS*, 2021.

[16] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 963–968, 2011.

[17] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6602–6611, 2017.

[18] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth prediction. In *The International Conference on Computer Vision (ICCV)*, October 2019.

[19] S Burak Gokturk, Hakan Yalcin, and Cyrus Bamji. A time-of-flight depth sensor-system description, issues and solutions. In *2004 conference on computer vision and pattern recognition workshop*, pages 35–35. IEEE, 2004.

[20] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *The International Conference on Computer Vision (ICCV)*, 2019.

[21] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[22] Nick Hawes, Christopher Burbridge, Ferdian Jovan, Lars Kunze, Bruno Lacerda, Lenka Mudrova, Jay Young, Jeremy Wyatt, Denise Hebesberger, Tobias Kortner, et al. The strands project: Long-term autonomy in everyday environments. *IEEE Robotics & Automation Magazine*, 24(3):146–156, 2017.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, June 2016.

[24] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *CVPR*, 2018.

[25] Massimiliano Iacono and Antonio Sgorbissa. Path following and obstacle avoidance for an autonomous uav using a depth camera. *Robotics and Autonomous Systems*, 106:38–46, 2018.

[26] Jeffrey Ichnowski, Yahav Avigal, Justin Kerr, and Ken Goldberg. Dexnerf: Using a neural radiance field to grasp transparent objects. *arXiv preprint arXiv:2110.14217*, 2021.

[27] Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. Dpsnet: End-to-end deep plane sweep stereo. In *ICLR*, 2019.

[28] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 2017–2025, Cambridge, MA, USA, 2015. MIT Press.

[29] Ping Jiang, Yoshiyuki Ishihara, Nobukatsu Sugiyama, Junji Oaki, Seiji Tokura, Atsushi Sugahara, and Akihito Ogawa. Depth image–based deep learning of grasp planning for textureless planar-faced objects in vision-guided robotic bin-picking. *Sensors*, 20(3):706, 2020.

[30] Tong Ke, Tien Do, Khiem Vuong, Kourosh Sartipi, and Stergios I. Roumeliotis. Deep multi-view depth estimation with predicted uncertainty. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[31] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations (ICLR)*, 2015.

[32] Georg Klein and David Murray. Parallel tracking and mapping on a camera phone. In *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, pages 83–86. IEEE, 2009.

[33] Klemen Kotar and Roozbeh Mottaghi. Interactron: Embodied adaptive object detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14840–14849, 2022.

[34] Tomáš Krajník, Jaime P Fentanes, Joao M Santos, and Tom Duckett. Fremen: Frequency map enhancement for long-term mobile robot autonomy in changing environments. *IEEE Transactions on Robotics*, 33(4):964–977, 2017.

[35] Lars Kunze, Nick Hawes, Tom Duckett, Marc Hanheide, and Tomáš Krajník. Artificial intelligence for long-term robot autonomy: A survey. *IEEE Robotics and Automation Letters*, 3(4):4023–4030, 2018.

[36] Hao-Yuan Kuo, Hong-Ren Su, Shang-Hong Lai, and Chin-Chia Wu. 3d object detection and pose estimation from depth image for robotic bin picking. In *2014 IEEE international conference on automation science and engineering (CASE)*, pages 1264–1269. IEEE, 2014.

[37] Bo Liu, Xuesu Xiao, and Peter Stone. A lifelong learning approach to mobile robot navigation. *IEEE Robotics and Automation Letters*, 6(2):1090–1096, 2021.

[38] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[39] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*, 2021.

[40] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[41] 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 *The European Conference on Computer Vision (ECCV)*, 2020.

[42] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *In IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[43] Raúl Mur-Artal and Juan D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[44] Vaishakh Patil, Wouter Van Gansbeke, Dengxin Dai, and Gool Luc Van. Don't forget the past: Recurrent depth estimation from monocular video. In *IEEE Robotics and Automation Letters*, 2020.

[45] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. *arXiv preprint arXiv:2011.13961*, 2020.

[46] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ICCV*, 2021.

[47] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[48] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[49] Johannes L. Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 501–518, Cham, 2016. Springer International Publishing.

[50] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[51] Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-metric loss for self-supervised learning of depth and egomotion. In *ECCV*, 2020.

[52] Stanford Artificial Intelligence Laboratory et al. Robotic operating system.

[53] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[54] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow (extended abstract). In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4839–4843. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Sister Conferences Best Papers.

[55] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[56] Dashuai Wang, Wei Li, Xiaoguang Liu, Nan Li, and Chunlong Zhang. Uav environmental perception and autonomous obstacle avoidance: A deep learning and depth camera combined solution. *Computers and Electronics in Agriculture*, 175:105523, 2020.

[57] Kaixuan Wang and Shaojie Shen. Mvdepthnet: Real-time multiview depth estimation neural network. In *3DV*, 2018.

[58] Rui Wang, Stephen M Pizer, and Jan-Michael Frahm. Recurrent neural network for (un-)supervised learning of monocular video visual odometry and depth. In *CVPR*, 2019.

[59] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *In IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[60] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The Temporal Opportunist: Self-Supervised Multi-Frame Monocular Depth. In *Computer Vision and Pattern Recognition (CVPR)*, 2021.

[61] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *ICCV*, 2021.

[62] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. DD-PPO: learning near-perfect pointgoal navigators from 2.5 billion frames. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

[63] Mitchell Wortsman, Kiana Ehsani, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Learning to learn how to learn: Self-adaptive visual navigation using meta-learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[64] Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson Env: real-world perception for embodied agents. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018.

[65] Jianwei Yang, Zhile Ren, Mingze Xu, Xinlei Chen, David Crandall, Devi Parikh, and Dhruv Batra. Embodied amodal recognition: Learning to move to perceive objects. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2040–2050, 2019.

[66] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *ECCV*, 2018.

[67] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Tsung-Yi Lin, Alberto Rodriguez, and Phillip Isola. Nerf-supervision: Learning dense object descriptors from neural radiance fields. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6496–6503. IEEE, 2022.

[68] Haokui Zhang, Chunhua Shen, Ying Li, Yuanzhouhan Cao, Yu Liu, and Youliang Yan. Exploiting temporal consistency for real-time video depth estimation. In *ICCV*, 2019.

[69] Wang Zhao, Shaohui Liu, Yezhi Shu, and Yong-Jin Liu. Towards better generalization: Joint depth-pose learning without posenet. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[70] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6612–6619, 2017.

APPENDIX

In this supplemental document, we provide more setting details (Section A), experimental details of our method (Section B) and additional visualization results (Section C).

A. Details in Embodied Exploration

a) Collision Detection.: To avoid collisions during exploration, we equip the agent with a base lidar sensor as a safety measure, given that the initial depth model may not be precise enough to navigate through obstacles effectively. In the event of an imminent collision predicted by the base sensor, we mark the collided location as occupied in the local occupancy grid map to adjust the agent's future exploration plans. This process helps prevent the agent from repeating failed actions and improves its exploration efficiency. In addition, to prevent any damage of the device in the real world, we have a base LiDAR Hokuyo UTM-30LX, and if obstacles are within 0.2 meters, we force the car to move backward as a safety measure.

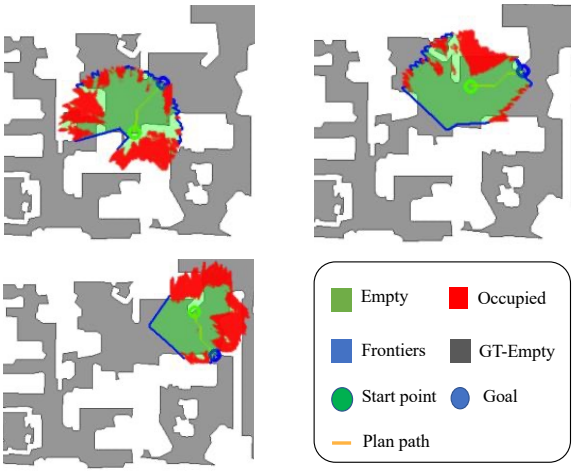
b) Depth Model Initialization.: To effectively execute frontier and inconsistency exploration strategies, a depth prediction model must have reasonable depth predictions. Thus to obtain initial reasonable depth predictions, we first gather images from the environment using random exploration to train our initial depth prediction network. Afterward, we use the initialized models to do our embodied exploration.

B. Experimental Details

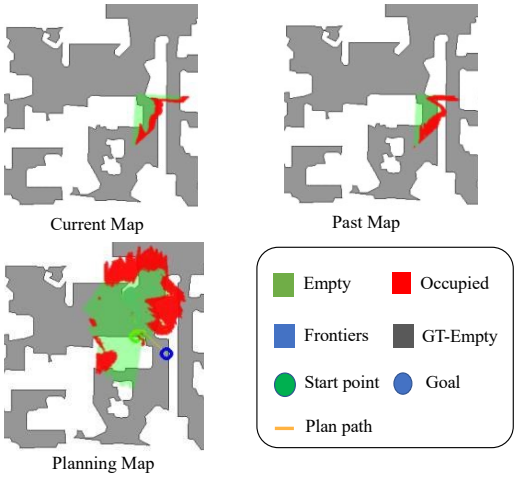
a) Agent Settings: The agent has a camera with the height of 0.8 meters and a 90-degree field. We also set the base Lidar range at 0.1 meters for collision detection, and the output depth ranges from 0.1 to 20 meters, which is suitable for indoor environment cases. The image size is [192, 640], which is a common resolution used in unsupervised depth prediction works for comparison.

In active exploration, to warm up the model, the agent first collects 6,000 frames using random exploration. We then begin active exploration, alternating between frontier-based exploration for 800 frames with depth-inconsistency-based exploration for 400 frames.

b) Network Architecture.: Our Embodied Depth Network (EDN) takes as input multiple RGB images and their corresponding camera poses and outputs an inverse depth map. To ensure a fair comparison with related networks, we used two past frames to predict the depth. The coarse depth map is obtained as explained in Section III-E, using a pretrained RAFT [54] model on KITTI [40] dataset. The refinement network architecture is based on a UNet [47], which comprises a ResNet18 [23] encoder to extract features from both the coarse depth map and the current RGB image. These features are then fused using point-wise addition and fed into the decoder, which is a DispNet similar to [70]. The output of the decoder has sigmoid activation layers, while ELU nonlinearities [6] are used elsewhere. We convert the sigmoid output x to depth D using $D = 1/(ax + b)$, where a and b are chosen to ensure that D falls between 0.1 and 20 meters.



(a) Illustration of Frontier Exploration. The agent sets the goal to the center of one randomly-picked frontier group which is based on the occupancy map.



(b) Illustration of Depth-Inconsistency Exploration. The agent checks the inconsistent areas between current and past occupancy maps and sets the goal to the center of the depth-inconsistency areas.

Fig. 9: **Active Exploration Demo.** We show the top-down maps displaying traversable areas (grey) and untraversable areas (white) to illustrate our method.

c) Training Details.: We implemented our approach using PyTorch with a backbone ResNet18 trained on ImageNet [7]. Additionally, we incorporated color augmentations with a 50% probability. These augmentations include random gamma, brightness, and color shifts, achieved by sampling from uniform distributions within the ranges of [0.8, 1.2] for gamma, brightness, and each color channel independently. We use Adam optimizer with a learning rate of 10^{-4}.

To train our model, we employ a photometric loss computed from the 4th frame to the current frame for simulations and the 15th frame for real-world data. We set the weight of the smoothness term to 10^{-3}. We use the ADAM optimizer [31] with a learning rate of 10^{-4} and a batch size of 12 on a single Nvidia GTX Titan X. We initialize the model with 6,000 frames of data for warm-up and then train the model every 3,000 frames of data. Once we reach a maximum dataset size of 30,000 frames, we perform an additional 3-epoch training.

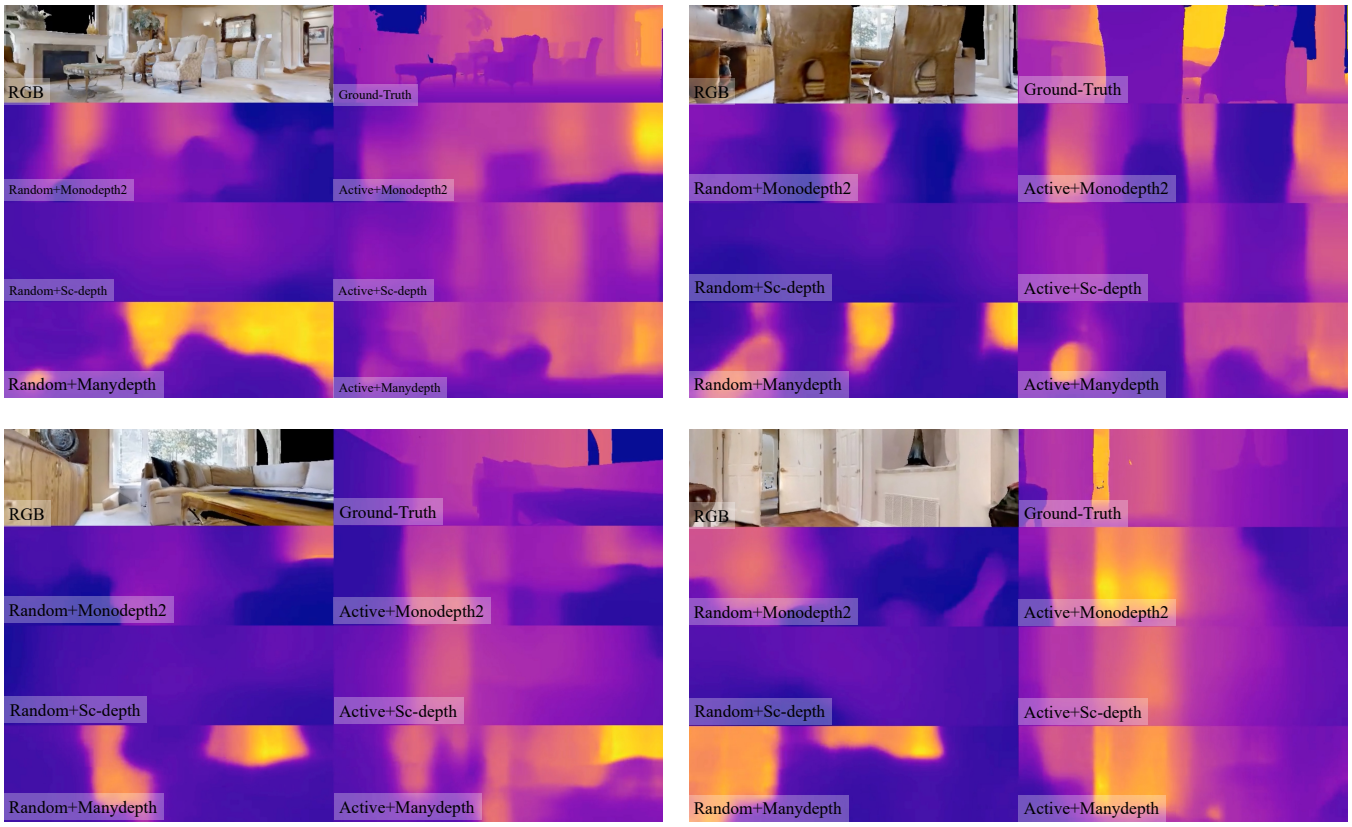


Fig. 10: **Depth Predictions in Different Models.** Our active policy improves the performance of all of the baseline models.

C. Additional Results

a) Active Data Collection.: We provide further visualization details on our active data collection strategy, which enables our method to effectively explore and select informative views to improve depth estimation accuracy. The frontier exploration expands the distribution of data by selecting views that are close to the current field of view but have not yet been observed. This strategy ensures that the network has access to a diverse set of viewpoints, which can help it learn to generalize better across different scenes. On the other hand, the depth-inconsistency exploration strategy aims to identify areas in the scene where the network is uncertain about its depth predictions. Our active strategy can guide the new data collection even in cases in which the network outputs an ambiguous prediction.

b) Depth Predictions in Different Models.: We compare the performance of different depth estimation models under random and active data collection policies in Fig. 10. Specifically, we visualize the depth predictions of three different models: Monodepth2[18], Sc-Depth[1], and ManyDepth[60]. We find that our active policy consistently improves the performance of all of the baseline models, producing depth predictions that are more accurate than those obtained with random data collection. Our results demonstrate the effectiveness of our proposed active data collection strategy in improving depth estimation accuracy across a range of different models.

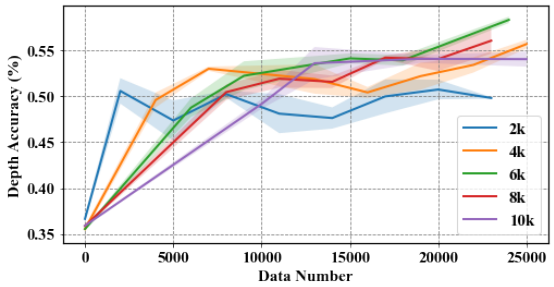


Fig. 11: **Performance vs Data Gathered on Warm-up Data.**

c) Warm-up Data Number.: Figure 11 demonstrates the impact of warm-up data on model training. In general, we found limited impact with using different values of data to warm-up training. Using a small number of warm-up data leads to unstable initialization, while an excessively large number may cause premature convergence and suboptimal performance. We found that 6k data gave the best performance.