Exponential Convergence of Infeasibility Proofs for Kinematic Motion Planning*

Sihui Li[1][0000−0003−1766−4316] and Neil T. Dantam[1][0000−0002−0907−2241]

Department of Computer Science, Colorado School of Mines, USA
{li,ndantam}@mines.com

Abstract. Proving motion planning infeasibility is an important part of a complete motion planner. Common approaches for high-dimensional motion planning are only probabilistically complete. Previously, we presented an algorithm to construct infeasibility proofs by applying machine learning to sampled configurations from a bidirectional sampling-based planner. In this work, we prove that the learned manifold converges to an infeasibility proof exponentially. Combining prior approaches for sampling-based planning and our converging infeasibility proofs, we propose the term *asymptotic completeness* to describe the property of returning a plan or infeasibility proof in the limit. We compare the empirical convergence of different sampling strategies to validate our analysis.

1 Introduction

A complete motion planner returns a plan when one exists and reports failure when no plan exists [32]. Such complete motion planners would be desirable for high-level planning problems such as task and motion planning [14,18] that must solve motion planning as a sub-problem; when no motion plan exists, analyzing the infeasibility of motion planning would help eliminate search branches and improve efficiency. Significant previous work focused on finding feasible or optimal motion plans [19,22,23,26,28,38,39,44,49]. This paper addresses the complementary issue: finding infeasiblity proofs when no motion plan exists.

In [34], we introduced an algorithm to construct infeasibility proofs in motion planning by learning a manifold in the obstacle region to separate the start and goal configurations. The algorithm has two main steps. In the learning step, we train a classifier between points in the start tree and points in the goal tree, which are sampled from RRT-connect[33]. After learning the manifold, in the validation step, we triangulate it to validate that the manifold is entirely in the obstacle region. If we can find such an obstacle region manifold, then this manifold is an infeasibility proof because its existence prevents any collision free path between the start and goal configurations.

This paper analyzes the convergence of learning infeasibility proof manifolds and proves that, when no motion plan exists, the learned manifold converges to

* This work is supported in part by the National Science Foundation under Grant No. IIS-1849348

an infeasibility proof exponentially with respect to the number of configurations in the planning graph. The probability of learning the infeasibility proof depends on the number of configurations sampled close to the obstacle region. We first prove that the probability of sampling the necessary regions converges to one in the limit and then show that this convergence is exponential in the number of samples (see section 4). We validate convergence empirically and compare uniform sampling and Gaussian sampling (see section 5). This analysis addresses the learning step; further analysis of the validation step is part of future work.

For robust convergence, we require certain properties of the configuration space. In particular, the obstacle region must be non-infinitesimal, which we define precisely as *entirely ε-blocked* (see Definition 3), complementary to the notion of ε-goodness [25] on the free space. An entirely ε-blocked obstacle region is necessary to obtain sufficient samples to learn the manifold.

Unlike the conventional definition of completeness, where a positive or negative result must be returned in finite time, our analysis shows that the previously proposed algorithm returns a plan or an infeasibility proof in the limit. To describe this property, we propose the term *asymptotic completeness* for convergence to either a motion plan or an infeasibility proof as the number of samples increases.

In this work, we consider kinematic motion planning problems without differential constraints. We assume that infeasibility is only caused by the obstacle region and configuration space limits. That is, we do not consider steering functions [30], dynamics [17], or implicit constraints [4,27]. This assumption is valid for many scenarios, including purely kinematic movement of robot arms. Infeasibility proofs for more general cases remain part of our future work.

2 Related work

Sampling-based motion planners are widely used for high-dimensional motion planning problems [1,12,13,19,21,23,26,28,29,31,35,42,43]. Many sampling-based planners offer probabilistic completeness and exponential convergence in the number of samples, effective properties for cases when there is a feasible plan [33]. However, if there is no plan, these algorithms run until timeout without any guarantee on plan non-existence. This drawback creates challenges for the application of sampling-based planners in higher level planning problems [3,9,15,41]. In this paper, we analyze convergence of infeasibility proofs for the framework presented in [34], addressing the other side of the story for completeness.

Previous works have addressed different aspects of infeasibility proofs. Some previous work addressed motion planning infeasibility proofs for single objects. [46] proves path non-existence for single, rigid objects in a 2D or 3D workspace with the notion of caging, which applies to stable grasping of robot hand. [2] considers the simplified problem of a rigid body passing through a narrow gate with a discretized orientation of the object. These works do not have completeness guarantees.

Other works offer complete motion planning based on space decomposition. [37] decomposes the obstacle region into alpha-shapes and then queries the

connectivity of two configurations. [47] and [48] combine cell decomposition with a probabilistic roadmap (PRM). These algorithms are resolution-complete due to the underlying cell decomposition.

There are also exact algorithms based on cell decomposition. Vertical cell decomposition [11] works on lower dimensional (2D and 3D) configuration spaces with piecewise linear obstacle regions. Cylindrical algebraic decomposition [10] works on general configuration spaces defined by semi-algebraic sets. Planning with roadmaps generated from these decompositions is complete. However, there are challenges to represent the configuration space exactly, especially for higher dimensional manipulators [32].

Deterministic sampling-based motion planning provides certain guarantees on plan non-existence [8,20]. Using low-dispersion sampling strategies, if such a planner does not find a plan, then either no solution exists or a solution exists only through some narrow passage. However, these algorithms are not complete since the infeasibility guarantee of low-dispersion sampling is not definite.

Visibility [45] and sparsity [16] based planners also provide some infeasibility information. These methods add sampling configurations to a roadmap only when the configurations improve the planning graph in certain ways. Planning terminates when no further configurations can be added for a certain number (M) of consecutive samples, and the percentage of the free space not covered by the roadmap is estimated as $1/M$, so that if no plan is found when the algorithm terminates, the problem may be considered to be infeasible [40]. However, these methods do not prove plan nonexistence definitively since the percentage of not covered free space approaches 0 but can never be 0 exactly. In contrast, this paper analyzes convergence properties of [34] to show that the framework offers a definite infeasibility proof in the limit.

To summarize, this paper extends [34] to now prove that the learned manifold converges exponentially to a definite infeasibility proof with increasing samples. For this property of returning positive or negative results in the limit, we propose the term *asymptotic completeness* (see Definition 7).

3 Problem Definition

A motion planning problem consists of a configuration space $\mathcal{C}$ of dimension n, a start configuration $\boldsymbol{q}_{\text{start}}$, and a goal configuration $\boldsymbol{q}_{\text{goal}}$ [32]. The configuration space $\mathcal{C}$ is the union of the closed set obstacle region $\mathcal{C}_{\text{obs}}$ and the open set free space $\mathcal{C}_{\text{free}}$.

A feasible plan is defined as a path σ such that $\sigma[0,1] \in \mathcal{C}_{\text{free}}$, and $\sigma[0] = \boldsymbol{q}_{\text{start}}$, $\sigma[1] = \boldsymbol{q}_{\text{goal}}$. The solution to the motion planning problem is a feasible plan if one exists, or an infeasibility proof if no motion plan exists.

3.1 Infeasibility Proofs

We define an infeasibility proof as a closed manifold that is contained entirely in $\mathcal{C}_{\text{obs}}$ and that separates the start $\boldsymbol{q}_{\text{start}}$ and goal $\boldsymbol{q}_{\text{goal}}$ [34].

Definition 1: Infeasibility Proof

A manifold $\mathcal{M}$ in $\mathcal{C}$ defined by a continuous function $f(\boldsymbol{q}) = 0$ is an *infeasibility proof* if and only if,

(I) $\mathcal{M}$ is a closed manifold,
(II) $\mathcal{M}$ separates the start and the goal, $f(\boldsymbol{q}_{\text{start}})f(\boldsymbol{q}_{\text{goal}}) < 0$,
(III) $\mathcal{M}$ is contained entirely in $\mathcal{C}_{\text{obs}}$, $\forall f(\boldsymbol{q}) = 0$, $\boldsymbol{q} \in \mathcal{C}_{\text{obs}}$.

No plan exists if and only if there exists an infeasibility proof according to Definition 1; the proof is in [34]. We note that the proof manifold in Definition 1 may be either a smooth or piecewise manifold in the configuration space.

3.2 Configuration space requirements

Compared to typical sampling-based motion planners, we introduce additional requirements on the configuration space to robustly construct infeasibility proofs and support our analysis.

First, we must add a requirement to $\mathcal{C}_{\text{obs}}$, which is similar to the requirements on $\mathcal{C}_{\text{free}}$ for robustness and probabilistic completeness of sampling-based planners. *ε-goodness* addresses the need for an adequately large open space anywhere in $\mathcal{C}_{\text{free}}$ [25]. Similarly, *δ-clearance* provides for a robustly feasible path, which is necessary for probabilistic completeness to ensure the possibility of sampling in narrow spaces of $\mathcal{C}_{\text{free}}$ [23]. We now introduce an analogous notion that applies to $\mathcal{C}_{\text{obs}}$.

Definition 2: ε-blocked

We say a state $\boldsymbol{q}$ in $\mathcal{C}_{\text{obs}}$ is *ε-blocked* for any $\varepsilon > 0$, if a closed ball with radius ε centered at $\boldsymbol{q}$ is contained entirely in $\mathcal{C}_{\text{obs}}$.

Definition 3: Entirely ε-blocked

We say $\mathcal{C}_{\text{obs}}$ is *entirely ε-blocked* if for any $\boldsymbol{q}$ in $\mathcal{C}_{\text{obs}}$, there exists a neighboring configuration $\mathbf{p}$ within ε distance of $\boldsymbol{q}$, such that $\mathbf{p}$ is ε-blocked.

The following definitions and proofs in this paper assume an entirely ε-blocked $\mathcal{C}_{\text{obs}}$. The ε-blocked property ensures sufficient "thickness" of the obstacle region to sample and construct an infeasibility proof in $\mathcal{C}_{\text{obs}}$. If any region in $\mathcal{C}_{\text{obs}}$ is not ε-blocked, then the probability of sampling configurations necessary to construct an infeasibility proof in those regions would be zero. Theoretically, the proof manifold can be contained in such infinitesimal regions, but our current approaches cannot find such manifolds. We claim that the requirement for an entirely ε-blocked $\mathcal{C}_{\text{obs}}$ is acceptable for many robot manipulators because physical obstacles have nonzero volume.

Second, our analysis tightens the typical requirement of sampling-based planners to have a *metric space*. Instead, we assume the configuration space is a Euclidean space: configurations are real-valued vectors with a Euclidean distance metric. This is necessary since we need a manifold in Euclidean space for the triangulation step (refer to subsection 3.3). We claim the restriction to a Euclidean configuration space is acceptable for many robot manipulators where configurations are defined by joint angles; considering inner hardware arrangements (e.g., cabling) and outside physical environment, joint angles θ and $\theta + 2k\pi$, for integer k, are often not equivalent.

Finally, the configuration space boundaries (e.g., joint limits) are a special case of $\mathcal{C}_{\text{obs}}$. We treat the boundaries as fixed obstacle regions with a positive thickness to ensure the existence of $\mathcal{M}$ in general. We discuss how to deal with joint limits in more detail in subsection 4.2.

3.3 Summary of Infeasibility Proof Algorithm

We summarize the key points of the infeasibility proof algorithm in [34] before proceeding to the analysis in section 4. Please refer to [34] for more details.

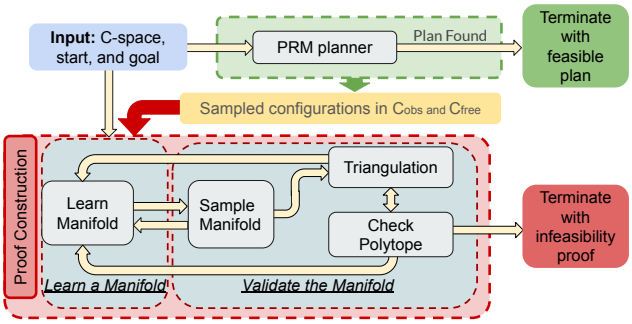


Fig. 1: Algorithm overview. The red block shows the infeasibility proof construction, which runs in parallel with a PRM planner.

3.3.1 Learn a Manifold Figure 1 shows the overall algorithm structure, which runs in parallel a sampling-based planner and the infeasibility proof construction, terminating with either a plan or an infeasibility proof. We start the infeasibility proof by learning a manifold using available sampled configurations. Compared to [34], which used RRT-connect as the sampling-based planner, this paper and analysis use a Probabilistic Roadmap (PRM) planner. RRT-connect is a bidirectional planner that grows two trees from the start and the goal. In contrast, a PRM is multi-directional. PRM construction samples all areas of the free space and includes all valid $\mathcal{C}_{\text{free}}$ samples in a graph, which applies to configuration spaces with multiple free space components. When $\mathcal{C}_{\text{free}}$ is disconnected, the PRM planner constructs a graph of free space configurations that has multiple, disconnected components (see supplementary material for an example comparing PRM with RRT-connect). We learn the manifold using supervised learning. The sampled configurations are the training data. Configurations in the graph

component containing the goal configuration are one class, and all configurations in other components are another class. Then, we train a two class classifier. Here, we use a support vector machine (SVM) with Radial Basis Function (RBF) kernel. Other classifiers may also fit within this framework, but the SVM has some notable benefits which we will discuss in subsection 4.1.

According to Definition 1, the manifold must be closed, continuous, and entirely in $\mathcal{C}_{\text{obs}}$. In section 4, we prove the learned manifold converges to meet these requirements with enough $\mathcal{C}_{\text{free}}$ samples. This convergence property is essential to all following steps.

3.3.2 Validate the Manifold After learning the manifold, we must verify that it is entirely in $\mathcal{C}_{\text{obs}}$. Directly proving that a manifold is inside an implicitly defined space ($\mathcal{C}_{\text{obs}}$) is difficult, so the idea is to tessellate the manifold, construct a reassembling polytope, then check that the polytope is entirely in $\mathcal{C}_{\text{obs}}$. For the tessellation, we sample points on the manifold and then use these points to construct a triangulation of the manifold with tangential Delaunay complexes [5,6]. The tangential Delaunay complexes construction requires a boundary-less underlying manifold. This triangulation step returns a polytope that approximates the manifold. Then, we check each facet of the polytope to prove it is entirely in $\mathcal{C}_{\text{obs}}$.

To summarize, our algorithm runs the multi-directional search and infeasibility proof construction in parallel. The proof construction learns a manifold and then checks whether the approximating polytope is contained in $\mathcal{C}_{\text{obs}}$. Next, we analyze the probability of the learned manifold becoming an infeasibility proof. In this paper, we focus on analysis of the learning part, the validation steps analysis is part of future work.

4 Convergence Analysis

In this section, we prove that the probability of the learned manifold being an infeasibility proof converges to one as more configurations are sampled in $\mathcal{C}_{\text{free}}$ and that convergence is exponential with the number of sampled configurations.

4.1 Requirements on the learning method

The infeasibility proof manifold is the decision boundary of a supervised learner. We summarize three requirements on the learning method, which is necessary for our proof of convergence.

First, the learning method must find a boundary-less manifold. This requirement is necessary for the manifold to satisfy Definition 1 (I). In our implementation [34], the triangulation algorithm [5] also requires a boundary-less manifold.

Second, the learning method must be able to find a decision boundary that completely separates the training data given that the data is separable. In other words, the learning method is flexible enough to fit all curvatures of the two classes' boundaries to achieve an accuracy of one on the training data. Unlike

a typical machine learning classification problem, over-fitting to training data is actually desirable in this case to completely separate the components of the configuration space. This requirement corresponds to Definition 1 (II).

Third, the learned manifold must be at least d distance away from any training data points, where d is a positive number. This requirement is necessary to "push" the manifold into $\mathcal{C}_{\text{obs}}$ to obtain the proofs in the following subsections. We use this requirement as a condition to prove that the manifold eventually satisfies Definition 1 (III). A positive d distance is possible because we require $\mathcal{C}_{\text{obs}}$ to be entirely ε-blocked (Definition 3). If $\mathcal{C}_{\text{obs}}$ is not entirely ε-blocked, at the place where $\mathcal{C}_{\text{obs}}$ becomes arbitrarily "thin", the two training classes' points could be arbitrarily close to each other, so a positive d distance cannot be maintained by any learning methods that completely separates the two classes.

Our implementation [34] uses a Radial Basis Function (RBF) kernel Support Vector Machine (SVM) to satisfy these three requirements. For the first requirement of a boundary-less manifold, the SVM directly provides a closed-form, differentiable, boundary-less function for the learned manifold. For the second requirement of complete separation, RBF kernel SVM has a hyper-parameter, the variance γ, which we can tune to achieve the necessary level of over-fitting (complete separation of training data). For the third requirement of positive distance from training data, the SVM enforces a margin between the two classes. In our case, the margin and d depends on the "thinnest" place of $\mathcal{C}_{\text{obs}}$.

4.2 Preliminary Definitions

We separate the configurations in the PRM into two classes to produce the training data. The first class contains all the configurations that are connectable to $\boldsymbol{q}_{\text{goal}}$, and the second class contains all other configurations. With this classification, we aim towards learning an infeasibility proof that encloses the goal region (Definition 4). Alternatively, we could learn an infeasibility proof that encloses the start region, and the following definitions and proofs would proceed similarly.

Definition 4 (Goal Region). *The* goal region $\mathcal{C}_{\text{goal}}$ *is the connected component of* $\mathcal{C}_{\text{free}}$ *that contains* $\boldsymbol{q}_{\text{goal}}$.

If no plan exists, then $\boldsymbol{q}_{\text{goal}}$ and $\boldsymbol{q}_{\text{start}}$ must be completely separated, i.e., $\boldsymbol{q}_{\text{start}} \notin \mathcal{C}_{\text{goal}}$. There are two cases that may cause this separation. In the first case, the separation is caused only by the obstacle region. In the second case, the separation is caused by a combination of the obstacle region and the configuration space boundaries.

To support learning the manifold and combine these two cases, we treat the configuration space boundaries as part of $\mathcal{C}_{\text{obs}}$ as stated in section 3. For the learning method to work for all kinds of configuration spaces, we define a region of virtual $\mathcal{C}_{\text{free}}$ to enclose the configuration space boundaries' obstacle region (see Figure 2). The virtual $\mathcal{C}_{\text{free}}$ regions and the configuration space boundaries' obstacle regions also need to be sampled.

Based on this modified configuration space, when $\boldsymbol{q}_{\text{goal}}$ and $\boldsymbol{q}_{\text{start}}$ are separated by $\mathcal{C}_{\text{obs}}$, the configuration space has three disconnected components: the goal

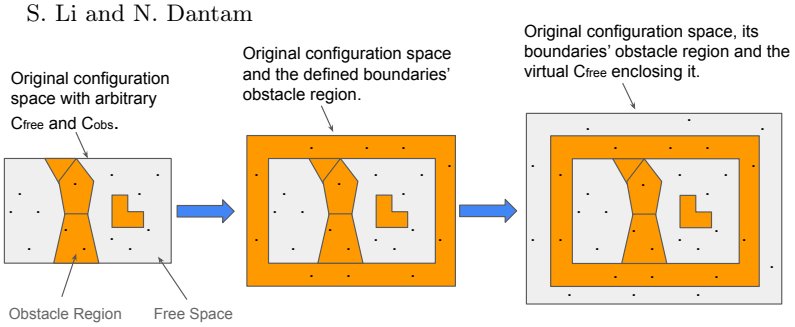


Fig. 2: Changes made to the configuration space boundaries to unify joint limits and $\mathcal{C}_{\text{obs}}$. With these modifications, we can treat the joint limits as a special obstacle region for the sake of learning the infeasibility proof.

region $\mathcal{C}_{\text{goal}}$, the obstacle region $\mathcal{C}_{\text{obs}}$, and the rest of the configuration space $\mathcal{C}_{\text{rest}}$. $\mathcal{C}_{\text{goal}}$ is a connected component according Definition 4. $\mathcal{C}_{\text{obs}}$ and $\mathcal{C}_{\text{rest}}$ are not necessarily connected components.

Next, we define the *inner boundary* and the *outer boundary*, which we will use for our convergence proof. Let $U(r, p)$ be a hyper-ball region centered at p with radius r.

Definition 5 (Inner Boundary (IB)). *The* inner boundary *is union of all configurations on $\mathcal{C}_{\text{obs}}$ boundaries such that there exists an arbitrarily small neighborhood U of the configuration that intersects the goal region,*

$$IB = \{\, p \in \mathcal{C}_{\text{obs}} \mid U(r,p) \cap \mathcal{C}_{\text{goal}} \neq \emptyset,\ \forall r > 0 \} \ .$$

Intuitively, the inner boundary consists of the obstacle region boundaries that are in contact with the goal region $\mathcal{C}_{\text{goal}}$. Similarly, we define the outer boundary.

Definition 6 (Outer Boundary (OB)). *The* outer boundary *is union of all configurations on $\mathcal{C}_{\text{obs}}$ boundaries such that there exists an arbitrarily small neighborhood U of the configuration that intersects $\mathcal{C}_{\text{rest}}$,*

$$OB = \{\, p \in \mathcal{C}_{\text{obs}} \mid U(r,p) \cap \mathcal{C}_{\text{rest}} \neq \emptyset,\ \forall r > 0 \} \ .$$

Intuitively, the outer boundary consists of the obstacle region boundaries that are in contact with rest of the configuration space $\mathcal{C}_{\text{rest}}$.

4.3 Proof of convergence

In this section, we prove that the probability of the learned manifold fulfilling the three requirements of an infeasibility proof defined in Definition 1 converges to one. Definition 1 (I) is satisfied by Lemma 2, Definition 1 (II) is satisfied by Proposition 1, and Definition 1 (III) is satisfied by Lemma 1. All following proofs are based on uniform sampling.

First, we address Definition 1 (II), ensuring that the learned manifold separates $\boldsymbol{q}_{\text{start}}$ and $\boldsymbol{q}_{\text{goal}}$.

Proposition 1. *The learned manifold always separates $\boldsymbol{q}_{\text{start}}$ and $\boldsymbol{q}_{\text{goal}}$.*

Proof. The separation of $\boldsymbol{q}_{\text{start}}$ and $\boldsymbol{q}_{\text{goal}}$ is a consequence of our requirement on the learning method in subsection 4.1. We assign $\boldsymbol{q}_{\text{start}}$ and $\boldsymbol{q}_{\text{goal}}$ to two different classes in the training data, so the requirement of a decision boundary that fully separates the training data ensures separation of $\boldsymbol{q}_{\text{start}}$ and $\boldsymbol{q}_{\text{goal}}$. □

Second, we address Definition 1 (III) by proving that the probability of the learned manifold being entirely in $\mathcal{C}_{\text{obs}}$ converges to one. We prove this convergence based on regions that, when sampled, are sufficient to learn the manifold contained in $\mathcal{C}_{\text{obs}}$. Then, we calculate the probability of failing to sample in all such regions for N sampled configurations, which converges to 0 as N gets larger. This approach parallels the convergence analysis of PRMs in [24], only we now analyze the convergence of infeasibility proofs.

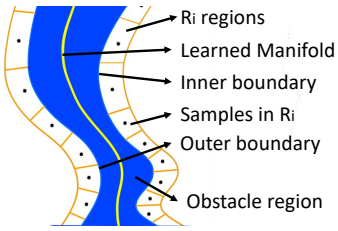


Fig. 3: Illustration of convergence proof in Lemma 1.

Lemma 1. *The probability of the learned manifold being entirely in $\mathcal{C}_{\text{obs}}$ converges to one as the number of sampled configurations in $\mathcal{C}_{\text{free}}$ approaches infinity.*

Proof. Based on the value of d described in subsection 4.1, we define a "coast" region C_r in $\mathcal{C}_{\text{free}}$ close to IB and OB,

$$C_r = \{\, p \in \mathcal{C}_{\text{free}} \mid \|p - q\| < d \ \wedge \ q \in IB \cup OB \} \ . \tag{1}$$

We decompose coast C_r into a set of small regions R_i such that $\bigcup_{i=1}^{N_R} R_i = C_r$. The number N_R depends on the smoothness of the coast. If the coast region is more complex, we need a finer decomposition and N_R needs to be larger. We decompose C_r finely enough such that when there is at least one configuration sampled in each R_i, the learned manifold cannot escape out of the R_is since the learned manifold must completely separate the samples (Figure 3). With the entirely ε-blocked requirement, a Lipschitz continuous manifold must exist in $\mathcal{C}_{\text{obs}}$, hich we can fit with a bounded, finite number of support vectors, so we claim N_r is a finite number.

According to the requirements on the learning method in subsection 4.1, the manifold must be at least d away from any data points. We also know that configurations sampled in each R_i are always less than d away from IB or OB by the definition of C_r. Combining these two conditions, if there is at least one configuration sampled in each R_i, it forces the learned manifold to be entirely in $\mathcal{C}_{\text{obs}}$. If some of the R_is contains no configurations, the learned manifold may not be contained in $\mathcal{C}_{\text{obs}}$. We note the probability that some of the regions are empty as P_{failure}; then, P_{success} is the probability that every region has at least one sampling configuration. When N configurations are sampled in $\mathcal{C}_{\text{free}}$, we calculate

P_{failure} as follows::

$$\begin{aligned} P_{\text{failure}} &\leq P_{\text{at least one of the } R_i\text{s is empty}} \\ &= P_{R_1 \text{ is empty}} + P_{R_1 \text{ is not empty}} * (P_{R_2 \text{ is empty}} \\ &\quad + P_{R_2 \text{ is not empty}} * (P_{R_3 \text{ is empty}} + \ldots)) \\ &\leq \sum_{i=1}^{N_R} P_{\text{R}_\text{i} \text{ is empty}} = \sum_{i=1}^{N_R} \left(1 - \frac{V(R_i)}{V(\mathcal{C}_{\text{free}})}\right)^N . \end{aligned} \tag{2}$$

For a given motion planning problem, N_R is a fixed number, V is a measurement for the sampling region volume, the probability of sampling in R_i is $V(R_i)/V(\mathcal{C}_{\text{free}})$ since we are assuming uniform sampling and each $V(R_i)/V(\mathcal{C}_{\text{free}})$ is a number between 0 and 1. When N satisfies $N > N_r$ and approaches infinity, P_{failure}'s upper bound approaches 0, so P_{failure} approaches 0. Then, we know $P_{\text{success}} = 1 - P_{\text{failure}}$ approaches 1. □

Third, we address Definition 1 (I) by proving that the learned manifold is a closed manifold when it is entirely in $\mathcal{C}_{\text{obs}}$.

Lemma 2. *The probability of the learned manifold being closed converges to one as the number of sampled configurations in $\mathcal{C}_{\text{free}}$ approaches infinity.*

Proof. We prove by contradiction, based on the requirements on the learning method in subsection 4.1 and Lemma 1. Assume that the probability of learning a closed manifold does not approach one. We require the learning methods to produce a manifold without boundary. If the manifold is open and boundaryless, the manifold would eventually escape from the finite obstacle region in the configuration space we defined, but Lemma 1 states the probability of the manifold being in $\mathcal{C}_{\text{obs}}$ approaches 1. Contradiction, the learned manifold must converge to a closed manifold. □

Theorem 1: Convergence

If no motion plan exists, then the probability that the learned manifold is an infeasibility proof converges to one as the number of sampled configurations in $\mathcal{C}_{\text{free}}$ approaches infinity.

Proof. Combining Proposition 1, Lemma 1, and Lemma 2, we conclude that as the number of sampled configurations in $\mathcal{C}_{\text{free}}$ approaches infinity, the probability that learned manifold satisfies all three conditions of Definition 1 converges to one. Thus, the learned manifold becomes an infeasibility proof. □

A *complete* motion planner returns a plan or reports plan non-existence in finite time. As we can see in the above analysis, learning of the infeasibility proof guarantees a probability of success approaching one as the number of sampled configurations approaches infinity, but at the same time, it does not guarantee a finite time result. This property is different from the definition of a complete

motion planner. It is also different from the definition of a *probabilistically complete* motion planner, since probabilistically complete motion planners never terminate (they may "time-out") when no plan exists. Based on these differences, we propose a different term, *asymptotically complete* motion planning.

Definition 7: Asymptotic Completeness

A motion planner is *asymptotically complete* if the probability to return a plan or correctly report plan non-existence converges to one as the number of sampled configurations approaches infinity.

The algorithm in [34] is asymptotically complete if validation step of the learned manifold always produces correct results given long enough time. In this paper, we focus on proving the learning part's convergence. The validation step's error depends on the triangulation step. According to [5] (Theorem 8.18), the triangulation with tangential Delaunay complexes has a bounded error which is parameterized by the sampling density. This means if we sample more points on the manifold, then the difference between the learned manifold and the triangulation becomes smaller, which would eventually generate a polytope entirely in $\mathcal{C}_{\text{obs}}$. A rigorous proof of the validation step is part of future work.

4.4 Proof of Exponential Convergence

Next, we analyze the rate of convergence and prove that the learned manifold converges exponentially to an infeasibility proof with the number of sampled configurations. Specifically, we prove that when no plan exists, P_{failure} decreases exponentially with the number of configurations sampled in $\mathcal{C}_{\text{free}}$. Extending the proof of Lemma 1, we calculate P_{failure} more precisely given the probability of sampling each of R_i regions. By viewing sampling as a sequence of independent trials, the convergence problem follows a binomial distribution. This approach parallels the exponential convergence proof for RRTs in [33], only we now show exponential convergence for infeasibility proofs.

Theorem 2: Exponential Convergence

If no motion plan exists, then the probability that the learned manifold is not an infeasibility proof after N configurations are sampled in $\mathcal{C}_{\text{free}}$ is at most $Ce^{-\frac{1}{2}Np}$, where C and p are constants.

Proof. Using uniform sampling, let the probability of sampling a configuration in R_i be p_i, then

$$p_i = \frac{V(R_i)}{V(\mathcal{C}_{\text{free}})} \geq \frac{V(R_{\min})}{V(\mathcal{C}_{\text{free}})} \ , \tag{3}$$

where $R_{\min}$ is the smallest R_i region.

Then, $p = V(R_{\min})/V(\mathcal{C}_{\text{free}})$ is a lower bound of sampling in any region R_i. After N configurations are sampled in $\mathcal{C}_{\text{free}}$, according to the binomial distribution model, we have

$$P_{\text{success}} \geq \binom{N}{N_R} p^{N_R} (1-p)^{N-N_R}, \tag{4}$$

where N_R is the number of R_i regions.

According to Chernoff bound on binomial distribution, $P(X \leq (1-\delta)\mu) \leq e^{-\mu\delta^2/2}$, where μ is the Binomial distribution's expected value, δ is a value between 0 and 1. In our case, $\mu = Np$. Here we want to calculate P_{failure}, which in Binomial distribution model form is $P(X \leq (N_R - 1))$. Let $\delta = 1 - (N_R - 1)/\mu = 1 - (N_R - 1)/Np$, when N is large enough, δ is between 0 and 1, we have the following,

$$\begin{aligned} P_{\text{failure}} &\leq P(X \leq (1-\delta)\mu) \\ &\leq e^{-\mu(1-(N_R-1)/\mu)^2/2} = e^{-Np/2+(N_R-1)-(N_R-1)^2/(2Np)} \\ &\leq e^{-N\frac{p}{2}+(N_R-1)} = \left(e^{N_R-1}\right)\left(e^{-\frac{1}{2}Np}\right) . \end{aligned} \tag{5}$$

Here, N_R and p are constants for a given motion planning problem, so P_{failure} has an upper bound of $Ce^{-\frac{1}{2}Np}$, where C is a constant. □

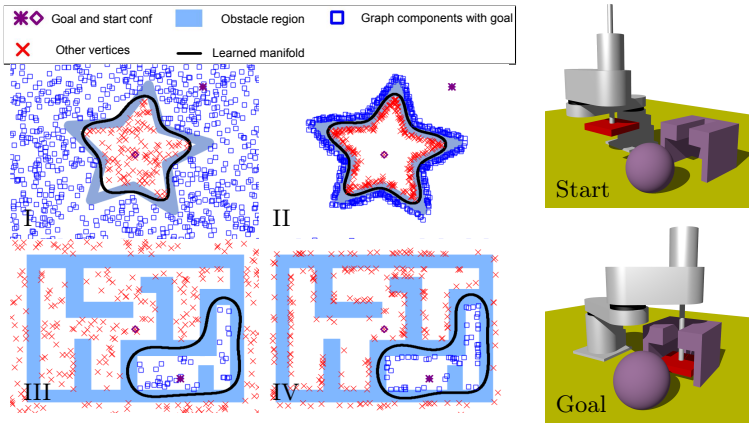
Theorem 2 shows that P_{failure} decreases exponentially with respect to the number of configurations in $\mathcal{C}_{\text{free}}$. In other words, the probability of the manifold becoming an infeasibility proof converges exponentially with the number of configurations sampled in $\mathcal{C}_{\text{free}}$.

The R_i regions come from a decomposition of the coast areas along IB and OB. This means if more configurations are sampled close to $\mathcal{C}_{\text{obs}}$, so that IB and OB is closely "covered" by sampled configurations, then the manifold is more likely to converge, i.e., converges faster. For this reason, sampling methods that find configurations close to $\mathcal{C}_{\text{obs}}$, such as Gaussian sampling [7], would offer faster convergence. We compare uniform and Gaussian sampling for manifold convergence in the experiments.

5 Experiments

In this section, we demonstrate Theorem 1 and Theorem 2 through experiments that learn the manifold with increasing number of sampled $\mathcal{C}_{\text{free}}$ configurations. We also compare uniform and Gaussian sampling to show the effect of different sampling distributions on manifold convergence.

We run experiments on two 2-DoF scenes (see Figure 4a) to visualize the impact of uniform vs. Gaussian sampling (sampling standard deviation is 0.1) and, and we run experiments for a 4-DoF manipulator (see Figure 4b) to validate the results on a practical (SCARA [36]) robot. We run 50 trials for each scene, where each trial increases the number of samples used for training and attempts to construct the infeasibility proof. Figure 5 shows the resulting success rates as the number of samples increase.



(a) 2 DoF scenes and the learned manifold.

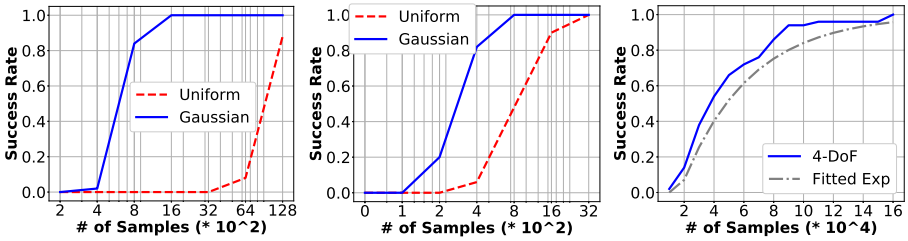
(b) 4 DoF scene.

Fig. 4: (a) 2-DoF scenes illustrating uniform vs. Gaussian sampling. (I) 800 uniform samples. (II) 800 Gaussian samples. (III) 300 uniform samples. (IV) 300 Gaussian samples. (b) 4-DoF scene using a SCARA [36] manipulator.

In Figure 5, we see that the ratio of successful infeasibility proofs approaches one with the increasing numbers of samples in $\mathcal{C}_{\text{free}}$, which supports Theorem 1, that the learned manifolds converges as more configurations are used for training. We can also see that the success rate approaches one exponentially (see Figure 5c, dotted grey line is a fitted $1 - b * e^{-ax}$ function as a comparison, here $a = 2.202782 * 10^{-5}, b = 1.442954$), which support Theorem 2, that the manifold converges exponentially.

Comparing Gaussian sampling with uniform sampling, as Figure 4a shows, Gaussian samples are close to $\mathcal{C}_{\text{obs}}$ while uniform samples are randomly distributed. In Figure 5a and Figure 5b, the blue line shows percentage of converged manifolds in 50 runs using Gaussian sampling, and the red line shows the results using uniform sampling. In the star scene (Figure 5a), all manifolds converges with 1600 configurations using Gaussian sampling, while some cases of uniform sampling are still not converged with 12800 configurations. In the maze scene, all Gaussian sampling cases converges at 800 configurations, while uniform sampling requires 3200 configurations to have the same percentage. These results supports our previous analysis that Gaussian sampling produces faster manifold convergence (subsection 4.4).

In addition, we see from Figure 5a and Figure 5b that the star scene needs more samples to converge in general. More samples are needed because the star-shaped obstacle has more sharp curves and thus needs a finer decomposition around the obstacle region's boundaries (more R_i regions). In the actual algorithm of [34], the manifold would converge with fewer samples because after we calculate manifold points, we heuristically add the manifold points in $\mathcal{C}_{\text{free}}$ back for training. These manifold points drive the learned manifold into $\mathcal{C}_{\text{obs}}$ faster since they are likely to be close to $\mathcal{C}_{\text{obs}}$. For example, in the star scene, the manifold points would exist at the inner star tips' curves, which would help retrain the manifold.



(a) The star scene (b) The maze scene (c) The 4 DoF manipulator

Fig. 5: Manifold converges as the number of samples increase. Comparing uniform sampling and Gaussian sampling, Gaussian sampling converges faster with respect to the number of configurations. 4-DoF scene uses Gaussian Sampling.

6 Conclusion and Discussion

We have proven that the learned manifolds converge exponentially to an infeasibility proof in the number of sampled configurations. To formalize the convergence analysis, we introduced two important concepts, ε-*blocked* and *asymptotic completeness*. Our analysis is supported by experiments showing success rates for increasing numbers of samples and comparing uniform and Gaussian sampling.

While the probability of the learned manifold being in the obstacle region converges to one with increasing samples, the algorithm to construct the infeasibility proof cannot immediately determine whether or not the manifold is in the obstacle region. Thus, after learning the manifold, we must validate that it is in the obstacle region. In our current implementation, validating the manifold dominates running time. Addressing this manifold validation is an important area of future work.

Furthermore, our current algorithm and analysis are limited to kinematic motion planning. Extending to a broader range of motion planning problems is part of future work.

References

1. Amato, N.M., Wu, Y.: A randomized roadmap method for path and manipulation planning. In: ICRA. vol. 1, pp. 113–120. IEEE (1996)
2. Basch, J., Guibas, L.J., Hsu, D., Nguyen, A.T.: Disconnection proofs for motion planning. In: ICRA. vol. 2, pp. 1765–1772. IEEE (2001)
3. Ben-Shahar, O., Rivlin, E.: Practical pushing planning for rearrangement tasks. T-RO **14**(4), 549–565 (1998)
4. Berenson, D., Srinivasa, S.S., Ferguson, D., Kuffner, J.J.: Manipulation planning on constraint manifolds. In: ICRA
5. Boissonnat, J.D., Chazal, F., Yvinec, M.: Geometric and Topological Inference. Cambridge University Press (2018), https://hal.inria.fr/hal-01615863, cambridge Texts in Applied Mathematics
6. Boissonnat, J.D., Ghosh, A.: Manifold reconstruction using tangential Delaunay complexes. Discrete & Computational Geometry **51**(1), 221–267 (2014)
7. Boor, V., Overmars, M.H., Van Der Stappen, A.F.: The Gaussian sampling strategy for probabilistic roadmap planners. In: RSS. vol. 2, pp. 1018–1023. IEEE (1999)

8. Branicky, M.S., LaValle, S.M., Olson, K., Yang, L.: Quasi-randomized path planning. In: ICRA. vol. 2, pp. 1481–1487. IEEE (2001)
9. Cambon, S., Alami, R., Gravot, F.: A hybrid approach to intricate motion, manipulation and task planning. IJRR **28**(1), 104–126 (2009)
10. Canny, J.: Computing roadmaps of general semi-algebraic sets. The Computer Journal **36**(5), 504–514 (1993)
11. Chazelle, B.: Approximation and decomposition of shapes. Algorithmic and Geometric Aspects of Robotics **1**, 145–185 (1985)
12. Şucan, I.A., Moll, M., Kavraki, L.E.: The open motion planning library. RAM **19**(4), 72–82 (2012)
13. Şucan, I.A., Kavraki, L.E.: A sampling-based tree planner for systems with complex dynamics. T-RO **28**(1), 116–131 (February 2012). https://doi.org/10.1109/tro.2011.2160466, http://dx.doi.org/10.1109/tro.2011.2160466
14. Dantam, N.T.: Task and Motion Planning. Springer Berlin Heidelberg (2020)
15. Dantam, N.T., Kingston, Z.K., Chaudhuri, S., Kavraki, L.E.: An incremental constraint-based framework for task and motion planning. IJRR **37**(10), 1134–1151 (2018)
16. Dobson, A., Bekris, K.E.: Sparse roadmap spanners for asymptotically near-optimal motion planning. IJRR **33**(1), 18–47 (2014)
17. Donald, B., Xavier, P., Canny, J., Reif, J.: Kinodynamic motion planning. JACM **40**(5), 1048–1066 (1993)
18. Garrett, C.R., Chitnis, R., Holladay, R., Kim, B., Silver, T., Kaelbling, L.P., Lozano-Pérez, T.: Integrated task and motion planning. Annual review of control, robotics, and autonomous systems **4**, 265–293 (2021)
19. Hsu, D., Kindel, R., Latombe, J.C., Rock, S.: Randomized kinodynamic motion planning with moving obstacles. IJRR **21**(3), 233–255 (2002)
20. Janson, L., Ichter, B., Pavone, M.: Deterministic sampling-based motion planning: Optimality, complexity, and performance. IJRR **37**(1), 46–61 (2018)
21. Janson, L., Schmerling, E., Clark, A., Pavone, M.: Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions. IJRR **34**(7), 883–921 (2015)
22. Kalakrishnan, M., Chitta, S., Theodorou, E., Pastor, P., Schaal, S.: STOMP: Stochastic trajectory optimization for motion planning. In: ICRA. pp. 4569–4574. IEEE (2011)
23. Karaman, S., Frazzoli, E.: Sampling-based algorithms for optimal motion planning. IJRR **30**(7), 846–894 (2011)
24. Kavraki, L.E., Kolountzakis, M.N., Latombe, J.C.: Analysis of probabilistic roadmaps for path planning. T-RO **14**(1), 166–171 (1998)
25. Kavraki, L.E., Latombe, J.C., Motwani, R., Raghavan, P.: Randomized query processing in robot path planning. JCSS **57**(1), 50–60 (1998)
26. Kavraki, L.E., Svestka, P., Latombe, J.C., Overmars, M.H.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. T-RO **12**(4), 566–580 (1996)
27. Kingston, Z., Moll, M., Kavraki, L.E.: Exploring implicit spaces for constrained sampling-based planning. IJRR **38**(10-11), 1151–1178 (2019)
28. Kuffner, J.J., LaValle, S.M.: RRT-connect: An efficient approach to single-query path planning. In: ICRA. vol. 2, pp. 995–1001. IEEE (2000)
29. Ladd, A.M., Kavraki, L.E.: Fast tree-based exploration of state space for robots with dynamics. In: Algorithmic Foundations of Robotics VI, pp. 297–312. Springer (2004)

30. Lafferriere, G., Sussmann, H.: Motion planning for controllable systems without drift. In: ICRA. pp. 1148–1149. IEEE Computer Society (1991)
31. LaValle, S.M.: Rapidly-exploring random trees: A new tool for path planning. Tech. Rep. TR-98-11, Computer Science Deptartment, Iowa State University (October 1998)
32. LaValle, S.M.: Planning algorithms. Cambridge university press (2006)
33. LaValle, S.M., Kuffner, J.J.: Randomized kinodynamic planning. IJRR **20**(5), 378–400 (2001)
34. Li, S., Dantam, N.T.: Learning proofs of motion planning infeasibility. In: RSS (2021)
35. Li, Y., Littlefield, Z., Bekris, K.E.: Asymptotically optimal sampling-based kinodynamic planning. IJRR **35**(5), 528–564 (2016)
36. Makino, H.: Assembly robot (July 1982), US Patent 4,341,502
37. McCarthy, Z., Bretl, T., Hutchinson, S.: Proving path non-existence using sampling and alpha shapes. In: ICRA. pp. 2563–2569. IEEE (2012)
38. Mukadam, M., Dong, J., Yan, X., Dellaert, F., Boots, B.: Continuous-time Gaussian process motion planning via probabilistic inference. IJRR **37**(11), 1319–1340 (2018)
39. Murray, S., Floyd-Jones, W., Qi, Y., Sorin, D.J., Konidaris, G.D.: Robot motion planning on a chip. In: RSS (2016)
40. Orthey, A., Toussaint, M.: Sparse multilevel roadmaps for high-dimensional robotic motion planning. In: ICRA. pp. 7851–7857 (2021). https://doi.org/10.1109/ICRA48506.2021.9562053
41. Ota, J.: Rearrangement of multiple movable objects-integration of global and local planning methodology. In: ICRA. vol. 2, pp. 1962–1967. IEEE (2004)
42. Otte, M., Frazzoli, E.: RRTX: Asymptotically optimal single-query sampling-based motion planning with quick replanning. IJRR **35**(7), 797–822 (2016)
43. Plaku, E., Bekris, K.E., Chen, B.Y., Ladd, A.M., Kavraki, L.E.: Sampling-based roadmap of trees for parallel motion planning. T-RO **21**(4), 597–608 (2005)
44. Schulman, J., Duan, Y., Ho, J., Lee, A., Awwal, I., Bradlow, H., Pan, J., Patil, S., Goldberg, K., Abbeel, P.: Motion planning with sequential convex optimization and convex collision checking. IJRR **33**(9), 1251–1270 (2014)
45. Siméon, T., Laumond, J.P., Nissoux, C.: Visibility-based probabilistic roadmaps for motion planning. Advanced Robotics **14**(6), 477–493 (2000)
46. Varava, A., Carvalho, J.F., Pokorny, F.T., Kragic, D.: Caging and path non-existence: a deterministic sampling-based verification algorithm. In: Robotics Research, pp. 589–604. Springer (2020)
47. Zhang, L., Kim, Y.J., Manocha, D.: A hybrid approach for complete motion planning. In: IROS. pp. 7–14. IEEE/RSJ (2007)
48. Zhang, L., Kim, Y.J., Manocha, D.: A simple path non-existence algorithm using c-obstacle query. In: Algorithmic Foundation of Robotics VII, pp. 269–284. Springer (2008)
49. Zucker, M., Ratliff, N., Dragan, A.D., Pivtoraiko, M., Klingensmith, M., Dellin, C.M., Bagnell, J.A., Srinivasa, S.S.: CHOMP: Covariant Hamiltonian optimization for motion planning. IJRR **32**(9-10), 1164–1193 (2013)