MAP2MOTION: BRAIN-INSPIRED PROSPECTIVE PLANNING FROM STRUCTURAL COGNITIVE MAPS TO EXECUTABLE MOTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Mammalian navigation suggests a computational regime in which compact structural knowledge is continuously grounded by prospective computation to support flexible long-horizon behavior. In contrast, conventional robotic navigation typically relies on dense metric representations, accurate localization, and globally specified trajectories, whose storage, computation, and metric-consistency requirements grow with environment scale. Inspired by structural cognitive maps and prospective neural sequences, we propose Map2Motion, a cognitive-map-guided prospective motion planning converting structural cognitive maps to executable motion. Map2Motion conditions a flow-based distribution of prospective motions on structural guidance and current perception, and learns a forward dynamics model to predict the generated motions' embodied future states and traversability for closed-loop selection and execution. Our experimental result on MP3D benchmark achieves an 83% success rate (SR) with only 4.7% of the global map storage of dense-map baselines, while retaining 64% SR under 0.2 m localization error. In $3,029.4\mathrm{m}^2$ real-world building navigation tasks, it achieves 78% SR with 19.32 ms average planning latency and only 2.6% of dense-map storage, indicating its efficiency and capability in practical scenarios. See https://anonymous.4open.science/r/Map-Motion-0D2E for details.

1 INTRODUCTION

Mammals navigate flexibly through large and changing environments by combining compact internal spatial knowledge with ongoing sensory evidence. Studies on cognitive-map theories Tolman (1948); O'keefe & Nadel (1978); Whittington et al. (2022) suggest that hippocampal–entorhinal system supports structured graph-like representations of environments that capture spatial structural relations Garvert et al. (2017); Bellmund et al. (2018); Whittington et al. (2022); Garvert et al. (2023), which imply that flexible navigation can exploit compact structural knowledge without explicitly reconstructing a globally dense geometric description for every movement.

In contrast, robot navigation commonly places much of the planning burden on dense metric geometry. The classical robot navigation systems commonly combine dense metric reconstruction and accurate localization with metric global path planning, local planning and control Cadena et al. (2016); Yang et al. (2023); Hughes et al. (2024). This paradigm is effective when accurate geometry is available, but maintaining dense maps, metric consistency, and repeatedly querying geometric representations becomes increasingly costly as environments scale Cadena et al. (2016); Hughes et al. (2024); Yang et al. (2023). Compact structural representations to some extent relieve this burden, but abstraction necessarily leaves local motion under-specified.

Recent findings in neuroscience suggests a complementary computational mechanism for transforming compact spatial knowledge into executable sensorimotor behavior Barnaveli et al. (2025). During navigation, hippocampal activity can prospectively represent possible future paths before movement, with these sequences modulated by the current goal Wikenheiser & Redish (2015); Yu et al. (2026); Tang et al. (2026). Studies of deliberation further support a search-and-evaluate view in which internally represented futures are assessed before behavioral commitment Redish (2016); Jensen et al. (2024). We draw from these findings two computational principles that structural rep-

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.

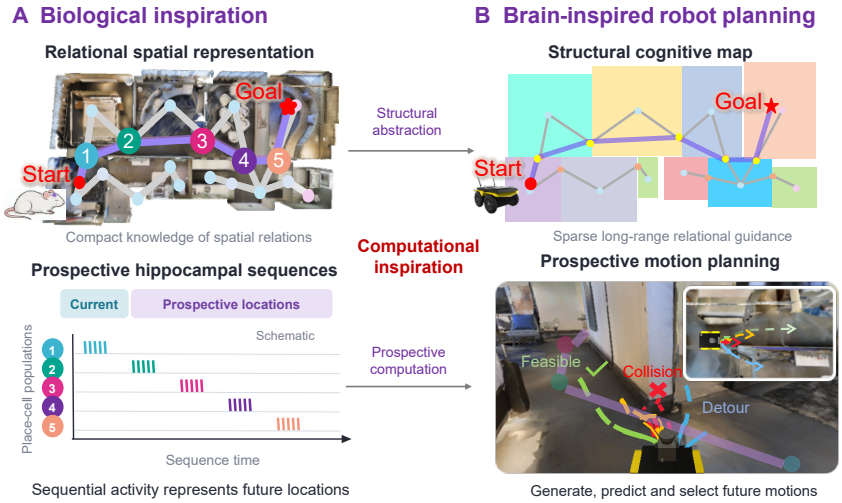

Figure 1: **Biological inspiration for structure-guided prospective motion planning.** (A) Conceptual illustration of relational spatial representations and prospective hippocampal sequences encoding possible future locations. (B) Map2Motion uses a sparse structural cognitive map for long-range guidance and prospectively generates and evaluates multiple future motions based on local perception, selecting an executable action before movement.

resentations organize large scale spatial cognition, and prospective computation considers possible futures before action.

Motivated by this perspective, we propose Map2Motion, a cognitive-map-guided navigation framework for prospective planning from structural cognitive maps to executable motion. A sparse layout-aware structural map provides long-range relational guidance rather than a metric trajectory to track. Rather than converting each structural node into an exact subgoal, our Map2Motion learns a structure-conditioned flow architecture to merge relational route information and perception throughout prospective motion generation. Every motion proposal is then grounded through a context-conditioned forward dynamics model that predicts its embodied future outcome for evaluation and selection. This yields a closed-loop prospective computation that converts under-specified structural cognitive map guidance into executable sensorimotor behavior.

The main contributions of this work are as follows:

1. **Structure-to-Motion Formulation.** We introduce the principles of structural cognitive map guidance and prospective computation from mammalian navigation into robot navigation, enabling efficient navigation with reduced map-storage and computational requirements.
2. **Generative–Predictive Planning.** We introduce Map2Motion, a prospective motion planning that converts structural cognitive-map guidance and current perception into executable motion through future generation and embodied forward dynamics prediction.
3. **Efficient and Robust Navigation.** Experiments show that the proposed Map2Motion achieves 83.47% navigation SR, remains above 64% at 0.2 m localization error in simulation, and uses a 42.7 KB planning map of 3,029.4 m^2 real-world multi-room environment reaching 78.3% SR in real-world navigation.

2 Related Work

2.1 Cognitive Maps and Prospective Navigation

Classical cognitive-map theories associate the hippocampal–entorhinal system with internal representations that support flexible spatial navigation Tolman (1948); O'keefe & Nadel (1978); Whittington et al. (2022). Evidence for graph-like organization further suggests that spatial knowledge

can be represented structurally rather than as point-wise geometry Garvert et al. (2017); Bellmund et al. (2018); Garvert et al. (2023). Complementary studies show prospective hippocampal sequences representing possible and goal-directed future paths before action Johnson & Redish (2007); Yu et al. (2026); Tang et al. (2026); Pfeiffer & Foster (2013); Wikenheiser & Redish (2015); Yu et al. (2026); Tang et al. (2026), while deliberation-related activity supports evaluation of internally represented futures Redish (2016); Jensen et al. (2024). Together, these findings motivate two computational principles underlying our framework: structural representation for long-range relational knowledge and prospective computation for generating and evaluating possible futures before execution.

2.2 Structural and Topological Robot Navigation

Topological map provides compact global representations for long-range navigation Blochliger et al. (2018); Kwon et al. (2021); Hughes et al. (2024). A large class of visual navigation methods constructs memories from previously observed states, demonstration trajectories, or learned reachability Kwon et al. (2021); Shah et al. (2021; 2023a;b; 2021; 2023b); Suomela et al. (2024); Garg et al. (2024); Sridhar et al. (2024); Gode et al. (2025). Their connectivity, however, largely follows observed or learned transitions, enabling effective reuse of traversable experience but making the resulting topology dependent on data-collection and exploration coverage. A complementary line of work derives topology from environmental geometry or structure Hughes et al. (2024); Ravichandran et al. (2022); Gu et al. (2024). TopoMap abstracts free-space geometry, while Topo-Field, HOV-SG, and LOG-Nav represent persistent room, object, and hierarchical structure Blochliger et al. (2018); Werby et al. (2024); Hou et al. (2025), and LOG-Nav Hou et al. (2026) further demonstrates that layout-aware topological maps can support hierarchical long-range navigation. These representations preserve more stable relational structure than point-wise free-space maps, but increasingly leave local executable motion under-specified.

Existing hierarchical systems commonly resolve this gap by converting topological guidance into local subgoals, metric trajectories, or learned local-planning objectives. TANGO combines global topology with local traversability and geometric planning, while LOG-Nav couples layout-level topology with a learned local planner Podgorski et al. (2025); Hou et al. (2026). In these approaches, abstract structural nodes are instantiated as local targets whose mismatch with current geometry must be handled by the downstream planner. Map2Motion instead preserves the structural route as under-specified relational guidance and uses it to condition a distribution over possible local futures, introducing a different structure-to-motion interface.

2.3 Learning-based Generative and Predictive Motion Planning

Generative navigation policies show that local motion can be represented as a multimodal distribution rather than a single deterministic trajectory. NoMaD uses diffusion to model diverse goal-conditioned behaviors, while FlowNav and related work use conditional Flow Matching for more efficient motion generation Sridhar et al. (2024); Gode et al. (2025); Nguyen et al. (2025). These methods establish generative future modeling as an effective navigation primitive, but their motion distributions are primarily conditioned on local observations, point/image goals, or traversal-derived memories rather than an abstract relational route, which limited their performance on sparser map.

Complementary to action generation, learned forward models enable predictive evaluation of candidate controls. Sampling-based controllers such as MPPI Williams et al. (2017) optimize candidate control sequences through dynamics rollouts and evaluation. Recent perceptive forward-dynamics models extend this idea by predicting platform-aware future states and failure likelihoods Roth et al. (2025). Map2Motion brings these capabilities together around a different interface: the structural cognitive map conditions a distribution of prospective controls, and forward prediction grounds each proposal in its embodied outcome, leveraging generation and prediction as the mechanism that transforms sparse relational knowledge into executable behavior.

3 Problem Statement

We study how abstract structural guidance can be transformed into executable sensorimotor behavior. Motivated by structural cognitive maps and prospective neural computation, we consider

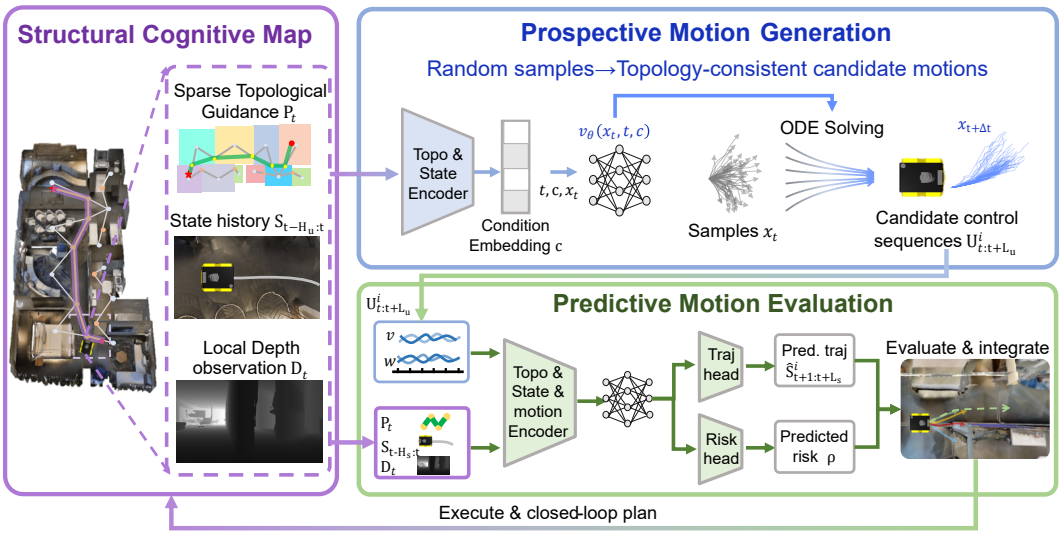

Figure 2: **Brain-inspired prospective motion planning.** A sparse structural cognitive map provides long-range relational guidance P_t. Conditioned on this guidance, current depth, and recent motion context, a structure-conditioned flow model generates multiple prospective control sequences $U_{t:t+L_u}$. A forward dynamics model then predicts the embodied future state $\hat{S}_{t:t+L_s}$ and risk ρ of each candidate for feasibility-aware selection. The selected motion is executed in closed loop, grounding abstract structural guidance into executable behavior.

a navigation setting in which compact structural knowledge provides long-range guidance, while executable motion is resolved online from current perception. The robot is given a sparse layout-aware structural map $M = (V, E)$, as its only global environment representation. V contains room and doorway nodes, each storing only its center 3D coordinate and spatial extent, while the room-door edges E encode traversable adjacency weighted by Euclidean distance. M contains no dense occupancy, obstacle geometry, visual memory, or executable reference trajectory.

Given the current robot pose in the room v_t and target pose in the room v_g, graph search on M yields a structural route $P_t = (v_t, \ldots, v_g)$. P_t specifies the sequence of structural transitions for long-range progress, but leaves the local motion between rooms and doors under-specified. At planning step t, the robot observes a local depth image D_t and a state estimate s_t containing its pose and motion state, and captures short-term motion context with recent state $s_{t-H_s:t-1}$ and control history $U_{t-H_u:t-1}$. The prospective planning problem is to infer, under structural guidance P_t and current sensory evidence, an executable future control sequence together with its anticipated motion outcome. We formulate this process as a conditional policy π_θ that convert structural guidance to control sequence distribution and predicted states conditioned with perception and motion context:

$$\pi_\theta(U_{t:t+L_u}, \hat{S}_{t:t+L_s} | D_t, P_t, s_t, s_{t-H_s:t-1}, U_{t-H_u:t-1}) \tag{1}$$

$U_{t:t+L_u}$denotes the selected future control sequence from time t to $t + L_u$, while $\hat{S}_{t:t+L_s}$ denotes its predicted state trajectory.The desired solution should advance along P_t, avoid locally observed obstacles, respect robot motion feasibility, and support efficient closed-loop replanning.

4 Method

4.1 Structural Cognitive Map Guidance

We use a sparse layout-aware topological map as our computational abstraction of a structural cognitive map. At planning step t, the robot pose determines its current room node r_t, while the task goal determines the target room r_g. We run A* over the room–doorway connectivity graph with Euclidean edge costs to obtain a structural route P_t. The coordinates of nodes along P_t are transformed into the robot-centric frame using the current pose, yielding structural guidance that specifies the upcoming rooms and doors and their relative locations.

4.2 PROSPECTIVE MOTION GENERATION

Since this sparse guidance contains no local obstacle geometry or directly trackable trajectory, executable motion must be further resolved online from current perception. Given the robot-centric structural guidance $\tilde{P}_t$, we define the planning context as

$$o_t = (D_t, P_t, s_t, S_{t-H_s:t-1}, U_{t-H_u:t-1}), \tag{2}$$

where depth image D_t provides local obstacle observations, P_t biases motion toward globally relevant passages, and the current state s_t, state history $S_{t-H_s:t-1}$, and control history $U_{t-H_u:t-1}$ provide short-term motion context. An encoder E_ψ, composed of a CNN for depth and GRUs for sequential state and control inputs, fuses these features into a condition embedding

$$c_t = E_\psi(D_t, P_t, s_t, S_{t-H_s:t-1}, U_{t-H_u:t-1}). \tag{3}$$

Considering that sparse structural guidance and local perception generally admit multiple feasible motions, we model a conditional distribution over future controls rather than a deterministic mapping. Specifically, we use a conditional flow-matching model learns a velocity field $v_\theta(x_\tau, \tau, c_t)$ implemented as an MLP over the concatenated intermediate sample x_τ, flow time τ, and condition c_t, which transports samples from a Gaussian prior x_0 to the context-conditioned motion distribution x_1. For each training sample with future control sequence

$$x_1 = U_t = \{u_t, \dots, u_{t+H_u-1}\}, \tag{4}$$

we sample $x_0 \sim \mathcal{N}(0, I)$ and $\tau \sim \mathcal{U}(0, 1)$, construct $x_\tau = (1-\tau)x_0 + \tau x_1$, and optimize

$$L_{\mathrm{FM}} = \mathbb{E}\Big[\|v_\theta(x_\tau, \tau, c_t) - (x_1 - x_0)\|_2^2\Big]. \tag{5}$$

At inference, the current context is encoded once, after which N independent samples $x_0^i \sim \mathcal{N}(0, I)$ are evolved under

$$\frac{dx_\tau^i}{d\tau} = v_\theta(x_\tau^i, \tau, c_t), \qquad i = 1, \dots, N, \tag{6}$$

yielding candidate control sequences $U_t^i = x_1^i$. Sampling from the conditioned motion distribution preserves multiple plausible responses to the same sparse guidance, providing robustness to structural sparsity and node-level localization errors while supplying diverse prospective motions for subsequent feasibility evaluation.

4.3 PREDICTIVE MOTION EVALUATION

The generated motion candidates are plausibly feasible but further evaluation is required. Motivated by prospective evaluation in biological navigation, where possible future outcomes are assessed before action, we learn a forward dynamics model (FDM) to predict the embodied consequences of each candidate motion. For a candidate control sequence U_t^i, the FDM predicts its future state trajectory $\hat{S}_t^i = \{\hat{s}_{t+1}^i, ..., \hat{s}_{t+H_s}^i\}$ and traversability $\hat{\rho}_t^i = \{\hat{\rho}_{t+1}^i, ..., \hat{\rho}_{t+H_s}^i\}$:

$$F_\phi(D_t, P_t, s_t, S_{t-H_s:t-1}, U_t^i) = (\hat{S}_t^i, \hat{\rho}_t^i). \tag{7}$$

A perception–motion encoder fuses depth D_t, structural guidance P_t, motion history $S_{t-H_s:t-1}$, and candidate controls U_t^i into a joint representation, which is processed by a GRU and two MLP heads for future-state and traversability prediction, respectively; architectural details are provided in the Appendix.

The FDM is trained on offline rollouts. Each sample contains the planning context o_t, the executed control sequence U_t, the resulting ground-truth future states S_t^{gt}, and traversability targets ρ_t^{gt} derived from the clearance along the recorded future trajectory. We jointly supervise future trajectory position with L_p, future trajectory orientation with L_q, and traversability prediction with L_ρ using the loss function

$$L_{\mathrm{FDM}} = \lambda_p L_p + \lambda_q L_q + \lambda_\rho L_\rho, \tag{8}$$

with the detailed loss definitions and traversability-label construction provided in the Appendix.

At inference, the same context of current planning step is paired with each candidate U_t^i, allowing the model to prospectively compare their predicted outcomes $(\hat{S}_t^i, \hat{\rho}_t^i)$ before execution. Each candidate U_t^i is scored by its predicted progress along the structural guidance, safety, and control smoothness,

$$J^i = w_{\mathrm{prog}} J_{\mathrm{prog}}^i(\hat{S}_t^i, P_t) + w_{\mathrm{safe}} J_{\mathrm{safe}}^i(\hat{\rho}_t) + w_{\mathrm{smooth}} J_{\mathrm{smooth}}^i(U_t^i, U_{t-H_u:t-1}), \tag{9}$$

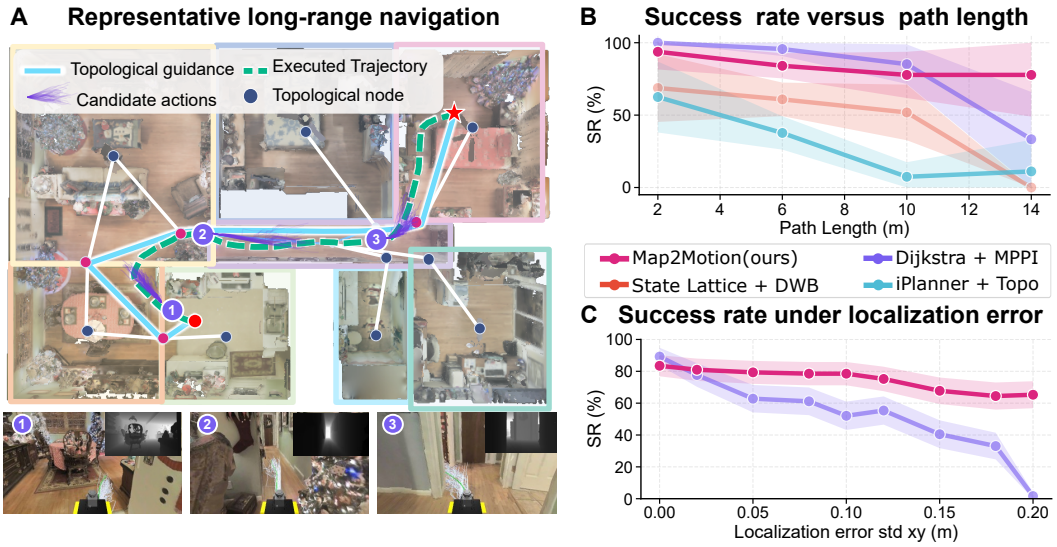

Figure 3: **Navigation example and navigation performance comparison.** **A.** A representative Matterport3D navigation episode, where sparse structural guidance specifies the long-range route and prospective motion planning generates executable local behavior from onboard perception. **B.** Navigation SR versus reference shortest-path length for Map2Motion and representative baselines; shaded regions denote 95% confidence intervals. **C.** Navigation success rate under increasing translational localization noise, compared with the dense-map Wavefront Dijkstra + MPPI baseline.

and the lowest-cost candidate is selected:

$$i^* = \arg\min_i J^i, \qquad U_t^* = U_t^{i^*}. \tag{10}$$

The selected controls are executed until the next planning step, when updated observations trigger a new proposal–prediction–selection cycle. This closed-loop process grounds abstract structural guidance into executable sensorimotor behavior.

5 Experiments

Baseline Protocol. To evaluate whether compact structural maps can support competitive long-horizon navigation when grounded into motion through prospective planning, we compare Map2Motion with representative metric-map-based and learning-based planners. The metric-map-based baselines perform global graph search on dense occupancy grids using Wavefront Dijkstra, A* Hart et al. (1968), or State Lattice Pivtoraiko et al. (2009), followed by MPPI Williams et al. (2017) or DWB Fox et al. (1997) for local control. From a graph-planning perspective, such grids can be viewed as densely discretized spatial graphs whose nodes encode locations and occupancy states, whereas Map2Motion plans over a much sparser layout-aware graph whose nodes represent persistent rooms and doorways by their locations and spatial extents. To isolate the structure-to-motion mechanism, we further include iPlanner+Topo, which uses the same topological node coordinates as sequential subgoals for a learned local planner Yang et al. (2023), following the hierarchical planning paradigm used in layout-aware navigation Hou et al. (2026). We restrict comparisons to coordinate-based global representations while varying their abstraction level and the mechanism used to convert global guidance into local motion, and therefore exclude visual-memory topological methods such as NoMaD Sridhar et al. (2024) and FlowNav Gode et al. (2025), whose nodes store traversal-derived images and rely on image-goal interfaces, introducing a different form of environment prior.

Setups. We evaluate Map2Motion in both simulation and the real world against classic grid-based and topology-guided baselines, in which we use the sparse layout-aware topological map constructed following Topo-Field Hou et al. (2025) as our computational abstraction of a structural cognitive

Table 1: Navigation performance and resource cost of representative methods.

Method	Map Prior	SR(%)↑	SPL(%)↑	Map (KB)↓	Plan (ms)↓	CPU (%)↓	GPU (%)↓
Dijkstra+MPPI	Occupancy	89.25	83.71	77.97	58.57	91.80	–
A*+MPPI		64.46	59.35	77.97	50.13	97.30	–
State Lattice+DWB		55.37	49.98	77.97	58.71	89.20	–
iPlanner	None	15.70	15.50	–	3.22	27.35	7.45
iPlanner+Topo	Structural topo.	32.23	31.73	3.66	4.38	29.44	6.90
Map2Motion		**83.47**	**79.34**	**3.66**	**6.70**	**35.18**	**4.96**

map. Simulations are conducted in Isaac Sim using Matterport3D Chang et al. (2017) scenes. Real-world experiments use a Clearpath Jackal equipped with an NVIDIA Jetson Orin for onboard planning. We collect about 50k training samples from demonstration rollouts. Each sample includes a 320×180 depth image, the topological route to the goal, a 16-step action command sequence at 0.05s intervals, an eight-step state history at 0.05s intervals for prospective motion generation training, and a 16-step state history at 0.5s intervals for FDM training. At inference, the Map2Motion generates 16 control commands at 0.05s intervals, while the FDM predicts eight future states at 0.5s intervals. This multi-resolution design supports high-frequency control and longer-horizon evaluation of dynamics, perception, and topology.

5.1 Navigation Performance

We report SR and SPL Anderson et al. (2018) as the primary navigation metrics, with collision and timeout rates used for failure and ablation analysis. As shown in Table 1, Wavefront Dijkstra + MPPI achieves the highest overall performance with 89.25% SR and 83.71% SPL benefiting from global dense occupancy map and LiDAR-based local perception. Map2Motion reaches 83.47% SR and 79.34% SPL using only a sparse structural topology. This requires only 4.7% of the global-map storage and reduces planning latency from 58.57 to 6.70 ms, while lowering CPU utilization from 91.80% to 35.18%. These results show that compact structural guidance can retain strong navigation performance while substantially reducing the global representation and planning burden.

The iPlanner+Topo baseline further isolates whether the same compact structural guidance can be exploited by directly treating topological nodes as sequential point goals. Adding topology improves iPlanner from 15.70% to 32.23% SR, but remains substantially below Map2Motion's 83.47%. This gap suggests that structural guidance alone is insufficient when instantiated only as exact local subgoals; reliable execution benefits from resolving the under-specified local motion online from both structure and current perception.

Figure 3B further examines performance as the reference path length increases. While the success rates of the baselines decrease substantially on longer routes, Map2Motion degrades more gradually and maintains a high success rate across the evaluated range. This trend indicates that long-range relational guidance coupled with online prospective motion resolution can sustain navigation over extended horizons without requiring a globally executable metric trajectory.

5.2 Computational Performance

We analyze how the number of candidate motions N and Flow Matching integration steps K affect the computational cost of prospective planning. Across all evaluated configurations, the mean end-to-end planning latency remains between 5.81 and 9.41 ms, supporting high-frequency closed-loop replanning. The dominant scaling term is prospective motion generation, which requires $2.21-5.80$ ms, whereas FDM rollout, candidate scoring, structural path extraction, and state-history processing remain nearly constant. Batch-parallel generation further keeps latency nearly unchanged for $N \leq 64$, with a noticeable increase only at $N = 128$.

As shown in Fig. 4A, navigation performance does not improve monotonically with additional computation. For small candidate sets ($N = 4$ or 8), increasing K from 1 to 4 raises SR from $76.86-78.51\%$ to 83.47%, while $K = 8$ provides no further gain. Increasing N beyond this range is similarly ineffective: at $K = 4$, SR decreases from 83.47% at $N = 4$ or 8 to 71.90% at $N = 128$.

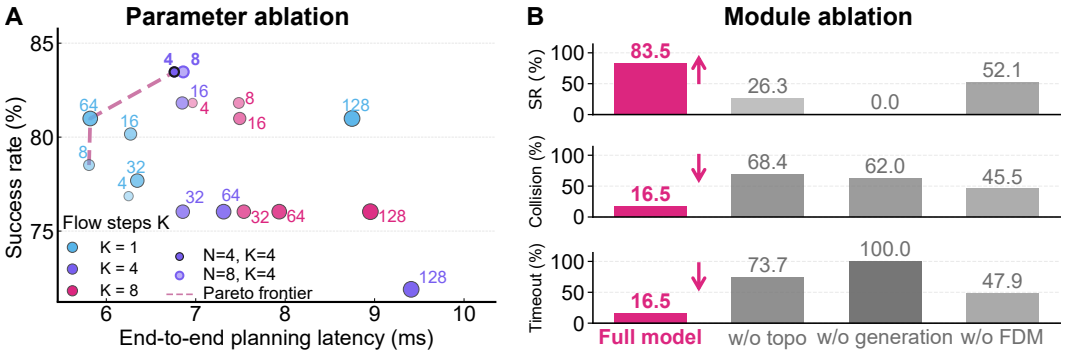

Figure 4: **Ablation analysis of Map2Motion. A.** Success rate versus end-to-end planning latency for different candidate sample sizes N and Flow Matching integration steps K; the dashed curve denotes the Pareto frontier. **B.** Module ablation of structural guidance, prospective motion generation, and forward feasibility prediction, evaluated by success, collision, and timeout rates.

The default $N = 4, K = 4$ setting therefore lies on the performance–latency Pareto frontier, achieving the highest SR with only 6.76 ms mean latency and fewer candidates than the equally successful $N = 8, K = 4$ setting. These results indicate that Map2Motion relies on a small set of informative prospective motions rather than exhaustive sampling, enabling sample-efficient closed-loop planning.

5.3 Robustness to Localization Error

To evaluate the robustness of the proposed motion planner to localization uncertainty, we inject Gaussian noise into the x-y coordinates of the localization estimate received by the planner. The noise is resampled at each planning step and affects the relative topological guidance used by the planner. The robot itself is still simulated in Isaac Sim using its true physical state. As shown in Fig. 3C, the success rate of the baseline decreases rapidly as the localization noise increases. In contrast, Map2Motion degrades more gradually and maintains a success rate above 64% even when the localization noise reaches $0.2\,\mathrm{m}$. This robustness can be attributed to the weaker dependence of Map2Motion on precise metric alignment. In dense-map-based pipelines, localization errors affect the consistency between the global occupancy map, local point cloud, planned path, and controller state. Since planning and control are handled by separate modules, these errors may accumulate and be amplified during execution. By contrast, Map2Motion uses the topological map only as coarse region-level guidance, while local depth observations provide immediate obstacle information. Moreover, its multi-candidate action generation and FDM-based rollout evaluation allow the planner to assess several feasible motion hypotheses before execution, further improving robustness under localization uncertainty.

5.4 Real-World Experiments

We evaluate Map2Motion deployed on a Clearpath Jackal equipped with an NVIDIA Jetson Orin to navigating in a $3029.4\,\mathrm{m}^2$ multi-room campus building containing approximately $121.2\,\mathrm{m}$ of corridors, a $120\,\mathrm{m}^2$ laboratory, a $440\,\mathrm{m}^2$ café, and a $548\,\mathrm{m}^2$ lounge area. Point-goal navigation episodes are generated from randomly sampled start–goal pairs and cover different lighting conditions, including daytime and nighttime, as well as naturally occurring pedestrian traffic.

Map2Motion uses only a Topo-Field-derived Hou et al. (2025) sparse layout-aware topological map for global guidance, while local motions are generated and evaluated online from depth observations and localization estimates. We compare against a classical Dijkstra+DWB Fox et al. (1997) pipeline operating on a pre-built dense occupancy map under the same long-range point-goal tasks. Map2Motion achieves a 78.3% success rate and 64.0% SPL, while requiring only 42.74 KB of planning-map storage, $38.36\times$ less than the dense occupancy map. Its average planning latency is 19.32 ms, compared with 221.58 ms for dense global path search in the large-scale environment. These results show that sparse structural guidance with closed-loop prospective computation trans-

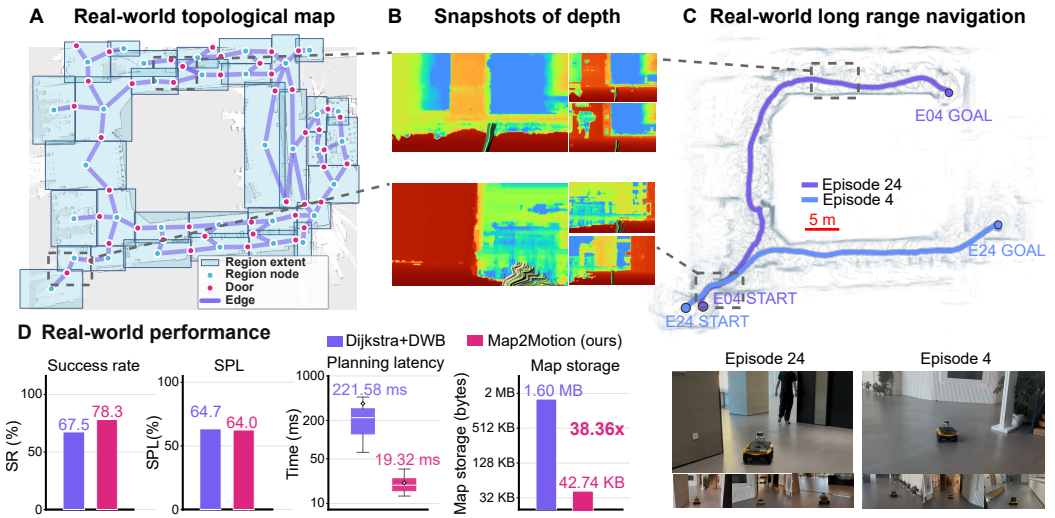

Figure 5: **Real world experiment of Map2Motion. A.** Sparse layout-aware topological map of the $3{,}029.4\,\mathrm{m}^2$ test environment, containing room nodes, doorway nodes, and their connectivity. **B.** Representative onboard depth observations and predicted future trajectories and traversability. **C.** Representative long-horizon navigation episodes showing the structural routes and executed trajectories from start to goal. **D.** Real-world comparison with the dense-map Dijkstra+DWB baseline in SR, SPL, planning latency, and planning-map storage.

fers effectively to real-world deployment while substantially reducing map storage and planning overhead.

5.5 Ablation Study

We conduct ablation experiments to evaluate the contribution of three key components: the structural topological map guidance, the prospective motion generation, the FDM motion prediction and evaluation. As shown in Figure 4 B, removing the motion generation leads to complete failure, indicating that random sampling cannot provide effective candidates under the tested computational budget, and that the FDM alone can only evaluate given actions rather than generate task-relevant proposals. Removing the FDM reduces SR to 52.07% and substantially increases the collision rate, suggesting that the Map2Motion can generate plausible actions but requires predictive rollout and traversability estimation for reliable action selection. Removing topological guidance further reduces SR to 26.32%, showing that final-goal direction alone is insufficient for long-horizon multi-room navigation, where local observations often do not reveal the correct passage toward the goal. These results demonstrate the complementarity of structural guidance, sample-efficient motion proposal, and predictive evaluation.

6 Conclusions and Future Work

Inspired by computational principles associated with mammalian navigation, we formulate robot navigation around structural cognitive-map guidance and prospective computation, enabling abstract and sparse environmental knowledge to be transformed into executable long-horizon behavior. Map2Motion learns a structure-conditioned flow model that generates prospective motions from relational guidance and current perception, together with a forward dynamics model that predicts their embodied future outcomes for closed-loop evaluation and selection.Experiments in simulation and the real world show that this formulation maintains strong long-horizon navigation performance while reducing planning-map storage and latency and remaining robust to localization perturbations. Extending this perspective from online inference to online adaptation, in future work we will explore prioritized neural replay Mattar & Daw (2018); Jensen et al. (2024) principle to realize more rapidly adapt prospective motion generation.

Reproducibility Statement

We provide the main formulation of Map2Motion, including structural guidance, prospective motion generation, forward prediction, and closed-loop action selection, in Sec. 4. Detailed network architectures, training objectives, temporal horizons, and candidate-evaluation costs are provided in the Appendix. The experimental setup, including simulator configuration, robot platform, sensing modalities, episode generation, success criteria, and baseline settings, is also documented in the Appendix, together with the full set of evaluated Nav2 baseline combinations. We additionally describe the construction of training samples and supervision used for both prospective motion generation and forward dynamics prediction. All implementation details and experimental configurations required to reproduce the reported results are provided in the Appendix and supplementary material.

References

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018.

Irina Barnaveli, Simone Viganò, Daniel Reznik, Patrick Haggard, and Christian F Doeller. Hippocampal-entorhinal cognitive maps and cortical motor system represent action plans and their outcomes. *Nature Communications*, 16(1):4139, 2025.

Jacob LS Bellmund, Peter Gärdenfors, Edvard I Moser, and Christian F Doeller. Navigating cognition: Spatial codes for human thinking. *Science*, 362(6415):eaat6766, 2018.

Fabian Blochliger, Marius Fehr, Marcin Dymczyk, Thomas Schneider, and Rol Siegwart. Topomap: Topological mapping and navigation based on visual slam maps. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3818–3825. IEEE, 2018.

Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

Dieter Fox, Wolfram Burgard, and Sebastian Thrun. The dynamic window approach to collision avoidance. *IEEE robotics & automation magazine*, 4(1):23–33, 1997.

Sourav Garg, Krishan Rana, Mehdi Hosseinzadeh, Lachlan Mares, Niko Sünderhauf, Feras Dayoub, and Ian Reid. Robohop: Segment-based topological map representation for open-world visual navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4090–4097. IEEE, 2024.

Mona M Garvert, Raymond J Dolan, and Timothy EJ Behrens. A map of abstract relational knowledge in the human hippocampal–entorhinal cortex. *elife*, 6:e17086, 2017.

Mona M Garvert, Tankred Saanum, Eric Schulz, Nicolas W Schuck, and Christian F Doeller. Hippocampal spatio-predictive cognitive maps adaptively guide reward generalization. *Nature Neuroscience*, 26(4):615–626, 2023.

Samiran Gode, Abhijeet Nayak, Débora NP Oliveira, Michael Krawez, Cordelia Schmid, and Wolfram Burgard. Flownav: Combining flow matching and depth priors for efficient navigation. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 17762–17768. IEEE, 2025.

Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5021–5028. IEEE, 2024.

Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

Jiawei Hou, Wenhao Guan, Longfei Liang, Jianfeng Feng, Xiangyang Xue, and Taiping Zeng. Topofield: Topometric mapping with brain-inspired hierarchical layout-object-position fields. *IEEE Robotics and Automation Letters*, 2025.

Jiawei Hou, Yuting Xiao, Xiangyang Xue, and Taiping Zeng. Log-nav: Efficient layout-aware object-goal navigation with hierarchical planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18297–18305, 2026.

Nathan Hughes, Yun Chang, Siyi Hu, Rajat Talak, Rumaia Abdulhai, Jared Strader, and Luca Carlone. Foundations of spatial perception for robotics: Hierarchical representations and real-time systems. *The International Journal of Robotics Research*, 43(10):1457–1505, 2024.

Kristopher T Jensen, Guillaume Hennequin, and Marcelo G Mattar. A recurrent network model of planning explains hippocampal replay and human behavior. *Nature neuroscience*, 27(7):1340–1348, 2024.

Adam Johnson and A David Redish. Neural ensembles in ca3 transiently encode paths forward of the animal at a decision point. *The Journal of neuroscience*, 27(45):12176–12189, 2007.

Obin Kwon, Nuri Kim, Yunho Choi, Hwiyeon Yoo, Jeongho Park, and Songhwai Oh. Visual graph memory with unsupervised representation for visual navigation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 15890–15899, 2021.

Marcelo G. Mattar and Nathaniel D. Daw. Prioritized memory access explains planning and hippocampal replay. *Nature Neuroscience*, 21:1609–1617, 2018. doi: 10.1038/s41593-018-0232-z.

Khang Nguyen, An T Le, Tien Pham, Manfred Huber, Jan Peters, and Minh Nhat Vu. Flowmp: Learning motion fields for robot planning with conditional flow matching. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11291–11297. IEEE, 2025.

John O'keefe and Lynn Nadel. *The hippocampus as a cognitive map*. Oxford university press, 1978.

Brad E Pfeiffer and David J Foster. Hippocampal place-cell sequences depict future paths to remembered goals. *Nature*, 497(7447):74–79, 2013.

Mihail Pivtoraiko, Ross A Knepper, and Alonzo Kelly. Differentially constrained mobile robot motion planning in state lattices. *Journal of Field Robotics*, 26(3):308–333, 2009.

Stefan Podgorski, Sourav Garg, Mehdi Hosseinzadeh, Lachlan Mares, Feras Dayoub, and Ian Reid. Tango: Traversability-aware navigation with local metric control for topological goals. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2399–2406. IEEE, 2025.

Zachary Ravichandran, Lisa Peng, Nathan Hughes, J Daniel Griffith, and Luca Carlone. Hierarchical representations and explicit memory: Learning effective navigation policies on 3d scene graphs using graph neural networks. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 9272–9279. IEEE, 2022.

A David Redish. Vicarious trial and error. *Nature Reviews Neuroscience*, 17(3):147–159, 2016.

Pascal Roth, Jonas Frey, Cesar Cadena, and Marco Hutter. Learned perceptive forward dynamics model for safe and platform-aware robotic navigation. *arXiv preprint arXiv:2504.19322*, 2025.

Dhruv Shah, Benjamin Eysenbach, Gregory Kahn, Nicholas Rhinehart, and Sergey Levine. Rapid exploration for open-world navigation with latent goal models. *arXiv preprint arXiv:2104.05859*, 2021.

Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. Gnm: A general navigation model to drive any robot. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7226–7233. IEEE, 2023a.

Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. Vint: A foundation model for visual navigation. *arXiv preprint arXiv:2306.14846*, 2023b.

Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 63–70. IEEE, 2024.

Lauri Suomela, Jussi Kalliola, Harry Edelman, and Joni-Kristian Kämäräinen. Placenav: Topological navigation through place recognition. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5205–5213. IEEE, 2024.

Wenbo Tang, Xiyu Mei, Ryan E Harvey, Estrella Carbajal-Leon, Talia Netzer, Hongyu Chang, Azahara Oliva, and Antonio Fernandez-Ruiz. Goal-directed hippocampal theta sweeps during memory-guided navigation. *Nature Neuroscience*, pp. 1–11, 2026.

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

Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024.

James CR Whittington, David McCaffary, Jacob JW Bakermans, and Timothy EJ Behrens. How to build a cognitive map. *Nature neuroscience*, 25(10):1257–1272, 2022.

Andrew M Wikenheiser and A David Redish. Hippocampal theta sequences reflect current goals. *Nature neuroscience*, 18(2):289–294, 2015.

Grady Williams, Andrew Aldrich, and Evangelos A Theodorou. Model predictive path integral control: From theory to parallel computation. *Journal of Guidance, Control, and Dynamics*, 40 (2):344–357, 2017.

Fan Yang, Chen Wang, Cesar Cadena, and Marco Hutter. iplanner: Imperative path planning. *arXiv preprint arXiv:2302.11434*, 2023.

Changmin Yu, Zilong Ji, Jake Ormond, John O'Keefe, and Neil Burgess. Hippocampal theta sweeps indicate goal direction during navigation. *Nature Neuroscience*, pp. 1–12, 2026.

AI USE STATEMENT

In this work, we used generative AI tools to assist with drafting and editing parts of the manuscript, searching for and identifying relevant literature, and creating or editing software code. All AI-assisted content was reviewed by the authors. In particular, LLM-generated code was manually inspected, verified, and tested for correctness by the authors before being used in the experiments. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

APPENDIX

PROSPECTIVE MOTION GENERATION CONFIGURATION

The prospective motion generator is trained on Isaac Sim rollout samples containing synchronized local observations, sparse structural guidance, motion history, and future control sequences. Each training sample consists of a single-channel depth image resized to 90×160, an 8-step state history containing local pose and velocity features, an 8-step future control sequence, and a structural topological route represented in the robot-centric frame. The state and control horizons are sampled at intervals of five simulation steps, providing compact temporal context for short-horizon motion generation.

The depth, state-history, and structural-route features are concatenated and processed by a two-layer GRU with hidden size 128 to obtain the condition embedding c_t. The intermediate flow sample x_τ

is separately projected to a 32-dimensional action feature and combined with c_t and the flow-time embedding τ as input to the velocity network v_θ. The Flow Matching network does not directly regress a motion proposal. Instead, it learns the context-conditioned velocity field

$$v_\theta(x_\tau, \tau, c_t),$$

where x_τ denotes an intermediate control-sequence sample along the flow trajectory and τ is the flow time. The intermediate action sample is embedded into a 32-dimensional feature and combined with the context and flow-time representation to predict the corresponding velocity in control-sequence space.

During training, the target future control sequence serves as the endpoint of the conditional flow-matching path. At inference, independent samples $x_0^i \sim \mathcal{N}(0, I)$ are initialized in the prior distribution and evolved by solving

$$\frac{dx_\tau^i}{d\tau} = v_\theta(x_\tau^i, \tau, c_t), \qquad \tau \in [0, 1].$$

The terminal states x_1^i obtained after ODE integration constitute the prospective control proposals U_t^i. Thus, the learned model parameterizes a topology- and perception-conditioned flow over future controls, while executable motion candidates emerge through numerical integration of this learned velocity field.

Predictive Motion Evaluation Configuration

The FDM adopts a multi-branch architecture to estimate future robot motion and trajectory feasibility conditioned on local observations, sparse topological guidance, historical states, and candidate control sequences. Specifically, the model consists of four modality-specific encoders, a recurrent fusion module, and two prediction heads. The historical state sequence is encoded by a two-layer GRU with a hidden dimension of 32, while the single-channel depth observation is processed by a ResNet-based visual encoder with $[2, 2, 2, 2]$ residual blocks and then projected to a 32-dimensional feature. The local topological path, represented by two-dimensional coordinates sequence, is encoded by a two-layer GRU with hidden dimension 32, and the future action sequence is embedded by an MLP into a 32-dimensional latent vector. The four encoded features are concatenated into a 128-dimensional representation and further processed by a two-layer recurrent fusion module with hidden dimension 128. Based on the fused latent state, two lightweight prediction heads are used to estimate future local state increments, future motion quantities, and per-step collision or traversability risks, respectively. The state head use hidden dimensions $[64, 32]$, and the collision head outputs a scalar feasibility estimate for each future step. In the reported experiments, the model is trained with an 8-step history and an 8-step prediction horizon, both sampled at intervals of 5 simulation steps, together with one local topological guidance route. Therefore, the FDM performs compact short-horizon rollout prediction over 8 future steps while maintaining a low-dimensional representation suitable for real-time candidate evaluation.

We intentionally use a prediction horizon longer than the candidate-control horizon. The candidate sequence specifies the near-term control intent, while the FDM is supervised with future states extending beyond the commanded interval. Consequently, the model learns to extrapolate longer-term, lower-resolution motion evolution from the current perception, structural guidance, motion history, and short-horizon control proposal. These extrapolated states are not directly executed; instead, they provide additional prospective evidence for evaluating whether a candidate is likely to make safe and structurally consistent progress beyond its immediate control horizon. The FDM is trained on offline rollouts collected from simulation. Each training sample comprises the planner observation $\mathbf{o}_t$, the executed action sequence $\mathbf{U}_{t:t+L_u}$, the corresponding ground-truth future trajectory $\mathbf{S}^{\mathrm{gt}}_{t+1:t+L_s}$, and the TSDF supervision signal $\boldsymbol{\rho}_t^{\mathrm{gt}}$. The trajectory targets are directly extracted from the recorded rollout, whereas the TSDF targets are generated by querying the offline scene-level TSDF map at the corresponding ground-truth trajectory positions, while the TSDF is not needed in inference. The overall training objective is defined as

$$L_{\mathrm{FDM}} = \lambda_p L_p + \lambda_q L_q + \lambda_\rho L_\rho, \tag{11}$$

where λ_p, λ_q, and $\lambda\rho$ are weighting coefficients for the position, orientation, and risk prediction losses, respectively. The position prediction loss is given by

$$L_p = \frac{1}{L_s} \sum_{k=1}^{L_s} \left\| \hat{\mathbf{p}}_{t+k} - \mathbf{p}_{t+k}^{\mathrm{gt}} \right\|_2^2, \tag{12}$$

and the orientation prediction loss is defined as

$$L_q = \frac{1}{L_s} \sum_{k=1}^{L_s} d_q \left(\hat{\mathbf{q}}_{t+k}, \mathbf{q}_{t+k}^{\mathrm{gt}} \right)^2, \tag{13}$$

The risk prediction loss is formulated as

$$L_\rho = \frac{1}{L_s} \sum_{k=1}^{L_s} \left\| \hat{\rho}_{t+k} - \rho_{t+k}^{\mathrm{gt}} \right\|_2^2. \tag{14}$$

Here, $\mathbf{p}$ and $\mathbf{q}$ denote the position and orientation, respectively; L_s denotes the prediction horizon; and $d_q(\cdot, \cdot)$ represents the distance measure defined over the orientation space.

During inference, each candidate generated by the actor is rolled out by the FDM and evaluated using a dynamics-aware cost:

$$J^i = w_{\mathrm{safe}} J^i_{\mathrm{safe}} + w_{\mathrm{prog}} J^i_{\mathrm{prog}} + w_{\mathrm{smooth}} J^i_{\mathrm{smooth}}. \tag{15}$$

. The safety cost penalizes predicted trajectories that approach obstacles:

$$J^i_{\mathrm{safe}} = \frac{1}{L_s} \sum_{k=1}^{L_s} \max(0, m - \hat{\rho}^i_{t+k})^2, \tag{16}$$

where m is a safety margin. The topological progress cost encourages the predicted trajectory to approach the upcoming structural route:

$$J^i_{\mathrm{prog}} = \sum_{\mathbf{w} \in \mathcal{W}'_t} \alpha_{\mathbf{w}} \min_{1 \le k \le H_s} \left\| \hat{\mathbf{p}}^i_{t+k} - \mathbf{w} \right\|_2. \tag{17}$$

The smoothness cost regularizes consistency with the previously executed action and discourages abrupt control changes:

$$J^i_{\mathrm{smooth}} = \left\| \mathbf{u}^i_t - \mathbf{u}_{t-1} \right\|_2^2 + \gamma \sum_{k=1}^{L_u - 1} \left\| \mathbf{u}^i_{t+k} - \mathbf{u}^i_{t+k-1} \right\|_2^2. \tag{18}$$

Experiments Settings

The simulation environment is built on the Isaac Lab ManagerBasedEnv framework with a differential-drive Clearpath Jackal robot as the default mobile platform. Each environment instance is equipped with onboard sensing modalities including an RGB-D camera, a third-person camera, an IMU, a 2D LiDAR, and contact sensors, enabling evaluation under both perception-driven planning and physics-based collision detection. The simulator supports warehouse, Gibson, and Matterport-style indoor scenes; all main experiments are conducted in Matterport3D environments imported as USD assets. For parallel simulation, different Matterport3D scenes can be assigned to different environment instances to improve scene diversity during data collection and evaluation. The physics simulation runs at a timestep of $1/120$ s with a control decimation factor of 2, resulting in an effective control frequency of 60 Hz, while rendering is performed every two physics steps. Each episode is initialized with randomly sampled start and goal locations. Traversable positions are selected using a 0.10 m-resolution occupancy map together with safety-radius filtering, ground-height constraints, and footprint-based collision checks. The robot is initialized through state teleportation followed by repeated resampling and short stability verification to ensure valid roll, pitch, velocity, and height conditions before execution. A navigation episode is considered successful when the robot reaches within 0.20 m of the goal; otherwise, it is terminated as a timeout failure after at most 6000 simulation steps.

Table 2: Comparison of navigation performance and resource consumption.

Method	Map Size (KB)↓	Step Time Cost (ms)↓	Success Rate (%)↑	Collision Rate (%)↓	Timeout Rate (%)↓
Wavefront Dijkstra + DWB	77.97	52.70	85.12	2.29	14.88
Wavefront Dijkstra + MPPI	77.97	58.57	89.25	6.61	10.74
Wavefront Dijkstra + RPP	77.97	52.52	89.25	10.74	10.74
A* + DWB	77.97	52.89	66.94	9.91	33.06
A* + MPPI	77.97	50.13	64.46	5.78	35.54
State Lattice + MPPI	77.97	57.91	70.24	15.70	29.75
State Lattice + DWB	77.97	58.71	55.37	18.18	44.63
iPlanner	0	3.22	15.70	27.27	84.30
iPlanner + Topo	3.66	4.38	32.23	19.83	67.77
Ours	3.66	6.70	83.47	16.52	16.52

SUPPLEMENTARY RESULTS

The full set of Nav2 combinations evaluation is reported in Tab.2.

ABLATION STUDY SETTINGS

We conduct ablation experiments to evaluate the contribution of three key components: the structural topological map guidance, the prospective motion generation, the FDM motion prediction and evaluation. In *w/o generation*, the prospective motion generation is replaced with 64 randomly sampled action sequences, while the trained FDM is retained for candidate evaluation. In *w/o FDM*, the trained flow is retained, but FDM prediction is removed; candidates are selected using a cost defined directly over the action sequence and topological target. In *w/o topo*, spatial cognitive map are removed, and the relative final-goal coordinate is used as the guidance input to the prospective motion planning.