

FloNa: Floor Plan Guided Embodied Visual Navigation

Jiaxin Li¹, Weiqi Huang¹, Zan Wang¹, Wei Liang^{1, 2*}, Huijun Di^{1*}, Feng Liu³

¹Beijing Institute of Technology, Beijing, China

²Yangtze Delta Region Academy of Beijing Institute of Technology, Jiaxing, China

³Beijing Racobit Electronic Information Technology Co., Ltd.

{lijiaxin, huangweiqi, wangzan, ajon, liangwei}@bit.edu.cn, liufeng@racobit.com

Abstract

Humans naturally rely on floor plans to navigate in unfamiliar environments, as they are readily available, reliable, and provide rich geometrical guidance. However, existing visual navigation settings overlook this valuable prior knowledge, leading to limited efficiency and accuracy. To eliminate this gap, we introduce a novel navigation task: **Floor Plan Visual Navigation (FloNa)**, the first attempt to incorporate floor plans into embodied visual navigation. While the floor plan offers significant advantages, two key challenges emerge: (1) handling the spatial inconsistency between the floor plan and the actual scene layout for collision-free navigation, and (2) aligning observed images with the floor plan sketch despite their distinct modalities. To address these challenges, we propose FloDiff, a novel diffusion policy framework incorporating a localization module to facilitate alignment between the current observation and the floor plan. We further collect 20k navigation episodes across 117 scenes in the iGibson simulator to support the training and evaluation. Extensive experiments demonstrate the effectiveness and efficiency of our framework in unfamiliar scenes using floor plan knowledge.

Introduction

An essential task in embodied AI is enabling agents to navigate in diverse environments toward a goal, represented as points (Savva et al. 2019), images (Zhu et al. 2017; Shah et al. 2023), objects (Chaplot et al. 2020a), or language instructions (Anderson et al. 2018b; Wang et al. 2023b). Recently, researchers have increasingly leveraged easily accessible prior knowledge (Shah et al. 2021) to improve efficiency and accuracy. Floor plans, in particular, serve as a valuable and widely available source of such knowledge, offering high-level semantic and geometric information that can assist agents in localizing and navigating in unfamiliar spaces. Moreover, incorporating the floor plan benefits various applications such as emergency response, search and rescue, and pathfinding in dynamic public environments.

Previous studies have explored integrating floor plans to facilitate localization and navigation, often relying on multi-sensor fusion (Li et al. 2021b) or imposing constraints on the floor plan structure (Boniardi et al. 2019b), which lim-

its their practical applicability. Inspired by the human ability to navigate efficiently in unfamiliar environments using minimal abstract information and visual cues, we aim to reduce both sensor dependence and structural constraints on the floor plan in navigation.

In this work, we introduce a novel navigation task, **Floor Plan Visual Navigation (FloNa)**, where agents navigate within an environment using an abstract floor plan and a series of RGB observations, as depicted in Fig. 1. Despite the valuable prior knowledge provided by the floor plan, this task remains challenging for two primary reasons. First, spatial inconsistency exists due to the significant difference between the floor plan and the actual observed layout, primarily caused by furniture placement in the scenes. This inconsistency can lead to collisions during navigation. Second, observation misalignment arises because the floor plan offers abstract topological information, whereas the RGB observations capture the appearance of natural scenes from a specific viewpoint. Such misalignment results in incorrect localization of the current observation within the floor plan, thereby hindering the effectiveness of planning.

To address these challenges, we develop FloDiff, a novel diffusion policy framework that exploits strong action distribution modeling capability to learn to handle the spatial inconsistency from extensive demonstrations implicitly. FloDiff also integrates an explicit localization module to align the observation and floor plan, resulting in two variants based on how current agent pose is derived: (1) Naive-FloDiff, which learns to predict the pose during training, and (2) Loc-FloDiff, which directly uses either ground truth poses or predictions from a pre-trained model. Both variants employ a Transformer backbone to fuse the floor plan and current observation, followed by passing the fusion into a policy network that learns to denoise the action sequence.

For benchmarking, we collect a dataset comprising approximately 20k navigation episodes across 117 distinct scenes using the iGibson simulator (Li et al. 2021a). The dataset includes around 3.3M images captured with a 45-degree field of view. We split the scenes into 67 for training and 50 for testing to assess the model’s generalization capability to unseen environments. Each scene comprises a floor plan, a traversability map, and sufficient navigation episodes. Each episode contains an A*-generated trajectory paired with corresponding RGB observations.

*Corresponding authors.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

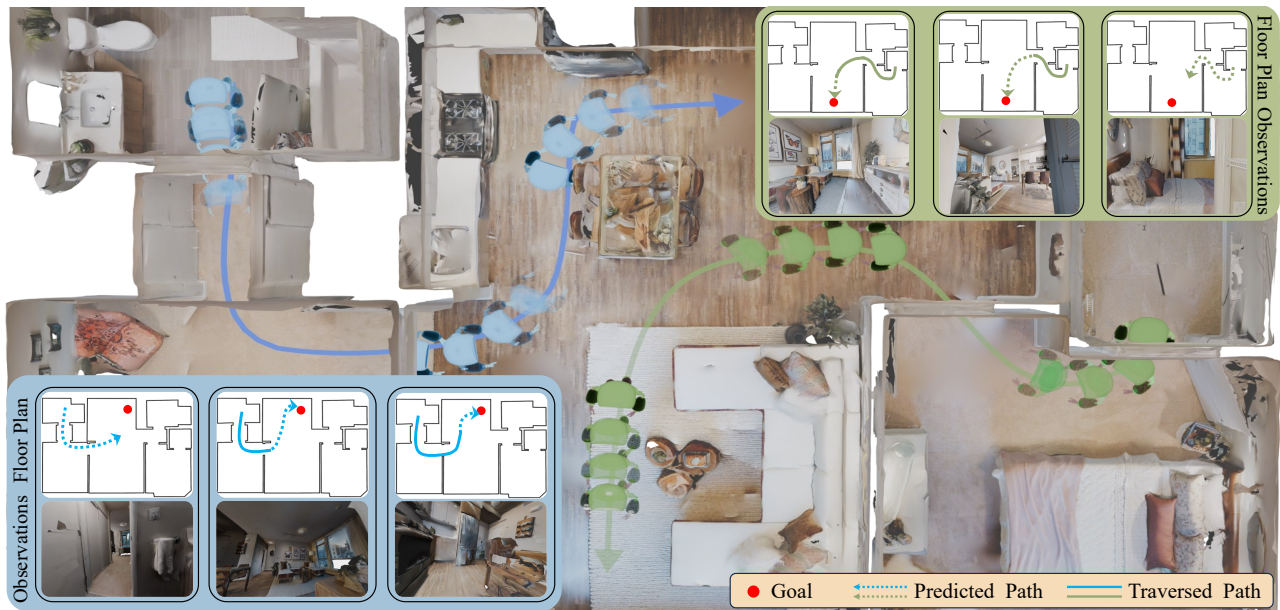


Figure 1: **Floor Plan Visual Navigation (FloNa)**: Given a floor plan with a marked goal indicated by the red dot, the agent’s task is to navigate to the corresponding target location in the environment using RGB observations. To tackle this task, we propose FloDiff, a novel diffusion policy-based framework that iteratively generates and refines the planned trajectory.

Extensive experiments demonstrate the effectiveness and efficiency of our method in navigating within unseen environments using a floor plan. Compared to baseline methods, our method achieves a higher Success Rate (SR) and greater efficiency, as measured by Success Weighted by Path Length (SPL). Additionally, we comprehensively analyze FloDiff’s different capabilities, including localization, collision avoidance, planning with diverse goals, and robustness. Real-world deployment on an Automated Guided Vehicle (AGV) without finetuning further highlights its robustness and generalization, proving its potential to handle unseen scenarios effectively in practical settings.

Our contributions are summarized as follows:

- We introduce FloNa, a novel task of navigating toward a goal using RGB observations and a floor plan, enriching the application scenarios in embodied visual navigation.
- We propose a novel end-to-end diffusion policy-based framework, *i.e.*, FloDiff, incorporating explicit agent localization to solve FloNa efficiently and effectively.
- We conduct extensive experiments on our curated dataset and thoroughly analyze FloDiff’s capabilities across various dimensions, demonstrating its superiority over baseline methods.

Related Work

Visual Navigation

Visual navigation aims at enabling robots or autonomous agents to navigate through environments using visual information. The definition of navigation tasks has evolved alongside the advancements in the field. Anderson et al. (2018a) are the first to explicitly categorize three navigation tasks based on goal types: PointGoal, ObjectGoal, and

AreaGoal. The PointGoal task requires the agent to navigate to a specific location, emphasizing the self-localization capabilities (Wijmans et al. 2019; Ramakrishnan et al. 2021; Zhao et al. 2021; Partsey et al. 2022; Tang et al. 2022). In contrast, the ObjectGoal and AreaGoal tasks involve commonsense knowledge of the environment, such as recognizing objects and understanding where they are typically located (Yadav et al. 2023; Ramakrishnan et al. 2022; Gadre et al. 2023). With advancements in image and natural language understanding, the field has shifted focus toward navigation tasks guided by high-level semantics. Tasks like image navigation (Zhu et al. 2017; Krantz et al. 2023; Yadav et al. 2023; Chaplot et al. 2020b) and language navigation (Anderson et al. 2018b; Song et al. 2023; Min et al. 2021; Wang et al. 2023a,b) have emerged, filling gaps in various real-world and natural application scenarios. Beyond vision and language, navigation using other modalities, such as sound, has also been explored (Chen et al. 2020; Gan et al. 2020; Chen et al. 2021). Despite the promising progress in prior studies, efficient and effective exploration remains a critical challenge in tasks involving unknown environments. In this work, we explore the potential of leveraging readily available prior information, *i.e.*, floor plans, to mitigate the dependence on time-consuming exploration, further enhancing the effectiveness and efficiency of navigation.

Floor Plan for Localization and Navigation

Floor plans, as stable structural information, have been widely explored in localization and navigation tasks. One line of research approaches localization within a traditional optimization framework, leveraging floor plans in combination with various sensor signals such as LiDAR (Bonardi et al. 2019a, 2017; Li, Ang, and Rus 2020; Mendez et al.

2020; Wang, Marcotte, and Olson 2019), images (Ito et al. 2014; Boniardi et al. 2019b), or visual odometry (Chu, Kim, and Chen 2015). Another branch adopts learning-based methods (Howard-Jenkins, Ruiz-Sarmiento, and Prisacariu 2021; Howard-Jenkins and Prisacariu 2022; Min et al. 2022; Chen et al. 2024) to solve the localization task. Some studies have also attempted to use augmented topological maps generated from sketch floor plans (Setalaphruk et al. 2003) or architectural floor plans (Li et al. 2021b) to aid the navigation. These works highlight the benefits of incorporating prior knowledge from floor plans for indoor localization and navigation. However, their dependence on multiple sensors hinders the practical applicability of the approach. To overcome this limitation, we investigate the feasibility of accomplishing the navigation task using only RGB observations and floor plans in this work.

Task Setting

Task Definition The goal of FloNa requires embodied agents to navigate from a starting position to a specific goal position in a 3D environment. During the navigation, the agent can only receive the ego-centric RGB image, the environment’s floor plan image, and a goal position, represented as a red dot in Fig. 3. Specifically, given the floor plan o_f , the agent receives the observation image o_t at each timestep t and produces action a_t to reach the goal g . A resulting episode $\{x_0, a_1, x_1, \dots, a_T, x_T\}$, where x_i comprises the agent’s location p_i and orientation r_i , is considered successful if the last position p_T is within a specified distance threshold τ_d from the goal and the number of collisions with the scene does not exceed a given threshold τ_c . $x_0 = \{p_0, r_0\}$ is the starting pose and T is the total step.

Simulator Setup Following Li et al. (2021a), we build the simulation environment upon the iGibson simulator and employ a Locobot as the embodied agent. The agent has a height of 0.85 meters and a base radius of 0.18 meters, and is equipped with an RGB camera with a resolution of 512×512 pixels. We define the action as $a_t := p_t - p_{t-1}$, $a_t \in \mathbb{R}^2$, representing continuous movement in the 2D plane.

Dataset Design

To facilitate the benchmarking, we collect a large-scale dataset comprising 20,214 navigation episodes across 117 static indoor scenes from Gibson (Xia et al. 2018), resulting in a total of 3,312,480 images captured with a 45-degree field of view. The Gibson scenes are reconstructed from the homes and offices with a Matterport device, which preserves the textures observed by the sensor, thus minimizing the sim-to-real gap. We split the dataset into training and testing sets, which comprise 67 scenes and 50 scenes, respectively.

For each scene, we provide a floor plan and a navigable map. We directly adopt the manually-annotated floor plans from Chen et al. (2024). For navigable maps, we employ a coarse-to-fine approach to obtain them by executing the following two steps. First, we generate a rough one from the scene’s mesh. Then, we manually refine it to address scanning artifacts; for example, the poorly reconstructed chair legs may lead to a navigable area. Based on scene size, we



Figure 2: **Typical scenes, navigable areas, navigation episodes in our collected dataset.** We show the short and long episodes using green and blue colors, respectively.

collect 150 episodes in small scenes, 180 in medium scenes, and 200 in large scenes. Before collecting each episode, we randomly sample two points at least 3 meters apart as the starting and target points. We then use the A* algorithm to search for a collision-free trajectory from the start to the target. We record the trajectory positions in the pre-defined world coordinate system and render the observation images during the search process. The episode trajectory length ranges from 4.53 to 42.03 meters. Fig. 2 presents some trajectory examples. Please refer to *supplementary material* for more details about the dataset construction.

Preliminary

Diffusion Model

The diffusion model (Ho, Jain, and Abbeel 2020; Sohl-Dickstein et al. 2015) is a probabilistic framework used in generative tasks. It learns to gradually denoise the sampled Gaussian noise to generate desired data through an iterative denoising process. Specifically, beginning with sampled noise x^K , the model performs K iterations of denoising to produce a series of intermediate poses x^{K-1}, \dots, x^0 . The process can be denoted as:

$$x^{k-1} = \alpha(x^k - \gamma\epsilon_\theta(x^k, k) + \sigma z), \quad (1)$$

where α, γ, σ are noise schedule functions, ϵ_θ is the noise prediction network with parameters θ and $z \sim \mathcal{N}(0, \mathbf{I})$.

Diffusion Policy

Diffusion policy (Chi et al. 2023) extends the concept of diffusion models, which are typically used for content generation, to policy learning, allowing an agent to make decisions based on learned probabilistic distributions. The policy usually incorporates an encoder to transform the observation into a lower-dimensional feature, serving as the condition for the denoising process. Eq. (1) can be adapted to:

$$\mathcal{A}_t^{k-1} = \alpha(\mathcal{A}_t^k - \gamma\epsilon_\theta(\mathcal{A}_t^k, \mathcal{O}_t, k) + \sigma z), \quad (2)$$

where \mathcal{A}_t and \mathcal{O}_t denote the action and the observation at each time step t , respectively.

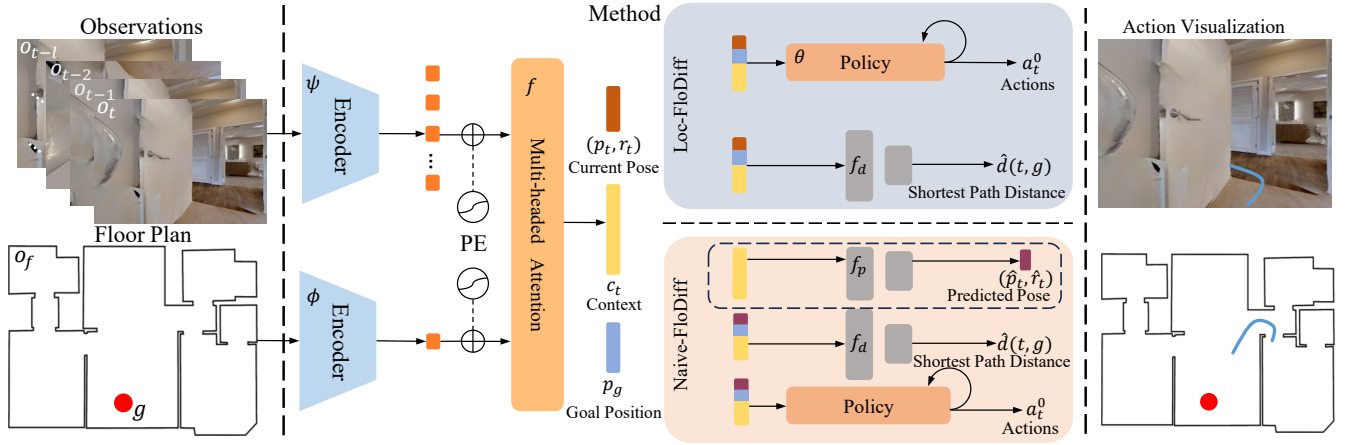


Figure 3: **Pipeline overview.** FloDiff employs an attention module to fuse features from visual observation and floor plan, yielding a context embedding c_t . Depending on how the current agent pose is derived, FloDiff has two variants: (1) Naive-FloDiff (below), which learns to predict the current pose (\hat{p}_t, \hat{r}_t) during policy learning; (2) Loc-FloDiff (above), which directly uses the ground truth pose or predictions from pre-trained models. The concatenation of the observation context c_t , goal position p_g , and current agent pose (p_t, r_t) is then fed into the policy network to generate actions.

Method

Building upon diffusion policy, we propose a novel end-to-end framework, *i.e.*, FloDiff, for solving FloNa. The overall architecture is illustrated in Fig. 3. Next, we will detail the FloDiff framework and the training process.

Diffusion Policy for FloNa

Framework Overview FloDiff consists of a transformer-based backbone that fuses visual observations with the floor plan and a policy network designed to navigate within the scene. As discussed, an ideal agent must be capable of (1) navigating to the goal without colliding with the scene and (2) aligning visual observations with the floor plan. To address the first challenge, we employ the powerful diffusion policy to learn navigation in crowded scenes from extensive demonstrations. For the second challenge, we explicitly predict the agent pose from observations or utilize a pre-trained model for pose prediction.

Observation Context In our work, the visual observation \mathcal{O}_t contains the observed images $\{o_{t-l}, o_{t-l+1}, \dots, o_t\}$, where l denotes the context length, and the floor plan image o_f . Inspired by a wealth of prior work on training high-capacity policies (Sridhar et al. 2023; Shah et al. 2023), we employ a transformer-based backbone to process the observation context. Specifically, for each observed image o_i , $i = t-l, \dots, t$, we employ a shared EfficientNet-B0 (Tan and Le 2019) as the encoder, denoted as ψ , to produce the visual latent features independently. Following Visual Navigation Transformer (ViNT) (Shah et al. 2023) that encodes the relative difference between the current observation and the goal image, we employ another EfficientNet-B0 as encoder ϕ to process the floor plan image. We train both ψ and ϕ from scratch. Subsequently, we apply multi-headed attention (Vaswani et al. 2017) layers $f(\cdot)$ to fuse the two branch

features. The final output feature serves as the observation context vector c_t .

Goal Position We compute goal position p_g in the world coordinate according to the marked goal point g . Precisely, we first determine the pixel coordinates of the goal on the floor plan image, denoted as u_g . We then convert the coordinates into the world coordinates using the floor plan resolution μ defined as the distance per pixel and the offset δ provided by Chen et al. (2024) through $p_g = \mu \cdot u_g + \delta$.

Policy Based on different ways of obtaining the current agent pose, we propose two variants of FloDiff.

First, as described above, $f(\cdot)$ fuses the observed images with the floor plan, indicating that c_t encodes valuable information for predicting the agent poses. To achieve this, we introduce a fully connected network f_p to predict the agent pose $\hat{x}_t = (\hat{p}_t, \hat{r}_t)$ from the observation context vector c_t . Following Shah et al. (2023), we also use an additional fully connected network f_d to predict the shortest path distance $\hat{d}(t, g)$ between the current position and the goal, as illustrated in Fig. 3. This design helps mitigate the policy’s tendency to move directly toward the goal, which could otherwise lead to dead ends. Finally, we concatenate the observation context vector c_t , goal position p_g , and predicted agent pose x_t as a conditional vector, from which the diffusion policy θ learns to denoise the sampled noise and generate the action sequence. We denote this variant as Naive-FloDiff.

Second, we can directly use the ground truth agent pose or predicted pose from a pre-trained model, *i.e.*, F³Loc (Chen et al. 2024). Unlike Naive-FloDiff, this design eliminates the need for f_p . We denote such FloDiff variants as Loc-FloDiff, including Loc-FloDiff (GT) and Loc-FloDiff (F³).

For all FloDiff variants, we model the predicted action sequence \mathcal{A}_t^0 at time step t as the future action sequence with a horizon of H_p , where the agent executes the first H_a steps.

During inference, we sample the predicted action sequence as described in Eq. (2) and compute the next position by adding the action vector to the current position.

Training

We train FloDiff on the training set, which consists of 67 indoor scenes, encompassing 11,575 episodes and approximately 26 hours of trajectory data. During training, we randomly sample trajectory segments with a fixed horizon from the episodes as training samples. To increase the diversity of target goals, we also randomly select a target position behind the sampled trajectory rather than using the episode endpoint as the target. We train Naive-FloDiff in an end-to-end manner using the following loss functions:

$$\begin{aligned} \mathcal{L}(\phi, \psi, f, \theta, f_d, f_p) = & \text{MSE}(\epsilon^k, \epsilon_\theta(c_t, \mathcal{A}_t^0 + \epsilon^k, k)) \\ & + \lambda_1 \cdot \text{MSE}(d(t, g), f_d(c_t, \hat{x}_t, p_g)) \quad (3) \\ & + \lambda_2 \cdot \text{MSE}(x_t, f_p(c_t)). \end{aligned}$$

Similarly, we train Loc-FloDiff using:

$$\begin{aligned} \mathcal{L}(\phi, \psi, f, \theta, f_d) = & \text{MSE}(\epsilon^k, \epsilon_\theta(c_t, \mathcal{A}_t^0 + \epsilon^k, k)) \quad (4) \\ & + \lambda_3 \cdot \text{MSE}(d(t, g), f_d(c_t, \hat{x}_t, p_g)). \end{aligned}$$

λ_1 , λ_2 , and λ_3 are hyperparameters for weighting different loss terms. $\text{MSE}(\cdot)$ computes mean square error.

In the implementation, FloDiff is trained for 5 epochs using AdamW (Loshchilov, Hutter et al. 2017) optimizer with a fixed learning rate of 0.0001. We empirically set $\lambda_1 = \lambda_3 = 0.001$ and $\lambda_2 = 0.005$. The attention layers are built using the native PyTorch implementation. The number of multi-head attention layers and heads are both 4. We set the dimension of the observation context vector c_t to 256. The diffusion policy is trained using the Square Cosine Noise Scheduler (Nichol and Dhariwal 2021) with $K = 10$ denoising steps. The noise prediction network ϵ_θ adopts a conditional U-Net architecture following (Janner et al. 2022) with 15 convolutional layers. We set the diffusion horizon as $H_p = 32$ and employ the first $H_a = 16$ steps to execute in each iteration. We train FloDiff using one NVIDIA RTX3090 GPU and assign a batch size of 256.

Experiments

This section begins with an introduction to the experimental setup, including the baseline methods and the evaluation metrics. We then present the results and the performance analysis regarding localization, collision avoidance, planning, and robustness, followed by a real-world deployment.

Experimental Setting

We evaluate our proposed method on 50 test indoor scenes. 10 pairs of start and end points are randomly selected in each scene, totaling 500 test pairs. Tab. 1 presents statistics on the straight-line distance and travel distance between the start and end points of the 500 test episodes. Collisions may occur because the floor plan does not account for furniture obstacles. To mitigate this, we adopt a strategy in which the agent will rotate 45 degrees clockwise and re-predict future actions when a collision happens.

	Min	Max	Mean	Median
straight-line distance	5.89	29.87	7.47	6.51
travel distance	6.97	34.72	9.15	8.11

Table 1: **Statistics of test episodes.** The units are in meters.

Baselines In our evaluation, in addition to Naive-FloDiff and Loc-FloDiff, we further design a modular baseline including a localization module and an explicit path-planning module, denoted as Loc-A*. The localization modules in both Loc-FloDiff and Loc-A* are implemented in two distinct ways, *i.e.*, using GT or F³Loc (Chen et al. 2024), comparing their effectiveness across different settings. The five methods are described as follows:

- **Loc-A* (F³):** In this baseline, the agent localizes itself using pre-trained F³Loc (Chen et al. 2024). Subsequently, the A* algorithm is employed to plan a trajectory from the predicted position to the goal on the floor plan and execute the mapped actions in the environment.
- **Loc-A* (GT):** Unlike **Loc-A* (F³)**, this baseline replaces the predicted pose with the ground truth pose.
- **Naive-FloDiff:** This method directly learns to predict the agent pose during policy training.
- **Loc-FloDiff (F³):** This method is based on our Loc-FloDiff model and utilizes the same localization module as **Loc-A* (F³)**.
- **Loc-FloDiff (GT):** Unlike **Loc-FloDiff (F³)**, the predicted poses are replaced by the ground truth poses.

Metrics The performance of the above navigation methods is evaluated using two primary metrics, *i.e.*,

- **Success Rate (SR):** $\text{SR} = \frac{1}{N} \sum_{i=1}^N S_i$, where N is the total number of test episodes. S_i equals 1 if episode i is successful; otherwise, it equals 0.
- **Success Weight by Path Length (SPL):** $\text{SPL} = \frac{1}{N} \sum_{i=1}^N S_i \cdot \frac{l_i}{\max(p_i, l_i)}$, where p_i is the length of the agent’s path, and l_i is the shortest path. The metric measures the efficiency of navigation.

Recall that a navigation task is successful if the final position is within a specified distance threshold τ_d from the goal and the number of collisions does not exceed a given threshold τ_c . Since the maximum shortest path length in the test set is 34.72 meters, any trajectory exceeding 100 is deemed a failure during the evaluation.

Results and Discussions

Main Results We present the main comparison results in Tab. 2, computing the SR and SPL under varying τ_c and τ_d . We observe that Loc-FloDiff (GT) consistently outperforms all other methods across varying conditions in both SR and SPL, indicating that the proposed diffusion policy-based method can effectively solve the FloNa. Below, we analyze the methods’ performance across various dimensions, including localization, collision avoidance, planning, and robustness. We provide additional experimental results and qualitative visualization in *supplementary material*.

Method		Loc-A* (F ³)			Loc-A* (GT)			Naive-FloDiff			Loc-FloDiff (F ³)			Loc-FloDiff (GT)			
		$\tau_d(m)$			$\tau_d(m)$			$\tau_d(m)$			$\tau_d(m)$			$\tau_d(m)$			
		0.25	0.30	0.35	0.25	0.30	0.35	0.25	0.30	0.35	0.25	0.30	0.35	0.25	0.30	0.35	
SR(%) \uparrow	τ_c	10	0.00	0.60	0.60	<u>16.6</u>	<u>25.4</u>	<u>38.6</u>	4.20	5.00	5.60	4.80	6.00	6.40	39.0	44.2	44.6
		30	0.00	0.60	0.60	<u>18.2</u>	<u>28.4</u>	<u>43.8</u>	9.20	11.8	13.0	11.6	14.0	15.0	53.4	60.8	61.0
		50	0.00	0.60	0.60	<u>18.5</u>	<u>29.4</u>	<u>45.6</u>	12.0	15.6	17.2	17.8	20.6	22.4	59.2	68.4	68.4
		∞	0.00	0.60	0.60	18.6	29.6	<u>45.8</u>	20.2	24.8	27.0	<u>26.8</u>	<u>30.8</u>	33.0	66.0	75.8	76.0
SPL(%) \uparrow	τ_c	10	0.00	0.59	0.59	<u>16.5</u>	<u>25.3</u>	<u>38.51</u>	3.31	4.15	4.68	4.02	5.24	5.54	36.3	41.4	42.1
		30	0.00	0.59	0.59	<u>18.1</u>	<u>28.3</u>	<u>43.51</u>	4.92	6.46	7.26	6.40	8.25	8.98	44.1	50.4	51.2
		50	0.00	0.60	0.60	<u>18.5</u>	<u>29.3</u>	<u>45.11</u>	5.54	7.31	8.30	8.07	9.98	10.8	46.6	53.4	54.1
		∞	0.00	0.60	0.60	<u>18.5</u>	<u>29.4</u>	<u>45.31</u>	6.82	8.71	9.74	9.58	11.7	12.5	48.2	55.2	56.0

Table 2: **SR (%) and SPL (%) of different models under various conditions.** The **bold** number indicates the best result, and the underlined number represents the second-best result.

Method	$\tau_c = 10$	$\tau_c = 30$	$\tau_c = 50$
Loc-A* (F ³)	9.94	29.82	49.71
Loc-A* (GT)	6.03	16.21	25.65
Naive-FloDiff	9.66	28.02	45.26
Loc-FloDiff (F ³)	9.52	27.40	43.66
Loc-FloDiff (GT)	6.44	15.57	22.45

Table 3: **Mean collision counts.** A small number is better.

Localization Analysis The results in Tab. 2 show that the success rate of Naive-FloDiff is lower than both Loc-FloDiff (GT) and Loc-FloDiff (F³). This suggests that the modular method of decoupling the localization module is more effective in tackling FloNa compared to the end-to-end method. It appears that the encoders face challenges in simultaneously encoding information for both localization and planning.

Notably, Loc-A* (F³) and Loc-FloDiff (F³) perform worse than their ground truth counterparts. Upon reviewing the planned trajectories, we observe that agents using F³Loc often become stuck due to collisions with obstacles in confined areas. This issue is primarily caused by inaccurate localization, which results in unreasonable path planning. We suppose this unsatisfactory localization stems from using F³Loc, which is pre-trained on data collected from an agent with a height of 1.70m, significantly different from our agent with 0.85m height. We also find that, even under similar unsatisfactory localization conditions, the success rate of LocFloDiff (F³) does not degrade as severely as that of Loc-A*. Furthermore, Loc-FloDiff (F³) even performs better than Loc-A* (GT) when $\tau_c = \infty$ and $\tau_d = 0.25m$ or $\tau_d = 0.3m$. This result suggests that while Loc-FloDiff is affected by localization accuracy, it demonstrates a certain robustness to noisy input, allowing it to perform reasonably even under less accurate localization.

Collision Avoidance To quantitatively evaluate the collision avoidance performance, we present the mean collision counts of successful cases with $\tau_d = 0.3m$ under varying τ_c in Tab. 3. The results demonstrate that Loc-FloDiff (GT) outperforms Loc-A* (GT), and Loc-FloDiff (F³) also results in fewer collisions than Loc-A* (F³). These findings suggest

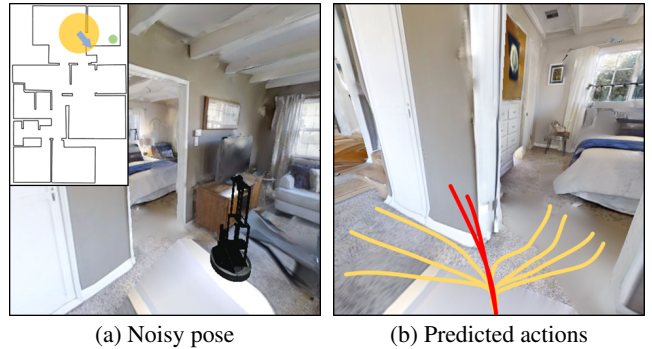


Figure 4: **Robustness of Loc-FloDiff (GT).** (a) The blue arrow is the ground truth agent pose, and the yellow circle indicates the noisy poses. (b) Our method can generate diverse collision-free paths (in yellow), even given the noisy poses. The red collision paths are generated by Loc-A* (GT).

that, when using the same localization module, Loc-FloDiff collides less than Loc-A*, highlighting its collision avoidance capability during navigation.

To qualitatively assess the collision avoidance capability of Loc-FloDiff, we conduct a diagnostic experiment using noisy ground truth localization. Specifically, the agent is initialized in front of a door, facing the door frame, as shown in Fig. 4a. The agent’s goal is to navigate to the nearest room through the door, with the green goal marked on the floor plan. The position p_t input to Loc-FloDiff is sampled from Gaussian noise centered on the ground truth position, with a variance of 1m, represented by the yellow circle in Fig. 4a. The orientation input r_t is sampled from a uniform distribution over 360°. Despite the noisy input, the agent successfully predicts collision-free paths most of the time, as shown with the yellow lines in Fig. 4b (although some predicted actions caused the agent to move away from the goal). In contrast, Loc-A* frequently predicts trajectories with collision, as indicated by the red lines in Fig. 4b.

Planning The SR and SPL results in Tab. 2 demonstrate that Loc-FloDiff (GT) achieves superior performance, highlighting its effectiveness and efficiency in path planning.

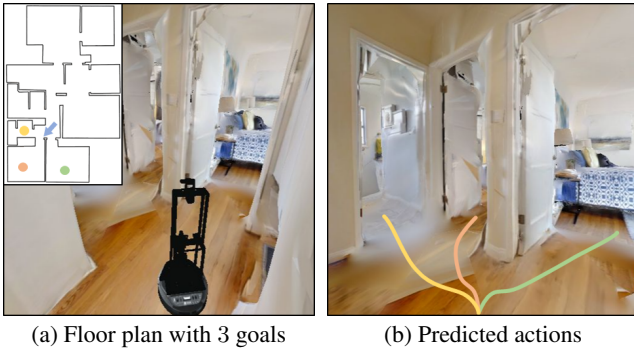


Figure 5: **Agent behavior varies given different goals.** (a) The agent starts from the same position but with three different goals. (b) Our approach predicts three distinct paths, each corresponding to a specific goal, as indicated by the respective colors.

Noise Variance	0.1m	0.3m	0.5m	GT
SR (%) \uparrow	26.0	20.0	24.0	68.4
SPL (%) \uparrow	10.8	8.21	12.2	53.4

Table 4: Results of Loc-FloDiff (GT) with noisy ground truth pose under different noise levels.

To qualitatively assess the planning ability of Loc-FloDiff (GT), we position the agent in front of three rooms, as shown in Fig. 5a, with the blue arrow indicating its position and heading direction. We assign three goal positions to the agent, each in a separate room and marked by a distinct color on the floor plan in Fig. 5a. As shown in Fig. 5b, the resulting action sequences, which align with the goal colors, successfully guide the agent to the correct rooms.

Robustness As previously mentioned, Fig. 4 qualitatively demonstrates that our method can robustly predict collision-free actions most of the time, even with noisy position inputs. To further evaluate the robustness of our method quantitatively, we conduct three experiments in which the Loc-FloDiff (GT) agent navigates to the goal with noisy pose inputs. In each experiment, the agent receives poses with Gaussian noise at a specific variance, *i.e.*, $var = 0.1m$, $var = 0.3m$, and $var = 0.5m$. All three experiments are evaluated with 50 cases, with one case per scene from the test set. The results, including SR and SPL metrics under the conditions $\tau_c = 50$ and $\tau_d = 0.3m$, are presented in Tab. 4. Although adding noise to the input pose leads to a decline in both SR and SPL, the performance does not degrade continuously as the noise variance increases. This stability highlights the model’s capability of handling uncertainties in localization while maintaining navigation performance.

Real-world Deployment

To validate the effectiveness of the model, we deploy our policy on an AGV. We observe that the agent surprisingly completes the navigation tasks in an **unseen real-world** environment even without finetuning, demonstrating its ro-

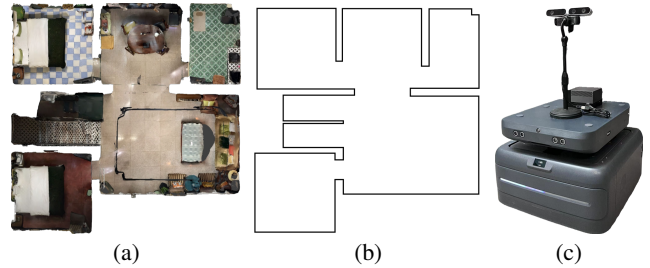


Figure 6: **Real-world Deployment.** (a) The top view of the real scene and (b) the floor plan of the scene. (c) The robot platform features a wheeled base, an RGB camera, and an NVIDIA Jetson AGX Orin.

bustness and generalization capabilities. **We recommend referring to our project website for the demonstration video of the planning results.**

We conduct the experiment in an unseen apartment with an area of about $108m^2$. Fig. 6a and Fig. 6b show the scene layout and corresponding floor plan. Fig. 6c shows the appearance of the AGV robot. The agent stands 0.36 meters tall, with a rectangular base measuring 0.46 meters in length and 0.42 meters in width. We additionally mounted an RGB camera on it and adjusted the camera height to 0.8m. The apartment is pre-scanned, and we evaluate our Loc-FloDiff (GT) in the scanned scene. The model generates actions based on RGB observations in simulation, which are then applied to the AGV and converted into linear and angular velocities using PD controller in the real world. Our model achieves an inference rate of approximately 1.88Hz when running on an NVIDIA Jetson AGX Orin. To provide the current pose for the agent, we use a single-line LiDAR-based odometry algorithm. Before the testing, we manually annotate the agent’s starting pose, represented as \mathcal{T}_0 , in the scene coordinate system. During navigation, the LiDAR odometry continuously calculates the agent’s pose relative to \mathcal{T}_0 and converts it to the scene coordinate system. At time t , the agent pose \mathcal{T}_t is calculated as $\mathcal{T}_t = \mathcal{T}_0 \circ \mathcal{T}_{t \rightarrow 0}$, where $\mathcal{T}_{t \rightarrow 0}$ represents the relative pose from t to 0 and \circ represents the coordinate transformation operation.

Conclusion

In summary, this work introduces FloNa, the first to incorporate floor plans into embodied visual navigation. To tackle the FloNa, we develop FloDiff, an efficient and effective diffusion policy framework integrating an explicit localization module, and curate a dataset for benchmarking. The superior performance of FloDiff over comparison baselines highlights the promising potential of our approach. By presenting this practical setting and solution, we aim to inspire further research in the visual navigation community and contribute to advancements in the field.

Acknowledgements

This project was supported by the National Natural Science Foundation of China (NSFC) under Grant No.62172043.

References

- Anderson, P.; Chang, A.; Chaplot, D. S.; Dosovitskiy, A.; Gupta, S.; Koltun, V.; Kosecka, J.; Malik, J.; Mottaghi, R.; Savva, M.; et al. 2018a. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*.
- Anderson, P.; Wu, Q.; Teney, D.; Bruce, J.; Johnson, M.; Sünderhauf, N.; Reid, I.; Gould, S.; and Van Den Hengel, A. 2018b. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Boniardi, F.; Caselitz, T.; Kümmerle, R.; and Burgard, W. 2019a. A pose graph-based localization system for long-term navigation in CAD floor plans. *Robotics and Autonomous Systems*, 112: 84–97.
- Boniardi, F.; Caselitz, T.; Kümmerle, R.; and Burgard, W. 2017. Robust LiDAR-based localization in architectural floor plans. In *International Conference on Intelligent Robots and Systems (IROS)*.
- Boniardi, F.; Valada, A.; Mohan, R.; Caselitz, T.; and Burgard, W. 2019b. Robot Localization in Floor Plans Using a Room Layout Edge Extraction Network. In *International Conference on Intelligent Robots and Systems (IROS)*.
- Chaplot, D. S.; Gandhi, D.; Gupta, A.; and Salakhutdinov, R. 2020a. Object Goal Navigation using Goal-Oriented Semantic Exploration. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Chaplot, D. S.; Salakhutdinov, R.; Gupta, A.; and Gupta, S. 2020b. Neural topological slam for visual navigation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Chen, C.; Jain, U.; Schissler, C.; Gari, S. V. A.; Al-Halah, Z.; Ithapu, V. K.; Robinson, P.; and Grauman, K. 2020. Soundspaces: Audio-visual navigation in 3d environments. In *European Conference on Computer Vision (ECCV)*.
- Chen, C.; Majumder, S.; Ziad, A.-H.; Gao, R.; Kumar Ramakrishnan, S.; and Grauman, K. 2021. Learning to Set Waypoints for Audio-Visual Navigation. In *International Conference on Learning Representations (ICLR)*.
- Chen, C.; Wang, R.; Vogel, C.; and Pollefeys, M. 2024. F³Loc: Fusion and Filtering for Floorplan Localization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Chi, C.; Feng, S.; Du, Y.; Xu, Z.; Cousineau, E.; Burchfiel, B.; and Song, S. 2023. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Robotics: Science and Systems (RSS)*.
- Chu, H.; Kim, D. K.; and Chen, T. 2015. You are here: Mimicking the human thinking process in reading floor-plans. In *International Conference on Computer Vision (ICCV)*.
- Gadre, S. Y.; Wortsman, M.; Ilharco, G.; Schmidt, L.; and Song, S. 2023. Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Gan, C.; Zhang, Y.; Wu, J.; Gong, B.; and Tenenbaum, J. B. 2020. Look, listen, and act: Towards audio-visual embodied navigation. In *International Conference on Robotics and Automation (ICRA)*.
- Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Howard-Jenkins, H.; and Prisacariu, V. A. 2022. LaLaLoc++: Global floor plan comprehension for layout localisation in unvisited environments. In *European Conference on Computer Vision (ECCV)*.
- Howard-Jenkins, H.; Ruiz-Sarmiento, J.-R.; and Prisacariu, V. A. 2021. LaLaLoc: Latent Layout Localisation in Dynamic, Unvisited Environments. In *International Conference on Computer Vision (ICCV)*.
- Ito, S.; Endres, F.; Kuderer, M.; Tipaldi, G. D.; Stachniss, C.; and Burgard, W. 2014. W-rgb-d: floor-plan-based indoor global localization using a depth camera and wifi. In *International Conference on Robotics and Automation (ICRA)*.
- Janner, M.; Du, Y.; Tenenbaum, J.; and Levine, S. 2022. Planning with Diffusion for Flexible Behavior Synthesis. In *International Conference on Machine Learning (ICML)*.
- Krantz, J.; Gervet, T.; Yadav, K.; Wang, A.; Paxton, C.; Mottaghi, R.; Batra, D.; Malik, J.; Lee, S.; and Chaplot, D. S. 2023. Navigating to objects specified by images. In *International Conference on Computer Vision (ICCV)*.
- Li, C.; Xia, F.; Martín-Martín, R.; Lingelbach, M.; Srivastava, S.; Shen, B.; Vainio, K. E.; Gokmen, C.; Dharan, G.; Jain, T.; Kurenkov, A.; Liu, K.; Gweon, H.; Wu, J.; Fei-Fei, L.; and Savarese, S. 2021a. iGibson 2.0: Object-centric simulation for robot learning of everyday household tasks. In *Conference on Robot Learning (CoRL)*.
- Li, J.; Chan, C. L.; Le Chan, J.; Li, Z.; Wan, K. W.; and Yau, W. Y. 2021b. Cognitive navigation for indoor environment using floorplan. In *International Conference on Intelligent Robots and Systems (IROS)*.
- Li, Z.; Ang, M. H.; and Rus, D. 2020. Online localization with imprecise floor space maps using stochastic gradient descent. In *International Conference on Intelligent Robots and Systems (IROS)*.
- Loshchilov, I.; Hutter, F.; et al. 2017. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5.
- Mendez, O.; Hadfield, S.; Pugeault, N.; and Bowden, R. 2020. SeDAR: reading floorplans like a human—using deep learning to enable human-inspired localisation. *International Journal of Computer Vision (IJCV)*, 128(5): 1286–1310.
- Min, S. Y.; Chaplot, D. S.; Ravikumar, P.; Bisk, Y.; and Salakhutdinov, R. 2021. Film: Following instructions in language with modular methods. *arXiv preprint arXiv:2110.07342*.
- Min, Z.; Khosravan, N.; Bessinger, Z.; Narayana, M.; Kang, S. B.; Dunn, E.; and Boyadzhiev, I. 2022. Laser: Latent space rendering for 2d visual localization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Nichol, A. Q.; and Dhariwal, P. 2021. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning (ICML)*.

- Partsey, R.; Wijmans, E.; Yokoyama, N.; Doboševych, O.; Batra, D.; and Maksymets, O. 2022. Is mapping necessary for realistic pointgoal navigation? In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Ramakrishnan, S. K.; Chaplot, D. S.; Al-Halah, Z.; Malik, J.; and Grauman, K. 2022. PONI: Potential Functions for ObjectGoal Navigation With Interaction-Free Learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Ramakrishnan, S. K.; Gokaslan, A.; Wijmans, E.; Maksymets, O.; Clegg, A.; Turner, J.; Undersander, E.; Galuba, W.; Westbury, A.; Chang, A. X.; et al. 2021. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*.
- Savva, M.; Kadian, A.; Maksymets, O.; Zhao, Y.; Wijmans, E.; Jain, B.; Straub, J.; Liu, J.; Koltun, V.; Malik, J.; Parikh, D.; and Batra, D. 2019. Habitat: A platform for embodied ai research. In *International Conference on Computer Vision (ICCV)*.
- Setalaphruk, V.; Ueno, A.; Kume, I.; Kono, Y.; and Kidode, M. 2003. Robot navigation in corridor environments using a sketch floor map. In *International Symposium on Computational Intelligence in Robotics and Automation*.
- Shah, D.; Eysenbach, B.; Kahn, G.; Rhinehart, N.; and Levine, S. 2021. ViNG: Learning Open-World Navigation with Visual Goals. In *International Conference on Robotics and Automation (ICRA)*.
- Shah, D.; Sridhar, A.; Dashora, N.; Stachowicz, K.; Black, K.; Hirose, N.; and Levine, S. 2023. ViNT: A Foundation Model for Visual Navigation. In *Conference on Robot Learning (CoRL)*.
- Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning (ICML)*.
- Song, C. H.; Wu, J.; Washington, C.; Sadler, B. M.; Chao, W.-L.; and Su, Y. 2023. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *International Conference on Computer Vision (ICCV)*.
- Sridhar, A.; Shah, D.; Glossop, C.; and Levine, S. 2023. Nomad: Goal masked diffusion policies for navigation and exploration. *arXiv preprint arXiv:2310.07896*.
- Tan, M.; and Le, Q. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*.
- Tang, T.; Du, H.; Yu, X.; and Yang, Y. 2022. Monocular camera-based point-goal navigation by learning depth channel and cross-modality pyramid fusion. In *AAAI Conference on Artificial Intelligence (AAAI)*.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Wang, H.; Liang, W.; Van Gool, L.; and Wang, W. 2023a. Dreamwalker: Mental planning for continuous vision-language navigation. In *International Conference on Computer Vision (ICCV)*.
- Wang, H.; Wang, W.; Liang, W.; Hoi, S. C.; Shen, J.; and Gool, L. V. 2023b. Active perception for visual-language navigation. *International Journal of Computer Vision (IJCV)*, 131(3): 607–625.
- Wang, X.; Marcotte, R. J.; and Olson, E. 2019. GLFP: Global localization from a floor plan. In *International Conference on Intelligent Robots and Systems (IROS)*.
- Wijmans, E.; Kadian, A.; Morcos, A.; Lee, S.; Essa, I.; Parikh, D.; Savva, M.; and Batra, D. 2019. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. *arXiv preprint arXiv:1911.00357*.
- Xia, F.; Zamir, A. R.; He, Z.; Sax, A.; Malik, J.; and Savarese, S. 2018. Gibson env: Real-world perception for embodied agents. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Yadav, K.; Majumdar, A.; Ramrakhya, R.; Yokoyama, N.; Baevski, A.; Kira, Z.; Maksymets, O.; and Batra, D. 2023. Ovrl-v2: A simple state-of-art baseline for imagenav and objectnav. *arXiv preprint arXiv:2303.07798*.
- Zhao, X.; Agrawal, H.; Batra, D.; and Schwing, A. G. 2021. The surprising effectiveness of visual odometry techniques for embodied pointgoal navigation. In *International Conference on Computer Vision (ICCV)*.
- Zhu, Y.; Mottaghi, R.; Kolve, E.; Lim, J. J.; Gupta, A.; Fei-Fei, L.; and Farhadi, A. 2017. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *International Conference on Robotics and Automation (ICRA)*.