Monte Carlo Methods for Motion Planning and Goal Inference

By

Jovana Kondic

B.S.E., Princeton University (2021)

Submitted to the Department of Electrical Engineering and Computer Science in Partial Fulfillment of the Requirements for the Degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2024

Authored by: Jovana Kondic
Electrical Engineering and Computer Science
January 26, 2024

Certified by: Dylan Hadfield-Menell
Assistant Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by: Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Monte Carlo Methods for Motion Planning and Goal Inference

by

Jovana Kondic

Submitted to the Department of Electrical Engineering and Computer Science
on January 26, 2024, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

Human cognition exhibits remarkable abilities in reasoning about the plans of others. Even infants can swiftly generate effective predictions from minimal observations. This capability largely stems from our ability to employ specific assumptions about others' decision-making, while considering potential alternative interpretations that align with reality. Such versatility is particularly crucial in navigation tasks, where multiple strategies exist for avoiding obstacles and reaching a target location. A sophisticated autonomous system should, therefore, be capable of: *(1) acknowledging the inherent uncertainty in various obstacle avoidance strategies*; and *(2) predicting motion plans in a way that recognizes the different possibilities in a given goal-driven navigation scenario*. To address these needs, we introduce a framework that captures the stochastic nature of motion planning and prediction through Monte Carlo sampling techniques. We ensure *(1)* by shifting the focus from pure trajectory optimization to generating a variety of near-optimal paths, and achieve *(2)* by developing a prediction method capable of capturing the inherent multimodality in the distribution over goal-driven trajectories. For the former, we utilize Markov Chain Monte Carlo (MCMC) methods to obtain trajectory samples that approximate the Boltzmann distribution, a common model for approximate rationality, which incorporates a cost function derived from trajectory optimization literature. For the latter, we develop a Bayesian model of the observed agent, and utilize Bayesian inference to reason about the underlying end goals of their movement. We propose a sequential Monte Carlo method that adapts the MCMC trajectory sampling to construct plausible hypotheses about the agent's motion plan and then updates these hypotheses in real-time with new observations. In experiments conducted within continuous, obstacle-laden environments, we demonstrate our framework's effectiveness for both diversity-aware motion planning and robust inference of latent goals from partial, noisy observations.

Acknowledgements

I owe my deepest gratitude to:

Prof. Dylan Hadfield-Menell, my thesis advisor, for his endless support and patience, and for the wisdom that instilled in me a North Star to steer my aspirations in research.
Tan-Zhi Xuan, my collaborator and mentor on this project, and a friend I highly look up to, for their inspiring technical expertise, mind-blowing intellect, and indispensable guidance.
Stewy Slocum, for his invaluable contributions to this project, and for the captivating conversations that never fail to refuel my mental energy and curiosity.
The Algorithmic Alignment Group, Prof. Josh Tenenbaum and the Computational Cognitive Science Group, and Prof. Vikash Mansinghka and the Probabilistic Computing Group, for their guidance in this project and beyond.
Prof. Russ Tedrake & the Fall 2022 edition of Robotic Manipulation, and Prof. Greg Wornell & the Fall 2021 edition of Algorithms for Inference, for turning the most challenging semesters into the most treasured experiences.
Prof. Leslie Kolodziejski, Prof. Bob Berwick, and the EECS Graduate Office, for believing in me when I doubted, and for always keeping their offices open.
Dr. Boris Ivanovic, for refueling my excitement for the world of trajectory prediction.
Katie, Maisy, Rui-Jie, Sarah, Jillian, Renato, Theo, Phill, Cas, Dora, Miroslav, Nikola, Gabe, RJ, John, and Ruairidh. I feel so fortunate to have my friends be my role models and to get to call my role models my friends.
Michael Psenka, for the math lessons, the coding tricks, the late-night brainstorming, the invaluable feedback, the stirringly selfless support, and the endless encouragement.
My family, for everything. A special shout-out goes to my dad for the unmatched (and dearly cherished) figure design contributions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> *We must see that music theory is not only about music, but about how people process it. To understand any art, we must look below its surface into the psychological details of its creation and absorption.*
>
> — Marvin Minsky

In 1960, American economist Thomas Schelling demonstrated that even in a vast and complex environment like New York City, individuals could successfully coordinate and find each other without direct communication (Schelling, 1958). He posited that this is achieved by reasoning about the other's thought process and choosing a target location that is most salient based on such reasoning (Schelling defines these salient solutions as *focal points*). Such coordination challenges are just one of many examples that illustrate human capacity for understanding and predicting the behavior of others. In developmental psychology, a multitude of research demonstrates that even preverbal infants can assign intentions to observed actions (Gergely et al., 1995), infer the goals of others (Woodward, 1998), and assist others to achieve them (Warneken and Tomasello, 2007).

This capacity for *inverse planning* – inferring other's hidden goal-oriented plans from observed actions – is largely facilitated by imposing assumptions about how people act. In Schelling's coordination games, for instance, participants are presumed to engage in team reasoning, acting as members of a team with the goal to maximize

the group's interest (Bacharach, 1999). More broadly, the understanding of how intentional agents choose actions has been extensively discussed across neuroscience and philosophy (Dennett, 1971; D'Andrade, 1987; Jones and Davis, 1965; Baron-Cohen et al., 2013). Frameworks like the Folk theory and the Theory of Mind (Churchland, 1988; Carruthers and Smith, 1996) were among the first ones to formalize understanding other people as the process of ascribing mental states to them. A common assumption accompanying such frameworks is rationality, commonly modeled as the aim to maximize the expected outcome value, as formalized by the expected utility theory (Briggs, 2014).

Choosing appropriate models for decision-making is also crucial in human-AI coordination. While intelligent agents, both human and artificial, are generally assumed to act efficiently towards their goals, assuming strictly optimal behavior can limit inference capabilities. For instance, informing a passive observer is sometimes better achieved through overemphasized (and, thus, inefficient) motions (Dragan et al., 2013). More broadly, cooperative settings often incentivize actions that might not be considered optimal under standard task objectives (Hadfield-Menell et al., 2016b; Fisac et al., 2020). In addition, reaching certain target states can involve multiple equivalently valid approaches, indicating that the assumption of optimality may not be insufficient for resolving ambiguity. This is particularly evident in path planning problems, where navigating around obstacles can be done with multiple equally valid strategies (Osa, 2020). Effective reasoning about others' latent states, therefore, requires both the development of a comprehensive simulation model and the ability to maintain uncertainty, taking into account various possible outcomes when making predictions.

In this work, we focus on planning and inverse planning in the domain of locomotion. We posit that a sophisticated autonomous system should be capable of: (1) acknowledging the inherent uncertainty in various obstacle avoidance strategies; and (2) predicting motion plans in a way that recognizes the different possibilities in a given goal-driven navigation scenario. To this end, we propose the utilization of Monte Carlo sampling methods, particularly the variants that leverage gradient information

from the target distribution.

This thesis makes the following contributions:

1. **Diversity-Aware Near-Optimal Motion Planning:** We employ Markov Chain Monte Carlo (MCMC) methods to generate trajectory samples that approximate the Boltzmann distribution, commonly used to model approximate rationality. By incorporating a cost function inspired by the trajectory optimization literature, our approach shifts the focus from solely optimizing trajectories to creating a range of near-optimal paths. These paths can serve as immediate solutions for navigation problems or be used to simulate others' planning processes.

2. **Multimodal Trajectory Prediction and Real-Time Goal Inference:** We introduce a Bayesian model of the observed agent and apply Bayesian inference to deduce the underlying end goals of their movement. To effectively capture the multimodal nature of goal-driven trajectory distributions, we propose a sequential Monte Carlo method that adapts the MCMC trajectory sampling for the construction and real-time updating of plausible motion plan hypotheses.

In the next chapter of the thesis, we will provide an overview of the foundational literature on trajectory generation and goal inference and highlight the works that are most related to the work presented here. In Chapter 3, we explore various formulations of the motion planning objective, then investigate MCMC sampling as a method to obtain a variety of near-optimal navigation solutions. In Chapter 4, we extend methods from Chapter 3 for the purposes of trajectory prediction and present a Bayesian framework for online inverse planning. Finally, Chapter 5 provides a brief summary of our insights and outlines future avenues for research.

Chapter 2

Background and Related Work

This chapter outlines the key approaches to motion planning and goal inference that serve as the foundation of our work, and highlights their main advantages and trade-offs. In Section 2.1, we focus on the works related to our trajectory generation framework covered in Chapter 3 and Section 4.2. In Section 2.2, we situate the literature within our inverse planning framework covered in Chapter 4.

2.1 Trajectory Planning

2.1.1 Sampling-based Motion Planning

Sampling-based algorithms have proven highly effective in quickly identifying feasible solutions within complex environments. Their success is attributed to the scalability and versatility achieved through random sampling of spaces and the incremental construction of feasible path representations (Elbanhawi and Simic, 2014; Orthey et al., 2024). These algorithms traditionally operate by generating samples from the state space. A pioneering contribution to this end was made by (Barraquand and Latombe, 1991), which leveraged Monte Carlo sampling to estimate the connectivity of obstacle-free configuration spaces. This approach laid the groundwork for subsequent techniques, such as the Probabilistic Roadmap Planner (PRM) (Sánchez and Latombe, 2003), which searches for navigation solutions in the space of graphs

constructed from randomly sampled configurations. Alternatively, methods like the Expansive-Space Trees (EST) (Hsu et al., 1997) and the Rapidly-exploring Random Tree (RRT) (LaValle, 1998; Kuffner and LaValle, 2000) use the random samples to incrementally grow a space-filling tree until it reaches an end goal. These methods can be optimized, for instance, by strategically biasing the probability of sampling from specific regions, as demonstrated by goal-oriented sampling methods (Kang et al., 2016). Recognized for their probabilistic completeness, these approaches also have their provably asymptotically optimal variants, such as PRM* and RRT* (Karaman and Frazzoli, 2011).

Our framework draws inspiration from research that shifts focus from state space sampling to trajectory space sampling (Kobilarov, 2012; Lee et al., 2018; Piché et al., 2019; Ratliff et al., 2009). In addition, the objective of our sampling framework extends beyond merely feasible solutions and aims to generate near-optimal trajectories. A notable work in this area takes inspiration from Monte Carlo optimization, and applies the cross-entropy method to importance sampling to guide the trajectory search towards optimal solution regions (Kobilarov, 2012). (Janson et al., 2015) present another innovative application of importance sampling, which integrates asymptotically optimal sampling-based motion planners with Monte Carlo methods with a focus on estimating collision probabilities under uncertainty.

A closer parallel to our methodology is formulated by (Lee et al., 2018), in that Monte Carlo sampling is used to generate a variety of optimized solutions to a navigation problem. Specifically, they extend the Metropolis Hastings algorithm to sample uniformly from the space of pareto-optimal trajectories, by defining a target distribution in terms of the Boltzmann distribution constrained to a Pareto frontier. In addition, our approach shares similarities with the framework introduced by (Piché et al., 2019), where planning is formulated as probabilistic inference over future near-optimal trajectories, and Sequential Monte Carlo is used to obtain multimodal policies. On the contrary, however, their method relies on a graphical model representation, which involves factorizing the posterior over trajectories using the forward-backward algorithm in a hidden Markov model.

2.1.2 (Stochastic) Trajectory Optimization

Our work also draws parallels to the optimization approaches to motion-planning problems. Given a measure of optimality (i.e., an objective or a cost function), these motion planning approaches perform numerical optimization in a high-dimensional parameterized trajectory space. Stochastic trajectory optimization techniques do this by injecting randomness into the process, as a way to trade accuracy for efficiency and escape local minima (Kochenderfer and Wheeler, 2019).

A foundational work in this space is CHOMP (Ratliff et al., 2009), which combines functional gradient techniques with Monte Carlo sampling as a way to avoid convergence to high-cost local minima and ensure probabilistic completeness. In particular, CHOMP optimizes trajectories based on an objective function that balances trajectory smoothness with obstacle avoidance. The locally optimal trajectories are then perturbed using Hamiltonian Monte Carlo, a sampling technique that leverages first-order gradients of the target distribution over trajectories.

Related subsequent works include STOMP (Kalakrishnan et al., 2011) which uses a gradient-free stochastic optimization method, and ITOMP (Park et al., 2012) which combines optimization with real-time re-planning to account for dynamic obstacles. Furthermore, Trajopt (Schulman et al., 2013) incorporates second-order derivative information into the trajectory optimization by leveraging sequential quadratic programming. Increasingly, optimization-based methods are interleaved with sampling-based approaches, for the purposes of generating good initial solutions and balancing global exploration with local optimization (Li et al., 2016; Kuntz et al., 2016).

Our method can be regarded as stochastic trajectory optimization, as the target distribution guiding our trajectory samples includes a cost function. Drawing inspiration from (Ratliff et al., 2009), we explore cost function formulations that effectively balance trajectory smoothness with obstacle avoidance. Similarly to (Schulman et al., 2013), we use second-order derivative information to steer our proposal generation. However, our method importantly diverges from the aforementioned ones by seeking multiple near-optimal navigation solutions, rather than a single optimal trajectory.

2.2 Trajectory Prediction and Goal Inference

For autonomous agents to collaborate fluidly with humans and other intelligents agents in physical spaces, they need to rapidly reason about others from low-level observations of their motion (Liu et al., 2018; Dragan, 2015; Wang et al., 2019). The tasks of motion prediction and goal inference have been tackled through a variety of approaches (Sukthankar et al., 2014), including deep learning (Min et al., 2014; Rabinowitz et al., 2018; Alahi et al., 2016; Ivanovic et al., 2020), inverse reinforcement learning (Ramachandran and Amir, 2007; Ziebart et al., 2008; Hadfield-Menell et al., 2016b), plan recognition (Ramırez and Geffner, 2009; Ramírez and Geffner, 2010; Kaminka et al., 2018), and Bayesian inverse planning (Baker et al., 2009; Albrecht et al., 2021; Zhi-Xuan et al., 2020; Alanqary et al., 2021; Albrecht et al., 2021; Zhi-Xuan et al., 2022).

Many of these approaches build upon the insight that intelligent agents act efficiently to achieve their goals, as formalized by the principles of rationality (Gergely and Csibra, 2003) and least effort (Zipf, 2016). Such satisficing behavior is commonly modeled by Boltzmann rationality, although other models such as bounded rationality have also been considered (Zhi-Xuan et al., 2020; Alanqary et al., 2021; Zhi-Xuan et al., 2022). According to the Boltzmann-rational model, the probability of an agent acting out a trajectory is inversely proportional to the costs associated with those trajectories. Under this assumption, a goal is more likely if the observed behavior matches an efficient plan that achieves that goal, as indicated by Bayesian inference (Baker and Tenenbaum, 2014). More broadly, inferring latent plans and goals from observations is closely related to reasoning about other's mental states as formalized by the theory of mind frameworks (Baker et al., 2017; Baker and Saxe, 2011; Scassellati, 2002)

In our framework, the observed agent is modeled as a Boltzmann-rational motion planner. Building upon work in legible motion planning (Dragan, 2015) and probabilistic programming (Cusumano-Towner et al., 2017; Seaman et al., 2018), we develop a Bayesian model of the agent, with which we can condition on observed trajectory

segments to infer the agent's underlying goal-driven motion plan. Unlike most goal inference approaches which are typically restricted to discretized (Ziebart et al., 2009) or symbolic domains (Ramírez and Geffner, 2010; Massardi et al., 2021), our model allows us to handle continuous environments and observations. In addition, in contrast to the previous approaches (Kaminka et al., 2018; Cusumano-Towner et al., 2017; Seaman et al., 2018), we allow for directly capturing the relationship that lower-cost paths are more probable, while also employing stochastic trajectory optimization to generate more plausible predictions about the underlying motion plans.

Chapter 3

Markov Chain Monte Carlo Sampling for Boltzmann-Rational Motion Planning

In this chapter, we explore Markov Chain Monte Carlo (MCMC) sampling algorithms as a method for obtaining a variety of near-optimal paths in obstacle-ridden environments. We begin by formalizing the problem of motion planning through the lens of trajectory optimization: in Section 3.1, we define a cost function that guides the path planning process and provide a parameterization of trajectories that allows for numerical optimization. In Section 3.2, we approach trajectory optimization from a probabilistic inference standpoint: we model path planning as a stochastic process and outline several MCMC algorithms as a way to guide trajectory samples towards optimal regions. The chapter concludes with extensive experiments in Section 3.3, where we examine the impact of various cost function formulations and then demonstrate the effectiveness of MCMC methods in motion planning.

3.1 Motion Planning Formalization and Trajectory Parameterization

We consider a trajectory γ to be a smooth function mapping time to agent configurations $\gamma : [0, 1] \to \mathcal{C} \subset \mathbb{R}^d$. Analogous to the approach in (Ratliff et al., 2009), we aim for the trajectories to conform to two distinct yet related aspects of motion planning: obstacle avoidance and smoothness of motion. Accordingly, we can optimize the trajectory function with respect to an objective that combines the obstacle avoidance and the smoothness criteria. In particular, we formulate a cost function $C(\gamma)$ that maps a trajectory γ in a space of trajectories Γ to a real number, which is a sum of the obstacle avoidance and the smoothness terms:

$$C(\gamma) = f_{obs}(\gamma) + \alpha f_{smooth}(\gamma), \tag{3.1}$$

weighted by a safety-smoothness trade-off hyperparameter, α. We define an obstacle avoidance term, f_{obs}, as the cumulative obstacle cost, $c : \mathbb{R}^d \to \mathbb{R}$, acquired throughout the trajectory:

$$f_{obs}(\gamma) = \int_0^1 c(\gamma(t))dt, \tag{3.2}$$

in order to encourage trajectories that maintain a safe distance away from obstacles. Furthermore, we define a smoothness term, f_{smooth}, in terms of dynamical properties of the trajectory. In particular, we consider f_{smooth} as a measure of the overall variation in speed over the course of the trajectory:

$$f_{smooth}(\gamma) = \int_0^1 \| \frac{d\gamma(t)}{dt} \|_2^2 dt. \tag{3.3}$$

This smoothness measure is notably invariant to the global speed of $\gamma(t)$, since the trajectory time is normalized between 0 and 1. To allow for numerically performing functional gradient descent on eq. (3.1), we opt for a uniform discretization that samples the trajectory function at consistent time intervals. In particular, we parameterize γ with a series of waypoints $\{x_i\}_{i=0}^T$, such that a continuous trajectory

is linearly interpolated between the waypoints, i.e.:

$$\gamma(t) = x_{i-1} + \frac{t - t_{i-1}}{\Delta t}(x_i - x_{i-1}), \tag{3.4}$$

where $t \in [t_{i-1}, t_i]$, and $\Delta t \coloneqq t_i - t_{i-1}$. Furthermore, we abstract the agent body to a point representation, so $x_i \in \mathcal{C} \subset \mathbb{R}^2$. This parameterization allows us to approximate the obstacle avoidance term in eq. (3.2) by the following discretization:

$$\begin{aligned} f_{obs}(\gamma) &= \int_0^1 c(\gamma(t))dt, \\ &\approx \frac{1}{T}\sum_{i=0}^{T} c\left(\gamma\left(\frac{i}{T}\right)\right), \\ &= \frac{1}{T}\sum_{i=0}^{T} c(x_i) \end{aligned} \tag{3.5}$$

In addition, it allows us to express the smoothness criterion in eq. (3.3) in terms of the Euclidean distances between consecutive waypoints, i.e., the displacement for each segment of the trajectory:

$$\begin{aligned} f_{smooth}(\gamma) &= \int_0^1 \|\gamma'(t)\|_2^2 dt, \\ &= \sum_{i=0}^{T} \int_{\frac{i}{T}}^{\frac{i+1}{T}} \left\| \frac{d\gamma(t)}{dt} \right\|_2^2 dt, \\ &= \sum_{i=0}^{T} \int_{\frac{i}{T}}^{\frac{i+1}{T}} \left\| \gamma\left(\frac{i+1}{T}\right) - \gamma\left(\frac{i}{T}\right) \right\|_2^2 \frac{1}{T} dt, \\ &= \frac{1}{T^2}\sum_{i=0}^{T} \|x_{i+1} - x_i\|_2^2. \end{aligned} \tag{3.6}$$

3.2 Near-Optimal Trajectories As Markov Chain Monte Carlo Samples

In Section 3.1, we defined a cost function akin to that in (Ratliff et al., 2009), which serves as the basis for optimizing trajectories. Drawing upon the insight that planning can formulated as inference (Botvinick and Toussaint, 2012; Levine, 2018; Piché et al., 2019), we now approach trajectory optimization from a probabilistic inference perspective, treating trajectories as outcomes of a stochastic process. This allows us to explore a multitude of possible trajectories instead of generating single optimal solution, thus accounting for multiple valid strategies to a planning problem (such as, for example, navigating around either side of an obstacle). We represent continuous-time trajectories as samples obtained from a distribution that favors trajectories with low cost. In particular, we use the Boltzmann distribution (also referred to as the maximum entropy distribution (Ziebart et al., 2008)), a common model for approximately rational decision making (Ziebart et al., 2009; Dragan, 2015), to obtain near-optimal, low-cost trajectories. Given a start and an end point, the probability of a trajectory is modelled as:

$$\pi(x_{0:T}|x_0, x_T) = \frac{1}{Z(x_0, x_T)} \exp[-\beta C(x_{0:T})], \tag{3.7}$$

where $Z(\cdot, \cdot)$ is an endpoint-dependent normalizing constant, and the cost function is as defined in eq. (3.1). Furthermore, β represents the rationality parameter used to influence the degree of stochasticity in trajectories: when $\beta = 0$ the sampled waypoints represent a random walk, and $\beta = \infty$ corresponds to pure deterministic optimization.

Importantly, evaluating the probability of a trajectory $x_{0:T}$ under the Boltzmann distribution π requires computing the normalizing constant $Z(x_0, x_T)$, which is intractable in continuous domains. We instead assume that motion plans $x_{0:T}$ are sampled according to a *Monte Carlo approximation* $\hat{\pi}$ of the true Boltzmann distribution π. In particular, we implement four Markov Chain Monte Carlo (MCMC) sampling algorithms to generate trajectory samples that approximate π: the generic Metropolis-Hastings (MH)

algorithm, Unadjusted Langevin Algorithm (ULA), Metropolis-Adjusted Langevin Algorithm (MALA), and Hamiltonian Monte-Carlo (HMC). We present the Metropolis-Hastings algorithm in Algorithm 1, and use it as a foundation for developing the rest of the MCMC algorithms.

Algorithm 1 Metropolis-Hastings Algorithm

procedure METROPOLIS-HASTINGS(x, $\pi(x)$, $g(x'|x)$, N)
  for $i \in [1, N]$ **do**
    $x' \sim g(x'|x)$ ▷ Sample new state x' from proposal distribution
    $u \sim \text{Unif}([0, 1])$ ▷ Sample u from uniform distribution
    $A \leftarrow \min\left(1, \frac{\pi(x')g(x|x')}{\pi(x)g(x'|x)}\right)$ ▷ Acceptance probability
    if $u \leq A$ **then**
      $x \leftarrow x'$ ▷ Accept new state
    end if
  end for
  return x ▷ Return final state
end procedure

The Metropolis-Hastings algorithm produces a Markov chain that asymptotically reaches the desired distribution – in our case, the Boltzmann distribution from eq. (3.7) – as its stationary (i.e., equilibrium) distribution. The Markov chain is uniquely defined by its state transition probabilities $p(x'|x)$, which are the product of the proposal distribution $g(x'|x)$ and the acceptance distribution $A(x', x)$. During the algorithm's execution, a subsequent sample x' is drawn from $g(x'|x)$ based on the current sample x, and $A(x', x)$ determines whether the proposed move to x' is accepted or the algorithm remains at x.

The choice of the proposal distribution influences the algorithm's ability to properly sample the entire state space. A narrowly focused g may lead to high acceptance, but the chain will fail to visit all regions of the state space with the proper frequency – resulting in a sample that is biased with respect to the target distribution. Conversely, a too broad g may suffer from low acceptance rates and inefficient exploration. To ensure that every state will be visited sufficiently often given a large number of steps, g must allow for the Markov chain to be ergodic, meaning that any state can be

reached from any other state in a finite number of steps, i.e., that there exists $n > 0$ such that the n-step transition probability from any initial state x to any state x' satisfies $p^n(x'|x) > 0$. A typical choice for g is the Gaussian distribution centered at x, i.e., $g(x'|x) = N(x, \epsilon I)$, which effectively transforms the sampling process into a random walk where ϵ controls the step size.

In addition to the ergodicity property, the Markov Chain produced by the Metropolis-Hastings algorithm also satisfies the detailed balance condition, which ensures that each state transition is reversible. This condition is a property of equilibrium in Markov chains, where for any two states x and x', the rate of transition $x \to x'$ is equal to the rate of transition $x' \to x$ when the system is at equilibrium. More formally, for the target distribution π, the state transition probabilities p, and the proposal distribution g, we can express the detailed balance condition as $\pi(x)p(x'|x) = \pi(x')p(x|'x)$. The acceptance probability $A(x', x) \coloneqq \min\left(1, \frac{\pi(x')g(x|x')}{\pi(x)g(x'|x)}\right)$ is derived from the detailed balance condition.

The detailed balance condition together with the ergodicity property facilitate convergence to the target distribution by respectively ensuring that there exists a stationary distribution and that it is unique. However, convergence to the target distribution can be slow in practice, especially in high dimensional settings where random proposals are less likely to maintain the target density π. To make more informed proposals for the next state of the Markov Chain (i.e., the next trajectory waypoint), we can follow the steepest ascent in the target density function by leveraging its gradient information. The remainder of the sampling algorithms we consider are all first-order MCMC algorithms.

The Unadjusted Langevin Algorithm (ULA) is based on the principles of Langevin dynamics (Langevin, 1908; Wright and Ma, 2022; Wu and Brooks, 2003; Hyvärinen and Dayan, 2005), a stochastic process used to describe the motion of particles subject to both deterministic forces and random fluctuations. A discretized version of the continuous Langevin equation is used as an ULA iteration step, $x_{t+1} \leftarrow x_t + \epsilon\nabla\pi(x_t) + \sqrt{2\epsilon}z$, where x_t represents the current state of the system (the position of the particle), $\nabla\pi(x_t)$ is the gradient of the target distribution at x_t (analogous

to the deterministic force), ϵ is a step size parameter, and $z \sim N(0, I)$ represents the random fluctuation due to noise. ULA can be interpreted as sampling from the proposal $g(x'|x) = N(x + \epsilon\nabla\pi(x), \epsilon I)$. Importantly, while an appropriate choice of ϵ can lead to a convergence to the target distribution, ULA does not correct for the errors introduced during the discretization of the Langevin dynamics, which can introduce bias to the trajectory samples.

To ensure that the samples asymptotically follow the target distribution despite the discretization errors, Metropolis-Adjusted Langevin Algorithm (MALA) adds the Metropolis acceptance step back to ULA, i.e., it follows the form of Algorithm 1 with the proposal distribution g of ULA (Rossky et al., 1978). While the acceptance step makes MALA slower per iteration compared to ULA, it may be more efficient overall if ULA requires many more iterations to achieve a comparable level of accuracy in approximating the target distribution.

Finally, we implement the Hamiltonian Monte Carlo (HMC) algorithm, which is also used for motion planning in (Ratliff et al., 2009). HMC is considered to be among the state-of-the-art MCMC algorithms due to the way it adapts physics-inspired methods to streamline exploration of the state space. In each iteration, the algorithm augments the state space with a momentum vector $p \sim N(0, I)$ in order to simulate a trajectory (x_t, p_t) through the phase space, the coordinate system of Hamiltonian mechanics. A discretized version of this trajectory is simulated using the Leapfrog integrator (Birdsall and Langdon, 2018; Yoshida, 1990), which updates x_t and p_t while approximately conserving the Hamiltonian, $H(x, p)$, representing the total energy of the system including the potential energy (a function of the target distribution) and the kinetic energy (a function of the momentum): $H(x, p) \coloneqq -log\pi(x) - \frac{1}{2}||p||^2$. After L steps of the simulated trajectory, an acceptance step is performed to correct for any discretization errors introduced by the Leapfrog integrator. In essence, HMC can be viewed as an extension of Algorithm 1 where the random walk proposals are replaced with proposals generated via a simulation of Hamiltonian mechanics, where the auxiliary momentum variable allows for larger moves in the state space that are still likely to be accepted. In addition, HMC replaces the Metropolis acceptance

probability with an acceptance probability that is a function of the Hamiltonian, $A(x', x) \coloneqq min\,(1, -H(x', p') + H(x, p))$.

3.3 Experiments

We employ sampling-based trajectory optimization with the primary goal of obtaining *a collection* of cost-efficient paths that represent a variety of valid motion plans. The obtained trajectories are a result of a complex interplay of numerous factors, including the inherent characteristics of the motion planning objective discussed in Section 3.1, and the statistical properties of the sampling techniques described in Section 3.2. To methodically examine the impact of these various design choices, we conduct an ablation study focusing on the following items:

1. We investigate the effects of the cost function $C(\gamma)$ presented in eq. (3.1) on trajectory optimization. We begin by examining the sensitivity of the generated trajectories to the safety-smoothness trade-off hyperparameter, α, in $C(\gamma)$. Subsequently, we explore how the formalization of obstacle avoidance influences the runtimes and convergence rates of the optimization framework.

2. Upon establishing a cost function and an obstacle avoidance method that allow for efficient and stable convergence, we evaluate the effectiveness of the sampling algorithms from Section 3.2 in generating desirable motion plans. In Section 3.3.2, we perform a visual analysis of the generated trajectories and investigate the computational complexity associated with the different sampling algorithms. In addition, we propose a criterion for quantitatively assessing the diversity of the generated motion plans and accordingly evaluate a selected trajectory sampling algorithm.

We perform the ablation study within a continuous two-dimensional motion planning domain, aiming to generate feasible trajectories that circumvent any obstructing obstacles. For our experimental setup, we employ Pymunk — a Python library dedicated to simulating 2D rigid body physics (Pymunk contributors, 2023). We model

the planning agent as a circle, characterized by a center $x_r \in \mathbb{R}^2$ and a radius $r > 0$. The agent navigates through a defined domain $D = \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1 \geq 0, x_2 \geq 0\}$, which is situated within the first quadrant of the Cartesian plane. Within this domain, we introduce obstacles in the form of squares, each defined by a center $o_i \in \mathbb{R}^2$ and a side length $s > 0$. The agent's movement is directed through velocity commands: for a generated trajectory γ, we construct a control input $u(t) = \frac{d}{dt}\gamma(t)$, enabling the agent to follow the planned path γ provided there are no impediments posed by obstacles. As described in Section 3.1, trajectories are parameterized as sets of waypoints, $\{x_i\}_{i=0}^{T}$. Given a fixed set of a start and a goal position, $x_0, x_T \in \mathbb{R}^2$, a continuous trajectory $\gamma(t) : [0, 1] \to \mathbb{R}^2$ is a linear interpolation between all the waypoints.

3.3.1 A Comparative Analysis of Obstacle Avoidance Methods

In Section 3.1, we defined the trajectory optimization problem with respect to a cost function that encourages trajectory smoothness and obstacle avoidance, as formulated by f_{smooth} and f_{obs} terms, respectively. Taking f_{smooth} as defined in eq. (3.6), we now investigate different formulations of the obstacle cost function in f_{obs}, as defined in eq. (3.5), and their effects on trajectory optimization. In the remainder of this section, we use deterministic optimization techniques as opposed to the stochastic, i.e., sampling-based methods described in Section 3.2, in order to be able to adequately capture the nuanced interplay between the cost components and the resulting trajectory.

Taking inspiration from the literature on control barrier functions (Nagumo, 1942; Ames et al., 2019, 2016), which are often defined in terms of the distance to the boundary of an unsafe region, we first consider an obstacle cost function in terms of the L^2 distance from the planning agent's position to the center of an obstacle:

$$c(x_i) = \sum_j -\|x_i - o_j\|_2^2, \tag{3.8}$$

where o_j represents the center position of an instance of an obstacle in the environment. Accordingly, f_{obs} is then formulated as $f_{obs}(\gamma) = \frac{1}{T}\sum_{i=0}^{T}\sum_j \|x_i - o_j\|_2^2$, as given by the eq. (3.5). We then generate trajectories by linearly interpolating between a sequence

of waypoints computed using vanilla gradient descent.

Among the generated trajectories, we observe a strong dependency on α, the safety-smoothness trade-off hyperparameter indicating the relative contributions of f_{smooth} compared to that of f_{obs} to the overall trajectory cost. In particular, for an obstacle cost function as defined in eq. (3.8), we observe that a low value of α leads trajectory waypoints far away from the obstacles and, ultimately, in the entirely opposite direction off the course. In addition, in spite of overfitting to f_{obs}, the final trajectory still intersects with obstacles as a result of the linear interpolation between the goal state and the waypoints generated on the opposite end of obstacles. On the other hand, for a high value of α, we observe an overfitting to the smoothness criterion. The generated waypoints reflect the shortest path between the start and the goal position, and the resulting trajectories turn out to be ignorant to the obstacle placement. We illustrate the findings in Figure 3-1, where the start position of the planning agent is depicted as the blue square, the goal state is depicted as the green square, the obstacles are depicted as red squares, and the red lines represent the linearly interpolated trajectories.

Importantly, we observe that very subtle variations in α lead to significant changes in the generated trajectories, suggesting that there is a complex interplay between path efficiency and obstacle avoidance – at least for the given choice of obstacle cost function. In response to this challenge, we explore two alternative formulations of the obstacle cost function: a modified L^2 distance from obstacles with a cutoff, and a Gaussian kernel approach.

For the L^2 norm with a cutoff, we take inspiration from (Ratliff et al., 2009) and formulate the obstacle cost function as follows:

$$c(x_i) = \sum_j \max\{0, r_j^2 - \|x_i - o_j\|_2^2\}, \tag{3.9}$$

where r_j represents the radius of influence around an obstacle j the center of which is positioned at o_j. This radius defines a circular region around the obstacle outside of which the obstacle will have no effect on the cost. This formulation of the obstacle

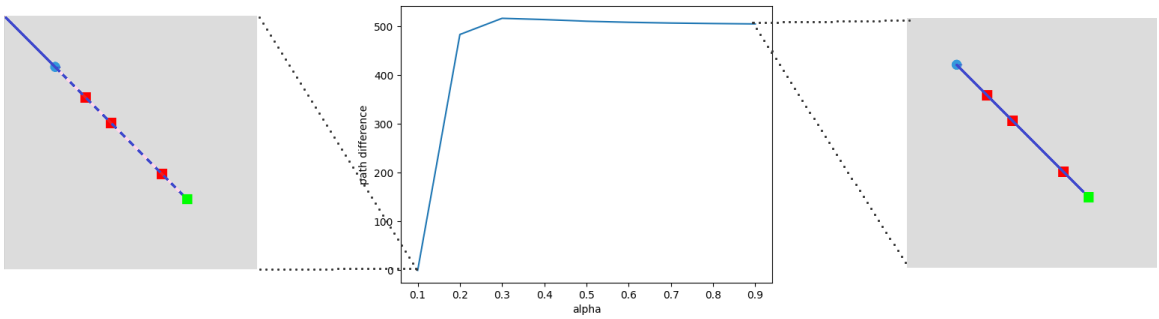


Figure 3-1: The impact of the safety-smoothness trade-off hyperparameter, α, on the generated trajectories. Plotted is the squared norm distance between the trajectory generated for $\alpha = 0.1$ and the trajectories generated at other values of α. The trajectories are generated with an obstacle avoidance method in the form of a cost function, f_{obs}, based on the L^2 distance from object centers. Start position, end goal, and obstacles are depicted in blue, green, and red, respectively.
We find that a low α value results in the motion planning objective, i.e., $C(\gamma)$, getting excessively influenced by obstacle avoidance, i.e., the f_{obs} term. This manifests with the waypoints getting generated excessively far away from obstacles, which, in the extreme, as depicted on the left, leads to a path that goes in a direction away from the obstacle-secluded goal (note that the trajectories are linearly interpolated between the waypoints which include the fixed end point, so this results in a linear interpolation through the obstacles).
Conversely, a high α value results in the motion planning objective getting dominated by the smoothness objective, i.e., f_{smooth}. In the extreme, as depicted on the right, this results in a trajectory that disregards obstacles and opts for the shortest, direct path towards the goal.
The sharp phase transition between the two extremes of overfitting, as indicated by the central plot, illustrates the practical challenges of tuning the safety-smoothness trade-off hyperparameter – especially with the L^2 distance-based obstacle avoidance.

cost function follows a rectified formulation, which can help simplify computations and enhance the predictability of the resulting trajectories in scenarios requiring clear demarcation between safe and unsafe areas.

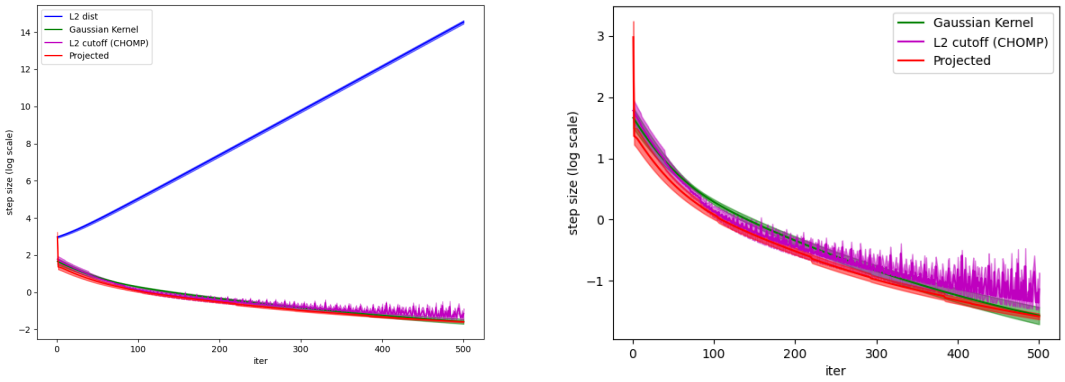
In addition, we consider an obstacle cost function in the form of a Gaussian kernel, noted for its effectiveness in hyperparameter tuning for SVM classifiers (Keerthi and Lin, 2003) and its previous applications to control barrier functions (Khan and Chatterjee, 2020). Given a bandwidth parameter, η, controlling the width of the Gaussian bell curve, we define the obstacle cost function as:

$$c(x_i) = \sum_j e^{-\eta \|x_i - o_j\|_2^2}. \tag{3.10}$$

This formulation allows for a smooth, exponentially decaying cost gradient that can lead to smoother trajectories and more stable convergence. In addition, with η we can control for how localized the effect of an obstacle is on the cost function.

Lastly, in addition to the above defined obstacle cost functions, we also consider a projection-based method for motion planning. This method can potentially simplify the motion planning process, as the obstacle avoidance term f_{obs} and its corresponding obstacle cost function are reduced to zero, and trajectory generation relies solely on the smoothness term, f_{smooth}, in the cost function from eq. (3.1). In order to enforce separation from obstacles while prioritizing for trajectory smoothness, we project individual waypoints x_i onto a safe zone located outside a predetermined safety margin — a fixed radius r — from the centers of the obstacles.

Overall, we consider four approaches to obstacle avoidance: the three obstacle cost function formulations for f_{obs} – the L^2 distance from obstacles, the L^2 distance with a cutoff, as inspired by (Ratliff et al., 2009), and the Gaussian kernel approach – in addition to the projection method that sets f_{obs} to zero. We evaluate these four approaches for their convergence rates within a motion planning framework based on vanilla gradient descent. As illustrated by Figure 3-2, we observe that the step sizes for each method tend to converge similarly over the iterations, with the notable exception of the L^2 distance-based obstacle cost, which diverges from this pattern as



(a) (b)

Figure 3-2: A comparative analysis of the obstacle avoidance methods focusing on the convergence of the trajectory optimization via vanilla gradient descent. Note that the plots depict the step size on a logarithmic scale.
The obstacle avoidance method in the form of a cost function based on the L^2 distance from the obstacle results in the maximization of a non-upper-bounded function. Consequently, we find that this obstacle avoidance method leads to an exponential increase in step size, as depicted in (a).
In (b), we focus the comparison on the three obstacle avoidance methods that exhibit convergence. These include a cost function in the form of a rectified alternative of the L^2 distance derived from (Ratliff et al., 2009), a cost function in the form of a Gaussian kernel, and a method that disregards the obstacle cost function and instead projects the generated waypoints into a safe region. We observe a notable stability improvement with the Gaussian kernel method and the projection-based method.

a result of being an unbounded function.

In addition to the convergence rates, we also contrast the four methods for obstacle avoidance based on their respective compute times. We conduct experiments using JAX on an Nvidia GeForce 1060 GPU, and present results in Figure 3-3. Despite its convergence rates being on par with those of the L^2 cutoff and Gaussian kernel methods, we find that the projection-based method stands out in terms of computational efficiency, requiring significantly less time to achieve convergence.

Using the projection-based method to ensure obstacle avoidance in the generated trajectories not only maximizes computational efficiency, but also uniquely circumvents sensitivity to the safety-smoothness trade-off hyperparameter, α, by eliminating the obstacle avoidance cost term (i.e., $f_{obs} = 0$). Owing to its engineering robustness, the

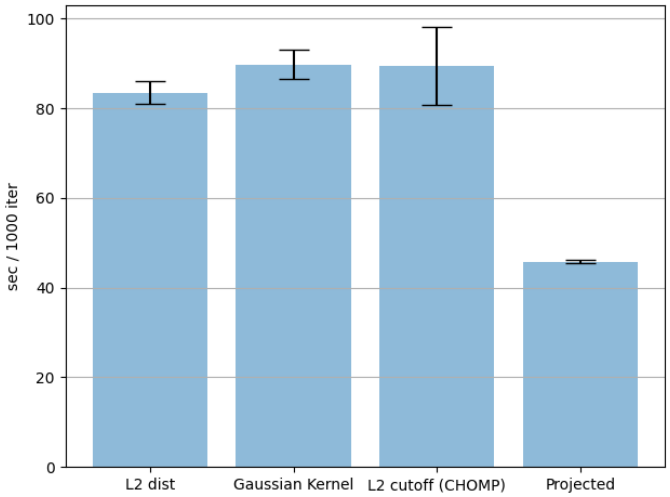


Figure 3-3: A comparative analysis of the obstacle avoidance methods focusing on the computation time required for trajectory optimization using vanilla gradient descent. The performance metric is quantified in terms of the time taken, reported in seconds, to complete 1000 iterations of gradient descent. Note that the plot illustrates the respective means and standard deviations of the four methods.
Since the projection-based method for obstacle avoidance circumvents the need for differentiation through an obstacle cost function, we observe a notable increase in computation speed as expected.

subsequent experiments all use the projection-based method for obstacle avoidance.

3.3.2 Performance Analysis of Markov Chain Monte Carlo Methods in Motion Planning

In Section 3.3.1, we generate trajectories using pure deterministic optimization in order to find the optimal cost function formulation, i.e., the optimal obstacle avoidance method. In this section, we use the projection-based method for obstacle avoidance, and generate trajectories using the sampling algorithms described in Section 3.2. We use Boltzmann distribution as the target distribution, and define the cost purely in terms of trajectory smoothness (i.e., the minimal squared path norm, as defined in eq. (3.6)).

First, we evaluate the time complexity for each of the three MCMC sampling algorithms from Section 3.2: the zero-th order method – the Metropolis-Hastings (MH) algorithm – which relies on random proposals, and the two first-order methods

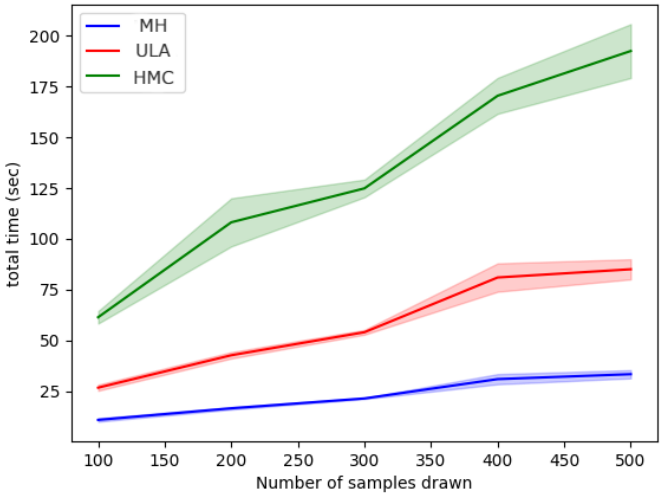
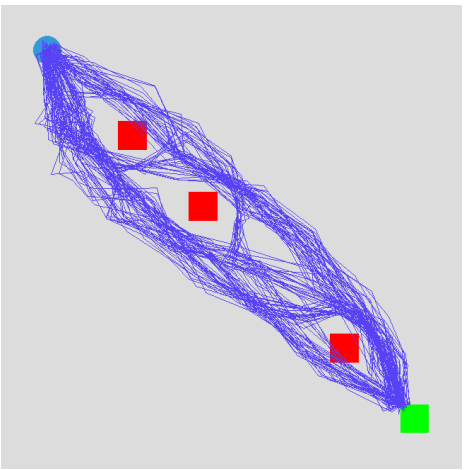


Figure 3-4: Runtime comparison among three Markov Chain Monte Carlo (MCMC) algorithms used for sampling trajectories: Metropolis-Hastings (MH), Unadjusted Langevin Algorithm (ULA), and Hamiltonian Monte Carlo (HMC).
As expected, the methods that leverage the gradient information of the target distribution, ULA and HMC, require more time to draw a specified number of samples compared to the zero-th order method, MH. HMC particularly shows a more pronounced increase in runtime due to the computational demands of the Leapfrog integrator. In the conducted experiments, the Leapfrog integrator within the HMC algorithm is configured to perform $L = 5$ steps per iteration. This implies that for every single iteration performed by ULA, the HMC algorithm executes five internal Leapfrog integration steps.

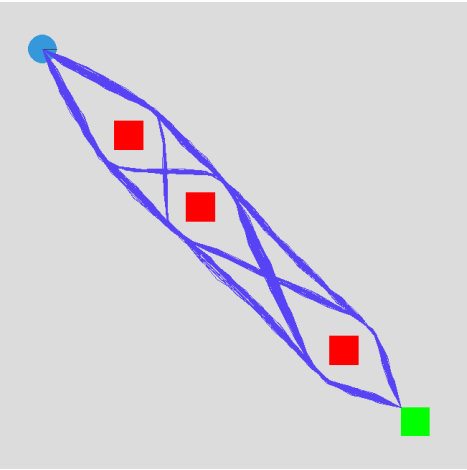
– the Unadjusted Langevin Algorithm (ULA), and the Hamiltonian Monte Carlo (HMC) algorithm – which leverage the first-order gradient information of the target distribution. In Figure 3-4, we show how the runtime of each of the three algorithms scales as a function of the sample sizes. As expected, the MH algorithm exhibits the lowest runtime, which increases linearly as more samples are drawn. The ULA shows a higher runtime compared to MH, with a steeper increase. Interestingly, we observe that ULA (which leverages the target distribution gradients) maintains a runtime that is about 3x that of MH (which doesn't use gradient information), which is consistent with automatic differentiation guarantees that gradients can be computed in at most 3 times the FLOPs needed to compute the function value itself (Bartholomew-Biggs et al., 2000). Finally, the HMC exhibits the highest runtime of all of the three methods.

It starts off close to ULA but then its runtime grows more rapidly as a result of the Leapfrog integration; for each step that ULA is taking HMC is internally integrating for L Leapfrog steps (in our experiments, $L = 5$).

Next, we qualitatively compare the trajectories generated using zeroth-order MCMC sampling with those generated by first-order MCMC sampling that incorporates gradient information. In Figure 3.3.2, we contrast set of trajectories generated using MH, in (a), with trajectories generated using ULA, in (b).



(a) Metropolis-Hastings (0th order)



(b) Unadjusted Langevin Algorithm (1st order)

Figure 3-5: A visual analysis of MCMC trajectory sampling methods. Start position, end goal, and obstacles are depicted in blue, green, and red, respectively.
In (a) we visualize the trajectories sampled via the Metropolis-Hastings algorithm, which explores the solution space via random proposals. While we observe a variety of valid paths, many of them exhibit indirect routes including several intersections with obstacles (note that the generated waypoints don't intersect the obstacles due to the projection-based obstacle avoidance, but the resulting linear interpolations do).
In (b) we visualize trajectories obtained using the Unadjusted Langevin Algorithm, which incorporates the first-order gradients of the target distribution to guide sampling. This results in a more focused exploration, with trajectories coherently distributed around the obstacles, demonstrating the algorithm's effectiveness in adhering closely to the optimal paths suggested by the Boltzmann distribution.
Consequently, in the subsequent investigation, we focus on the gradient-based MCMC methods, indicative of an enhanced optimization process in trajectory planning.

We find that even the simplest algorithm, the MH, is sufficient to obtain a variety of valid, collision-free trajectories. Moreover, the trajectories sampled with MH exhibit

greater diversity in the way they traverse the environment compared to ULA. However, there are also a number of MH trajectories indicative of greater path norms (and, therefore, greater costs) as a result of less smooth paths characterized by zigzagging in open spaces, as well as distant waypoints. In some cases, the distant waypoints compromise collision avoidance as indicated by the paths that intersect with the first and the last obstacle. In contrast, the ULA leverages the gradient information of the target distribution to optimize the cost function, which results in trajectories that are better optimized for minimizing the path norm and ensuring smoothness. Importantly, while the ULA generates more optimal trajectories, it still allows for a diversity of motion plans, as evidenced by the samples that cover every viable route through the obstacle field.

For subsequent experiments, we concentrate on the Langevin-based MCMC algorithms for motion planning, given the quality of the produced trajectories and their favorable computational cost, particularly compared to other first-order algorithms. Lastly, we introduce a formal metric for characterizing the diversity of trajectories, and we use it to evaluate the trajectories sampled with ULA.

Trajectory diversity is a topic studied in a variety of domains, including navigation in uncertain environments (Branicky et al., 2008), multi-agent coordination (Lupu et al., 2021), and traffic forecasting (Lupu et al., 2021). Various formulations of measures for trajectory diversity have been considered. (Branicky et al., 2008) defines path diversity as the probability that there exists at least one path in a set of paths over all possible environments that is not blocked by obstacles, (Ma et al., 2020) characterizes diversity based on a distance between alternative trajectories, namely the spatial separation between trajectory endpoints, and (Lupu et al., 2021) considers the similarity between the underlying probability distributions characterized using Jensen-Shannon (JS) divergence, a symmetric and smoothed version of the Kullback-Leibler (KL) divergence.

For our setting, we also introduce a probabilistic measure, leveraging our use of stochastic optimization, i.e., the MCMC sampling algorithms for motion planning. In particular, we formulate trajectory diversity as the variance of the distribution used

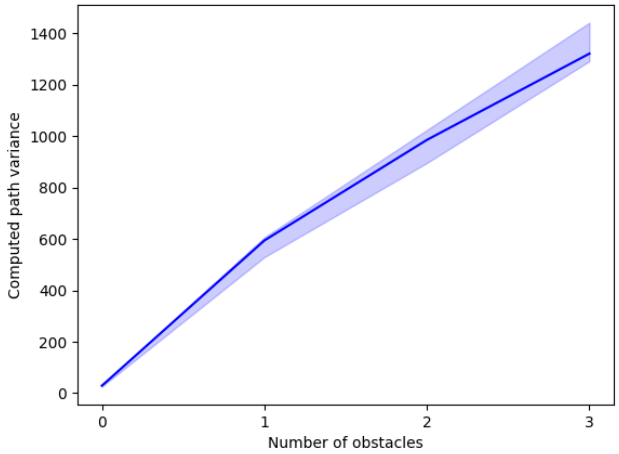


Figure 3-6: The correlation between trajectory variance, derived from the target Boltzmann distribution, and the number of obstacles in the environment. Trajectories are sampled with ULA, a gradient-based MCMC algorithm, with varying Boltzmann rationality parameter values, β, set to $\{10^{-5}, 10^{-1}, 10^{4}\}$ at each obstacle count $N = \{1, 2, 3\}$. The trajectory variance increases in a linear fashion with the number of obstacles, indicating that this quantity may serve as an effective metric for assessing the complexity of a motion planning problem. Moreover, given that the number of obstacles likely affects the variety of possible avoidance strategies, the trajectory variance can also be interpreted as a reflection of the motion plan diversity.

to sample trajectory waypoints:

$$\mathbb{V}(P) = \mathop{\mathbb{E}}_{\gamma \sim P}[d^2(\gamma, \bar{\gamma})], \tag{3.11}$$

where $P \sim e^{-\beta C(\gamma)}$ is the Boltzmann distribution as defined in eq. (3.7) with a trajectory cost formulated in terms of smoothness, i.e., the path norm in eq. (3.6), projected outside of a radius from the obstacles. Furthermore, the term $\bar{\gamma}$ represents the expected value of the trajectory under the distribution P, $\bar{\gamma} = \mathbb{E}_P[\gamma]$, and the squared path metric $d^2(\cdot, \cdot)$ is derived from the L^2 norm: $d^2(\gamma_1, \gamma_2) = \int_0^1 \|\gamma_1(t) - \gamma_2(t)\|_2^2 dt$.

In Figure 3.3.2, we show how the our trajectory diversity metric scales with the complexity of the environment, determined by the number of included obstacles. We observe that the variance measure increases as we increase the number of obstacles. This trend suggests that our variance-based metric for trajectory diversity, though derived from heuristic considerations, empirically aligns with our expectations for

such a measure. In addition, we can conclude that sampling with ULA, using the Boltzmann distribution as the target distribution, and trajectory smoothness as the cost, we can generate valid, obstacle-free trajectories for various degrees of difficulty of the 2D motion planning problem.

Chapter 4

Sequential Monte Carlo Sampling for Multimodal Trajectory Prediction and Online Goal Inference

By framing the motion planning problem within probabilistic inference, we can both generate diverse sets of motion plans for the planning agent and infer the latent states that the agent might possibly pursue in the future. In other words, by leveraging the Monte Carlo methods, we can engage in both *forward planning*, i.e., generating a complete trajectory given a goal state, and *inverse planning*, i.e., estimating the underlying motion plans and goal states from observed trajectory segments. In Chapter 3, we focused on Monte Carlo methods for forward planning and presented a framework for generating a diverse set of near-optimal trajectories using MCMC sampling. In this chapter[1], our focus shifts to Monte Carlo methods for inverse planning. In Section 4.1, we develop a Bayesian model of a goal-directed planning agent. In Section 4.2, we extend the MCMC methods for motion planning from Chapter 3 to plausibly simulate the agent's latent motion plans. Finally, in Section 4.3, we present a Sequential Monte Carlo (SMC) framework for dynamically predicting agent trajectories and their underlying goal points using Bayesian inference.

[1]The work presented in this chapter is a product of a collaborative effort, with substantial contributions and leadership from Tan Zhi-Xuan, and in collaboration with Stewy Slocum.

4.1 A Bayesian Model of Agents as Goal-Directed Motion Planners

To capture the sequential nature of decision-making and remain flexible in dynamic environments, we adopt a Bayesian approach to model the observed agent. This allows us to progressively refine predictions of the agent's future states and end goals as new observations arrive. We begin by defining a uniform prior over a set of possible goal regions G, reflecting an initial belief that, in the absence of further information, all goal regions are equally likely to be the actual target of the agent. Similarly, for each goal region $g \in G$, we assign a uniform prior over potential trajectory endpoints x_T within that goal region, assuming that all points within a given goal region are equally probable destinations for the agent. In accordance with Chapter 3, we then assume that the agent's motion plan represented as waypoints $x_{0:T}$ is drawn from a distribution $P(x_{0:T}|x_0, x_T)$ over low-cost, obstacle-avoiding trajectories with (known) start and (sampled) end points, (x_0, x_T). Finally, for each timestep $t \in [0, T]$, we model the observation of the agent's location, o_t, as being normally distributed around the true location at that timestep, x_t, with standard deviation σ representing potential noise due to any environmental factors.

$$\textit{Goal Prior:} \qquad g \sim \text{Uniform}(G) \tag{4.1}$$

$$\textit{Endpoint Prior:} \qquad x_T \sim \text{Uniform}(g) \tag{4.2}$$

$$\textit{Motion Planning:} \qquad x_{1:T} \sim P(x_{0:T}|x_0, x_T) \tag{4.3}$$

$$\textit{Noisy Observations:} \qquad o_t \sim \text{Normal}(x_t, \sigma) \tag{4.4}$$

Our aim is to infer the agent's goal, g, as well as the complete underlying motion plan, $x_{0:T}$, given observations of its trajectory so far, $o_{0:\tau}$, for some $\tau < T$, and a known initial location, x_0:

$$P(g, x_{0:T}|x_0, o_{0:\tau}). \tag{4.5}$$

We can represent this desired posterior distribution in terms of the previously defined

joint distributions and the likelihood of the observed agent states:

$$P(g, x_{0:T}|x_0, o_{0:\tau}) \propto P(g|x_T)P(x_{0:T}|x_0, x_T)\prod_{t=0}^{\tau} P(o_t|x_t). \tag{4.6}$$

To derive this equation, we first factor the full joint distribution of the goal, motion plan, and the observed trajectory segment, utilizing the chain rule and independence of individual observations:

$$\begin{aligned}
P(g, x_{1:T}, o_{0:T}|x_0) &= P(g|x_0)P(x_{1:T}|g, x_0)P(o_{0:\tau}|x_{1:T}, g, x_0), && (4.7)\\
&= P(g)P(x_{1:T}|g, x_0)P(o_{0:\tau}|x_{0:T}), && (4.8)\\
&= P(g)P(x_{1:T-1}|x_T, g, x_0)P(x_T|g, x_0)P(o_{0:\tau}|x_{0:\tau}), && (4.9)\\
&= P(g)P(x_{1:T}|x_T, x_0)P(x_T|g)\prod_{t=0}^{\tau} P(o_t|x_t). && (4.10)
\end{aligned}$$

We can then re-apply the chain rule to obtain the following:

$$\begin{aligned}
P(g, x_{0:T}|x_0, o_{0:\tau}) &= \frac{P(g, x_{1:T}, o_{0:\tau}|x_0)}{P(o_{0:\tau}|x_0)}, && (4.11)\\
&= \frac{P(g)P(x_T|g)P(x_{1:T}|x_T, x_0)\prod_{t=0}^{T} P(o_t|x_t)}{P(o_{0:\tau}|x_0)}, && (4.12)\\
&\propto P(g|x_T)P(x_{0:T}|x_0, x_T)\prod_{t=0}^{\tau} P(o_t|x_t), && (4.13)
\end{aligned}$$

where the proportionality is taken with respect to the goal g and the true motion plan $x_{0:T}$, and the marginal likelihood of the observations given the start point, $P(o_{0:\tau}|x_0)$, is constant with respect to g and $x_{0:T}$.

In the subsequent section, we adapt the methods presented in Chapter 3 to simulate potential motion plans of the observed agent. Following this, in Section 4.3, we introduce a framework for sequentially approximating the desired posterior distribution, as defined in eq. (4.6).

4.2 Sequential Monte Carlo Sampling for Trajectory Simulation

In the previous section, we modeled the observed agent as a goal-directed motion planner. In this section, we introduce a model to simulate plausible motion plans that the agent might pursue.

In Chapter 3, we explored the techniques to sample trajectories according to a Monte Carlo approximation of the Boltzmann distribution. This provided us with a framework to generate diverse motion plans in accordance with the principle of rational action. However, unlike in Chapter 3, where our aim was to generate trajectories for a planning agent, here we take the observer-centric approach, for the purposes of generating *hypotheses* about the agent's underlying motion plans. As a result, we not only need to generate samples from the motion planning distribution, $P(x_{0:T}|x_0, x_T)$, but we also need to evaluate it – which the MCMC algorithms in Chapter 3 can't do alone.

To generate a set of motion plan hypotheses, we propose the following approach. For each hypothesis, instead of sampling a trajectory using just one MCMC chain, we sample a set of weighted MCMC chains. Then, we return the result of one of those MCMC chains as the hypothesis. The main benefit of the weighted MCMC chains is that it gives us a way to estimate $P(x_{0:T}|x_0, x_T)$ by estimating the normalizing constant.

We formulate this method as a variant of the Sequential Monte Carlo (SMC) algorithm, characterized by the reweighting and resampling steps. Below, we outline the method, termed Sequential Monte Carlo for Boltzmann Trajectory Optimization (SMC-BoltzmannTrajOpt), and provide the pseudocode in Algorithm 2.

To enhance the plausibility of the sampled hypotheses, we generate a set of M candidate trajectories for each hypothesis within the total set of N hypotheses. Each of the M candidate trajectories is iteratively adapted to the target distribution via a

series of gradient-based MCMC kernels[2] (denoted as $K(\cdot)$). As discussed in Chapter 3, these kernels are stochastic analogues to the trajectory optimization steps (Ratliff et al., 2009; Schulman et al., 2013), so our approximation method can be viewed as an algorithm for stochastic trajectory optimization. We identify the most representative trajectory candidate by assigning importance weights to each. Initially, these weights correspond to the relative probability of a candidate trajectory under the unnormalized target distribution, P, compared to the proposal distribution Q. Over iterations, we adjust the weights in accordance with the likelihoods of the consecutive trajectory proposals, while maintaining the detailed balance condition of the Markov Chain, as detailed in Section 3.2. To counteract the bias introduced by sampling from the forward MCMC kernel, we multiply the weights by the reverse MCMC kernel (represented as $K(\cdot|\cdot)$ and $L(\cdot|\cdot)$, respectively, in Algorithm 2). The most likely trajectory candidate is then selected as one of the hypotheses through a resampling process. Specifically, we draw one out of M candidate trajectories based on a multinomial distribution, where the probability of selecting each trajectory is proportional to its normalized importance weight.

Algorithm 2 SMC for Boltzmann Trajectory Optimization (SMC-BoltzmannTrajOpt)

procedure SMC-BoltzmannTrajOpt(x_0, x_T, N, M)
  for $i \in [1, N]$ **do**
    $x_{0:T}^{i,0} \sim Q(x_{0:T}|x_0, x_T)$ ▷ Initialize from proposal distribution
    $w^{i,0} \leftarrow \frac{P(x_{0:T}^{i,0}|x_0,x_T)}{Q(x_{0:T}^{i,0}|x_0,x_T)}$ ▷ Compute importance weight
    $\hat{Z}^i \leftarrow 0$ ▷ Initialize normalizing constant estimate for particle i
    for $j \in [1, M]$ **do**
      $x_{0:T}^{i,j} \sim K(x_{0:T}|x_{0:T}^{i,j-1})$ ▷ Apply a series of MCMC kernels
      $w^{i,j} \leftarrow w^{i,j-1} \frac{P(x_{0:T}^{i,j}|x_0,x_T)}{P(x_{0:T}^{i,j-1}|x_0,x_T)} \frac{L(x_{0:T}^{i,j-1}|x_{0:T}^{i,j})}{K(x_{0:T}^{i,j}|x_{0:T}^{i,j-1})}$ ▷ Update importance weight
      $\hat{Z}^i \leftarrow \hat{Z}^i + w^{i,j}$ ▷ Update normalizing constant estimate
    end for
    $j \sim \text{Multinomial}(1, \frac{w^{i,\cdot}}{\hat{Z}^i})$ ▷ Resample one out of M trajectory indices
    $x_{0:T}^i \leftarrow x_{0:T}^{i,j}$ ▷ Select the resampled trajectory
  end for
  return $\{x_{0:T}^i, \hat{Z}^i\}_{i=1}^N$ ▷ Return resampled trajectories and normalizing constants
end procedure

[2]In addition to the MCMC methods from Chapter 3, we also consider MCMC kernels that incorporate second-order gradient information to enhance the trajectory proposals.

In addition to generating hypotheses about the motion plan, $x_{0:T}$, of the observed agent, we utilize SMC-BoltzmannTrajOpt to estimate the normalizing constant Z of the target distribution, $P(x_{0:T}|x_0, x_T)$. We derive unbiased estimates $\hat{Z}$, such that $\mathbb{E}[\hat{Z}] = Z(x_0, x_T)$, from the importance weight computation step of Algorithm 2. By taking the expected value of the sum of the importance weights, where P is the unnormalized target distribution and Q is the proposal that generates independent samples, we obtain the following relationship:

$$\mathbb{E}[\hat{Z}] = \mathbb{E}\left[\sum_{i=0}^{N} w^{i,0}\right], \tag{4.14}$$

$$= \mathbb{E}_{x_{0:T}^{1:N} \sim Q(\cdot)}\left[\sum_{i=1}^{N} \frac{P(x_{0:T}^i)}{Q(x_{0:T}^i)}\right], \tag{4.15}$$

$$= \sum_{i=1}^{N} \mathbb{E}_{x_{0:T}^i \sim Q(\cdot)}\left[\frac{P(x_{0:T}^i)}{Q(x_{0:T}^i)}\right], \tag{4.16}$$

$$= \sum_{i=1}^{N} \int \frac{P(x_{0:T}^i)}{Q(x_{0:T}^i)} Q(x_{0:T}^i) dx, \tag{4.17}$$

$$= \sum_{i=1}^{N} \int P(x_{0:T}^i) dx, \tag{4.18}$$

$$= \sum_{i=1}^{N} Z^i = Z. \tag{4.19}$$

Therefore, by using SMC-BoltzmannTrajOpt, we can acquire unbiased estimates of the normalized probability of a motion plan, $\hat{P}(x_{0:T}|x_0, x_T)$, which is sufficient for sound Bayesian inference (Cusumano-Towner et al., 2017; Lew et al., 2022).

4.3 Inverse Planning with Sequential Monte Carlo Samples

In Section 4.1, we introduced a Bayesian model for the observed agent. This model forms the basis for utilizing Bayesian inference for inverse planning, which involves estimating the agent's underlying motion plan and end goal, by reasoning about their likelihoods under the observed trajectory segments. Inverse planning, in this context, becomes a problem of recursive approximation of a desired posterior distribution (as defined in eq. (4.6)) under incoming noisy observations.

This task aligns exactly with the broader objective of Sequential Monte Carlo (SMC) methods: by employing a set of particles to represent hypotheses about the state of a system, SMC methods iteratively update and refine these hypotheses based on new observations, effectively tracking the posterior distribution over time (Doucet et al., 2001). In this section, we develop a SMC algorithm called Sequential Monte Carlo for Inverse Motion Planning (SMC-IMP) for sequentially approximating the posterior distribution over the agent's goals given observations of partially completed trajectories. We outline the pseudocode for SMC-IMP in Algorithm 3.

Algorithm 3 SMC for Inverse Motion Planning (SMC-IMP)

procedure SMC-IMP(x_0, $o_{0:\tau}$, N)

  $(g, x_{0:T})^i \sim P(g, x_{0:T}|x_0)$ **for** $i \in [1, N]$ ▷ Sample N hypotheses

  $w^i \leftarrow P(o_0|x_0)$ **for** $i \in [1, N]$ ▷ Initialize weights

  for $t \in [1, \tau],\ i \in [1, N]$ **do**

    $\tilde{x}_t^i \sim K(x_t; o_t)$ ▷ Propose new $\tilde{x}_t$ close to o_t

    $\tilde{x}_{0:T}^i \leftarrow (x_{0:t-1}^i, \tilde{x}_t, x_{t+1:T}^i)$ ▷ Replace x_t with new $\tilde{x}_t$

    $w^i \leftarrow w^i \frac{L(\tilde{x}_t^i; o_t)}{K(x_t^i; \tilde{x}_{0:T}^i)} \frac{P(g^i, \tilde{x}_{0:T}^i|x_0)}{P(g^i, x_{0:T}^i|x_0)} P(o_t|\tilde{x}_t^i)$ ▷ Reweight hypotheses

    $x_{0:T}^i \sim \text{MALA}(\cdot, \text{NMC}(\cdot; \tilde{x}_{0:T}^i))$ ▷ Rejuvenate via MCMC

  end for

  return $\{(g, x_{0:T})^i, w^i\}_{i=1}^N$ ▷ Return weighted hypotheses

end procedure

In line with the standard particle filtering algorithms, SMC-IMP begins with a

set of N hypotheses about the agent's goal-driven motion plan, $\{(g, x_{0:T})^i\}_{i=1}^N$. In particular, SMC-IMP obtains the initial set of motion plan hypotheses from the SMC-BoltzmannTrajOpt algorithm, as outlined in Section 4.2. Initially, these hypotheses are all weighted by the observation prior, in accordance with eq. (4.4). At each timestep t, as the new observation o_t of the agent's state arrives, the hypotheses are adjusted to better align with o_t. Similarly as before, this is achieved using the MCMC kernels (denoted as K).

In addition to the hypotheses themselves, the weights of the hypotheses are also updated. Similarly to the approach in Algorithm 2, we mitigate the bias introduced by sampling from the forward MCMC kernel by adjusting the weights using the reverse MCMC kernel. These kernels, taking the form of 2D Gaussians, are again represented as $K(\cdot|\cdot)$ and $L(\cdot|\cdot)$, respectively. In addition, the weights are adjusted by the ratio of conditional probabilities, $P(g, \tilde{x}_{0:T}|x_0)$ and $P(g, x_{0:T}|x_0)$, which reflects whether the newly proposed motion plan aligns more closely with the goal than the previous one. These conditional probabilities factor into $P(g, \tilde{x}_{0:T}|x_0) = P(g)P(x_T|g)P(\tilde{x}_{0:T}|x_0)$ and $P(g, x_{0:T}|x_0) = P(g)P(x_T|g)P(x_{0:T}|x_0)$, respectively, where $P(g)$ and $P(x_T|g)$ are the goal and the endpoint priors from eqs. (4.1) and (4.2).

The terms $P(\tilde{x}_{0:T}|x_0)$ and $P(x_{0:T}|x_0)$ are not directly known due to the unknown normalizing constant. However, as discussed in Section 4.2, SMC-BoltzmannTrajOpt allows us to estimate $P(x_{0:T}|x_0)$ as $\hat{P}(x_{0:T}|x_0) = \frac{w}{\hat{Z}}$, where w is the weight of the returned motion plan and $\hat{Z}$ is the estimate of the normalizing constant. Similarly, $P(\tilde{x}_{0:T}|x_0)$ can be approximated as $P(\tilde{x}_{0:T}|x_0) \approx \tilde{w}/\hat{\tilde{Z}}$, where $\hat{\tilde{Z}} \coloneqq \hat{Z} - w + \tilde{w}$. Consequently, the hypothesis reweighing step becomes $w^i \leftarrow w^i \frac{K(\tilde{x}_t^i; o_t)}{L(x_t^i; \tilde{x}_{0:T}^i)} \frac{\tilde{w}/(\hat{Z}-w+\tilde{w})}{w/\hat{Z}} P(o_t|\tilde{x}_t^i)$.

Lastly, to prevent the degeneracy of weights and introduce more diversity into the set of hypotheses, we further adjust the hypotheses with rejuvenation moves (Chopin, 2002). We rejuvenate the hypotheses using gradient-based MCMC kernels, including a first-order method, MALA, as described in Section 3.2, and a second-order method, Newtonian Monte Carlo (NMC), which uses the Hessian in addition to the gradient of the log-posterior to propose new states (Arora et al., 2019). This combined approach allows us to facilitate both the global exploration of the state space as well

as fine-tuning within the high probability regions. The resulting weighted collection of hypotheses $\{(g, x_{0:T})^i, w^i\}_{i=1}^N$ at timestep t represents a discrete approximation to the posterior over the agent's goal and motion plan, $P(g, x_{0:T}|x_0, o_{0:t})$.

4.4 Experiments

We evaluate the inverse planning capabilities of our framework on 45 agent trajectories across 5 different scenes in a 2D environment. Each of the 5 scenes contains 3 possible goal regions with varying obstacle layouts, including smaller scattered, irregularly-shaped obstacles, a maze, and a tunnel. Analogously to Chapter 3, we design these various scenes in order to test our framework's ability to generate diverse motion plan hypotheses. In addition, the various scenes test our inference algorithm's ability to handle multimodal posterior distributions over goal-driven motion plans, i.e., the ability to consider a variety of likely outcomes. In each scene, we generate a set of three trajectories (of length $T = 20$) per goal, corresponding to different strategies for navigating around the obstacles.

Our implementation uses Gen (Cusumano-Towner et al., 2019), an open-source probabilistic programming system embedded in Julia (Bezanson et al., 2017). We configure the SMC-BoltzmannTrajOpt algorithm to sample $N = 600$ particles and use a rationality parameter of $\beta = 20$ for the target distribution defined in eq. (3.7). We compare the SMC-IMP goal inference framework against two baselines: a greedy distance-based heuristic and a Laplace approximation to the posterior over goals akin to the approach in (Dragan, 2015).

We define the greedy heuristic baseline to operate under a simplified assumption of agent behavior, in which the closer a goal region, g, is to the last observed agent location, o_t, the higher the likelihood that it represents the agent's intended destination. We express this mathematically as:

$$P(g|o_{0:t}) \propto e^{(\text{min-dist}(o_t, g))}, \tag{4.20}$$

where min-dist(o_t, g) is the minimum Euclidan distance from the observed agent location o_t to each of the goal regions $g \in G$.

Next, we implement the Laplace approximation baseline in the following way. Consider a generic Boltzmann distribution-based model of trajectories, $P(x_{0:T}) = e^{-c(x_{0:T})}$. Further, consider a second-order Taylor expansion of c about a local minimum, $x^*_{0:T}$, where $\nabla_{x_{0:T}} c(x_{0:T}) = \mathbf{0}$:

$$c(x_{0:T}) \approx \hat{c}(x_{0:T}) \coloneqq c(x^*_{0:T}) + \frac{1}{2}(x_{0:T} - x^*_{0:T})^\top \left(\nabla^2 c(x^*_{0:T})\right)(x_{0:T} - x^*_{0:T}), \tag{4.21}$$

and $\nabla^2 c(x^*_{0:T})$ is the Hessian of c at $x^*_{0:T}$. Note that approximating $P(x_{0:T}) \approx e^{-\hat{c}(x_{0:T})} \propto e^{-\frac{1}{2}(x_{0:T}-x^*_{0:T})^\top H (x_{0:T}-x^*_{0:T})}$, where $H \coloneqq (\nabla^2 c(x^*_{0:T}))$, is a Gaussian. This allows us to compute the desired distribution, $P(g|o_{0:t})$. For example, in the noiseless case, we get the following expression for $P(g|x_{0:\tau})$ proportional to g:

$$\begin{aligned} P(g|x_{0:\tau}) &= P(x_{0:\tau}|g)\frac{P(g)}{P(x_{0:\tau})}, && (4.22)\\ &\propto P(x_{0:\tau}|g)P(g), && (4.23)\\ &= \frac{e^{-\hat{c}(x_{0:\tau})}\int_{x_{\tau+1:T}} e^{-\hat{c}(x_{\tau+1:T})} dx_{\tau+1:T}}{\int_{x_{0:T}} e^{-\hat{c}(x_{\tau+1:T})} dx_{\tau+1:T}} P(g), && (4.24) \end{aligned}$$

where all integrals are Gaussian integrals and are thus explicitly computable.

In Table 4.1, we compare the performance of SMC-IMP to the two baselines. We report the posterior probability each method assigns to the true goal at $\frac{1}{5}$, $\frac{1}{4}$, $\frac{1}{3}$, and $\frac{1}{2}$ the length of the trajectory (rounded to the nearest timestep). The reported probabilities are averaged across all trajectories in the dataset. In addition, we provide the Brier score at the same timesteps, which reflects how well-calibrated the predictions are (lower values indicate higher accuracy) (Brier et al., 1950).

As expected, SMC-IMP outperforms the greedy distance-based baseline. Compared to the Laplace approximation, SMC-IMP tends to assign lower probability to the true goal, but with a better Brier score at earlier timesteps. This reflects that SMC-IMP is better at maintaining uncertainty when the data is ambiguous, whereas the Laplace approximation is over-confident at earlier timesteps, assigning high probabilities to

the wrong goals. Crucially, SMC-IMP is the only method of the three that accounts for the inherent multimodality of the distribution over trajectories stemming from a multitude of strategies for navigating around obstacles.

	$\mathbf{P}(g_{\mathbf{true}}\|o_{1:t})$				**Brier Score**			
Method	$t = T/5$	$T/4$	$T/3$	$T/2$	$t = T/5$	$T/4$	$T/3$	$T/2$
Greedy	0.42	0.45	0.51	0.62	0.67	0.65	0.59	0.45
Laplace	**0.60**	**0.61**	**0.73**	**0.85**	0.68	0.66	0.45	**0.25**
SMC-IMP	0.53	**0.61**	0.70	0.79	**0.48**	**0.43**	**0.40**	0.33

Table 4.1: A comparative evaluation of the goal inference methods based on the probability assigned to the true goal and the predictive accuracy as indicated by the Brier score (lower is better).
These metrics are computed at various timesteps t expressed as fractions of the full trajectory length T, and then averaged across multiple trials.
We contrast our method, SMC-IMP, with a greedy distance-based heuristic and a goal inference method via Laplace approximation inspired by (Dragan, 2015).
Our findings show that SMC-IMP surpasses the greedy heuristic, as anticipated. In comparison with the Laplace approximation, SMC-IMP tends to be more conservative in its predictions, often ascribing a lower probability to the true goal. Nonetheless, SMC-IMP demonstrates superior accuracy when the number of observations is very limited.

In Figure 4-1, we illustrate our framework's performance in dynamic trajectory prediction and goal inference. The inference is based on noisy observations of partially completed trajectories (represented as black points) within an environment characterized by irregularly shaped obstacles (in grey) and three distinct goal regions (blue, red, and green). The top of the figure shows how the motion plan hypotheses, initially sampled by SMC-BoltzmannTrajOpt, get updated based on the incoming observations. The bottom part illustrates how SMC-IMP progressively refines its predictions about the agent's intended goal over time.

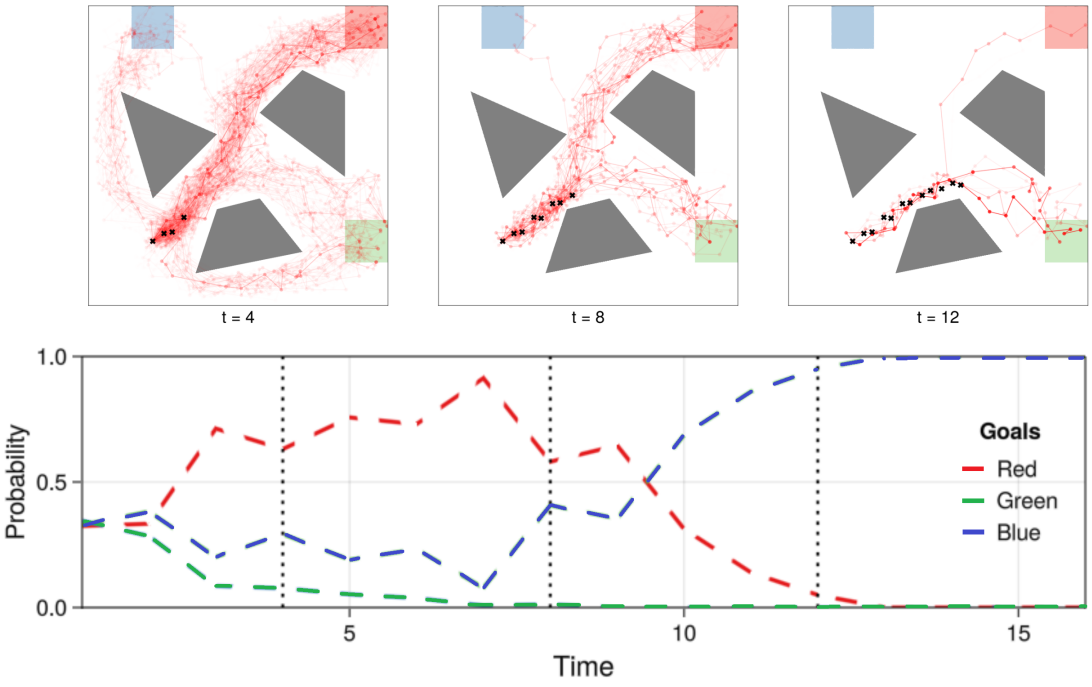


Figure 4-1: A visual analysis of the motion plan predictions (above), alongside the goal inference (below). The evaluation is performed within a continuous environment featuring three irregularly shaped obstacles (shown in grey) and three potential goal regions (colored in blue, red, and green). Black markers denote the noisy observations, while the hypothesised motion plans are illustrated in red.

Chapter 5

Discussion

We began this work with the vision of partaking in the enhancement of human-AI interaction by contributing to the development of fitter forward and inverse planning methods. Our approach is multidisciplinary, drawing upon insights from mobile robotics, decision theory, and computational cognitive science. We provide a unified trajectory planning and inference framework that jointly leverages Monte Carlo methods, highlighting their versatility and effectiveness in both efficiently approximating desired behavior as well as making informed predictions in real-time within a Bayesian framework. In the following sections of this chapter, we outline our primary technical contributions and suggest future directions of exploration.

5.1 Summary of Contributions

First, our investigation focuses on the development of tailored Markov Chain Monte Carlo (MCMC) algorithms for motion planning. By using MCMC samples to approximate the Boltzmann distribution, we show that we can generate a variety of near-optimal trajectories instead of just a single, strictly optimal solution to the navigation problem. Ensuring such diversity-aware planning is essential for decision-making under uncertainty (Lupu et al., 2021; Zhao and Wildes, 2021; Huang et al., 2020), and we propose a variance-based metric to characterize the diversity of the generated motion plans. We adapt both the conventional Metropolis-Hastings algorithm as well

as the first-order methods that incorporate gradient information about the target distribution. We set up a continuous, two-dimensional navigation environment to assess the algorithms' performance based on computational complexity, motion plan feasibility, and trajectory diversity. Extensive analysis reveals that even the simplest gradient-based sampling methods reliably produce smooth, collision-free trajectories while efficiently exploring high-probability regions of the state space. We also show that the variance of the generated trajectories correlates well with the amount of the obstacle avoidance strategies. Furthermore, we find that a simple projection technique, which translates infeasible samples back into free space, provides more robust and computationally efficient obstacle avoidance compared to the penalty-based methods akin to those used in (Ratliff et al., 2009).

In the latter half of our study, we shift focus from an actor-centric to an observer-centric approach. Rather than generating a variety of near-optimal navigation solutions for a planning agent, we apply our diversity-aware motion planning framework to enable robust inferences about observed motion. We develop a model for the observed agent in which we adopt a Bayesian approach to reason about the agent's motion plan and end goal in real-time: by modeling agents as approximately rational planners, we estimate latent states by simulating probable behaviors. Our proposed method, Sequential Monte Carlo for Boltzmann Trajectory Optimization (SMC-BoltzmannTrajOpt), enables approximate Bayesian inference while constructing a set of optimized simulations of the agent's motion plan. We incorporate this method into a novel inverse planning framework, Sequential Monte Carlo for Inverse Motion Planning (SMC-IMP), which predicts the agent's goal and motion plan in real-time by dynamically optimizing the simulated trajectories with MCMC kernels. We assess our framework across diverse 2D navigation scenarios and find that it surpasses distance-based goal inference. In addition, our framework's distinctive ability to capture the inherent multimodality of posterior distributions over the motion plans results in an improved prediction accuracy in ambiguous situations, compared to a Laplace approximation to the posterior over goals akin to the approach in (Dragan, 2015).

5.2 Limitations and Future Work

While our framework shows promise for advancing trajectory simulation and goal inference, many challenges remain. Extensions of our approach may require improving the runtime and accuracy of the algorithm through better Monte Carlo approximations of the Boltzmann distribution. This may involve the use of smarter proposal distributions or specialized solvers to generate candidate motion plans that integrate any existing knowledge about the topology of the environment.

Additionally, it would be interesting to explore the possibility of integrating our framework with existing algorithms for inverse task-level planning (Zhi-Xuan et al., 2020), with the eventual hope of performing goal inference and trajectory prediction over task-and-motion plans (Garrett et al., 2021). This would provide a significant contribution to the current state-of-the-art in the field of human-robot collaboration, particularly in relation to tasks involving navigation around movable obstacles and manipulation.

Adapting our method to human-robot collaboration problems would also involve scaling to higher-dimensional domains. For a rigorous and more comprehensive assessment of the practicality of MCMC motion planners in more complex settings, it would be necessary to conduct further experiments, for example, involving contact and narrow spaces, as in (Shkolnik and Tedrake, 2011). However, we are optimistic about the potential of our method to scale to 3D motion planning problems since our trajectory sampling algorithm in Section 4.3 can be viewed as a probabilistic modification to the well-established Sequential Quadratic Programming (SQP)-based motion planners (Hadfield-Menell et al., 2016a).

Furthermore, our method is also applicable to human motion prediction for collision avoidance, which remains a major challenge in the development of safe and highly-autonomous collaborative systems (Ajoudani et al., 2018). Currently, behavior forecasting is dominated by deep learning approaches (Schydlo et al., 2018; Alahi et al., 2016; Martinez et al., 2017) that typically lack interpretability and require large amounts of data. In contrast, a method like ours offers a white-box approach

grounded in cognitive theories of goal-directed behavior, and does not necessitate hours of data collection and retraining for each new problem. By leveraging pose estimation techniques, we could represent observed human motion in a way that is compatible with our trajectory parameterization and motion planning formulations. The trajectories predicted by our framework could then be used as inputs to the robot's cost function and problem definition, which can in turn be solved using standard robotic planning methods, especially probabilistic algorithms like belief-space planning (Kaelbling and Lozano-Pérez, 2013; Hadfield-Menell et al., 2015).

Nonetheless, it is crucial to take into account safety concerns when doing behavior prediction in domains where humans and machines would interact closely. Fundamentally, there is a discrepancy between the real world and simulated environments that rely on simplistic noise representations and limited models of rational decision-making. Consequently, inaccurate estimations of internal states could result in inappropriate responses and coordination failures. This can be especially detrimental in safety-critical tasks, such as in manufacturing, medicine, and disaster response.

Another crucial aspect of safety involves adhering to physical constraints and ensuring effective obstacle avoidance. Our motion planning method relies on projecting samples onto the feasible set for navigating constrained environments. While these projections are typically small in practice, it would be valuable to develop a more principled approach that avoids the possibility of failure in tightly constrained spaces and maintains the property of reversibility (Brubaker et al., 2012). Deployment in complex physical spaces, thus, may require considering more rigorous obstacle avoidance assurances and verification (Mitsch et al., 2017).

Bibliography

A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib. Progress and prospects of the human–robot collaboration. *Autonomous Robots*, 42: 957–975, 2018.

A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016.

A. Alanqary, G. Z. Lin, J. Le, T. Zhi-Xuan, V. K. Mansinghka, and J. B. Tenenbaum. Modeling the mistakes of boundedly rational agents within a bayesian theory of mind. *arXiv preprint arXiv:2106.13249*, 2021.

S. V. Albrecht, C. Brewitt, J. Wilhelm, B. Gyevnar, F. Eiras, M. Dobre, and S. Ramamoorthy. Interpretable goal-based prediction and planning for autonomous driving. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1043–1049, 2021. doi: 10.1109/ICRA48506.2021.9560849.

A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2016.

A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada. Control barrier functions: Theory and applications. In *2019 18th European control conference (ECC)*, pages 3420–3431. IEEE, 2019.

N. S. Arora, N. K. Tehrani, K. D. Shah, M. Tingley, Y. L. Li, N. Torabi, D. Noursi, S. A. Masouleh, E. Lippert, and E. Meijer. Newtonian monte carlo: a second-order gradient method for speeding up mcmc. 2019.

M. Bacharach. Interactive team reasoning: A contribution to the theory of cooperation. *Research in Economics*, 53(2):117–147, 1999. ISSN 1090-9443. doi: https://doi.org/10.1006/reec.1999.0188. URL `https://www.sciencedirect.com/science/article/pii/S1090944399901886`.

C. Baker and R. Saxe. Bayesian theory of mind: Modeling joint belief-desire attribution. *Proceedings of the Thirty-Third Annual Conference of the Cognitive Science Society*, 01 2011.

C. L. Baker and J. B. Tenenbaum. Modeling human plan recognition using bayesian theory of mind. 2014. URL https://api.semanticscholar.org/CorpusID:5878846.

C. L. Baker, R. Saxe, and J. B. Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009.

C. L. Baker, J. Jara-Ettinger, R. Saxe, and J. B. Tenenbaum. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, 1, 2017. URL https://api.semanticscholar.org/CorpusID:3338320.

S. Baron-Cohen, M. Lombardo, and H. Tager-Flusberg. *Understanding Other Minds: Perspectives from developmental social neuroscience*. Oxford University Press, 08 2013. ISBN 9780199692972. doi: 10.1093/acprof:oso/9780199692972.001.0001. URL https://doi.org/10.1093/acprof:oso/9780199692972.001.0001.

J. Barraquand and J.-C. Latombe. Robot motion planning: A distributed representation approach. *The International Journal of Robotics Research*, 10(6):628–649, 1991. doi: 10.1177/027836499101000604. URL https://doi.org/10.1177/027836499101000604.

M. Bartholomew-Biggs, S. Brown, B. Christianson, and L. Dixon. Automatic differentiation of algorithms. *Journal of Computational and Applied Mathematics*, 124 (1-2):171–190, 2000.

J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017. URL https://doi.org/10.1137/141000671.

C. K. Birdsall and A. B. Langdon. *Plasma physics via computer simulation*. CRC press, 2018.

M. Botvinick and M. Toussaint. Planning as inference. *Trends in cognitive sciences*, 16(10):485–488, 2012.

M. S. Branicky, R. A. Knepper, and J. J. Kuffner. Path and trajectory diversity: Theory and algorithms. In *2008 IEEE International Conference on Robotics and Automation*, pages 1359–1364. IEEE, 2008.

G. W. Brier et al. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.

R. A. Briggs. Normative theories of rational choice: Expected utility. 2014.

M. Brubaker, M. Salzmann, and R. Urtasun. A family of mcmc methods on implicitly defined manifolds. In *Artificial intelligence and statistics*, pages 161–172. PMLR, 2012.

P. Carruthers and P. K. Smith, editors. *Theories of Theories of Mind*. Cambridge University Press, New York, 1996.

N. Chopin. A sequential particle filter method for static models. *Biometrika*, 89(3): 539–552, 2002.

P. M. Churchland. Folk psychology and the explanation of human behavior. *Philosophical Perspectives*, 3:225–241, 1988. doi: 10.2307/2214269.

M. F. Cusumano-Towner, A. Radul, D. Wingate, and V. K. Mansinghka. Probabilistic programs for inferring the goals of autonomous agents. *arXiv preprint arXiv:1704.04977*, 2017.

M. F. Cusumano-Towner, F. A. Saad, A. K. Lew, and V. K. Mansinghka. Gen: A general-purpose probabilistic programming system with programmable inference. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI 2019, pages 221–236, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6712-7. doi: 10.1145/3314221.3314642. URL `http://doi.acm.org/10.1145/3314221.3314642`.

R. D'Andrade. A folk model of the mind. In D. Holland and N. Quinn, editors, *Cultural Models in Language and Thought*, pages 112–148. Cambridge University Press, 1987.

D. C. Dennett. Intentional systems. *Journal of Philosophy*, 68(February):87–106, 1971. doi: 10.2307/2025382.

A. Doucet, N. de Freitas, and N. J. Gordon. An introduction to sequential monte carlo methods. In A. Doucet, N. de Freitas, and N. J. Gordon, editors, *Sequential Monte Carlo Methods in Practice*, Statistics for Engineering and Information Science, pages 3–14. Springer, 2001. doi: 10.1007/978-1-4757-3437-9_1. URL `https://doi.org/10.1007/978-1-4757-3437-9_1`.

A. D. Dragan. *Legible robot motion planning.* PhD thesis, Carnegie Mellon University, 2015.

A. D. Dragan, K. C. Lee, and S. S. Srinivasa. Legibility and predictability of robot motion. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 301–308, 2013. doi: 10.1109/HRI.2013.6483603.

M. Elbanhawi and M. Simic. Sampling-based robot motion planning: A review. *IEEE Access*, 2:56–77, 2014. doi: 10.1109/ACCESS.2014.2302442.

J. F. Fisac, M. A. Gates, J. B. Hamrick, C. Liu, D. Hadfield-Menell, M. Palaniappan, D. Malik, S. S. Sastry, T. L. Griffiths, and A. D. Dragan. Pragmatic-pedagogic value alignment. In N. M. Amato, G. Hager, S. Thomas, and M. Torres-Torriti, editors, *Robotics Research*, pages 49–57, Cham, 2020. Springer International Publishing. ISBN 978-3-030-28619-4.

C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez. Integrated task and motion planning. *Annual*

Review of Control, Robotics, and Autonomous Systems, 4(1):265–293, 2021. doi: 10.1146/annurev-control-091420-084139. URL https://doi.org/10.1146/annurev-control-091420-084139.

G. Gergely and G. Csibra. Teleological reasoning in infancy: The naıve theory of rational action. *Trends in cognitive sciences*, 7(7):287–292, 2003.

G. Gergely, Z. Nádasdy, G. Csibra, and S. Bíró. Taking the intentional stance at 12 months of age. *Cognition*, 56(2):165–193, 1995. ISSN 0010-0277. doi: https://doi.org/10.1016/0010-0277(95)00661-H. URL https://www.sciencedirect.com/science/article/pii/001002779500661H.

D. Hadfield-Menell, E. Groshev, R. Chitnis, and P. Abbeel. Modular task and motion planning in belief space. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4991–4998. IEEE, 2015.

D. Hadfield-Menell, C. Lin, R. Chitnis, S. Russell, and P. Abbeel. Sequential quadratic programming for task plan optimization. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5040–5047. IEEE, 2016a.

D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan. Cooperative inverse reinforcement learning. *Advances in neural information processing systems*, 29, 2016b.

D. Hsu, J.-c. Latombe, and R. Motwani. Path planning in expansive configuration spaces. *International Journal of Computational Geometry Applications*, 09, 03 1997. doi: 10.1142/S0218195999000285.

X. Huang, S. G. McGill, J. A. DeCastro, L. Fletcher, J. J. Leonard, B. C. Williams, and G. Rosman. Diversitygan: Diversity-aware vehicle motion prediction via latent semantic sampling, 2020.

A. Hyvärinen and P. Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

B. Ivanovic, K. Leung, E. Schmerling, and M. Pavone. Multimodal deep generative models for trajectory prediction: A conditional variational autoencoder approach. *IEEE Robotics and Automation Letters*, 6:295–302, 2020. URL https://api.semanticscholar.org/CorpusID:221090012.

L. Janson, E. Schmerling, and M. Pavone. Monte carlo motion planning for robot trajectory optimization under uncertainty, 2015.

E. E. Jones and K. E. Davis. From acts to dispositions the attribution process in person perception11much of the research reported herein was supported by national science foundation grants 8857 and 21955 to the first author. volume 2 of *Advances in Experimental Social Psychology*, pages 219–266. Academic Press, 1965. doi: https://doi.org/10.1016/S0065-2601(08)60107-0. URL https://www.sciencedirect.com/science/article/pii/S0065260108601070.

L. P. Kaelbling and T. Lozano-Pérez. Integrated task and motion planning in belief space. *The International Journal of Robotics Research*, 32(9-10):1194–1227, 2013.

M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal. Stomp: Stochastic trajectory optimization for motion planning. In *2011 IEEE International Conference on Robotics and Automation*, pages 4569–4574, 2011. doi: 10.1109/ICRA.2011.5980280.

G. Kaminka, M. Vered, and N. Agmon. Plan recognition in continuous domains. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. doi: 10.1609/aaai.v32i1.12097. URL `https://ojs.aaai.org/index.php/AAAI/article/view/12097`.

G. Kang, Y. Kim, W. S. You, Y. Lee, H. Oh, H. Moon, and H. Choi. Sampling-based path planning with goal oriented sampling. pages 1285–1290, 07 2016. doi: 10.1109/AIM.2016.7576947.

S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning, 2011.

S. S. Keerthi and C.-J. Lin. Asymptotic behaviors of support vector machines with gaussian kernel. *Neural computation*, 15(7):1667–1689, 2003.

M. Khan and A. Chatterjee. Gaussian control barrier functions: Safe learning and control. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 3316–3322. IEEE, 2020.

M. Kobilarov. Cross-entropy motion planning. *The International Journal of Robotics Research*, 31(7):855–871, 2012. doi: 10.1177/0278364912444543. URL `https://doi.org/10.1177/0278364912444543`.

M. J. Kochenderfer and T. A. Wheeler. *Algorithms for optimization*. Mit Press, 2019.

J. Kuffner and S. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 2, pages 995–1001 vol.2, 2000. doi: 10.1109/ROBOT.2000.844730.

A. Kuntz, C. Bowen, and R. Alterovitz. Interleaving optimization with sampling-based motion planning (ios-mp): Combining local optimization with global exploration, 2016.

P. Langevin. *On the theory of Brownian motion*. 1908.

S. M. LaValle. Rapidly-exploring random trees : a new tool for path planning. *The annual research report*, 1998. URL `https://api.semanticscholar.org/CorpusID:14744621`.

J. Lee, D. Yi, and S. S. Srinivasa. Sampling of pareto-optimal trajectories using progressive objective evaluation in multi-objective motion planning. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9, 2018. URL `https://api.semanticscholar.org/CorpusID:53450230`.

S. Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

A. K. Lew, M. Cusumano-Towner, and V. K. Mansinghka. Recursive monte carlo and variational inference with auxiliary variables. In *Uncertainty in Artificial Intelligence*, pages 1096–1106. PMLR, 2022.

L. Li, X. Long, and M. A. Gennert. Birrtopt: A combined sampling and optimizing motion planner for humanoid robots. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 469–476, 2016. doi: 10.1109/HUMANOIDS.2016.7803317.

C. Liu, J. B. Hamrick, J. F. Fisac, A. D. Dragan, J. K. Hedrick, S. S. Sastry, and T. L. Griffiths. Goal inference improves objective and perceived performance in human-robot collaboration, 2018.

A. Lupu, B. Cui, H. Hu, and J. Foerster. Trajectory diversity for zero-shot coordination. In *International Conference on Machine Learning*, pages 7204–7213. PMLR, 2021.

Y. J. Ma, J. P. Inala, D. Jayaraman, and O. Bastani. Likelihood-based diverse sampling for trajectory forecasting. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13259–13268, 2020. URL `https://api.semanticscholar.org/CorpusID:237487995`.

J. Martinez, M. J. Black, and J. Romero. On human motion prediction using recurrent neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4674–4683, 2017. doi: 10.1109/CVPR.2017.497.

J. Massardi, M. Gravel, and Beaudry. Error-tolerant anytime approach to plan recognition using a particle filter. *Proceedings of the International Conference on Automated Planning and Scheduling*, 29(1):284–291, May 2021. doi: 10.1609/icaps.v29i1.3490. URL `https://ojs.aaai.org/index.php/ICAPS/article/view/3490`.

W. Min, E. Ha, J. Rowe, B. Mott, and J. Lester. Deep learning-based goal recognition in open-ended digital games. *Proceedings of the 10th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE 2014*, 10:37–43, 01 2014. doi: 10.1609/aiide.v10i1.12717.

S. Mitsch, K. Ghorbal, D. Vogelbacher, and A. Platzer. Formal verification of obstacle avoidance and navigation of ground robots. *The International Journal of Robotics Research*, 36(12):1312–1340, 2017. doi: 10.1177/0278364917733549. URL `https://doi.org/10.1177/0278364917733549`.

M. Nagumo. Über die lage der integralkurven gewöhnlicher differentialgleichungen. *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, 24:551–559, 1942.

A. Orthey, C. Chamzas, and L. E. Kavraki. Sampling-based motion planning: A comparative review. *Annual Review of Control, Robotics, and Autonomous Systems*, 7(1):null, 2024. doi: 10.1146/annurev-control-061623-094742. URL `https://doi.org/10.1146/annurev-control-061623-094742`.

T. Osa. Multimodal trajectory optimization for motion planning. *The International Journal of Robotics Research*, 39(8):983–1001, 2020. doi: 10.1177/0278364920918296. URL `https://doi.org/10.1177/0278364920918296`.

C. Park, J. Pan, and D. Manocha. Itomp: Incremental trajectory optimization for real-time replanning in dynamic environments. 06 2012.

A. Piché, V. Thomas, C. Ibrahim, Y. Bengio, and C. Pal. Probabilistic planning with sequential monte carlo methods. In *International Conference on Learning Representations*, 2019.

Pymunk contributors. Pymunk: Easy-to-use pythonic 2d physics library, 2023. URL `https://www.pymunk.org/en/latest/pymunk.html`. Accessed: 2023-11-13.

N. Rabinowitz, F. Perbet, F. Song, C. Zhang, S. A. Eslami, and M. Botvinick. Machine theory of mind. In *International conference on machine learning*, pages 4218–4227. PMLR, 2018.

D. Ramachandran and E. Amir. Bayesian inverse reinforcement learning. In *IJCAI*, volume 7, pages 2586–2591, 2007.

M. Ramırez and H. Geffner. Plan recognition as planning. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence.*, pages 1778–1783. Citeseer, 2009.

M. Ramírez and H. Geffner. Probabilistic plan recognition using off-the-shelf classical planners. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa. Chomp: Gradient optimization techniques for efficient motion planning. In *2009 IEEE International Conference on Robotics and Automation*, pages 489–494. IEEE, 2009.

P. J. Rossky, J. D. Doll, and H. L. Friedman. Brownian dynamics as smart monte carlo simulation. *The Journal of Chemical Physics*, 69(10):4628–4633, 1978.

G. Sánchez and J.-C. Latombe. A single-query bi-directional probabilistic roadmap planner with lazy collision checking. In R. A. Jarvis and A. Zelinsky, editors, *Robotics Research*, pages 403–417, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-540-36460-3.

B. Scassellati. Theory of mind for a humanoid robot. *Autonomous Robots*, 12:13–24, 01 2002. doi: 10.1023/A:1013298507114.

T. C. Schelling. The strategy of conflict. prospectus for a reorientation of game theory. *Journal of Conflict Resolution*, 2(3):203–264, 1958.

J. Schulman, J. Ho, A. X. Lee, I. Awwal, H. Bradlow, and P. Abbeel. Finding locally optimal, collision-free trajectories with sequential convex optimization. *Robotics: Science and Systems IX*, 2013.

P. Schydlo, M. Rakovic, L. Jamone, and J. Santos-Victor. Anticipation in human-robot cooperation: A recurrent neural network approach for multiple action sequences prediction. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5909–5914. IEEE, 2018.

I. R. Seaman, J.-W. van de Meent, and D. Wingate. Nested reasoning about autonomous agents using probabilistic programs. *arXiv preprint arXiv:1812.01569*, 2018.

A. C. Shkolnik and R. Tedrake. Sample-based planning with volumes in configuration space. *ArXiv*, abs/1109.3145, 2011.

G. Sukthankar, C. Geib, H. Bui, D. Pynadath, and R. Goldman. *Plan, activity, and intent recognition: Theory and practice*. Morgan Kaufmann, Apr. 2014. ISBN 9780123985323.

L. Wang, R. Gao, J. Váncza, J. Krüger, X. Wang, S. Makris, and G. Chryssolouris. Symbiotic human-robot collaborative assembly. *CIRP Annals*, 68(2):701–726, 2019. ISSN 0007-8506. doi: https://doi.org/10.1016/j.cirp.2019.05.002. URL `https://www.sciencedirect.com/science/article/pii/S0007850619301593`.

F. Warneken and M. Tomasello. Helping and cooperation at 14 months of age. *Infancy : the official journal of the International Society on Infant Studies*, 11 3:271–294, 2007. URL `https://api.semanticscholar.org/CorpusID:14240397`.

A. L. Woodward. Infants selectively encode the goal object of an actor's reach. *Cognition*, 69(1):1–34, 1998. ISSN 0010-0277. doi: https://doi.org/10.1016/S0010-0277(98)00058-4. URL `https://www.sciencedirect.com/science/article/pii/S0010027798000584`.

J. Wright and Y. Ma. *High-dimensional data analysis with low-dimensional models: Principles, computation, and applications*. Cambridge University Press, 2022.

X. Wu and B. R. Brooks. Self-guided langevin dynamics simulation method. *Chemical Physics Letters*, 381(3-4):512–518, 2003.

H. Yoshida. Construction of higher order symplectic integrators. *Physics letters A*, 150(5-7):262–268, 1990.

H. Zhao and R. P. Wildes. Where are you heading? dynamic trajectory prediction with expert goal examples. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7609–7618, 2021. doi: 10.1109/ICCV48922.2021.00753.

T. Zhi-Xuan, J. Mann, T. Silver, J. Tenenbaum, and V. Mansinghka. Online bayesian goal inference for boundedly rational planning agents. *Advances in Neural Information Processing Systems*, 33:19238–19250, 2020.

T. Zhi-Xuan, N. Gothoskar, F. Pollok, D. Gutfreund, J. B. Tenenbaum, and V. K. Mansinghka. Solving the baby intuitions benchmark with a hierarchically bayesian theory of mind. *arXiv preprint arXiv:2208.02914*, 2022.

B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, et al. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa. Planning-based prediction for pedestrians. In *ICRIS*, pages 3931–3936. IEEE, 2009.

G. K. Zipf. *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio Books, 2016.