Learned Critical Probabilistic Roadmaps for Robotic Motion Planning

Brian Ichter and Aleksandra Faust
Robotics at Google, Mountain View, CA, USA
ichter@google.com, sandrafaust@google.com

Robot motion planning computes a collision-free, dynamically feasible, and low-cost trajectory from an initial configuration to goal region [1]. Sampling-based motion planning (SBMP) approaches, such as probabilistic roadmaps (PRMs) [2], efficiently solve complex planning problems through a set of probing samples to construct a roadmap, an implicit topological graph of the robot’s configuration space. To connect the initial state and goal region, PRMs search the roadmap and identify a sequence of configurations and local connections which the robot may traverse. Though these algorithms can form arbitrarily accurate representations as the number of samples increases to infinity, in practice, only a few critical states are necessary to parameterize solution trajectories. Often these critical states enjoy significant structure, for example entries to narrow passages, yet are only identified through exhaustive sampling [3].

We present a method that learns to recognize critical configurations and use them to construct a hierarchical PRM. These critical configurations are quantified through betweenness centrality [4], a graph-theoretic measure of centrality based on a sample’s importance to shortest paths through a graph, followed by a smoothing step that retains critical samples that are necessary for planning. Online, given a new planning problem, we construct a Critical Probabilistic Roadmap, which samples a small number of critical configurations and large number of non-critical configurations. The non-critical samples are connected locally, preserving the theoretical guarantees of SBMP, while the critical configurations are connected to all samples, providing critical edges through the graph. We show two orders of magnitude improvement in solving a narrow passage problem.

Related Work. Learned sample distributions for robotic motion planning have been studied recently [3], [5], [6], [7], [8] and [9] use offline solution trajectories to learn a distribution of samples and bias sampling towards regions where an optimal solution might lie. [6] learns a local sampler to bias sampling towards. Finally [7] learns to identify critical regions for sampling based on images of successful solutions and leverages these samples by growing trees from them. In this work, we identify critical regions from graph theoretical techniques in the configuration space and incorporate these samples into a Critical Roadmap, which allows the critical samples to connect throughout the configuration space.

Several approaches exist to compute a sample’s “criticality”; the most promising we identified were: label propagation [8] (Fig. 1a), minimum k-cuts [9] (Fig. 1c), and betweenness centrality [4] (Fig. 1d). Label propagation finds reasonable results for very narrow passage problems, but the results are unstable and poorly defined for problems with less constrained bottlenecks (Fig 1b). Minimum k-cuts can identify many critical samples, but require a fixed number of cuts be provided which can result in too few cuts, thus ignoring critical regions, or too many, thus identifying non-critical regions, e.g., corners. Fig. 1c. Furthermore, minimum k-cuts is significantly slower than the other approaches. Ultimately, we selected betweenness centrality, which captures the number of all-pairs shortest paths that pass through a given node, as it is fast to compute, stable, and principled.

Fig. 1: Approaches for identification of critical samples: (1a-1b) Label propagating approaches fail with moderately narrow passages. (1c) Minimum k-Cuts identify several non-critical samples. (1d) Betweenness centrality was found to be principled and perform well. The size and color of each sample is proportional to criticality. (1e) The learned criticality prediction. Critical PRM states are sampled proportional to their criticality.

I. CRITICAL SAMPLE IDENTIFICATION AND LEARNING

The first phase of the algorithm learns to identify critical samples from a set of PRMs generated for a family of training environments. This phase generates the critical sample dataset and then trains a predictor deep neural net model conditioned on the planning environment.

Dataset Creation. For a given free configuration space $X_{\text{free}}$, we construct a standard PRM $G$ with samples $\{x_i\}_{i \in [1..n]} \in X_{\text{free}}$ and edges $(x_i, x_j)$ for $i, j \in [1..n]$ if and only if the trajectory from sample $x_i$ to $x_j$ is collision free and the samples are within a connection radius $r_n$, as defined in [10]. With these roadmaps in hand, we compute the criticality of each sample via the betweenness centrality – the number of all-pairs shortest paths in a graph that pass through a given node. We make two alterations to betweenness centrality to adapt it to the motion planning problem and the
complexity of PRMs. First, we only compute an approximate value by solving \( m \) shortest path problems with a randomly chosen initial node to all other graph nodes (note that if \( m = n \), this is exact). Each time a node is used in a shortest path, its centrality score is incremented. Secondly, we add a smoothing step to discount samples that can be skipped along the shortest path. Essentially, for a collision-free path that traverses nodes \( x_1, x_{i+1}, x_{i+2} \), if the connection between node \( x_1 \) and \( x_{i+2} \) is collision-free, then node \( x_{i+1} \) is not critical to the path, and thus its score should not be incremented. This step is necessary to eliminate samples that are simply in the free space trajectory between critical samples and used due to the limited \( r_n \) connection radius.

**Training.** The computed centrality values become labels for training a neural network \( h_\theta(x, y) \), parameterized by \( \theta \), where \( x \in \mathcal{X}_{\text{free}} \) is a configuration sample and \( y \) is a representation of \( \mathcal{X}_{\text{free}} \). Herein, \( y \) is the list of workspace coordinates of the obstacles. Alternatively, it can be an occupancy grid, or another free space representation. We minimize \( L_2 \) loss to learn \( \theta \).

**Algorithm 1 Online Critical PRM Construction**

1. **Input:** Planning problem \( (\mathcal{X}_{\text{free}}, x_{\text{init}}, x_{\text{goal}}) \), \( \lambda \), \( n \)
2. Sample \( n \) configurations and compute criticality with \( h_\theta(x_i, y) \).
3. Select \( \lambda \log(n) \) critical samples proportional to criticality.
4. Connect critical samples to all samples.
5. Connect non-critical samples within an \( r_n \) radius.
6. Connect \( x_{\text{init}} \) and \( x_{\text{goal}} \) globally into the Critical PRM.
7. Search Critical PRM for shortest path from \( x_{\text{init}} \) to \( x_{\text{goal}} \).

**Complexity:** The complexity of Critical PRM remains \( \mathcal{O}(n \log(n)) \) as the \( n - \lambda \log(n) \) uniform samples are locally connected and maintain the standard \( \mathcal{O}(n \log(n)) \) complexity [10]. The \( \lambda \log(n) \) critical samples are connected to all \( n \) neighbors, requiring no nearest neighbor lookup and \( n \lambda \log(n) \) constant time connections and collision checks.

**Probabilistic Completeness and Asymptotic Optimality:**

The theoretical guarantees of probabilistic completeness and asymptotic optimality from [11], [12], [10] hold for this method by adjusting any references to \( n \) (the number of samples) to \( (n - \lambda \log(n)) \) (the number of uniform samples in our methodology). This result is detailed in Appendix D of [11] and Section 5.3 of [12], which show that adding samples can only improve the solution.

**III. RESULTS AND FUTURE WORK**

**Results.** As a proof of concept, we consider the narrow passage problem shown in Fig. 2 in which the space is divided into four regions connected by three narrow passages (with randomly generated locations). The Critical PRM achieves a two order of magnitude improvement in success rate and one order of magnitude in path length (Fig. 3). While we acknowledge this narrow passage dominated environment is well suited towards Critical PRM, we also expect that as the complexity of the planning problem increases (due to dimensionality or differential constraints) the effects of Critical PRM will also increase.

**Future Work.** We plan to address this last statement by applying Critical PRMs to complex systems and differentially constrained robots. We further plan to investigate to what degree critical samples can be generated based only on local features. Finally, long-term, we plan to explore to what extent similarities between critical samples in SBMP can be exploited in reinforcement learning with these techniques.

**ACKNOWLEDGEMENTS**

The authors thank Vincent Vanhoucke, Chase Kew, Edward Schmerling, James Harrison, and Marco Pavone for insightful discussions.
REFERENCES


