2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Learning-informed Long-Horizon Navigation under Uncertainty for Vehicles with Dynamics

Abhish Khanal Hoang-Dung Bui Erion Plaku Gregory J. Stein

***Abstract*— We present a novel approach to learning-augmented, long-horizon navigation under uncertainty in large-scale environments in which considering the robot dynamics is essential for informing good behavior. Our approach tightly integrates sampling-based motion planning, which computes dynamically feasible routes to the goal through different unexplored boundaries, and a high-level planner that leverages predictions about unseen space to select a route that best makes progress toward the unseen goal. Owing to its ability to understand the impacts of the robot's dynamics on how it should attempt to reach the goal, our approach achieves both higher reliability and improved navigation performance compared to competitive learning-informed and non-learned baselines in simulated office-building-like environments.**

I. Introduction

We consider the navigation problem for a robot with dynamics operating in unknown environments, where the objective is to reach an unseen point goal while avoiding collisions and reducing the overall travel distance. As the robot moves, it uses its sensor to detect nearby obstacles and available free space, updating its partial map of the environment and replanning when necessary. Effective navigation in such scenarios requires considering differential constraints imposed by the robot dynamics, which limit the velocity, acceleration, turning radius, and motion directions posing significant challenges.

Sampling-based motion planning is well-suited for navigation tasks. However, such planners are designed under the assumption that the environment is fully known [1], [2], [3], [4]. To adapt them to work in an unknown environment, a common strategy is to assume that all the unseen space is unoccupied, plan a trajectory to the goal, and then re-plan when obstacles are observed [5], [6], [7]. However, as it does not consider what lies in unseen space, a robot planning under this optimistic assumption will often encounter tight spaces, corners, or dead-ends that increase overall navigation time or get the robot unrecoverably stuck.

To plan effectively in unknown environments, the robot must infer what lies in the unseen space so that it may determine which routes will be most likely to reach the unseen goal. This objective falls under the umbrella of *navigation under uncertainty*—often modeled as a Partially Observable Markov Decision Process (POMDP) [8], [9]—yet planning using this model is computationally intensive, as determining the best route in general requires envisioning all possible configurations of unseen space and imposes an intractable challenge.

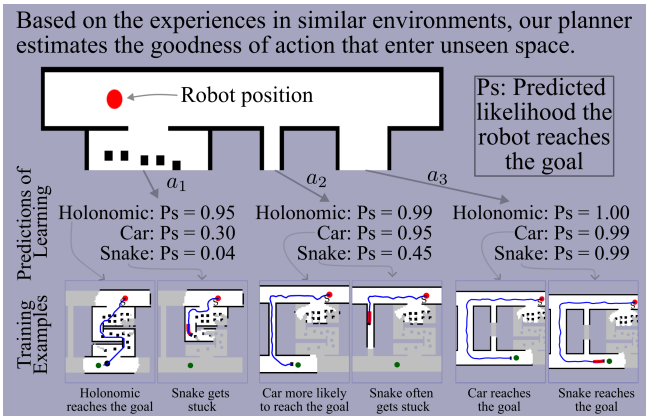

Fig. 1. **Our planner uses dynamics-aware predictions about unseen space to avoid poor routes to the goal.** Learning provides vehicle-specific predictions about the goodness of actions entering unseen space and so can reach the unseen goal more quickly and reliably than competitive baselines.

To overcome the difficulties associated with dynamics-aware planning in a partial map, learning can be used [10], [11] to predict which routes entering unseen space are less likely to encounter unexpected resistance or get the robot stuck. By estimating the likelihood that the robot will be able to reach its goal via one region of unseen space or another, a high-level planner could guide the robot to select a dynamically-feasible trajectory expected to reduce cost and best ensure that the robot reaches its target.

To guide the robot via predictions from learning, we take inspiration from the Learning over Subgoals Planning (LSP) abstraction [12], in which the robot uses learning to select among a set of *frontiers*—each a boundary between free and unknown space—through which the robot can travel in an effort to reach the unseen point goal. Informed by predictions from learning, the high-level LSP planner encourages the robot to navigate through frontiers that are more likely to connect to the unseen goal. However, LSP is concerned merely with the *structure* of unseen space and so planning does not consider (i) how the robot dynamics impact traversal through known space or (ii) how predictions regarding the goodness of actions entering unseen space strongly depend on the robot's dynamics. Instead, performing well requires an approach to long-horizon planning under uncertainty that in-

A. Khanal, H. Bui, and G. J. Stein are with the Department of Computer Science, George Mason University, Fairfax, VA, 22030, USA. E. Plaku is with the National Science Foundation, Alexandria, VA 22314 USA.

The work by E. Plaku is supported by (while serving at) the National Science Foundation. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

corporates vehicle dynamics and so guides the robot towards routes through unseen space well-suited to its embodiment.

In this paper, we present a hierarchical planner that uses (i) sampling-based motion planning to plan trajectories through each frontier into the unseen space to the goal, and (ii) a high-level planner that leverages dynamics-aware learning-generated predictions about travel through unseen space to select the trajectory that will allow the robot to best make progress towards its goal. To compute the trajectory through the frontiers, we use the sampling-based motion planner introduced by Khanal et al. [5]. Meanwhile, the high-level planner, inspired by LSP [12], predicts dynamics-aware estimates of frontiers (e.g., the likelihood that the robot can reach the unseen goal) and uses those predictions to seek out routes well suited for the robot's constraints.

In experiments in simulated office-inspired environments, we demonstrate high reliability and reduced average cost compared to a non-learned baseline approach; we also demonstrate the poor reliability of a learned baseline that plans without considering robot dynamics, emphasizing the importance of our approach for good performance.

II. Related Work

a) Sampling-based Motion Planning: Sampling-based motion planning is commonly used for robots with dynamics because of its computational efficiency [13], [14]. These planners generate vast motion trees whose branches correspond to collision-free and dynamically-feasible trajectories. RRT [1], [15] and its variants [16], [17] rely on nearest neighbors to guide the expansion. Other methods exploit cell decompositions [18] or introduce a discrete layer to guide the motion-tree expansion [19], [20], [21]. Machine learning has also been used to better guide the expansion [22], [23], [24], [25], [26], [27], [28], [29]. These approaches are typically designed for fully-known environments and exhibit slower replanning in unknown environments. Khanal et al. [5] leverages adaptive grid subdivisions and prior solutions to guide the motion-tree expansion to significantly reduce the replanning time when navigating in an unknown environment. However, the planner does not infer what lies beyond unseen space. Consequently, it may explore areas that do not lead to the goal, potentially increasing the overall cost or even becoming trapped in narrow spaces.

b) Planning under Uncertainty and Learning for Planning: Planning under uncertainty, often modeled as POMDP [30], [8], [31] is incredibly difficult in practice and so learning is often used to help inform good behavior in such domains. Learning has been applied in navigation under uncertainty settings where vehicle dynamics strongly determine behavior, yet most such approaches [32], [10], [33], [34] reason only a few time steps into the future and are thus not well suited to *long-horizon* navigation domains. There additionally exist model-free planning approaches, often trained via deep reinforcement learning, capable of handling vehicle dynamics. Such approaches have shown impressive successes for planning under uncertainty [35], [36], [37], [38], [39], though have onerous data requirements and can be brittle in large scale environments [40]. The LSP abstraction of Stein et al. [12] proposes a high-level model-based planning abstraction for long-horizon planning under uncertainty, though it does not consider vehicle dynamics, limiting performance for dynamically-constrained robots.

Fig. 2. The snake-like robot model (with second-order dynamics) used in our experiments. Image re-used from Khanal et al. [5]

III. Problem Formulation

The robot aims to navigate a partially-mapped environment to reach the goal in minimum expected time, updating its map via a range- and occlusion-limited planar laser scanner as it travels. The motions must be collision-free and dynamically-feasible. Effective performance also requires the robot to make predictions about the goodness and potential feasibility of trajectories that enter unseen space, so that it may avoid likely dead ends or becoming stuck.

a) Robot Model: The robot, denoted as $\mathcal{R} = \langle \mathcal{P}, \mathcal{S}, \mathcal{U}, f \rangle$, is characterized by its shape $\mathcal{P}$, state space $\mathcal{S}$, control space $\mathcal{U}$, and dynamics represented as differential equations $f : \mathcal{S} \times \mathcal{U} \to \dot{\mathcal{S}}$. In the experiments, a snake-like robot (as depicted in Fig. 2) is employed, modeled as a car towing N trailers [14]. The snake comprises a head and N interconnected rectangular links, each having a length $L = 1m$, a width $W = 0.6m$, and connected with a hitch distance $H = 0.01m$. The robot state $s = (x, y, v, \psi, \theta_0, \theta_1, \ldots, \theta_N)$ encompasses position (x, y), velocity (v with $|v| \leq 2$ m/s), steering angle (ψ with $|\psi| \leq 1.5$ radians), head orientation (θ_0), and orientation (θ_i) for each individual link. The control input $u = (u_a, u_\omega)$ comprises acceleration (u_a with $|u_a| \leq 2m/s^2$) and steering rate (u_ω with $|u_\omega| \leq 3$ radians/s). The differential equations f for the dynamics are defined as

$$\begin{aligned} \dot{x} = v\cos(\theta_0)\cos(\psi), \quad \dot{y} = v\sin(\theta_0)\cos(\psi), \\ \dot{\theta_0} = v\sin(\psi)/L, \quad \dot{v} = u_a, \quad \dot{\psi} = u_\omega, \\ \forall i \in \{1, \ldots, N\} : \\ \dot{\theta_i} = \frac{v}{H}(\sin(\theta_{i-1}) - \sin(\theta_0))\textstyle\prod_{j=1}^{i-1}\cos(\theta_{j-1} - \theta_j). \end{aligned} \tag{1}$$

b) Computing Dynamically-Feasible Trajectories: The robot dynamics f determine the new state $s_{\text{new}} \in \mathcal{S}$ that is obtained when a control $u \in \mathcal{U}$ is applied to a state $s \in \mathcal{S}$. The new state s_{new} is computed by a function $\textsc{simulate}(s, u, f, dt)$, which numerically integrates f for a time step dt, i.e.,

$$s_{\text{new}} \leftarrow \textsc{simulate}(s, u, f, dt). \tag{2}$$

Applying a sequence of controls $\langle u_0, \ldots, u_{\ell-1} \rangle$ in succession gives rise to a dynamically-feasible trajectory $\zeta : \{0, \ldots, \ell\} \to \mathcal{S}$, where $\zeta(0) \leftarrow s$ and $\forall i \in \{1, \ldots, \ell\}$:

$$\zeta(i+1) \leftarrow \textsc{simulate}(\zeta(i), u_i, f, dt). \tag{3}$$

c) Navigation Under Uncertainty: Formally, this problem can be represented as a POMDP [8], [30]. The expected cost Q under this model can be written via a belief space b_t variant of the Bellman equation [31]:

$$Q(b_t, a_t) = \sum_{b_{t+1}} P(b_{t+1}|b_t, a_t)\big[R(b_{t+1}, b_t, a_t) + \min_{a_{t+1}\in\mathcal{A}(b_t+1)} Q(b_{t+1}, a_{t+1})\big], \quad (4)$$

where $R(b_{t+1}, b_t, a_t)$ is the cost accumulated by reaching belief state b_{t+1} from b_t by executing primitive action a_t, consisting of a one-second-long dynamically-feasible motion corresponding to the execution of a control $u \in \mathcal{U}$.

IV. Preliminaries

In this section, we present the salient details of the high-level planner of Stein et al. [12], whose learning-augmented model-based planning abstraction we build upon to determine where the robot should enter unseen space in an effort to reach the unseen goal, and of the low-level sampling-based trajectory planner from Khanal et al. [5], which we use to compute motion costs needed to inform high-level navigation planning and for making progress through known space.

A. Planning in a Partial Map via Learning over Subgoals

Using the POMDP model of Eq. (4) to plan is both computationally intractable and requires access to a distribution over environments too difficult to obtain in general. To mitigate these challenges, the LSP approach introduces an action abstraction such that each high-level action corresponds to a *subgoal*: a point on a frontier (a contiguous boundary between free and unseen space) through which the robot can enter unseen space and try to reach the goal. Thus, a high-level action consists of (1) navigating to a *subgoal* and (2) exploring beyond it in an effort to reach the goal. Under the LSP approach, learning is used to estimate the properties associated with high-level actions—compact statistics of unseen space referred to *subgoal properties*—to inform high-level, model-based planning.

Under LSP, planning is done over an abstract belief state: a tuple $b_t = \langle m_t, \mathcal{A}_t, s_t\rangle$, where m_t is the (partial) map of the environment, $\mathcal{A}_t$ is the set of unattempted high-level actions, and s_t is the robot state. Upon selecting a high-level action $a_t \in \mathcal{A}_t$ the robot travels a distance $D(m_t, s_t, a_t)$ in known space to reach the *subgoal*. High-level actions have binary outcomes: with probability $P_S(a_t)$, the robot *succeeds* in reaching the goal accumulating *success cost* $R_S(a_t)$, the expected cost of the robot reaching the goal; with a probability $1 - P_S(a_t)$, the robot *fails* to reach the goal and accumulates an *exploration cost* $R_E(a_t)$, the cost of exploring that region. Fig. 3 shows a schematic of the LSP action abstraction.

The terms P_S, R_S, and R_E—too difficult to compute exactly—are estimated from images collected on board the robot via learning. Problematically, these subgoal properties as well as the known-space distance D are computed only for a holonomic robot and so do not reflect the actual costs and likelihood of success for a dynamically-constrained vehicle, leading to poor performance. *We address this key limitation in this work.*

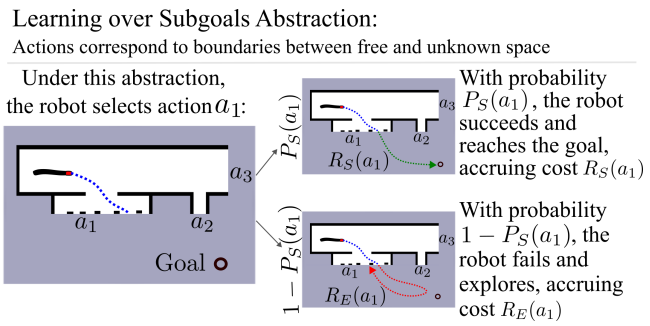

Fig. 3. The Learning over Subgoals Planning approach [12] uses learning to predict statistics of unseen space associated with temporally extended actions entering unseen space.

Upon executing a high-level subgoal-action a_t and failing to reach the goal, the robot can select another action, defining a model-based planning problem for the LSP abstract belief and action abstraction. After completing a_t during planning, the updated abstract belief state reflects that the robot has moved to the location of the subgoal $s(a_t)$ and that the set of unattempted high-level actions has updated: $b_{t+1} = \langle m_t, \mathcal{A}_t\backslash\{a_t\}, s(a_t)\rangle$. Upon failing to reach goal via action a_t, the robot selects another action $a_{t+1} \in \mathcal{A}_t\backslash\{a_t\}$ from the updated set of unattempted subgoal-actions and planning continues.

The LSP planner selects the action $a_t \in \mathcal{A}_t$ that minimizes overall expected cost computed via a Bellman equation:

$$Q(b_t, a_t \in \mathcal{A}_t) = D(m_t, s_t, a_t) + P_S(a_t)R_S(a_t) + (1 - P_S(a_t))\left[R_E(a_t) + \min_{a_{t+1}\in\mathcal{A}_t\backslash\{a_t\}} Q(b_{t+1}, a_{t+1}))\right] \quad (5)$$

B. Execution and Planning Framework

We build upon the execution and planning framework from Khanal et al. [5]. The framework consists of two modules: an execution module (EM) and a planning module (PM). The execution module progressively executes the planned trajectory, while the planning module's role is to generate trajectories that are both collision-free and dynamically-feasible based on the partial map. When the robot's sensor detects new obstacles, the execution module engages the planning module to replan, allowing the robot to avoid the newly detected obstacles effectively.

PM employs discrete search over an adaptive grid subdivision and leverages previous solutions to guide the sampling-based expansion of a motion tree from the current state toward the goal. PM was shown to significantly reduce the planning time, enhance solution consistency, thereby minimizing oscillatory behavior, and increase clearance from the obstacles, which in turn improves overall robustness.

V. Methodology

We propose a novel planning approach for improved dynamics-aware, learning-informed long-horizon navigation

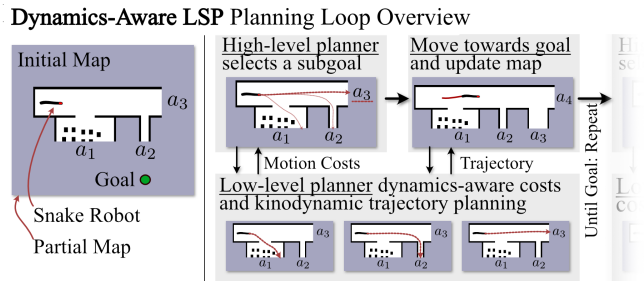

Fig. 4. **Overview of the Dynamics-Aware LSP Planning Loop** Our planner tightly integrates high-level planning, which directs the robot towards the most promising region of unseen space, and low-level motion planning, which provides costs and trajectories to inform high-level planning and movement through known space.

in partially mapped environments. Our approach consists of two integrated modules: a high-level subgoal-selection planner and a low-level trajectory planner, as shown in Fig. 4. Critically, though our high-level planner relies upon the LSP abstraction, Eq. (5), our method instead uses motion costs D and estimated statistics of unseen space (the subgoal properties: P_S, R_S, and R_E) that both reflect the underlying vehicle dynamics, made possible by the low-level trajectory planner. As such, our planner affords improved behavior in complex, cluttered environments, in which the goodness of an action that enters unseen space strongly depends on the vehicle's dynamics.

We first discuss how high-level action selection incorporates vehicle dynamics in Sec V-A. The high-level planner relies on predictions about the robot's ability to traverse unseen space, estimated via a learned model trained via data that incorporates the robot's dynamics, as discussed in Sec. V-B. Finally, Sec. V-C describes how our robot makes progress towards the goal once a subgoal-action is chosen.

A. Dynamics-Aware, High-Level Planning

For high-level planning, we leverage the LSP abstraction discussed in Sec. IV-A. Planning via Eq. (5) depends on being able to compute motion costs through known space D and estimate compact statistics of unseen space that depend on both the partial map m_t and (unique to this work) the vehicle's dynamics f: (i) P_S, the likelihood that the robot can successfully reach the goal via a particular route through unseen space, (ii) R_S, the expected cost of success for that robot to reach the goal if the goal can be reached, and (iii) R_E, the expected cost of exploration for that robot to reveal the space and turn back. The known space distances $D(f, m_t, s_t, a_t)$ are computed via the low level trajectory planner discussed in Sec. V-C, and so travel distances to each subgoal and between subgoals reflect the vehicle's kinodynamic constraints. The subgoal properties, $P_S(f, a_t)$, $R_S(f, a_t)$, and $R_E(f, a_t)$, also depend on the robot's dynamics, though are too difficult to compute directly and so are estimated via learning, as described in Sec. V-B.

B. Learning: Estimating Dynamics-Aware Statistics of Unseen Space

So as to select the subgoal expected to best afford quick progress towards the goal, we require an estimator that can predict the dynamics-aware subgoal properties: $P_S(f, a_t)$, $R_S(f, a_t)$, and $R_E(f, a_t)$. This section details how we generate the training data needed to inform these predictions about unseen space and also defines the learned model (a convolutional neural network) and details our training procedure.

a) Generating Data to Estimate Dynamics-Aware Statistics of Unseen Space: Data is generated during an offline training phase from trials collected onboard each robot that navigates via a non-learned, optimistic strategy through previously unseen environments. For all steps of every trial, whenever the robot reveals more of the map, new frontiers—boundaries between free and unknown space that each correspond to a subgoal and associated exploratory action—give rise to training data. For each newly updated frontier, we use the underlying "known map" to generate labeled data for the subgoal properties for the frontier. When the known map is available, we can determine whether it is possible to reach the goal via the selected boundary for the robot of interest and the associated cost of doing so.

Thus, if a dynamically-feasible trajectory can be found through the subgoal of interest and the frontier to which it corresponds, the label for the probability of success is $P_S = 1$ and the label for R_S is the associated cost of the part of the trajectory in unseen space. If the goal cannot be reached, the success probability label is $P_S = 0$ and the cost of exploration R_E is the trajectory cost to the farthest reachable point in unseen space, also computed via our motion planner. Inputs to the learned model, described in more detail in the next section, consist primarily of cropped, top-down views of the partial map centered on the subgoal of interest and are recorded alongside the labels for the dynamics-aware subgoal properties.

We collect training data separately for all robots and train a separate learned model for each. By using the learned model trained specifically for each robot, we ensure the the predictions about unseen space are tailored to the robot's dynamics; in particular this allows robots with stringent kinodynamic constraints to avoid cluttered or tightly-constrained regions in which it might otherwise become stuck and prefer routes more likely to allow it to successfully reach the unseen goal.

b) Estimating the Subgoal Properties P_S, R_S and R_E: Our estimator is a convolutional neural network, which consumes a 32×32 pixel crop of the occupancy grid centered around the subgoal of interest, as well as a $32 \times 32 \times 2$ grid that encodes both (i) the relative distance to the center of the subgoal and (ii) the relative distance to the goal for every pixel in the grid. The $32 \times 32 \times 3$ input is passed through five two-layer convolutional encoder blocks, each followed by a max-pool operation. The output is flattened and then passed through an additional three fully-connected layers to produce a three-element output: the logits for P_S and the costs R_S and R_E. LeakyReLU activation functions are used

Model	Non-learned		Holonomic		Dynamics-Aware	
Vehicle	Success	Cost	Success	Cost	Success	Cost
Holonomic	100%	131.8	100%	120.7	100%	120.7
Car	99%	230.9	59%	131.1	100%	160.8
Snake	95%	234.6	57%	127.4	99%	162.1

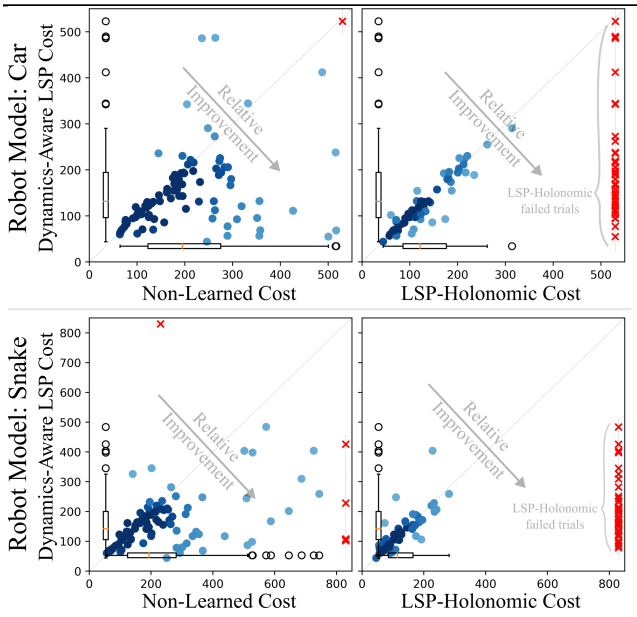

Fig. 5. **Results: Two Hallway Experiments** We conduct 100 trials in the Two Hallway environment for each planner {Non-Learned, LSP-Holonomic, Dynamics-Aware LSP} for each robot model {Holonomic, Car, Snake}. The results demonstrate that our Dynamics-Aware LSP planner achieves both low cost and high reliability. See Sec. VI-A for details.

after each neural layer. The loss function consists of a cross-entropy loss for P_S and L1-loss (regression) for the two cost terms. All models are trained for four epochs. During deployment, the trained model for each robot dynamics model estimates the dynamics-aware subgoal properties for each subgoal individually so that they can be used by the high-level planner to inform planning.

C. Low-level Planning: Progressing Towards the Goal

Once the subgoal is chosen using high-level planner, we use the low-level trajectory planner [5] to compute a dynamically feasible trajectory through that subgoal to the goal. To make sure the trajectory passes through the chosen subgoal, we mask all the other subgoal (marking them as obstacles in the map) and then apply our low-level motion planner through the masked grid.

Once the trajectory is computed, the robot moves one time step along the trajectory, and updates its partial map using the sensor information it collects via LIDAR. Whenever the map is updated, high-level planning is rerun and a new subgoal may be selected. The planning and navigation process repeats until the goal is reached or the robot becomes stuck, which is reported as a failure.

VI. Experimental Results

Our experiments consist of simulated trials in procedural office-like environments (as shown in Fig. 6 and 8), where the robot is placed in previously unmapped spaces and

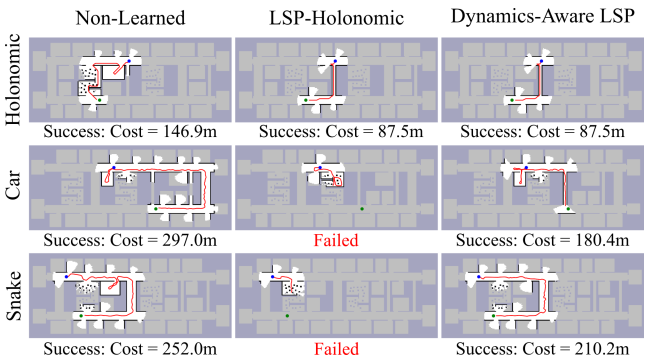

Fig. 6. **Example Trials: Two Hallway Environment** These trials show the ability of our Dynamics-Aware LSP planner to both improve performance and achieve high reliability.

expected to quickly and successfully reach an unseen point goal. For each trial, we deploy three different types of robot: (1) a *holonomic vehicle*, which has no dynamic constraints on its motion (2) a *car*, as described in Sec. III, and (3) a *snake*, modeled as a car pulling 5 trailers as described in Sec. III.

Our procedural office-like spaces consist of parallel hallways, each surrounded by multiple dead-end "office" rooms and three different types of passages that connect the hallways: (1) clutter-filled winding passages, (2) narrow corridors, and (3) wide corridors. The varied procedurally generated passages are meant to be more or less amenable to robots of varied dynamics—e.g., the tight confinement of the clutter-filled winding passages is most difficult for snake-like robots, moderately challenging for a car-like robot, and relatively easy for a holonomic robot. Notably, predictions about unseen space that do not take the robot embodiment into consideration are likely to lead dynamically-constrained vehicles to pursue options at which they are unlikely to succeed. The start and goal locations for each trial are randomly sampled from opposite ends of the map so that the robots must traverse the passages to succeed. We evaluate following approaches for comparison:

Optimistic Planner (Non-Learned) This planner assumes all unseen space is unoccupied and uses the motion planner of Sec. IV-B to navigate, updating the map as it travels.

LSP-Holonomic (Holonomic Learned Model) This planner uses learned model trained on holonomic robot to make predictions about unseen space, a mismatch with the dynamically-constrained vehicles, yet known-space distances D rely on the low-level motion planner and so respect vehicle dynamics. Once a subgoal is chosen, the motion planner provides a dynamically-feasible trajectory for the robot to follow that navigates towards the selected subgoal.

Dynamics-Aware LSP Planner (Ours) This planner uses learned model trained on robot itself to estimate the properties of unseen space, thus considering the vehicle dynamics for all aspects of high-level and for low-level planning.

Model	Non-learned		Holonomic		Dynamics-Aware	
Robot	Success	Cost	Success	Cost	Success	Cost
Holonomic	100%	208.3	100%	199.8	100%	199.8
Car	91%	402.5	37%	225.8	100%	281.4
Snake	90%	389.5	33%	237.2	96%	280.5

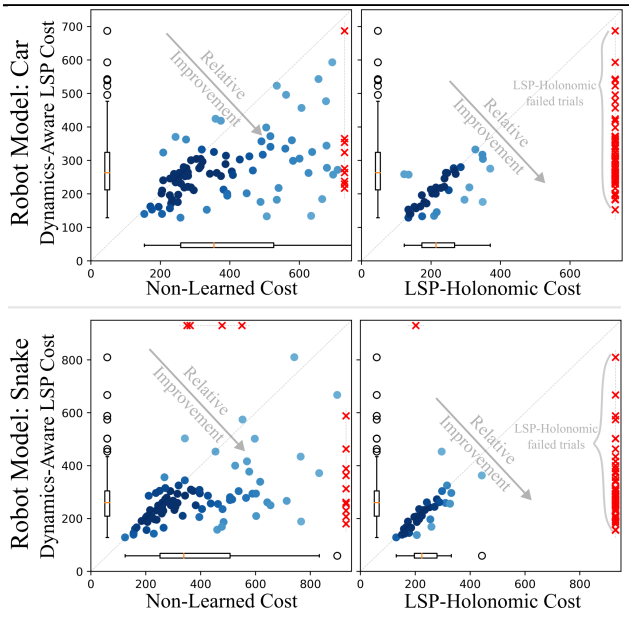

Fig. 7. **Results: Three Hallway Experiments** We conduct 100 trials in the Two Hallway environment for each planner and robot model. The results demonstrate that our Dynamics-Aware LSP planner achieves both low cost and high reliability. See Sec. VI-B for details.

A. Two Hallway Results

We first evaluate the performance of all planners and robots in *Two-Hallway* Environments, in which only two parallel hallways exist, connected by passages of various types. Fig 5 includes results collected for 100 trials for all three robot types—holonomic, car, and snake—and additionally show scatter plots for all trials in Fig. 5.

Our results show the effectiveness of our Dynamics-Aware LSP planner, which achieves both near 100% reliability and 30.5% and 30.9% lower average cost than the non-learned baseline for the car and the snake robots, respectively, owing to its ability to quickly seek out promising routes to the goal. Moreover, unaware of the vehicle dynamics, the LSP-Holonomic planner routinely leads the vehicles towards tightly constrained passages that the snake and car are less likely to successfully traverse than the holonomic model upon which it is trained. By contrast, our Dynamics-Aware LSP planner succeeds 100% and 99% of trials for the car and snake, respectively—considerably more reliable than the LSP-Holonmic planner, which succeeds only 59% and 57% for the car and snake, respectively—as it is trained with the vehicle dynamics in mind and so has learned to avoid such regions. This effect can be seen in Fig. 6.

B. Three Hallway Results and Generalization Performance

We additionally demonstrate results in a Three Hallway office-like environment. Critically, to demonstrate the ability of our approach to generalize its predictions and scale to

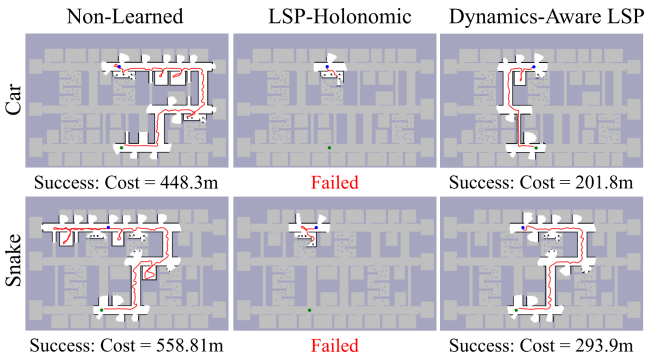

Fig. 8. **Example Trials: Three Hallway Environment** These trials show the ability of our LSP Dynamics planner to both improve performance and achieve high reliability.

larger environments, we use the same learned model as in the Two Hallway environment, which has never specifically seen any Three Hallway maps.

The results in our Three Hallway environments, shown in both the scatterplots and data table of Fig. 7, are consistent with those of the Two Hallway maps from the previous section, demonstrating that our Dynamics-Aware LSP planner achieves both higher reliability and improved performance compared to the non-learned baseline. We also observe significantly higher reliability than the LSP-Holonomic learned baseline, which succeeds in only 37% and 33% of the car and snake trials, respectively, compared to the 100% and 96% success rates of our Dynamics-Aware LSP planner. Fig. 8 shows two trials that illustrate the representative performance of the three planners across trials for each of the car and snake robots. Moreover, these results highlight the consistent performance and reliability of our approach even in environments larger and more complex than those it has seen during training.

VII. Conclusion

We have demonstrated Dynamics-Aware LSP Planner, a learning-augmented model-based planning approach that tightly couples the high-level Learning over Subgoals planning abstraction of Stein et al. [12] and the sample-efficient low-level motion planner of Khanal et al. [5] to achieve improved long-horizon navigation in partially mapped environments. We demonstrate performance in two- and three-hallway simulated office-inspired environments in which good behavior requires considering robot dynamics. Our results show improved reliability and lower expected cost over competitive baselines and achieves effective generalization to larger environments than were seen during training. In future work, we aim to extend our approach to support learning from rich, high-dimensional sensor information—e.g., vision or semantic segmentation—and demonstrate performance in large-scale real world environments on a physical platform.

References

[1] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[2] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal on Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[3] ——, "Optimal kinodynamic motion planning using incremental sampling-based methods," in *IEEE Conference on Decision and Control*, Atlanta, GA, 2010, pp. 7681–7687.

[4] L. E. Kavraki, P. Švestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[5] A. Khanal, H.-D. Bui, G. J. Stein, and E. Plaku, "Guided sampling-based motion planning with dynamics in unknown environments," in *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, 2023, pp. 1–8.

[6] S. Koenig and M. Likhachev, "D* lite," in *Eighteenth national conference on Artificial intelligence*, 2002, pp. 476–483.

[7] F. Yang, C. Cao, H. Zhu, J. Oh, and J. Zhang, "Far planner: Fast, attemptable route planner using dynamic visibility update," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 9–16.

[8] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, 1998.

[9] ——, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, pp. 99–134, 1998.

[10] C. Richter, W. Vega-Brown, and N. Roy, "Bayesian learning for safe high-speed navigation in unknown environments," *Robotics Research: Volume 2*, pp. 325–341, 2018.

[11] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive mapping and planning for visual navigation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2616–2625.

[12] G. J. Stein, C. Bradley, and N. Roy, "Learning over subgoals for efficient navigation of structured, unknown environments," in *Conference on robot learning*. PMLR, 2018, pp. 213–222.

[13] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005.

[14] S. M. LaValle, *Planning Algorithms*. Cambridge, MA: Cambridge University Press, 2006.

[15] ——, "Motion planning: The essentials," *IEEE Robotics & Automation Magazine*, vol. 18, no. 1, pp. 79–89, 2011.

[16] D. Devaurs, T. Simeon, and J. Cortés, "Enhancing the transition-based rrt to deal with complex cost spaces," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 4120–4125.

[17] A. Shkolnik, M. Walter, and R. Tedrake, "Reachability-guided sampling for planning under differential constraints," in *IEEE International Conference on Robotics and Automation*, 2009, pp. 2859–2865.

[18] I. A. Şucan and L. E. Kavraki, "A sampling-based tree planner for systems with complex dynamics," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 116–131, 2012.

[19] E. Plaku, "Region-guided and sampling-based tree search for motion planning with dynamics," *IEEE Transactions on Robotics*, vol. 31, pp. 723–735, 2015.

[20] E. Plaku, E. Plaku, and P. Simari, "Clearance-driven motion planning for mobile robots with differential constraints," *Robotica*, vol. 36, pp. 971–993, 2018.

[21] ——, "Direct path superfacets: An intermediate representation for motion planning," *IEEE Robotics and Automation Letters*, vol. 2, pp. 350–357, 2017.

[22] J. Wang, T. Zhang, N. Ma, Z. Li, H. Ma, F. Meng, and M. Q.-H. Meng, "A survey of learning-based robot motion planning," *IET Cyber-Systems and Robotics*, vol. 3, no. 4, pp. 302–314, 2021.

[23] J. Wang, W. Chi, C. Li, C. Wang, and M. Q.-H. Meng, "Neural RRT*: Learning-based optimal path planning," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1748–1758, 2020.

[24] W. J. Wolfslag, M. Bharatheesha, T. M. Moerland, and M. Wisse, "RRT-CoLearn: towards kinodynamic planning without numerical trajectory optimization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1655–1662, 2018.

[25] H.-T. L. Chiang, J. Hsu, M. Fiser, L. Tapia, and A. Faust, "RL-RRT: Kinodynamic motion planning via learning reachability estimators from RL policies," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4298–4305, 2019.

[26] R. Yonetani, T. Taniai, M. Barekatain, M. Nishimura, and A. Kanezaki, "Path planning using neural A* search," in *International Conference on Machine Learning*, 2021, pp. 12 029–12 039.

[27] A. H. Qureshi, Y. Miao, A. Simeonov, and M. C. Yip, "Motion planning networks: Bridging the gap between learning-based and classical motion planners," *IEEE Transactions on Robotics*, vol. 37, no. 1, pp. 48–66, 2020.

[28] O. Arslan and P. Tsiotras, "Machine learning guided exploration for sampling-based motion planning algorithms," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 2646–2652.

[29] X. Xiao, B. Liu, G. Warnell, and P. Stone, "Motion planning and control for mobile robot navigation using machine learning: a survey," *Autonomous Robots*, vol. 46, no. 5, pp. 569–597, 2022.

[30] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling, "Learning policies for partially observable environments: Scaling up," in *Machine Learning Proceedings*, 1995.

[31] J. Pineau and S. Thrun, "An integrated approach to hierarchy and abstraction for pomdps," Carnegie Mellon University, Tech. Rep., 2002.

[32] A. Elhafsi, B. Ivanovic, L. Janson, and M. Pavone, "Map-predictive motion planning in unknown environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8552–8558.

[33] K. D. Katyal, A. Polevoy, J. Moore, C. Knuth, and K. M. Popek, "High-speed robot navigation using predicted occupancy maps," in *International Conference on Robotics and Automation (ICRA)*, 2021, pp. 5476–5482.

[34] Z. A.-H. Santhosh Kumar Ramakrishnan and K. Grauman, "Occupancy anticipation for efficient exploration and navigation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[35] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. J. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu, D. Kumaran, and R. Hadsell, "Learning to navigate in complex environments," 2017.

[36] A. Khan, C. Zhang, S. Li, J. Wu, B. Schlotfeldt, S. Y. Tang, A. Ribeiro, O. Bastani, and V. Kumar, "Learning safe unlabeled multi-robot planning with motion constraints," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 7558–7565.

[37] J. Zhang, J. T. Springenberg, J. Boedecker, and W. Burgard, "Deep reinforcement learning with successor features for navigation across similar environments," 2017.

[38] G. Wayne, C.-C. Hung, D. Amos, M. Mirza, A. Ahuja, A. Grabska-Barwinska, J. Rae, P. Mirowski, J. Z. Leibo, A. Santoro, M. Gemici, M. Reynolds, T. Harley, J. Abramson, S. Mohamed, D. Rezende, D. Saxton, A. Cain, C. Hillier, D. Silver, K. Kavukcuoglu, M. Botvinick, D. Hassabis, and T. Lillicrap, "Unsupervised predictive memory in a goal-directed agent," 2018.

[39] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra, "Decentralized distributed PPO: Learning near-perfect pointgoal navigators from 2.5 billion frames," in *International Conference on Learning Representations (ICLR)*, 2020.

[40] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.