MUTUAL GUIDANCE DIFFUSION FOR OBJECT NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Modular object-goal navigation (ObjectNav) typically constructs a semantic map and then plans from that map as a fixed condition. Under partial observability, however, route selection and beliefs about unobserved space are coupled: a route determines which future regions matter, while plausible future layouts constrain which routes are executable. We introduce **`MGNav`**, a dual-branch diffusion framework that jointly generates future waypoints and waypoint-aligned semantic layouts. At every denoising step, gated bidirectional cross-attention lets evolving route hypotheses constrain layout imagination and lets imagined free space, obstacles, and semantic context refine the route. This interaction targets the mismatch between scene reasoning and planning that arises when the map is consumed as a fixed condition, rather than merely adding an auxiliary prediction task. Experiments on Gibson and Matterport3D show consistent gains over a trajectory-diffusion baseline and over controlled concatenation and ungated cross-attention alternatives. We also state the additional trajectory and layout supervision required by our method and analyze the assumptions and failure modes of coupled generation.

1 INTRODUCTION

In Object Navigation (ObjectNav), an agent, starting each navigation episode in an unseen environment, navigates to a target object based on visual observations (*e.g.*, a remote control). The episode is considered successful if it reaches sufficient proximity within a fixed time limit. In contrast to classical navigation tasks (DeSouza & Kak, 2002; Thrun, 2000), where the agent is guided solely by geometric signals to freely explore the environment, ObjectNav presents a partially observable problem that often leads to ineffective behaviors (*e.g.*, repeated backtracking, oscillation, or getting trapped at dead ends).

As a routine activity in human daily life, when the target is invisible most of the time, *past* experiences serve as critical clues for inferring its location and planning *future* waypoints. Inspired by this, reinforcement learning (RL) approaches (Zhu et al., 2017; Mirowski et al., 2017; Mousavian et al., 2019; Ramrakhya et al., 2023) establish a mapping from information-rich RGB-D observations to low-level actions in an end-to-end manner, enabling robot navigation through carefully engineered rewards and fixed policies. While achieving commendable success, fine-grained scene context details are lost by such *implicit* encoding. Humans do not learn simple stimulus-response mappings; instead, they construct a "cognitive map" (Tolman, 1948; Brady et al., 2008; Eichenbaum & Cohen, 1997; Smith et al., 1975) that binds inter-object relationships, goals, and the layout of contextual objects. Based on this, subsequent solutions adopted a module-based framework (Chaplot et al., 2020a; Ramakrishnan et al., 2022; Zhai & Wang, 2023; Zhang et al., 2024; Zhu et al., 2022) consisting of an *object detector*, a *map construction* module (either online (Chaplot et al., 2020a) or offline (Ramakrishnan et al., 2022)), a *waypoint predictor*, and several *policy refinement* processes.

This *perception-first, planning-next* regime has led subsequent improvements to focus on enhancing individual modules, *e.g.*, boosting perception performance, adopting information-rich maps, or introducing auxiliary tasks to refine the policy, while largely overlooking how coordination across modules at the framework level affects navigation performance and efficiency. Human navigation does not proceed in a strictly sequential manner. Instead, humans form expectations about future scenes on the fly while observing, and use these expectations to decide which direction to approach next. For instance, when searching for a remote control, a person does not first scan the entire living

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

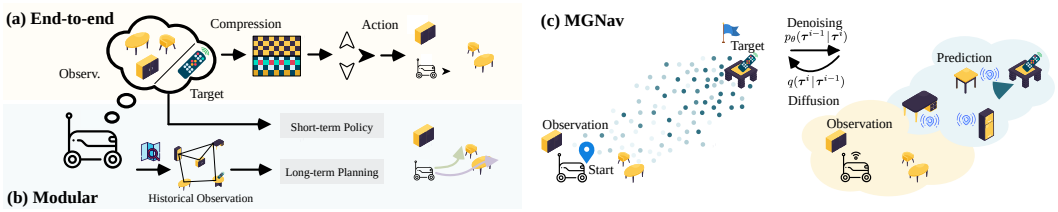


Figure 1: **Comparison of different ObjectNav paradigms.** (a) End-to-end methods directly map observations to actions, but often lack explicit spatial structure and long-horizon planning consistency. (b) Modular *perceive-then-plan* methods decompose navigation into separate perception and planning stages, but neglect their dynamic coordination during decision making. (c) **MGNav** jointly models future waypoint planning and future semantic layout imagination, allowing route hypotheses and future scene constraints to be progressively refined together.

room to build a complete map before planning. Rather, they continuously leverage interactions with environmental objects as new constraints to revise their path. This process is highly flexible and exhibits multi-modal behavior. Such coordination allows humans to achieve robust navigation even when individual capabilities, such as object perception, are imperfect.

Diffusion models capture complex, multi-modal distributions by learning to reverse a gradual noising process, and their iterative sampling has two properties that suit this coordination problem. ❶ It admits flexible conditioning, so auxiliary guidance signals can steer sampling. ❷ It refines a coarse hypothesis progressively, so information exchanged at intermediate states can still change the outcome. These properties make diffusion a natural substrate for coupling perception and planning within one generative process.

Motivated by this observation, we propose **MGNav**, a **M**utual **G**uidance diffusion framework for ObjectNav that jointly models future trajectory planning and future semantic layout imagination. Specifically, **MGNav** contains two coupled diffusion branches: a trajectory branch that generates a sequence of future waypoints, and a semantic layout branch that predicts a sequence of waypoint-aligned future local semantic layouts. The two branches are connected through a ***mutual guidance mechanism*** that exchanges intermediate denoising states at every diffusion step, as shown in Figure 1. In this way, imagined future layouts provide spatial constraints such as free space, obstacles, and semantic context for route generation, while the evolving route hypothesis provides intention-aware anchors that guide future layout prediction. Rather than treating scene understanding as a fixed condition for planning, **MGNav** allows planning and future scene imagination to be refined jointly throughout the generation process.

Our contribution is threefold. First, we formulate modular ObjectNav as *joint future imagination*, making the dependence between route hypotheses and unobserved local structure explicit. Second, we introduce a coupled diffusion model whose gated, bidirectional exchange occurs throughout denoising rather than through one-way conditioning or late fusion. Third, we evaluate the mechanism with component, fusion, and layout-representation ablations on Gibson and Matterport3D, while explicitly documenting the supervision and simulator assumptions under which the comparison holds. The central claim is therefore not that sequence prediction or look-ahead planning is new, but that progressively coupling route and layout hypotheses improves modular ObjectNav under partial observability.

2 Related Work

Visual Navigation. Classic visual navigation aims to find a collision-free path to a given location (Wijmans et al., 2020; DeSouza & Kak, 2002). In ObjectNav, the agent must navigate to an object category in an unseen environment, which further requires semantic reasoning under partial observability. According to network architecture design, existing ObjectNav methods can be broadly categorized into end-to-end learning and modular methods.

End-to-end methods (Ye et al., 2021; Ramrakhya et al., 2022; Khandelwal et al., 2022) directly map raw visual observations or learned visual representations to navigation actions. Such methods are

flexible and can benefit from advances in representation learning (Yadav et al., 2022; Radford et al., 2021). They also avoid explicitly designing intermediate mapping or planning modules. However, since scene structure and spatial memory are often encoded implicitly, it can be difficult for these policies to maintain long-horizon consistency, diagnose failure cases, or explicitly reason about unexplored regions. Many such policies rely on recurrent architectures (Hochreiter & Schmidhuber, 1997; Chung et al., 2014), which may accumulate errors over long episodes; recent spatio-temporal Transformer policies such as NaviFormer (Xie et al., 2025) alleviate this forgetting, but scene structure remains implicit in their representations, so unexplored regions and route feasibility cannot be reasoned about explicitly.

Modular methods (Chaplot et al., 2020a; Ramakrishnan et al., 2022; Zhu et al., 2022) instead maintain explicit spatial memory, typically by extending occupancy maps with semantic channels. A planner or goal-prediction module then selects navigation targets, and a local policy executes low-level actions. This design improves interpretability and makes it easier to incorporate geometry, object semantics, and historical observations. However, most modular methods still follow a sequential *perceive-then-plan* paradigm: perception first builds or updates a map, and planning then consumes this representation as a fixed condition. Later improvements often focus on individual modules, such as better semantic maps, improved goal prediction, or policy refinement. In contrast, **MGNav** focuses on the coordination between planning and future scene reasoning. Rather than treating the map as a static input to the planner, **MGNav** jointly generates future waypoints and waypoint-aligned future semantic layouts, enabling the two predictions to refine each other during denoising.

Diffusion Models. Diffusion models (DDPMs (Ho et al., 2020)) learn a data distribution by reversing a gradual noising process, with strong results in image generation (Ramesh et al., 2022; Saharia et al., 2022; Peebles & Xie, 2023), super-resolution (Rombach et al., 2022; Wu et al., 2023), inpainting (Lugmayr et al., 2022; Xie et al., 2023), and editing (Brooks et al., 2023; Sheynin et al., 2023), implemented with UNet (Rombach et al., 2022; Saharia et al., 2022) or Transformer (Peebles & Xie, 2023; Ramesh et al., 2022; Wu et al., 2023) backbones. In robotics and sequential decision making, Diffuser (Janner et al., 2022), Crossway Diffusion (Li et al., 2023), and Diffusion Policy (Chi et al., 2023) show that diffusion can represent multi-modal behavior distributions and generate whole action or trajectory sequences, which is attractive for ObjectNav, where several plausible routes may coexist under partial observation.

Generative planning and world models. T-Diff (Yu et al., 2024) diffuses ObjectNav trajectories and Diffusion-as-Reasoning (Ji et al., 2024) diffuses semantic maps; each treats the other quantity as fixed. Our work is also related to diffusion-based joint navigation (Niedoba et al., 2023), active guidance of diffusion policies with learned dynamics (Du & Song, 2025), and generative scene reasoning for ObjectNav (Zhang et al., 2024). These methods establish that future-state prediction or external dynamics can guide action generation. **MGNav** addresses a different coupling: it jointly denoises a route and a sequence of route-aligned semantic layouts, with each branch conditioning the intermediate states of the other. Unlike a one-way conditional planner, both predictions remain editable throughout sampling; unlike late fusion, consistency is imposed before either output is finalized. We therefore compare direct concatenation, ungated cross-attention, and gated cross-attention as exchange mechanisms in the diagnostic analysis. This positioning also clarifies that the layout branch is a local, non-autoregressive world model specialized to planning-relevant semantics rather than a general environment simulator.

3 METHODOLOGY

3.1 JOINT FUTURE IMAGINATION FOR OBJECT NAVIGATION

At each step the agent only holds an incomplete semantic map built from past observations, while the target may lie behind occlusions or outside the explored area. Conventional modular methods plan from this fixed map, implicitly assuming it suffices to choose the next waypoint; in partially observed scenes, however, a feasible route also depends on what structure is likely to appear along the intended path. We therefore formulate ObjectNav as *joint future imagination*: **MGNav** predicts both *where to move* and *what local scene structure may be encountered along the route*. This is not a multi-task extension, because the two variables are coupled: the route determines which regions should be imagined, and the imagined layouts constrain whether the route is plausible.

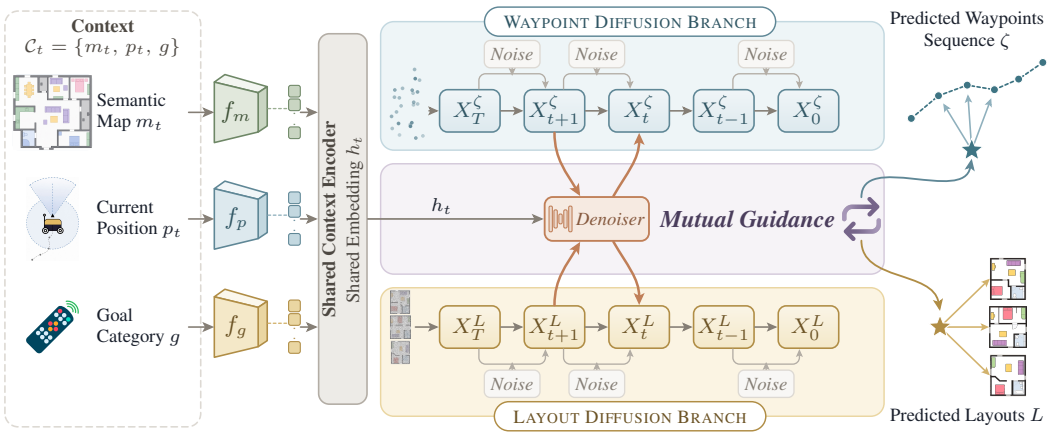


Figure 2: **Overview of MGNav.** We formulate ObjectNav as a joint future imagination problem. Given the current semantic map, pose, and target category, **MGNav** jointly generates a future waypoint sequence and waypoint-aligned future semantic layouts. The trajectory and layout branches exchange intermediate denoising states through mutual guidance, allowing route hypotheses and future spatial constraints to progressively refine each other. The resulting trajectory is executed in a receding-horizon manner.

At navigation timestep t, the agent maintains a semantic map m_t (explored free space, obstacles, unknown regions, and semantics), observes its current pose p_t, and is given a target category g; we write the navigation context as $\mathcal{C}_t = \{m_t, p_t, g\}$. From this context, **MGNav** predicts two future variables over a horizon of K steps. The first is a future waypoint sequence $\xi_t = \{w_1, \ldots, w_K\}$ with $w_k \in \mathbb{R}^{d_w}$, which describes where the agent intends to move. The second is a sequence of waypoint-aligned future semantic layouts $\mathcal{L}_t = \{l_1, \ldots, l_K\}$ with $l_k \in \mathbb{R}^{H \times W \times C}$, where each l_k is a top-down semantic patch centered at w_k that describes the local structure the agent expects to find there, including traversable space, obstacles, unknown regions, and semantic occupancy. Both sequences share the index k, so the k-th layout is always the imagined surrounding of the k-th waypoint. We model the coupled conditional distribution

$$p(\xi_t, \mathcal{L}_t \mid \mathcal{C}_t). \tag{1}$$

This distribution captures two complementary forms of uncertainty: the multi-modal uncertainty of future routes and the uncertainty of future scene structures under partial observation. A factorized alternative, $p(\xi_t \mid \mathcal{C}_t)\, p(\mathcal{L}_t \mid \mathcal{C}_t)$, would treat the two predictions as conditionally independent given the current map. This is exactly the assumption that breaks down under partial observability: which layout matters depends on the chosen route, and which route is executable depends on the layout likely to be found along it. By modeling the two variables jointly, **MGNav** allows route hypotheses and future spatial constraints to be refined together before the agent physically commits to a subgoal.

3.2 Context Encoding

We encode the navigation context into a shared representation:

$$h_t = f_{\text{ctx}}(m_t, p_t, g) = \big[f_m(m_t); f_p(p_t); f_g(g)\big], \tag{2}$$

where $[\cdot;\cdot]$ denotes concatenation and $f_m(\cdot)$, $f_p(\cdot)$, and $f_g(\cdot)$ denote the map, pose, and goal encoders. The map encoder is a ResNet-18 whose first layer accepts the semantic-map channels, the pose encoder is a multilayer perceptron, and the goal encoder is a category embedding followed by a projection (Appendix C.1). The shared context h_t is used by both diffusion branches at every denoising step. Sharing one encoding has two consequences. First, trajectory generation and future layout imagination are grounded in exactly the same current evidence, so any disagreement between them reflects different hypotheses about the future rather than different views of the present, while the two branches can still maintain distinct latent hypotheses. Second, h_t is computed once per replanning step and reused across all denoising iterations, so the cost of encoding the map does not grow with the number of diffusion steps.

3.3 MUTUAL-GUIDED DUAL-BRANCH DIFFUSION

To model Equation (1), we introduce a dual-branch diffusion framework. One branch generates future waypoints, while the other imagines future semantic layouts. A straightforward design would denoise the two variables independently and fuse the final outputs. However, such late fusion is insufficient for navigation: the trajectory branch would not receive future spatial constraints during generation, and the layout branch would not know which regions are relevant to the agent's route intention. A one-way alternative, in which the layout is generated first and then used as a condition for the planner, recovers the perceive-then-plan structure at a finer granularity and still prevents the route from influencing what is imagined.

MGNav instead performs *mutual guidance* throughout the reverse diffusion process. At diffusion step τ, let ξ_t^τ denote the noisy waypoint sequence and Z_t^τ denote the noisy latent layout sequence. The two branches first compute branch-specific hidden states:

$$H_\xi^\tau = F_\xi(\xi_t^\tau, \pi(\tau), h_t), \qquad H_L^\tau = F_L(Z_t^\tau, \pi(\tau), h_t), \tag{3}$$

where $\pi(\tau)$ is the diffusion timestep embedding, and $F_\xi(\cdot)$ and $F_L(\cdot)$ are the trajectory and layout denoising backbones, implemented with DiT-style Transformer blocks that process waypoints and latent layouts as token sequences. The mutual guidance block then performs bidirectional information exchange:

$$\bar{H}_\xi^\tau = \mathrm{CA}_\xi(H_\xi^\tau, H_L^\tau), \qquad \bar{H}_L^\tau = \mathrm{CA}_L(H_L^\tau, H_\xi^\tau), \tag{4}$$

where CA_ξ and CA_L are cross-attention operators whose queries come from their own branch and whose keys and values come from the other. The layout-to-trajectory direction allows waypoint tokens to attend to imagined traversability, obstacle, and semantic cues. The trajectory-to-layout direction provides route-aware anchors, encouraging layout imagination to focus on regions that are relevant to the evolving route hypothesis. Attention is a natural choice for this exchange because the relevant correspondence is not fixed in advance: a waypoint approaching a doorway may need the layout of the next room rather than the patch with the same index, and an ambiguous layout patch may need to consult whichever waypoints pass closest to it.

Because early denoising states are noisy and may contain unreliable hypotheses, directly injecting all cross-branch features can be harmful. We therefore use gated residual fusion:

$$\tilde{H}_\xi^\tau = H_\xi^\tau + G_\xi^\tau \odot \bar{H}_\xi^\tau, \qquad \tilde{H}_L^\tau = H_L^\tau + G_L^\tau \odot \bar{H}_L^\tau, \tag{5}$$

where G_ξ^τ and G_L^τ are learnable sigmoid gates. The gates adaptively control how much information is transferred at each denoising stage. Intuitively, early steps can rely more on coarse self-branch hypotheses, while later steps can exploit stronger cross-branch constraints when both route and layout predictions become sharper. Because the exchange enters as a residual, a gate close to zero reduces each branch to an ordinary single-variable diffusion model; mutual guidance can therefore add evidence the network has learned to trust without overwriting a branch's own hypothesis. Appendix C.4 illustrates the complete block.

The result is a closed loop: the layout branch discourages waypoints that conflict with imagined occupied regions, and the trajectory branch reduces layout ambiguity by specifying where the agent intends to move, turning a serial pipeline into two mutually constrained denoising processes. Since the exchange is repeated at every denoising step, a route hypothesis can be revised many times before it is finalized, which is the second property of iterative sampling discussed in Section 1.

3.4 TRAJECTORY BRANCH: FUTURE ROUTE GENERATION

The trajectory branch denoises a waypoint sequence to generate a short-horizon route segment. Unlike step-wise action prediction, this branch predicts a complete future trajectory segment, which helps preserve long-horizon consistency and reduces the accumulation of local prediction errors. We represent the clean future trajectory as $\xi_t^0 = [w_1, \ldots, w_K] \in \mathbb{R}^{K \times d_w}$ with $d_w = 2$. Each waypoint is expressed relative to the current pose p_t and normalized by the local planning range, which keeps the regression targets on a comparable scale across episodes; when fewer than K future positions remain on the expert path, the last one is repeated (Appendix B.2). Both branches share the same Gaussian forward process,

$$\xi_t^\tau = \sqrt{\bar{\alpha}_\tau}\, \xi_t^0 + \sqrt{1 - \bar{\alpha}_\tau}\, \epsilon_\xi, \qquad Z_t^\tau = \sqrt{\bar{\alpha}_\tau}\, Z_t^0 + \sqrt{1 - \bar{\alpha}_\tau}\, \epsilon_L, \qquad \epsilon_\xi, \epsilon_L \sim \mathcal{N}(0, I), \tag{6}$$

where $\bar{\alpha}_\tau$ is the cumulative noise schedule and Z_t^0 is the latent layout sequence defined in Section 3.5.

The trajectory denoiser $\epsilon_\theta^\xi(\xi_t^\tau, \pi(\tau), h_t, \tilde{H}_\xi^\tau)$ predicts the injected noise from its guided state $\tilde{H}_\xi^\tau$, which already carries the layout-to-trajectory exchange of Equation (5). It is a Transformer that treats each waypoint as a token, capturing route continuity, turning behavior, and long-range goal direction; because $\tilde{H}_\xi^\tau$ contains the gated cross-attention over H_L^τ, the denoiser sees intermediate future-layout constraints rather than the current map alone. Denoising the whole segment at once also makes the waypoints consistent with one another by construction: an early waypoint and a later one are refined under the same attention, which discourages routes whose parts would individually be plausible but jointly infeasible.

3.5 Semantic Layout Branch: Future Scene Imagination

The semantic layout branch predicts top-down future semantic structures along the generated route. Rather than generating raw future RGB observations, we predict semantic layouts because they are compact, planner-friendly, and directly encode traversability and semantic context. Future RGB frames would spend model capacity on appearance, lighting, and texture, none of which decides whether a route is executable, whereas a semantic layout contains exactly the quantities a planner consumes.

For each future waypoint w_k, the layout l_k is obtained by cropping a local semantic patch in a frame centered at w_k, and $\mathcal{L}_t^0 = \{l_1, \ldots, l_K\}$ denotes the clean layout sequence. This waypoint-centered construction is crucial: it aligns the k-th route hypothesis with the k-th imagined local structure, making future scene prediction route-aware rather than free-form.

Direct diffusion over high-dimensional semantic patches is expensive and tends to overemphasize local pixel-level errors. We therefore perform layout diffusion in the latent space of an autoencoder with encoder E_ϕ and decoder D_ϕ: each patch is mapped to $z_k^0 = E_\phi(l_k)$, the sequence $Z_t^0 = [z_1^0, \ldots, z_K^0]$ is diffused with the forward process of Equation (6), and a layout is recovered as $D_\phi(z_k^0)$. Symmetrically to the trajectory branch, the layout denoiser $\epsilon_\theta^L(Z_t^\tau, \pi(\tau), h_t, \tilde{H}_L^\tau)$ predicts the latent noise from its guided state $\tilde{H}_L^\tau$. To preserve semantic fidelity after decoding, the autoencoder is trained with a reconstruction loss $\mathcal{L}_{\text{rec}} = \sum_{k=1}^K \mathcal{D}\big(D_\phi(z_k^0), l_k\big)$, where $\mathcal{D}$ is a channel-wise reconstruction loss. Decodability matters beyond interpretability: the obstacle channel of the decoded layout is what the geometric consistency term in Section 3.6 queries, so a latent space that could not be decoded would leave the two branches coupled only through attention.

3.6 Joint Optimization and Receding-Horizon Deployment

The two branches are trained jointly. Each denoiser is supervised with the standard noise-prediction objective on its own variable,

$$\begin{aligned} \mathcal{L}_{\text{traj}} &= \mathbb{E}_{\xi_t^0, \epsilon_\xi, \tau} \big\| \epsilon_\xi - \epsilon_\theta^\xi(\xi_t^\tau, \pi(\tau), h_t, \tilde{H}_\xi^\tau) \big\|_2^2, \\ \mathcal{L}_{\text{layout}} &= \mathbb{E}_{Z_t^0, \epsilon_L, \tau} \big\| \epsilon_L - \epsilon_\theta^L(Z_t^\tau, \pi(\tau), h_t, \tilde{H}_L^\tau) \big\|_2^2, \end{aligned} \tag{7}$$

where both expectations share the same diffusion step τ, so that the two branches are always exchanged at matching noise levels. To explicitly couple predicted routes with imagined layouts, we further introduce a geometric consistency term. Let $\hat{w}_k$ and $\hat{z}_k$ denote the clean estimates recovered from the noisy states and the predicted noise at step τ, let $\hat{O}_k$ denote the obstacle channel of the decoded predicted layout $\hat{l}_k = D_\phi(\hat{z}_k)$, and let $\Pi(\hat{w}_k)$ map the predicted waypoint into the coordinate frame of the k-th patch. We penalize trajectories that pass through imagined occupied regions:

$$\mathcal{L}_{\text{geo}} = \sum_{k=1}^K \hat{O}_k\big(\Pi(\hat{w}_k)\big). \tag{8}$$

The value $\hat{O}_k(\Pi(\hat{w}_k))$ is obtained by differentiable bilinear sampling. Because each training patch is anchored at the ground-truth waypoint w_k (Appendix B.3), $\Pi(\hat{w}_k)$ shifts with the prediction error $\hat{w}_k - w_k$, so the loss yields a non-zero gradient that pushes the route out of imagined obstacles and makes the coupling between the two branches explicit during training. We also impose a first-order smoothness prior $\mathcal{L}_{\text{smooth}} = \sum_{k=2}^K \|\hat{w}_k - \hat{w}_{k-1}\|_2^2$, which discourages large jumps between

Table 1: **ObjectNav results** on Gibson (Xia et al., 2018) and MP3D (Chang et al., 2017). A check in "Extra sup." denotes training signals beyond online interaction and standard pretrained detectors, such as expert or human demonstrations, ground-truth semantic maps, or large-scale pretrained visual encoders. Values are taken from the respective papers under their own protocols and are not necessarily directly comparable across rows; **bold** marks the best result among modular methods.

Method	Hist. Info.	Extra sup.	Gibson (Xia et al., 2018)			MP3D (Chang et al., 2017)		
			SR↑	SPL↑	DTS↓	SR↑	SPL↑	DTS↓
Random			0.4	0.4	3.89	0.5	0.5	8.05
End-to-end methods								
DD-PPO(Wijmans et al., 2020)			15.0	10.7	3.24	8.0	1.8	6.94
Red-Rabbit(Ye et al., 2021)			-	-	-	34.6	7.9	-
THDA(Maksymets et al., 2021)			-	-	-	28.4	11.0	5.62
SSCNav(Liang et al., 2021)		✓	-	-	-	27.1	11.2	5.71
EmbCLIP(Khandelwal et al., 2022)		✓	68.1	39.5	1.15	29.2	10.1	5.40
Habitat-Web(Ramrakhya et al., 2022)		✓	-	-	-	35.4	10.2	-
ENTL(Kotar et al., 2023)			-	-	-	17.0	5.0	-
NaviFormer(Xie et al., 2025)	✓		82.6	-	-	-	-	-
Modular methods								
ANS(Chaplot et al., 2020b)	✓		67.1	34.9	1.66	27.3	9.2	5.80
SemExp(Chaplot et al., 2020a)	✓		71.1	39.6	1.39	28.3	10.9	6.06
PONI(Ramakrishnan et al., 2022)	✓	✓	73.6	41.0	1.25	31.8	12.1	5.10
L2M(Georgakis et al., 2022)	✓	✓	-	-	-	32.1	11.0	5.12
Stubborn(Luo et al., 2022)	✓		-	-	-	31.2	13.5	5.01
PEANUT(Zhai & Wang, 2023)	✓	✓	-	-	-	**40.5**	**15.8**	-
3D-aware(Zhang et al., 2023)	✓		74.5	42.1	1.16	34.0	14.6	**4.78**
L3MVN(Yu et al., 2023)	✓		76.9	38.8	1.01	-	-	-
SGM(Zhang et al., 2024)	✓	✓	78.0	44.0	1.11	37.7	14.7	4.93
MGNav (Ours)	✓	✓	**80.3**	**45.5**	**0.98**	40.2	15.5	5.05

consecutive waypoints and yields routes that are easier for the local planner to follow. The final training objective is

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{traj}} + \lambda_2 \mathcal{L}_{\text{layout}} + \lambda_3 \mathcal{L}_{\text{rec}} + \lambda_4 \mathcal{L}_{\text{geo}} + \lambda_5 \mathcal{L}_{\text{smooth}}. \tag{9}$$

Each term has a distinct role: the two diffusion losses teach each branch to generate its own variable, $\mathcal{L}_{\text{rec}}$ keeps the layout latent decodable, $\mathcal{L}_{\text{geo}}$ ties the route to the imagined obstacles, and $\mathcal{L}_{\text{smooth}}$ keeps the route executable.

At test time, both branches are initialized from Gaussian noise, $\xi_t^T \sim \mathcal{N}(0, I)$ and $Z_t^T \sim \mathcal{N}(0, I)$, and for $\tau = T, \ldots, 1$ **MGNav** performs mutual-guided denoising to produce $\hat{\xi}_t$ and $\hat{\mathcal{L}}_t$. For execution, the agent follows a receding-horizon protocol. Instead of committing to the entire generated trajectory, it selects an intermediate waypoint $\hat{w}_{k_g}$ as a short-horizon subgoal and executes it with a local planner. After several environment steps, the semantic map is updated with new observations and **MGNav** replans from the updated context. This preserves the benefit of sequence-level future reasoning while retaining closed-loop reactivity under partial observability. The imagined layouts are never written into the map: they act only as intermediate variables that shape the route, and the map itself is updated exclusively from real observations.

4 Experiment

4.1 Experimental Setup

Datasets. We evaluate on Gibson (Xia et al., 2018) and Matterport3D (MP3D) (Chang et al., 2017) in Habitat. For Gibson, we use the 25-train/5-validation-scene tiny split and 1,000 validation episodes with six target categories, following the modular protocol of Ramakrishnan et al. (2022). For MP3D, we use 56 training and 11 validation scenes and evaluate on 2,195 validation episodes with 21 target categories. Thus, our reported MP3D results concern these 67 train/validation scenes rather than all 90 MP3D scenes; they should only be compared with results using the same validation protocol. The detailed goal categories are provided in the appendix.

Implementation Details. Training uses additional simulator supervision that is unavailable to purely online end-to-end agents: expert/shortest-path future trajectories supervise the waypoint branch, and

ground-truth waypoint-aligned future semantic layouts supervise the layout branch. We mark this distinction in Table 1. We sample 84k and 465k training pairs from Gibson and MP3D, respectively, with an 85%/15% training/validation split. Semantic maps are cropped and resized to 224×224 before encoding (Appendix B.1). We implement both denoisers with DiT-style Transformer blocks (Peebles & Xie, 2023); the current semantic-map context is encoded by a ResNet-18 (He et al., 2016) whose first layer matches the map channels. Each denoiser contains $N = 8$ Transformer blocks. We train for 200 epochs with AdamW (Kingma & Ba, 2015; Loshchilov & Hutter, 2019), a base learning rate of 10^{-4}, 1,000-step linear warmup, cosine decay, and EMA decay up to 0.9999. The maximum diffusion timestep is $T = 100$, the trajectory length is $K = 32$, and the executed subgoal index is $k_g = 28$. The local controller converts this subgoal into 30-degree turns and 25-cm forward actions; replanning occurs every five environment steps, with a 500-step episode limit. Appendix C specifies the branch operators and Appendix E gives training and deployment pseudocode.

Evaluation Metric. We adopt three standard ObjectNav metrics following (Ramakrishnan et al., 2022; Zhai & Wang, 2023; Chaplot et al., 2020a). SR (Success Rate) denotes the proportion of successful episodes. SPL (Success weighted by Path Length) measures navigation efficiency relative to the shortest path. DTS (Distance To Goal) reports the final distance to the goal when an episode terminates.

4.2 Comparison with State-of-the-Arts

Main Results. We compare **MGNav** with representative end-to-end and modular ObjectNav systems on Gibson and MP3D. Because these systems differ in supervision, perception, and data splits, Table 1 is evidence of competitive performance rather than a universal state-of-the-art claim.

On Gibson, **MGNav** reaches **80.3** SR and **45.5** SPL, improving over the directly compared modular SGM configuration by **2.3** and **1.5** points, respectively, while reducing Distance to Goal (DTS) to **0.98**. NaviFormer reports a higher Gibson SR under its own end-to-end protocol; we include it to avoid conflating protocol-specific improvements with an overall ranking. Our controlled ablations below are the primary evidence for attributing gains to mutual guidance.

On MP3D, **MGNav** obtains 40.2 SR and 15.5 SPL. It improves over the reported SGM values but is essentially tied with PEANUT in SR and slightly below PEANUT in SPL. These comparisons reinforce the need to evaluate methods under matched supervision and splits. Within our fixed protocol, the trajectory-only, concatenation, cross-attention, and gated-cross-attention variants provide a cleaner test of the proposed mechanism.

Discussion. We attribute these improvements to the mutual-guidance design in Section 3.3: instead of conditioning planning on a fixed semantic map, **MGNav** lets route and layout hypotheses constrain each other throughout denoising, which matters most in partially observed scenes where a consistent plan depends on plausible estimates of unobserved structure.

4.3 Diagnostic Analysis

We ablate **i)** the layout branch and the mutual guidance mechanism, **ii)** the cross-branch exchange mechanism, **iii)** the layout branch design, and **iv)** the training objectives, all on Gibson under the main-result setting.

Key Component Analysis. We first examine the essential components of **MGNav** in Table 2. Here, Baseline denotes a trajectory diffusion planner without future semantic layout imagination and without cross-branch mutual guidance. Adding the semantic layout branch in Section 3.5 alone already brings clear improvements on Gibson, indicating that imagined future spatial structure provides useful constraints for long-horizon planning. Introducing the mutual guidance mechanism in Section 3.3 further improves performance, showing that simply predicting future layouts is not sufficient; instead, trajectory planning and future scene imagination should interact dynamically during generation. By combining both components, the full **MGNav** achieves the best performance across all metrics, validating the necessity of jointly modeling and jointly refining the two future variables.

Guidance Mechanism. To assess how the two diffusion branches should interact, we compare different guidance mechanisms in Table 3. Without cross-branch guidance, the two branches are

Table 2: Detailed analysis of essential components of **MGNav** on Gibson.

Algorithm Component	SR↑	SPL↑	DTS↓
BASELINE	76.4	41.9	1.17
+ Semantic Layout Branch only	78.2	43.1	1.09
+ Mutual Guidance only	78.8	43.6	1.05
MGNav	**80.3**	**45.5**	**0.98**

Table 3: Analysis of guidance mechanism for cross-branch interaction on Gibson.

Guidance Mechanism	SR↑	SPL↑	DTS↓
None	77.1	42.6	1.13
Direct Concatenation	78.0	43.4	1.08
Cross-Attention	79.2	44.3	1.02
Gated Cross-Attention (Ours)	**80.3**	**45.5**	**0.98**

Table 4: Analysis of semantic layout branch design on Gibson.

Layout Branch Design	SR↑	SPL↑	DTS↓
No Layout Branch	76.4	41.9	1.17
Pixel-space Layout Prediction	77.5	42.8	1.12
Latent Layout Prediction	78.7	43.9	1.06
+ Reconstruction Loss	79.5	44.7	1.01
+ Mutual Guidance (Ours)	**80.3**	**45.5**	**0.98**

Table 5: Analysis of training objectives for **MGNav** on Gibson.

Training Objective Analysis	SR↑	SPL↑	DTS↓
$\mathcal{L}_{\mathrm{traj}}$ only	76.1	41.7	1.18
$+\mathcal{L}_{\mathrm{layout}}$	77.8	43.0	1.10
Full Objective	**80.3**	**45.5**	**0.98**

effectively decoupled and yield the weakest performance. Direct concatenation improves over this baseline, suggesting that exchanging information between future route hypotheses and future scene constraints is beneficial. Cross-attention further enhances performance by enabling more structured feature interaction. Finally, gated cross-attention achieves the best results, demonstrating that adaptive feature fusion is important for controlling how much information should be transferred at different denoising stages.

Semantic Layout Branch Design. We study different layout branch designs in Table 4. Removing the layout branch leads to a clear drop, confirming the importance of future scene imagination. Pixel-space prediction brings limited gains, while latent layout diffusion is more effective due to its compact representation. Adding reconstruction supervision further improves semantic decodability. The full design with mutual guidance performs best, showing that future layouts are most useful when they interact with trajectory generation.

Training Objective. We analyze the grouped objectives in Table 5. Using only $\mathcal{L}_{\mathrm{traj}}$ gives the weakest performance, and adding $\mathcal{L}_{\mathrm{layout}}$ improves all metrics. The contribution of $\mathcal{L}_{\mathrm{rec}}$ is isolated separately in Table 4, whereas $\mathcal{L}_{\mathrm{geo}}$ and $\mathcal{L}_{\mathrm{smooth}}$ are added jointly in the full objective; Table 5 therefore supports these two regularizers as a pair rather than individually.

5 LIMITATIONS AND SCOPE

The evidence in this paper is limited to simulated indoor ObjectNav with an explicit semantic map, simulator-provided expert trajectories, and ground-truth future layouts during training. The method therefore requires stronger supervision than purely online end-to-end reinforcement-learning agents, and the aggregate table should not be interpreted as a fully controlled ranking across supervision regimes. At deployment, errors in semantic segmentation, depth, pose, or the low-level controller can corrupt both branches. Because mutual guidance is bidirectional, a confident but incorrect layout may also reinforce an infeasible route; receding-horizon replanning reduces but does not eliminate this risk. Finally, two coupled diffusion processes add iterative inference cost relative to one-shot waypoint predictors, and our experiments do not establish performance in dynamic, outdoor, or real-robot environments, where a static semantic map may not be constructible at all.

6 CONCLUSION

We presented **MGNav**, which reformulates modular ObjectNav as joint future imagination: a trajectory branch and a semantic layout branch are denoised together, and gated bidirectional exchange at every denoising step lets route hypotheses and imagined local structure refine each other rather than remain isolated in a serial perceive-then-plan pipeline. On Gibson and MP3D this coupling yields consistent gains over a trajectory-only diffusion planner and over concatenation and ungated alternatives, at the cost of additional supervision and iterative inference.

AI USE STATEMENT

In this work, we used generative AI tools to edit the manuscript for readability and to check formatting, citation consistency, and correspondence between the text and the reported experiments. We did not use generative AI tools to generate synthetic data, develop the theoretical or conceptual framework, formulate or prove mathematical claims, propose or refine hypotheses, design or provide feedback on the experimental methodology, or implement the method; the remaining tasks with required disclosure are not applicable to this work. All AI-assisted edits were reviewed by the authors, who take responsibility for the final content of this work, including its text, claims, and artifacts.

REPRODUCIBILITY STATEMENT

`MGNav` is implemented in PyTorch and trained on four NVIDIA A40 GPUs (48 GB each). Section 4.1 lists the datasets, scene and episode splits, training-pair counts, optimizer, schedule, diffusion horizon, trajectory length, subgoal index, and local-controller parameters. Appendix B describes how future waypoint sequences and waypoint-aligned layouts are constructed, Appendix C specifies the branch operators, and Appendix E gives training and deployment pseudocode. Code and configuration files will be released upon publication.

REFERENCES

Timothy F. Brady, Talia Konkle, George A. Alvarez, and Aude Oliva. Visual long-term memory has a massive storage capacity for object details. *Proceedings of the National Academy of Sciences*, 105(38):14325–14329, 2008.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023.

Angel X. Chang, Angela Dai, Thomas A. Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *3DV*, 2017.

Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *NeurIPS*, 2020a.

Devendra Singh Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. Neural topological slam for visual navigation. In *CVPR*, 2020b.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning*, 2014.

Guilherme N. DeSouza and Avinash C. Kak. Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267, 2002.

Maximilian Du and Shuran Song. DynaGuide: Steering diffusion policies with active dynamic guidance. In *Advances in Neural Information Processing Systems*, 2025.

Howard Eichenbaum and Neal J. Cohen. *From conditioning to conscious recollection: Memory systems of the brain*. Oxford University Press, 1997.

Georgios Georgakis, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, and Kostas Daniilidis. Learning to map for active semantic goal navigation. In *ICLR*, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *ICML*, 2022.

Yiming Ji, Kaijie Yun, Yang Liu, Zhengpu Wang, Boyu Ma, Zongwu Xie, and Hong Liu. Diffusion as reasoning: Enhancing object navigation via diffusion model conditioned on LLM-based object-room knowledge. *arXiv preprint arXiv:2410.21842*, 2024.

Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: Clip embeddings for embodied ai. In *CVPR*, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Klemen Kotar, Aaron Walsman, and Roozbeh Mottaghi. Entl: Embodied navigation trajectory learner. In *ICCV*, 2023.

Xiang Li, Varun Belagali, Jinghuan Shang, and Michael S. Ryoo. Crossway diffusion: Improving diffusion-based visuomotor policy via self-supervised learning. *arXiv preprint arXiv:2307.01849*, 2023.

Yiqing Liang, Boyuan Chen, and Shuran Song. Sscnav: Confidence-aware semantic scene completion for visual semantic navigation. In *ICRA*, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Andreas Lugmayr, Martin Danelljan, Andrés Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022.

Haokuan Luo, Albert Yue, Zhang-Wei Hong, and Pulkit Agrawal. Stubborn: A strong baseline for indoor object navigation. In *IROS*, 2022.

Oleksandr Maksymets, Vincent Cartillier, Aaron Gokaslan, Erik Wijmans, Wojciech Galuba, Stefan Lee, and Dhruv Batra. Thda: treasure hunt data augmentation for semantic navigation. In *ICCV*, 2021.

Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J. Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, et al. Learning to navigate in complex environments. In *ICLR*, 2017.

Arsalan Mousavian, Alexander Toshev, Marek Fisher, Jana Košecká, Ayzaan Wahid, and James Davidson. Visual representations for semantic target driven navigation. In *ICRA*, 2019.

Marek Niedoba, Jonathan W. Lavington, Yunpeng Liu, Vasileios Lioutas, Jacob Sefas, Xiaoxuan Liang, and Frank Wood. A diffusion-model of joint interactive navigation. In *Advances in Neural Information Processing Systems*, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Santhosh K. Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. Poni: Potential functions for objectgoal navigation with interaction-free learning. In *CVPR*, 2022.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. In *arXiv preprint arXiv:2204.06125*, 2022.

Ram Ramrakhya, Eric Undersander, Dhruv Batra, and Abhishek Das. Habitat-web: Learning embodied object-search strategies from human demonstrations at scale. In *CVPR*, 2022.

Ram Ramrakhya, Dhruv Batra, Erik Wijmans, and Abhishek Das. Pirlnav: Pretraining with imitation and rl finetuning for objectnav. In *CVPR*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022.

Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. *arXiv preprint arXiv:2311.10089*, 2023.

Steven M. Smith, Arthur M. Glenberg, and Robert A. Bjork. Context-dependent memory in two natural environments: on land and underwater. *British Journal of Psychology*, 66(3):325–331, 1975.

Sebastian Thrun. Probabilistic algorithms in robotics. In *AI Magazine*, volume 21, pp. 93–109, 2000.

Edward C. Tolman. Cognitive maps in rats and men. *Psychological Review*, 55(4):189–208, 1948.

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 *ICLR*, 2020.

Chanyue Wu, Dong Wang, Yunpeng Bai, Hanyu Mao, Ying Li, and Qiang Shen. Hsr-diff: Hyperspectral image super-resolution via conditional diffusion models. In *ICCV*, 2023.

Fei Xia, Amir Roshan Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *CVPR*, 2018.

Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *CVPR*, 2023.

Wei Xie, Haobo Jiang, Yun Zhu, Jianjun Qian, and Jin Xie. NaviFormer: A spatio-temporal context-aware transformer for object navigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 14708–14716, 2025.

Karmesh Yadav, Ram Ramrakhya, Xavier Puig, Eric Undersander, Mrinal Idrees, Joel Ye, Dhruv Batra, and Abhishek Das. Offline visual representation learning for embodied navigation. *arXiv preprint arXiv:2204.13226*, 2022.

Joel Ye, Dhruv Batra, Abhishek Das, and Erik Wijmans. Auxiliary tasks and exploration enable objectgoal navigation. In *ICCV*, 2021.

Bangguo Yu, Hamidreza Kasaei, and Ming Cao. L3mvn: Leveraging large language models for visual target navigation. In *IROS*, 2023.

Xinyao Yu, Sixian Zhang, Xinhang Song, Xiaorong Qin, and Shuqiang Jiang. Trajectory diffusion for objectgoal navigation. In *NeurIPS*, 2024.

Albert J. Zhai and Shenlong Wang. Peanut: Predicting and navigating to unseen targets. In *ICCV*, 2023.

Jiazhao Zhang, Liu Dai, Fanpeng Meng, Qingnan Fan, Xuelin Chen, Kai Xu, and He Wang. 3d-aware object goal navigation via simultaneous exploration and identification. In *CVPR*, 2023.

Sixian Zhang, Xinyao Yu, Xinhang Song, Xiaohan Wang, and Shuqiang Jiang. Imagine before go: Self-supervised generative map for object goal navigation. In *CVPR*, 2024.

Mengmeng Zhu, Rui Li, Tao Sun, Yansong Li, and Junchi Yan. Navigating to objects in unseen environments by distance prediction. In *IROS*, 2022.

Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J. Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *ICRA*, 2017.