Improving Efficiency of Sampling-based Motion Planning via Message-Passing Monte Carlo

Makram Chahine
CSAIL, MIT
chahine@mit.edu

T. Konstantin Rusch
ELLIS Institute Tübingen &
Max Planck Institute for Intelligent Systems &
Tübingen AI Center
tkrusch@tue.ellis.eu

Zach J. Patterson
Case Western Reserve University
Mechanical and Aerospace Engineering
zpatt@case.edu

Daniela Rus
CSAIL, MIT
rus@csail.mit.edu

Abstract: Sampling-based motion planning methods, while effective in high-dimensional spaces, often suffer from inefficiencies due to irregular sampling distributions, leading to suboptimal exploration of the configuration space. In this paper, we propose an approach that enhances the efficiency of these methods by utilizing low-discrepancy distributions generated through Message-Passing Monte Carlo (MPMC). MPMC leverages Graph Neural Networks (GNNs) to generate point sets that uniformly cover the space, with uniformity assessed using the the $\mathcal{L}_p$-discrepancy measure, which quantifies the irregularity of sample distributions. By improving the uniformity of the point sets, our approach significantly reduces computational overhead and the number of samples required for solving motion planning problems. Experimental results demonstrate that our method outperforms traditional sampling techniques in terms of planning efficiency.

Keywords: Graph Neural Networks, Discrepancy Theory, Motion Planning

1 Introduction

Sampling-based motion planning is a key method for robot navigation in complex environments. From Probabilistic Roadmaps (PRMs) [1] to Rapidly-exploring Random Trees (RRTs) [2], the focus of much of the research in this area has been on developing more efficient and optimal motion planning algorithms [3, 4] that can handle high-dimensional spaces [5, 6], various types of constraints [7, 8], and dynamic environments [9, 10].

Instead of refining motion planning algorithms, our work targets the fundamental building block underlying these approaches—the core sampling process itself. We introduce a novel sampling strategy based on Message-Passing Monte Carlo (MPMC) [11], a graph neural network architecture trained to generate low-discrepancy point sets on unit hypercubes of arbitrary dimension. We expand the generic MPMC algorithm with the introduction of a novel training objective tailored to high-dimensional spaces, ensuring that generated points are optimally distributed and scalable.

This leads to a potent unbiased state sampling technique, that can be seamlessly integrated into any sampling-based planner, and that requires neither conditioning on the workspace description, nor a steer function, nor past examples, nor start and goal information. This approach provides strong theoretical guarantees and it also outperforms traditional techniques across various benchmarks, in environments of varying complexity and dimensionality.

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

Our contributions are stated as follows:

- We introduce the first application of MPMC neural network point set generation in motion planning, a significant novelty for unbiased sampling techniques.
- We propose a novel training objective tailored to high-dimensional configuration spaces, ensuring the generated points are well-suited for complex planning problems.
- We support our approach with rigorous theoretical justification, linking the training objective of MPMC point sets to a tighter upper bound on the distance from the optimal path.
- We establish superior planning efficiency against the current gold standard sampling approach in motion planning on a variety of PRM benchmarks, including challenging high-dimensional environments.
- We demonstrate the effectiveness of our sampling technique on a real-world UR5 robot arm, showing its potential for practical deployment in robotics.

Our MPMC-based sampling strategy offers a powerful and versatile alternative to traditional sampling methods. By targeting the sampling process directly, we open up new possibilities for improving the efficiency of a wide array of motion planning algorithms in robotic systems.

2 Related work

2.1 Learning for sampling-based planners

The incorporation of machine learning into motion planning has opened up new ways to accelerate pathfinding by learning from past experiences. One common strategy involves storing and reusing previously computed paths or solutions. For example, methods such as path libraries [12], sparse roadmaps [13], and local obstacle roadmaps [14] allow a robot to retrieve and adapt previously successful solutions to new, but similar, planning problems. These approaches reduce the computation time by narrowing the search space using knowledge from past instances.

Another line of research enhances the sampling process by learning distributions that guide planners toward more promising regions of the configuration space. Some methods employ problem-invariant distributions [15], while others adapt based on the workspace environment [16]. Deep learning further extends these concepts by learning from prior planning tasks, enabling distributions that condition on both the workspace and specific start-goal configurations of new problems [17]. These learned distributions effectively bias the sampling process, improving convergence rates and solution quality in new planning scenarios.

In contrast, our method departs from these past learning-based approaches by introducing a deterministic sampling strategy obtained via neural network training that does not depend on conditioning or prior knowledge.

2.2 Low-discrepancy constructions

Over the past century, numerous constructions of low-discrepancy point sets and sequences have been proposed. Most constructions are deeply rooted in number theory and abstract algebra. A widely used building block of low-discrepancy constructions is the one-dimensional van der Corput sequence [18] in base b, which is generalized to a higher dimensional setting via the Halton sequence [19]. Each of the d coordinates in a Halton sequence corresponds to a distinct van der Corput sequence in base b, with the bases selected to be co-prime. Faure sequences, as introduced in [20], offer a similar construction to Halton sequences but incorporate permutations of the digits in base b.

Another broad class of low-discrepancy constructions are today known as digital (t, s)-sequences, first introduced in [21]. These constructions include the widely known Sobol sequence [22] which is constructed using tools from linear algebra involving primitive polynomials and well-chosen generating matrices defined over finite fields.

A further distinct approach, rooted in a different branch of number theory, emerged with Korobov's introduction of the good-lattice method [23]. This technique utilizes modular arithmetic and prime number properties to construct a structured, grid-like set of integration nodes. Since its introduction, lattice rules have undergone significant extensions and refinements. Comprehensive discussions and valuable reviews of these developments are available in [24, 25, 26, 27].

Several approaches have recently emerged targeting the construction of points for fixed dimension and number of points. In [28], new low-discrepancy point sets were proposed by optimizing permutations applied to the Halton sequence. Another approach, known as subset selection, was introduced to select $k < N$ points from an N-element set that minimize the discrepancy. An exact algorithm for this selection was presented in [29], while a swap-based heuristic approach was employed in [30]. Additionally, [31] proposed a non-linear programming method to generate point sets with optimal star-discrepancy for fixed dimension and number of points. However, this approach faces significant computational challenges, limiting its practical application to finding optimal sets for only up to 21 points in two dimensions and 8 points in three dimensions. MPMC significantly differs from all previous approches by its explicit use of machine learning. Moreover, it has been shown in [11] that MPMC generates point sets with significantly better distributional properties compared to any previous method, reaching optimal or near-optimal discrepancy.

3 Methods

Message-Passing Monte Carlo (MPMC) is a machine learning approach designed to generate low-discrepancy point sets, which are the key for efficiently covering space in a uniform manner. MPMC leverages Graph Neural Networks (GNNs) and tools from Geometric Deep Learning to generate these point sets. The method focuses on the geometric properties needed to ensure uniformity and it is highly versatile for generating points across different dimensions. In this section we describe MPMC and its extensions for the motion planning application domain.

3.1 Message-Passing Monte Carlo Sampling

Message-Passing Monte Carlo (MPMC) [11] leverages Graph Neural Networks (GNNs) to generate point sets that cover the space in a uniform manner (for an extension to non-uniform distributions see [32]). This uniformity can be assessed through measures of irregularity termed discrepancy. While there exist a plethora of different uniformity measures, we focus on the $\mathcal{L}_p$-discrepancy here. That is, given a set of points $\{\mathbf{X}_i\}_{i=1}^N$ in the unit hypercube $[0,1]^d$ and $p \geq 1$, the $\mathcal{L}_p$-discrepancy is defined as,

$$\mathcal{L}_p^p(\{\mathbf{X}_i\}_{i=1}^N) := \int_{[0,1]^d} \left| \frac{\#(\{\mathbf{X}_i\}_{i=1}^N \cap [0,\mathbf{x}))}{N} - \mu([0,\mathbf{x})) \right|^p d\mathbf{x}, \tag{1}$$

where $\#(\{\mathbf{X}_i\}_{i=1}^N \cap [0,\mathbf{x}))$ counts how many points of $\{\mathbf{X}_i\}_{i=1}^N$ fall inside the box $[0,\mathbf{x}) = \prod_{i=1}^d [0,x_i)$ for $\mathbf{x} = (x_1,\ldots,x_d) \in [0,1]^d$, and $\mu(\cdot)$ denotes the usual Lebesgue measure. A differentiable closed-form solution to the high-dimensional integral in (1) for the case of $p = 2$, known as Warnock's formula [33], can be computed in $\mathcal{O}(N^2 d)$. Consequently, MPMC utilizes the $\mathcal{L}_2$-discrepancy as the training loss to generate point sets with low-discrepancy. The MPMC model transforming random input points into low-discrepancy sets via deep GNNs is depicted in Fig. 1.

3.1.1 Fast MPMC training in higher dimensions

As described in [11], the $\mathcal{L}_2$-discrepancy fails to distinguish random from highly uniform points in high dimensions for moderate amount of points. Therefore, it was suggested in [11] to minimize the Hickernell $\mathcal{L}_2$-discrepancy $D_{H,2}$ [34] in this case, that sums over all $\mathcal{L}_2$-discrepancies of k-dimensional projections, with $1 \leq k \leq d$. A straightforward implementation of $D_{H,2}$ has a computational complexity of $\mathcal{O}((2^d - 1)N^2 d)$, which would involve evaluating the $\mathcal{L}_2$-discrepancy for billions of projections for $d \geq 30$. To mitigate this issue, the authors in [11] limit the summation to smaller, randomly selected subsets of projections. In contrast, in this work we leverage the closed-

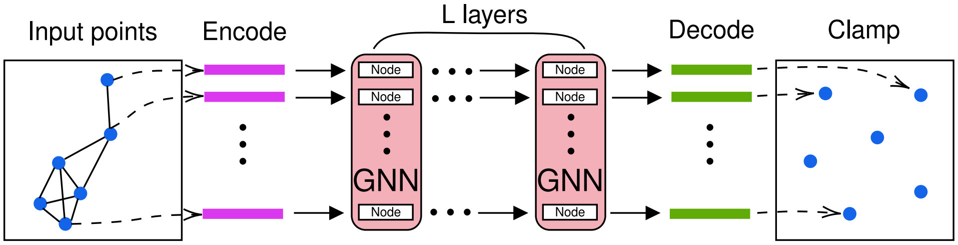

Figure 1: Schematic of the MPMC model reproduced from [11]. First, random input points are encoded to a high dimensional representation. Second, the encoded representations are passed through a deep GNN, where the underlying computational graph is constructed based on nearest neighbors using the positions of the initial input points. Finally, the node-wise output representations of the final GNN layer are decoded and clamped back into the unit hypercube.

form solution of the Hickernell $\mathcal{L}_2$-discrepancy from [35] that can be computed again in $\mathcal{O}(N^2 d)$ instead of $\mathcal{O}((2^d - 1)N^2 d)$ [36]. More concretely,

$$\begin{aligned} D^2_{H,2}(\{\mathbf{X}_i\}_{i=1}^N) &= \sum_{\emptyset \neq s \subseteq \{1,\dots,d\}} \mathcal{L}_2^2(\{\mathbf{X}_i^{(s)}\}_{i=1}^N) \\ &= \left(\frac{4}{3}\right)^d - \frac{2}{N}\sum_{i=1}^N \prod_{k=1}^d \left(\frac{3}{2} - \frac{\mathbf{X}_{i,k}^2}{2}\right) + \frac{1}{N^2}\sum_{i=1}^N\sum_{j=1}^N\prod_{k=1}^d [2 - \max(\mathbf{X}_{i,k}, \mathbf{X}_{j,k})], \end{aligned}$$

where $\mathbf{X}_{i,k}$ is the k-th entry of $\mathbf{X}_i$, and $\{\mathbf{X}_i^{(s)}\}_{i=1}^N$ is the projection of $\{\mathbf{X}_i\}_{i=1}^N$ onto $[0,1]^{|s|}$. This enables very fast training in higher dimensions, minimizing the discrepancy over all projections.

3.2 Motion planning via Probabilistic Road Maps

Algorithm 1 ps-PRM Algorithm

```
$S \leftarrow$ Sample($N$)
$S_v \leftarrow$ PruneInvalid($S$)
$V \leftarrow \{\mathbf{X}_{\text{start}}, \mathbf{X}_{\text{goal}}\} \cup S_v$
$E \leftarrow \emptyset$
for all $v \in V$ do
    $P_{\text{near}} \leftarrow$ Near($V \setminus \{v\}, v, r_N$)
    for all $p \in P_{\text{near}}$ do
        if CollisionFree($v, p$) then
            $E \leftarrow E \cup \{(v,p)\} \cup \{(p,v)\}$
        end if
    end for
end for
return ShortestPath($\mathbf{X}_{\text{start}}, \mathbf{X}_{\text{goal}}, V, E$)
```

In this work, the focus is on the efficiency of sampling-based motion planning in the sense of task success with respect to number of points sampled. Furthermore, for convenience in dealing with pre-trained sets of points of fixed number, we use a particular instantiation of the Probabilistic Roadmap (PRM) algorithm, which we refer to as ps-PRM (for pre-sampled PRM). This specific version of the PRM algorithm is described in Algorithm 1.

The algorithm begins by sampling N points from the entire space to form the set S. Next, we prune out milestones from the initial set S that do not fall in free space, keeping only valid samples S_v, which are combined with the start and goal $\{\mathbf{X}_{\text{start}}, \mathbf{X}_{\text{goal}}\}$ to obtain V. For each node v in V, the algorithm identifies nearby nodes within a radius r_N, with the set of such points denoted P_{near}. For each neighbor p in P_{near}, the algorithm checks for collision-free paths between v and p. If the path is free of collisions, a bidirectional edge is added between v and p to the edge set E. Finally, the algorithm attempts to find a shortest path from $\mathbf{X}_{\text{start}}$ to $\mathbf{X}_{\text{goal}}$ and returns the path if one exists; otherwise, it indicates failure.

3.3 Theoretical guarantee

Next, we outline a theoretical justification of minimizing discrepancy as a means to improving efficiency of sampling-based motion planning. To this end, we introduce another uniformity measure known as dispersion, which is commonly used in assessing the efficiency of sampling-based motion planning. Given a point set $\{\mathbf{X}_i^s\}_{i=1}^N$ in $[0,1]^d$ and $p \geq 1$, the l_p-dispersion is then defined as,

$$\mathcal{D}_p(\{\mathbf{X}_i\}_{i=1}^N) = \sup_{\mathbf{s}\in[0,1]^d} \min_{1\leq i\leq N} \|\mathbf{s} - \mathbf{X}_i\|_p.$$

Dispersion is closely related to discrepancy through the following inequality established in [37],

$$\mathcal{D}_\infty(\{\mathbf{X}_i^s\}_{i=1}^N) \leq \mathcal{L}_\infty(\{\mathbf{X}_i^s\}_{i=1}^N)^{1/d},$$

with the $\mathcal{L}_\infty$-discrepancy. Based on this and following [38], one can provide a deterministic sampling guarantee. Namely, using the PRM planning algorithm, assuming some mild assumptions are satisfied (e.g., feasible path δ-clearance), and choosing a radius r_N of the PRM algorithm based on N samples according to,

$$r_N = 2\alpha\sqrt{d}\mathcal{L}_\infty(\{\mathbf{X}_i^s\}_{i=1}^N)^{\frac{1}{d}},$$

for some $\alpha > 1$ and r_N satisfying conditions corresponding to the path δ-clearance, one can guarantee that the cost of the returned path is within a factor of $\frac{1}{\alpha-1}$ of the optimal δ-clear path. Clearly, a low-discrepancy induces a guarantee of the resulting path to be close to the optimal δ-clear path.

Note. The MPMC points are generated using the Hickernell $\mathcal{L}_2$-discrepancy loss, while our guarantee is related to the $\mathcal{L}_\infty$-discrepancy. Though the relation between $\mathcal{L}_\infty$ and $\mathcal{L}_2$ remains an open question in discrepancy theory, empirical evidence suggests the two are very close to each other at least for medium numbers of points [11]. Elsewhere, a clarification on $\mathcal{D}_\infty$ dispersion used in the referenced result from [38] and its relationship to grids of points is provided in Appendix A.3.

4 Experimental Evaluation

We evaluate our motion planning solution in simulation, using a variety of environments ranging from 2D mazes to higher dimensional spaces and in a physical setting with a UR5 robot arm. We assess performance according to the number of sampled points and success rate in finding a path.

4.1 Sampling methods

The performance of our method is evaluated against Uniform sampling and also against Halton [39] and Sobol [40] (for the 2-D maze setup only) sequences, two widely-used low-discrepancy Quasi-Monte Carlo methods (see Appendix A.1 for 2D-visualization). Sobol sequences are designed to efficiently cover spaces by minimizing gaps between sample points, making them ideal for complex problems. Halton sequences generate points using prime number bases, providing effective uniform coverage. However, both approaches suffer from correlation issues in higher dimensions [11].

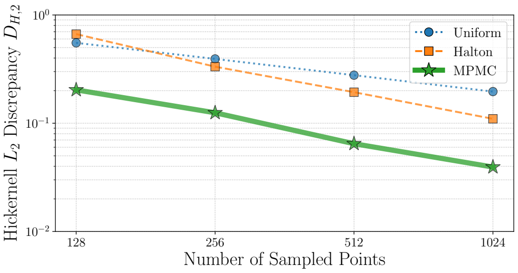

Figure 2: Hickernell $\mathcal{L}_2$-discrepancy in $d = 10$ for Uniform, Halton and MPMC sampling.

To obtain a statistical measure of performance across repeated runs, we randomize the selection of the MPMC points from the pool of trained point sets (batches can contain 8, 16 or 32 point sets based on the training instance). Similarly for the deterministic Halton and Sobol sequences, we initialize a new sequence where the previous one ended. Hence, when sampling point sets of size N, we take the points indexed from $(i-1)\,N+1$ to $i * N$ for the i-th run, with i taking integer values from 1 to the number of runs fixed for the experiment.

To illustrate differences between the sampling techniques considered in the context of this work, we provide the Hickernell $\mathcal{L}_2$-discrepancies of point sets of sizes $\{128, 256, 512, 1024\}$ in 10 dimensions in Figure 2. Halton sequences never surpass a 2 times advantage over Uniform sampling, and can even offer worse discrepancy for smaller N. MPMC points, on the other hand, reach substantially lower Hickernell $\mathcal{L}_2$-discrepancy, consistently close to 3 times lower than that of Halton and achieving up to a 5-fold improvement over Uniform sampling for larger point sets.

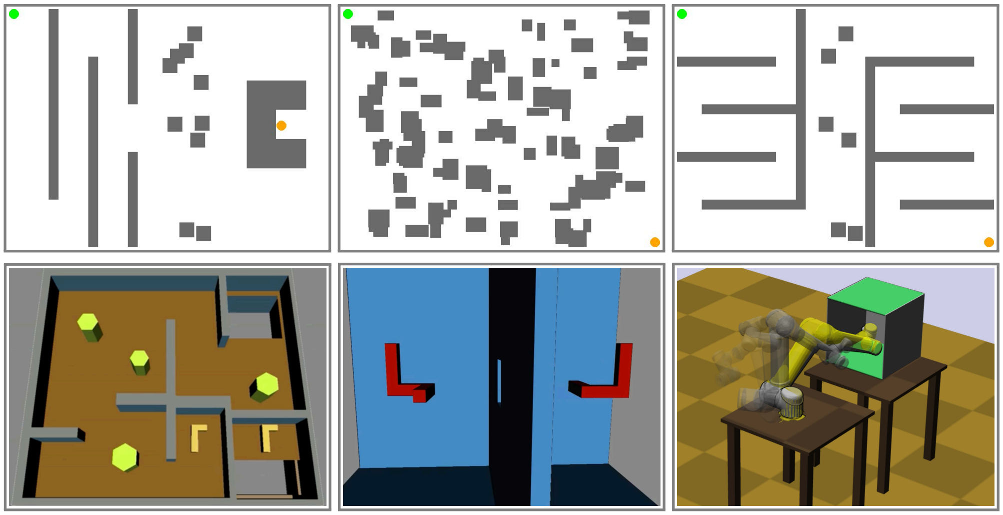

Figure 3: Experiments of increasing task complexity: 2D mazes (top) with start (green) and goal (orange) for basic navigation, followed by SE(3) (bottom left/center) for complex object rotation/translation, and 5-D UR5 (bottom right) for constrained robotic manipulation.

4.2 Experiments

2-D maze. The goal is to navigate a series of three 2-D mazes depicted in Figure 3. These are designed only to be increasingly difficult to solve for PRM with Uniform sampling before comparing to the other sampling techniques to avoid introducing any bias favoring a specific scheme. The maps are of size 640 by 480 and the agent is assumed to be a disk of radius 6 for collision checking. A single start and end goal is considered and 50 planning attempts are run by level, number of sampled points and sampling technique. The shortest path solver used is `A*` [41].

OMPL benchmarks All of the Uniform, Halton and MPMC sampling schemes are put to the test on experiments from the popular Open Motion Planning Library (OMPL) [42]. The base code PRM implementation is only modified to account for our fixed number pre-sampling. We ensure sufficient compute time to reach failure and run 50 iterations per sampling method, per number of points sampled and per scenario described below:

1. SE(3) rigid body puzzles. The Special Euclidean Group in 3 Dimensions SE(3) is composed of a 3-D translational component and a 3-D rotational component. The latter is often conveniently sampled in quaternion space, with uniform sampling on the 4D unit sphere manifold. In keeping with standard practice, we evaluate samplers' efficient coverage only on the translational 3-D Euclidean space component (discussion on this point provided in Appendix A.4. Two puzzles, Cubicle and Twistycool, are benchmarked from the OMPL. For the latter, we reduce the bounds to allow this harder problem to be solved with the same N values. Both examples can be visualized in Figure 3

2. d-D hypercube corridor. On the d-dimensional hypercube, the valid region is defined such that there exists an index k with the following constraints: for all dimensions $i < k$, the i-th coordinate must be less than or equal to a threshold edge width λ, and for all dimensions $i > k$, it must be greater than or equal to $1 - \lambda$. This results in a valid subspace that resembles narrow passageways along the hypercube edges, leading from one corner of the hypercube to the diagonally opposite corner. For dimension $d \in \{2, 3, 10\}$, the value of the edge width is tuned to ensure a meaningful comparison between samplers (respectively $\lambda \in \{0.1, 0.2, 0.37\}$).

3. 10-*D kinematic chain.* The experiment features a robot with multiple links in a 10-dimensional configuration space. The robot must navigate through an environment with obstacles represented as line segments. Initially, the robot's first link is at zero radians, and subsequent links are arranged with a specific angular offset. The goal is to reach a target configuration where the first link aligns nearly with a desired orientation, while avoiding collisions with the environment and itself.

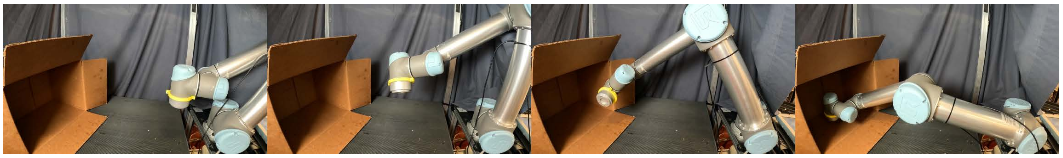

Figure 4: Demonstration of a motion plan on hardware. The goal is to reach into the box without making contact.

UR5 robot arm The UR5 robot arm is a popular collaborative robot with a 6 degree of freedom workspace. In this benchmark, our goal is to reach into a sideways box without contacting the table or the box. We forgo an end effector and thus we do not utilize the last DOF of the robot (which rotates the end effector and in this setting does not effect the workspace of the manipulator). Simulation, sampling, and planning is implemented in the Klamp't software package. We compare our sampling method to sampling from the uniform distribution. For a visualization of the task in simulation see Fig. 3. We also demonstrate our results on real hardware.

5 Results

5.1 2-D maze

The performance of the different sampling techniques for all three levels of the 2-D mazes are presented in Figure 5. On all three levels and for all fixed number of points, MPMC offers the best value per sampled point set size on all but two instances (outperformed by Halton by 2% on level 1 with $N = 32$ and by Sobol by 8% on level 3 with $N = 128$). Indeed, MPMC's efficiency superiority is all the more established on the harder level, where it can reach close to 90% with only 256 points sampled, a feat that takes Halton and Sobol twice the number of points to reach, and a performance level Uniform sampling fails to reach even with 4 times that number.

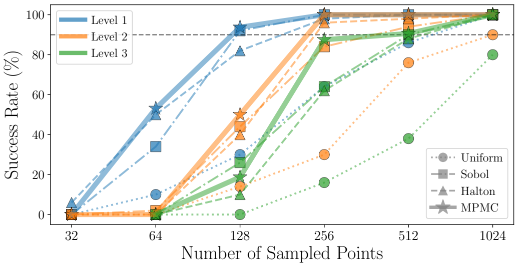

Figure 5: Success rates on the 3 levels of 2-D mazes versus the number of points sampled. Per-level results are grouped by color and the data by sampler is identified by marker and line styles.

In fact, Uniform sampling is substantially less efficient as it requires sampling at least 4 times the number of points of low-discrepancy methods to reach the 90% mark on all levels. Elsewhere, Sobol and Halton sequences seem to offer comparable performance overall, with relative ranking seemingly contingent on the level and number of points sampled. They often fail to meet the performance of MPMC with significant margins (e.g. 34% success for Halton on level 1 with $N = 64$ against MPMC's 54%, under 64% for both Halton and Sobol on level 3 with $N = 256$ to MPMC's 88%).

5.2 OMPL and UR5 benchmarks

The benchmarking success rates are provided in Table 1, along with the average number of valid points sampled by each scheme and its standard deviation.

On the SE(3) Cubicle and Twistycool experiments, MPMC globally achieves the best performance, although with a marginal advantage. This tightness of margins can be linked to the importance of the SO(3) angle quaternion sampling which is done uniformly in all setups. Halton and Uniform schemes appear to be relatively more contingent on the scenario and are harder to clearly rank.

However, MPMC outperforms both Uniform and Halton samplers more consistently and significantly when used on all the samples' components. Indeed, over the 2 and 3 dimensional hypercube

Table 1: Benchmark results on the OMPL suite and UR5 robot arm comparing performance in terms of success rate (SR in %) versus N the number of points sampled $\{128, 256, 512, 1024\}$. Also provided are the mean and standard deviation of the number of valid milestones in each experiment (denoted $|V|$)

Experiment	**Space**	**Sampler**	**128**		**256**		**512**		**1024**	
			SR (%)	**|V|**	**SR (%)**	**|V|**	**SR (%)**	**|V|**	**SR (%)**	**|V|**
Cubicle	SE(3)	Uniform	0	56.4 ± 5.0	0	112.9 ± 7.2	2	221.8 ± 13.6	26	412.6 ± 60.8
		Halton	0	57.4 ± 3.4	0	112.3 ± 4.8	4	221.1 ± 8.9	36	382.4 ± 98.9
		MPMC	0	58.8 ± 3.3	0	112.3 ± 3.6	**10**	221.2 ± 14.5	**46**	367.4 ± 100.5
Twistycool	SE(3)	Uniform	8	89.2 ± 14.7	4	180.4 ± 22.3	26	332.7 ± 71.6	70	485.0 ± 221.7
		Halton	6	89.9 ± 10.8	**16**	171.4 ± 30.2	20	336.0 ± 63.0	56	584.1 ± 193.4
		MPMC	**10**	89.5 ± 11.5	8	175.2 ± 27.6	**30**	327.7 ± 67.1	**72**	499.3 ± 212.7
Hypercube	$\mathbb{R}^2$	Uniform	24	15.8 ± 13.5	24	19.1 ± 24.1	44	56.4 ± 50.0	60	104.1 ± 85.6
		Halton	24	7.9 ± 11.1	36	19.2 ± 23.5	36	47.0 ± 48.9	36	67.5 ± 90.7
		MPMC	**48**	15.4 ± 12.4	**60**	31.6 ± 24.4	**68**	69.3 ± 45.5	**84**	161.0 ± 71.0
Hypercube	$\mathbb{R}^3$	Uniform	24	7.9 ± 7.5	34	17.1 ± 14.9	44	29.5 ± 27.4	54	64.1 ± 51.5
		Halton	20	5.3 ± 6.5	32	15.0 ± 14.2	34	27.6 ± 27.3	28	42.4 ± 52.3
		MPMC	**56**	10.3 ± 6.3	**64**	22.5 ± 12.7	**76**	40.6 ± 22.6	**96**	103.1 ± 20.1
Hypercube	$\mathbb{R}^{10}$	Uniform	**2**	7.0 ± 2.0	12	9.7 ± 2.5	62	17.7 ± 4.3	88	33.9 ± 5.5
		Halton	**2**	6.6 ± 1.8	12	10.4 ± 2.3	56	17.7 ± 2.9	98	32.6 ± 3.4
		MPMC	0	6.3 ± 1.7	**16**	10.7 ± 2.4	**80**	18.9 ± 1.9	**100**	31.1 ± 2.9
Kinematic Chain	$\mathbb{R}^{10}$	Uniform	4	30.2 ± 4.8	12	57.5 ± 9.6	22	103.5 ± 26.3	54	172.5 ± 64.9
		Halton	2	30.0 ± 4.7	14	55.5 ± 9.1	22	103.7 ± 21.4	**62**	156.0 ± 64.5
		MPMC	**6**	31.5 ± 4.3	**24**	51.8 ± 7.5	**26**	102.3 ± 21.5	60	161.7 ± 71.9
UR5	$\mathbb{R}^5$	Uniform	0	47.6 ± 5.8	10	92.2 ± 7.8	6	184.3 ± 11.5	12	368.6 ± 16.1
		Halton	2	49.4 ± 6.9	4	93.1 ± 7.5	14	185.8 ± 10.8	18	369.2 ± 15.6
		MPMC	**6**	49.1 ± 3.0	**31**	97.4 ± 3.5	**75**	181.4 ± 3.8	**81**	360.6 ± 4.3

examples, MPMC sampling solves around twice as many runs as its best competitor with 128 and 256 point regimes and maintains an advantage in the region of 50% with larger point sets.

The previous observation remains, to a solid extent, valid in higher dimensions. Although relative performance gains are in multiple cases well reduced, MPMC sampling maintains top-performing status. Furthermore, there remains instances where it offers large performance gains.

This is evident on the 10-D kinematic chain test, where MPMC reaches 24% success rate with 256 samples, compared to only 12% for Uniform and 14% for Halton. Similarly, a significant gap is established between MPMC (80%) and its counterparts (Halton 56% and Uniform 62%) with 512 points on the 10-D hypercube benchmark. In the realistic scenario of the UR5 planning task (5-D), MPMC consistently surpasses Uniform and Halton sampling (3 to 8-fold higher success rates). We also demonstrate that our approach is readily deployed to real hardware (Figure 4).

6 Discussion

This work introduces the application of MPMC neural network point set generation in motion planning, offering a novel, unbiased approach to sampling. Our proposed method includes a custom training objective specifically designed for high-dimensional configuration spaces, ensuring that the generated points are optimized for complex planning tasks. Moreover, we rigorously show that MPMC point sets offer a tighter upper bound on the distance from the optimal path.

Through extensive experiments on PRM benchmarks, we demonstrate that our method significantly improves planning efficiency over commonly used sampling methods, especially in high-dimensional and challenging environments. Furthermore, we validate the real-world applicability of our technique by successfully implementing it on a UR5 robot arm, highlighting its potential for deployment in practical robotic systems.

Limitations and future work

One limitation of our approach is the need to retrain the MPMC model for each specific number of points N and dimensionality d to ensure optimal performance. This retraining requirement can be computationally intensive, particularly for applications where the configuration space frequently changes. Future work will focus on refining the neural network architecture and training procedure to achieve generalization across both N and d, reducing the need for retraining and making the method more flexible across different planning scenarios. Possible avenues to mitigate this limitation are discussed in Appendix A.2.

Moreover, while we have demonstrated the benefits of MPMC within the PRM framework, future research will explore the potential advantages of integrating this technique into more sophisticated sampling-based planners, potentially extending its impact across a broader class of planning algorithms. Finally, an exciting avenue for future exploration lies in adapting our technique to compute-critical applications, such as acrobatic flight or high-speed driving, where planners require real-time performance and could greatly benefit from efficient, high-quality sampling methods.

Acknowledgments

This work was supported in part by the Postdoc.Mobility grant P500PT-217915 from the Swiss National Science Foundation, the National Science Foundation EFRI program under grant number 1830901, the Gwangju Institute of Science and Technology, and the Department of the Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Department of the Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996. doi:10.1109/70.508439.

[2] S. M. LaValle and J. James J. Kuffner. Randomized kinodynamic planning. *The International Journal of Robotics Research*, 20(5):378–400, 2001. doi:10.1177/02783640122067453. URL `https://doi.org/10.1177/02783640122067453`.

[3] L. Janson, E. Schmerling, A. Clark, and M. Pavone. Fast marching tree: a fast marching sampling-based method for optimal motion planning in many dimensions, 2015. URL `https://arxiv.org/abs/1306.3532`.

[4] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *CoRR*, abs/1105.1186, 2011. URL `http://arxiv.org/abs/1105.1186`.

[5] A. Orthey and M. Toussaint. Sparse multilevel roadmaps for high-dimensional robotic motion planning. In *2021 IEEE International Conference on Robotics and Automation (ICRA 2021)*, pages 7851–7857, Piscataway, NJ, 2021. IEEE. doi:10.1109/ICRA48506.2021.9562053.

[6] A. Orthey and M. Toussaint. Rapidly-exploring quotient-space trees: Motion planning using sequential simplifications, 2019. URL `https://arxiv.org/abs/1906.01350`.

[7] S. Karaman and E. Frazzoli. Sampling-based optimal motion planning for non-holonomic dynamical systems. *2013 IEEE International Conference on Robotics and Automation*, pages 5041–5047, 2013. URL `https://api.semanticscholar.org/CorpusID:9370488`.

[8] A. Yershova and S. M. LaValle. Motion planning for highly constrained spaces. In K. R. Kozłowski, editor, *Robot Motion and Control 2009*, pages 297–306, London, 2009. Springer London. ISBN 978-1-84882-985-5.

[9] M. Phillips and M. Likhachev. Sipp: Safe interval path planning for dynamic environments. *2011 IEEE International Conference on Robotics and Automation*, pages 5628–5635, 2011. URL `https://api.semanticscholar.org/CorpusID:16210040`.

[10] M. W. Otte and E. Frazzoli. Rrtx: Asymptotically optimal single-query sampling-based motion planning with quick replanning. *The International Journal of Robotics Research*, 35:797 – 822, 2016. URL `https://api.semanticscholar.org/CorpusID:20802196`.

[11] T. K. Rusch, N. Kirk, M. M. Bronstein, C. Lemieux, and D. Rus. Message-passing monte carlo: Generating low-discrepancy point sets via graph neural networks. *Proceedings of the National Academy of Sciences*, 121(40):e2409913121, 2024.

[12] D. Berenson, P. Abbeel, and K. Goldberg. A robot path planning framework that learns from experience. In *2012 IEEE International Conference on Robotics and Automation*, pages 3671–3678, 2012. doi:10.1109/ICRA.2012.6224742.

[13] D. Coleman, I. A. Sucan, M. Moll, K. Okada, and N. Correll. Experience-based planning with sparse roadmap spanners, 2014. URL `https://arxiv.org/abs/1410.1950`.

[14] J.-M. Lien. *Planning Motion in Environments with Similar Obstacles*, pages 89–96. 07 2010. ISBN 9780262289801. doi:10.7551/mitpress/8727.003.0013.

[15] P. Lehner and A. Albu-Schäffer. The repetition roadmap for repetitive constrained motion planning. *IEEE Robotics and Automation Letters*, 3(4):3884–3891, 2018. doi:10.1109/LRA.2018.2856925.

[16] C. Chamzas, Z. Kingston, C. Quintero-Pena, A. Shrivastava, and L. E. Kavraki. Learning sampling distributions using local 3d workspace decompositions for motion planning in high dimensions. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2021. doi:10.1109/icra48506.2021.9561104. URL `http://dx.doi.org/10.1109/ICRA48506.2021.9561104`.

[17] C. Chamzas, A. Cullen, A. Shrivastava, and L. E. Kavraki. Learning to retrieve relevant experiences for motion planning, 2022. URL `https://arxiv.org/abs/2204.08550`.

[18] J. van der Corput. Verteilungsfunktionen i–ii. *Proc. Akad. Amsterdam*, 38:813–821, 1058–1066, 1935.

[19] J. H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numer. Math.*, 2:84–90, 1960.

[20] H. Faure. Discrepance de suites associées à un système de numération (en dimension s). *Acta Arithmetica*, 41:337–351, 1982.

[21] H. Niederreiter. Point sets and sequences with small discrepancy. *Monatshefte für Mathematik*, 104(4):273–337, 1987.

[22] I. Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112, 1967. ISSN 0041-5553.

[23] N. Korobov. Number-theoretic methods of approximate analysis. *Fitzmatgiz, Moscow*, 1963. In Russian.

[24] S. Haber. Numerical evaluation of multiple integrals. *SIAM Review*, (12):481–526, 1970.

[25] I. H. Sloan. Lattice methods for multiple integration. *Journal of Computational and Applied Mathematics*, 12-13:131–143, 1985. ISSN 0377-0427.

[26] D. Nuyens. *The construction of good lattice rules and polynomial lattice rules*, pages 223–256. De Gruyter, Berlin, Boston, 2014.

[27] J. Dick, P. Kritzer, and F. Pillichshammer. *Constructions of Lattice Rules*, pages 95–139. Springer International Publishing, Cham, 2022.

[28] C. Doerr and F.-M. De Rainville. Constructing low star discrepancy point sets with genetic algorithms. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, page 789–796, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450319638.

[29] F. Clément, C. Doerr, and L. Paquete. Star discrepancy subset selection: problem formulation and efficient approaches for low dimensions. *J. Complexity*, 70:Paper No. 101645, 34, 2022. ISSN 0885-064X,1090-2708.

[30] F. Clément, C. Doerr, and L. Paquete. Heuristic approaches to obtain low-discrepancy point sets via subset selection. *J. Complexity*, 83:Paper No. 101852, 2024. ISSN 0885-064X,1090-2708.

[31] F. Clément, C. Doerr, K. Klamroth, and L. Paquete. Constructing optimal $\mathcal{L}_\infty$ star discrepancy sets. *Preprint*, 2023. https://arxiv.org/abs/2311.17463.

[32] N. Kirk, T. K. Rusch, J. Zech, and D. Rus. Low stein discrepancy via message-passing monte carlo. *arXiv preprint arXiv:2503.21103*, 2025.

[33] T. T. Warnock. Computational investigations of low-discrepancy point sets. In *Applications of number theory to numerical analysis*, pages 319–343. Elsevier, 1972.

[34] F. Hickernell. A generalized discrepancy and quadrature error bound. *Mathematics of computation*, 67(221):299–322, 1998.

[35] S. Joe. *Formulas for the computation of the weighted L2 discrepancy*. Citeseer, 1997.

[36] F. Clément, N. Kirk, A. B. Owen, and T. K. Rusch. On the optimization of discrepancy measures. *arXiv preprint arXiv:2508.04926*, 2025.

[37] H. Niederreiter. *Random number generation and quasi-Monte Carlo methods*. SIAM, 1992.

[38] L. Janson, B. Ichter, and M. Pavone. Deterministic sampling-based motion planning: Optimality, complexity, and performance. *The International Journal of Robotics Research*, 37(1): 46–61, 2018.

[39] J. H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2:84–90, 1960. URL `https://api.semanticscholar.org/CorpusID:122173470`.

[40] I. Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112, 1967. ISSN 0041-5553. doi:https://doi.org/10.1016/0041-5553(67)90144-9. URL `https://www.sciencedirect.com/science/article/pii/0041555367901449`.

[41] P. Hart, N. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. doi:10.1109/tssc.1968.300136. URL `https://doi.org/10.1109/tssc.1968.300136`.

[42] M. Moll, I. A. Şucan, and L. E. Kavraki. Benchmarking motion planning algorithms: An extensible infrastructure for analysis and visualization. *IEEE Robotics & Automation Magazine*, 22(3):96–102, September 2015. doi:10.1109/MRA.2015.2448276.

A Appendix

A.1 2D point set visualizations

Figure 6 depicts point sets generated by the various sampling schemes used in the 2D-maze experiments namely, Uniform, MPMC, Halton, and Sobol for different numbers of points. Uniformly sampled points exhibit large gap regions for all numbers of points. The sequential constructions of Halton and Sobol are more evenly spread but are surpassed in discrepancy terms by the learned MPMC points.

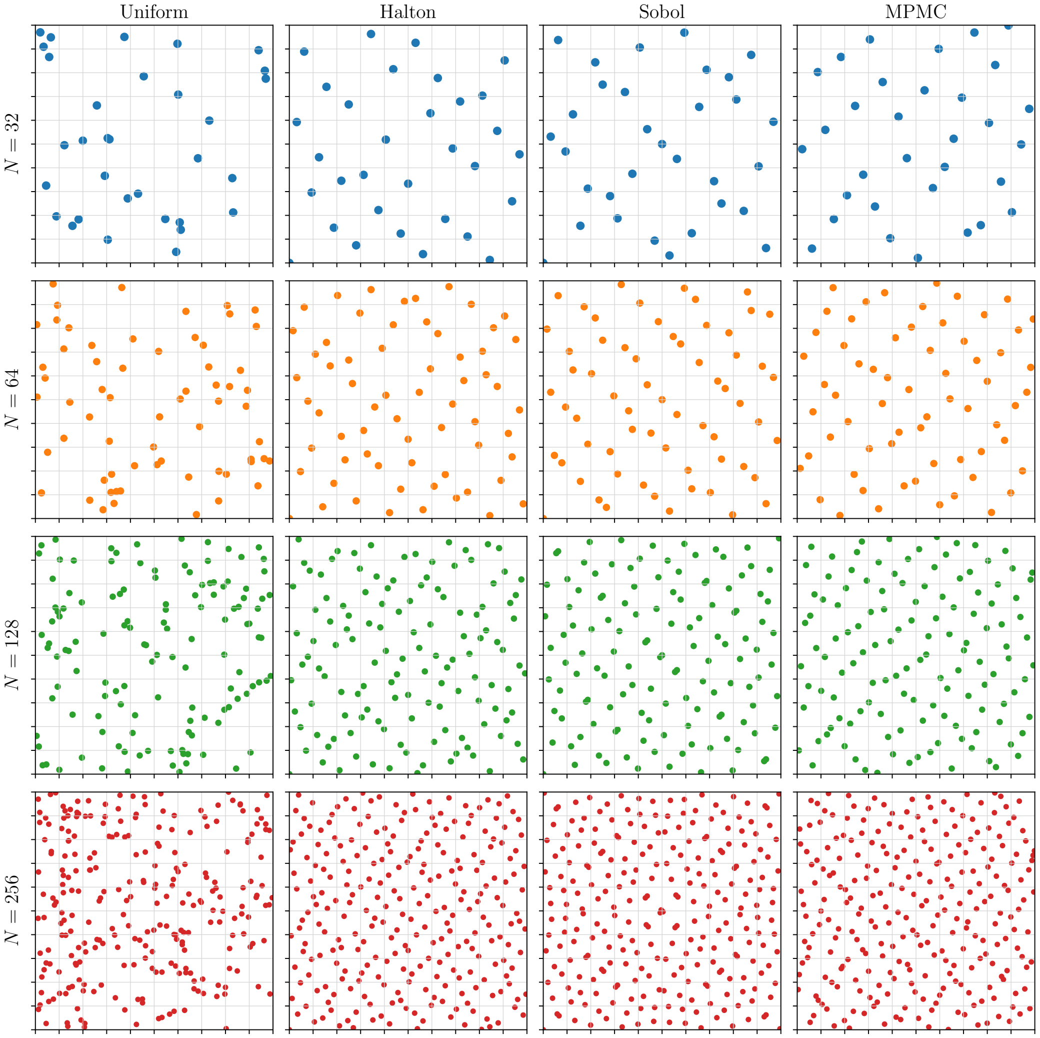

Figure 6: Points sampled on the unit square for $N \in \{32, 64, 128, 256\}$ and sampling distributions among Uniform, Halton, Sobol, and MPMC.

A.2 MPMC point (re)generation

The solution as presented in this work requires retraining a GNN point cloud for each (N, d) pair. Indeed, we focus on optimized point sets (not sequences) to maintain the clearest link between low-discrepancy sampling and improved PRM planning efficiency. There are, however, many practical solutions for mitigating this practicality obstacle and adapting this approach to sequential point generation:

1. For sequences over the number of points N, we propose training a large MPMC point set that can be transformed into a low-discrepancy sequence using a greedy approach, i.e., successively adding points that minimize the discrepancy over all choices in the point set. This further obviates the need for pruning.
2. Inductive graph learning is another solution to N generalization, where a GNN can be trained to perform across a varying range of number of nodes (i.e., points). This approach has proven its efficacy in many other contexts.
3. To deal with varying d, a high dimensional point cloud can be used for lower dimensional problems via canonical projection of the MPMC points. Indeed, it has been shown in [11] that MPMC points yield low-discrepancy in particular for lower dimensional projections of the point sets.

A.3 Dispersion and grids

Sukharev grids are structured sampling patterns that achieve uniform coverage by placing one point at the center of each cell in an evenly spaced Cartesian grid. In d dimensions, the domain $[0, 1]^d$ is divided into k equal intervals along each axis, producing k^d hypercubes, and one sample is placed at the center of each hypercube. Sukharev grids are $\mathcal{D}_\infty$-dispersion optimal when $N = k^d$, with k an integer. For arbitrary N, however, they can be wasteful: for example, with $N = 100$ and $d = 3$, the best the grid can do is $4 \times 4 \times 4 = 64$ points on the lattice, leaving 36 points that do not contribute to an improvement in dispersion. This inefficiency becomes increasingly severe in higher dimensions.

A.4 The case of non-Euclidean manifolds

MPMC can be extended to general manifolds by developing appropriate objective functions. As an example, MPMC can be extended to S^n by minimizing the total euclidean distance between all pairs of points. However, a general framework of MPMC on manifolds will require a separate investigation to measure the best approach and performance gains in such cases.