

SUPPLEMENTARY MATERIAL FOR LLM-GUIDED EVOLUTIONARY PROGRAM SYNTHESIS FOR QUASI-MONTE CARLO DESIGN

Anonymous authors

Paper under double-blind review

A IMPLEMENTATION AND COMPUTATION COMPLEXITY

A.1 INITIAL PROGRAMS

The initial parent program in 2D implemented a simple shifted Fibonacci lattice (Listing 1) and in 3D implemented a scrambled Sobol’ sequence (Listing 2).

Listing 1 Initial Program for 2D Point Set Search

```
1 def construct_star():
2     A=np.zeros((N,2))
3     phi=(math.sqrt(5)-1)/2
4     for i in range(N):
5         A[i,0]=(i+0.5)/N
6         A[i,1]=((i*phi)%1+(0.5/N))%1
7     return A
```

Listing 2 Initial Program for 3D Point Set Search

```
1 def construct_star():
2     A=Sobol(d=3, scramble=True, seed=42).random(n=N)
3     return A
```

A.2 LLM PROMPTS

This appendix provides the full instructional text of the prompts provided to the Large Language Model (LLM) within the OpenEvolve framework. These LLM prompts were used to generate programs to construct point sets directly (Fig. 1), iteratively optimize point sets (Fig. 2), and directly construct Sobol’ direction numbers (Fig. 3).

It is important to note that the text shown in the figures below constitutes the instructional component of a larger prompt. This text is dynamically combined with the source code of a “parent” program selected for mutation and often includes code from other high-performing “inspiration” programs to encourage crossover of ideas.

A.3 IMPLEMENTATION DETAILS

Our evolutionary algorithm treats each candidate solution as a full program (Python module) that implements either a point-set generator (for the 2D star-discrepancy experiments) or a Sobol’ direction-number constructor (for the QMC experiments). Each program exposes a fixed entry point with a prescribed signature (e.g., `construct_points(N)` or `construct_sobol_sequence()`), and is treated as a black box: we execute it in an isolated subprocess, validate its output, and then score it using the appropriate numerical objective.

Population structure and archive. We maintain a population of $P = 60$ candidate programs partitioned into four “islands” arranged in a ring topology. Each island thus holds 15 programs. The

LLM Prompt for Direct Construction

You are an expert mathematician specializing in the construction of QMC sampling points in a square. Your task is to improve a constructor function that directly outputs the position of 16 points on a unit square $[0, 1] \times [0, 1]$ in a way that minimizes the star discrepancy.

The star discrepancy is a measure of how uniformly distributed the points are in the square. It is defined as the supremum of the absolute value of the difference between the fraction of points and the area.

Focus on designing an explicit constructor that specifies the position of each point (x, y) in the unit square, rather than an iterative search algorithm.

It should output the position of each point (x, y) in the square $[0, 1] \times [0, 1]$. 0.0 and 1.0 are included in the square.

Figure 1: The full prompt provided to the LLM for generating programs that directly construct point sets with minimum star discrepancy.

LLM Prompt for Iterative Optimization

You are an expert mathematician specializing in the construction of QMC sampling points in a 2D square. Your task is to improve a constructor function that finds the position of 16 points on a unit square $[0, 1] \times [0, 1]$ in a way that minimizes the star discrepancy.

The star discrepancy is a measure of how uniformly distributed the points are in the square. It is defined as the supremum of the absolute value of the difference between the fraction of points and the area.

Use scipy optimization routines such as `scipy.optimize.minimize` to fine-tune the construction. The optimization routine and its initialization is critically important.

It should output the position of each point (x, y) in the square $[0, 1] \times [0, 1]$. 0.0 and 1.0 are included in the square.

Figure 2: The full prompt provided to the LLM for generating programs that iteratively optimize point sets to have minimum star discrepancy.

search proceeds in asynchronous iterations: at each iteration we (i) select one parent program, (ii) mutate it via an LLM call to obtain a single child, and (iii) evaluate and insert the child back into the island population. Every 25 iterations, we perform migration: one elite individual from each island is sent clockwise and replaces the worst program at the destination. In addition to the islands, we maintain a global elite archive storing the top 25 programs ever seen, ranked first by fitness and then lexicographically by (i) shorter program length and (ii) earlier discovery time. This archive is used both for parent selection and for constructing the LLM context.

Parent selection and LLM-based mutation. At each iteration, we choose a single parent for mutation. With probability 0.7, the parent is sampled from the elite archive; with probability 0.3, it is sampled from the union of all island populations. This biases search toward exploitation of high-performing programs while retaining exploration.

LLM Prompt for Sobol' Direction Numbers Search

You are an expert mathematician specializing in the construction of QMC sampling points in a square. Your task is to improve a constructor function that directly outputs the direction numbers for dimensions 2 to 32 of a Sobol Sequence. Your goal is to minimize the approximation error of a 32 dimensional Asian option price. The dimensions 1, 2, and 3 explain roughly 97% of the variance of the price. The Sobol sequence is defined by a polynomial of degree s , with coefficients represented as an integer a , and direction numbers m_i for each dimension i . The direction numbers must be odd integers and within the specified range. You must return a list of 31 dictionaries for directions 2 to 32, each containing the following keys:

- "s" (int): The degree of the polynomial. $1 \leq s \leq 30$
- "a" (int): The coefficients of the polynomial, represented as an integer. $0 \leq a < 2^{s-1}$
- "m_i" (list[int]): The direction numbers for the Sobol sequence, represented as a list of integers of length s . Each integer should be in the range $[0, 2^{i+1}]$ and has to be odd.

Focus on designing an explicit constructor that specifies these parameters, rather than an iterative search algorithm.

Figure 3: The full prompt provided to the LLM for generating a program that directly specifies Sobol' direction numbers to have lower integration error for an Asian option.

The mutation operator is a call to a Large Language Model (Google Gemini 2.0 Flash) configured as a program rewriter. Rather than applying small, local edits, we ask the LLM to produce a *complete rewrite* of the core implementation functions, conditioned on several pieces of context:

- the full source code of the parent program;
- the source code of the three best-performing programs available ("inspiration programs"), which may come from any island or from the archive;
- a system prompt specifying the required interfaces, validity constraints (e.g., dimensions, box $[0, 1]^d$, odd direction numbers), and the target objective (minimizing 2D star discrepancy or randomized QMC MSE for the Asian option).

This setup acts as a soft form of multi-parent crossover: the LLM is free to borrow subroutines and structural ideas from multiple high-quality programs but must output a single coherent child program. We use Gemini 2.0 Flash with temperature 0.7, top- $p = 0.95$, and a maximum context length of 8192 tokens. In practice, most calls use around 4000 tokens (prompt plus completion). We do not tune these LLM hyperparameters per task or perform any RL based finetuning or training on the LLM; all experiments share the same configuration.

Program execution and validity checks. Every newly generated program is executed in a dedicated subprocess via a small wrapper script. The wrapper:

1. imports the program as a module;
2. calls the required entry function (e.g., `construct_sobol_sequence()`);
3. serializes the returned object to a temporary file.

The main process then deserializes this object and performs a strict validity check. For Sobol' direction-number search, we require: 32 total dimensions; each dimension entry to contain fields s , a , and m ; $1 \leq s \leq 30$; $0 \leq a < 2^{s-1}$; $|m| = s$; each direction number satisfies $1 \leq m_i < 2^s$ and is odd.

If any of these checks fails, if the program crashes, or if execution exceeds a 600 second wall-clock timeout, the candidate is marked invalid and assigned zero fitness. For the 2D star discrepancy experiments, we analogously check that the program returns an N -point set in $[0, 1]^2$ with the correct shape and bounds; a wall-clock timeout of 600 seconds is used.

Fitness evaluation. For valid programs, we compute fitness using the same numerical objectives as in the main experiments:

- **2D star discrepancy.** Given an N -point set $P \subset [0, 1]^2$ returned by the program, we compute the star discrepancy $D^*(P)$ and define fitness as a monotone decreasing function of $D^*(P)$: $f = \frac{1}{1+D^*(P)}$. This preserves the ranking of candidate programs while providing a scalar objective for evolution.
- **Sobol’ direction numbers (randomized QMC MSE).** For Sobol’, the program outputs a list of 31 parameter triples $(s, a, \{m_i\})$ defining dimensions $2, \dots, 32$. To score a candidate A , we estimate the mean squared error (MSE) of a randomized QMC estimator for the 32-dimensional Asian call option used throughout our QMC experiments. Each evaluation draws $R = 1000$ independent randomizations (digital shifts and lower-triangular scrambling matrices) and, for each randomization, generates $N = 8192$ Sobol’ points in $d = 32$ via a compiled C++ generator. For each point set we evaluate the Asian payoff and price estimator $\hat{C}$, accumulate the squared errors $(\hat{C} - C_0)^2$ relative to the known ground-truth price C_0 , and define

$$f = \frac{1}{1 + \sum_{r=1}^R (\hat{C}^{(r)} - C_0)^2}$$

Search budget and hardware. For each task (2D star discrepancy and Sobol’ direction numbers), we run at most 2000 evolutionary iterations, each evaluating exactly one new candidate program. Including the initial population, this yields at most a few thousand distinct programs per run. In practice, a nontrivial fraction of LLM-generated programs fail validation (e.g., due to syntax errors or invalid solutions) and are quickly discarded with zero fitness; the bulk of computation is spent on valid candidates that survive to be evaluated. All numerical evaluations (star discrepancy, Sobol’ generation, QMC payoffs) are executed on CPU. Experiments were carried out on a workstation with an AMD EPYC 7763 CPU and 64 GB of RAM. A full Sobol’ direction-number run at this budget took approximately 96 hours of wall-clock time. We did not perform any hyperparameter tuning of the evolutionary algorithm; population size, archive size, migration schedule, parent selection probabilities, and LLM parameters are fixed across all experiments. Code and all generated point sets and Sobol’ parameters will be released at `anonymous`.

A.4 COMPUTATIONAL COMPLEXITY

The total computational cost of our method has three main components: (i) LLM-based mutation; (ii) program execution and validation; and (iii) numerical fitness evaluation. We analyze each in turn.

LLM mutation. Let T_{prompt} and T_{output} denote the number of tokens in the input and output of one LLM call. Each mutation prompt consists of: a fixed system instruction block, the source code of the parent program, and the source of three top “inspiration” programs. If the typical program size is L lines of code, then both the prompt and the completion are $\mathcal{O}(L)$ in length, so that one mutation costs $\mathcal{O}(T_{\text{prompt}} + T_{\text{output}}) = \mathcal{O}(L)$ tokens. With at most $C_{\text{LLM}} = 2000$ LLM calls per run, the total LLM complexity per run is

$$\mathcal{O}(C_{\text{LLM}} \cdot L).$$

In practice, we cap the context length at 8192 tokens, while observed calls average around 4000 tokens (prompt plus completion), so the token budget scales linearly with the number of iterations.

Program execution and validation. Program execution cost per candidate scales with program length and the cost of constructing its output. Under our design, programs generate their outputs symbolically (using explicit formulas and loops) rather than by performing expensive internal sampling. For the tasks considered:

- in 2D star-discrepancy experiments, point generation costs $\mathcal{O}(N)$ per program;
- in Sobol’ direction-number experiments, the program simply returns a list of 31 parameter triples, which is $\mathcal{O}(1)$.

Validity checks for Sobol’ parameters are linear in the size of this list (constant with respect to N), and validity checks for 2D point sets are linear in the number of points N . Compared to fitness evaluation, these costs are negligible.

Numerical fitness evaluation. The dominant cost lies in computing the numerical objective for each valid program.

- **2D star discrepancy.** For a point set P with N points in $d = 2, 3$, we use a star discrepancy routine that scales exponentially with dimension $\mathcal{O}(N^{1+d/2})$. Let $C_{\text{disc}}(N)$ denote this cost. If at most K distinct programs are evaluated in one run, the total cost for star discrepancy is

$$\mathcal{O}(K \cdot C_{\text{disc}}(N)).$$

For our ranges of N and d , this term is modest relative to the Sobol’ QMC experiments.

- **Sobol’ direction numbers (QMC MSE).** For Sobol’, a single fitness evaluation of a candidate requires $R = 1000$ randomized QMC estimators, each with $N = 8192$ points and dimension $d = 32$. The compiled C++ Sobol’ generator produces N points in $\mathcal{O}(Nd)$ time per randomization, and the Asian payoff evaluation likewise costs $\mathcal{O}(Nd)$: each path requires a cumulative sum over d steps, exponentials, and a payoff computation. Thus one fitness evaluation costs

$$\mathcal{O}(R \cdot N \cdot d).$$

With $R = 1000$, $N = 8192$, and $d = 32$, each evaluation processes $R \times N \approx 8.2 \times 10^6$ 32-dimensional points, corresponding to on the order of 2.6×10^8 scalar operations. If at most $K \leq 2000$ distinct Sobol’ programs are evaluated, the total numerical cost per run is

$$\mathcal{O}(K \cdot R \cdot N \cdot d),$$

which in our setting corresponds to roughly $K \times R \times N \approx 1.6 \times 10^{10}$ 32-dimensional QMC samples and Asian payoffs per full evolutionary run.

Overall complexity. Combining the above, the total computational complexity per run can be summarized as

$$\underbrace{\mathcal{O}(C_{\text{LLM}} \cdot L)}_{\text{LLM mutations}} + \underbrace{\mathcal{O}(K)}_{\text{program execution / validation}} + \underbrace{\mathcal{O}(K \cdot C_{\text{fit}})}_{\text{fitness evaluations}},$$

where $C_{\text{fit}} = C_{\text{disc}}(N)$ for 2D discrepancy and $C_{\text{fit}} = RNd$ for Sobol’. In all our Sobol’ experiments the $KRNd$ term dominates, while for 2D discrepancy the LLM term and the discrepancy term are of comparable size. Crucially, this entire cost is incurred once, *offline*. Using the resulting point sets or Sobol’ parameters in downstream applications has the same asymptotic cost as using a standard hand-designed formula: generating N Sobol’ points is still $\mathcal{O}(Nd)$, and evaluating QMC estimators with them is unchanged.

B GENERATED POINT SETS

This appendix provides the full numerical results for the star discrepancy (D_N^*) values of the point sets discovered by LLM evolutionary search. These tables serve as a comprehensive record of the performance of our method, establishing new state-of-the-art benchmarks and offering a detailed comparison against established low-discrepancy construction methods.

B.1 STAR DISCREPANCY EVALUATION

We evaluate D_N^* by enumerating all anchored boxes whose upper corners lie on the Cartesian product of per-dimension grids $G_j = \{\text{unique } x_{i,j}\} \cup \{1\}$, with the lower corner fixed at the origin. For each candidate corner $\mathbf{y} \in G_1 \times \dots \times G_d$, we compute its volume $\prod_j y_j$ and two point counts:

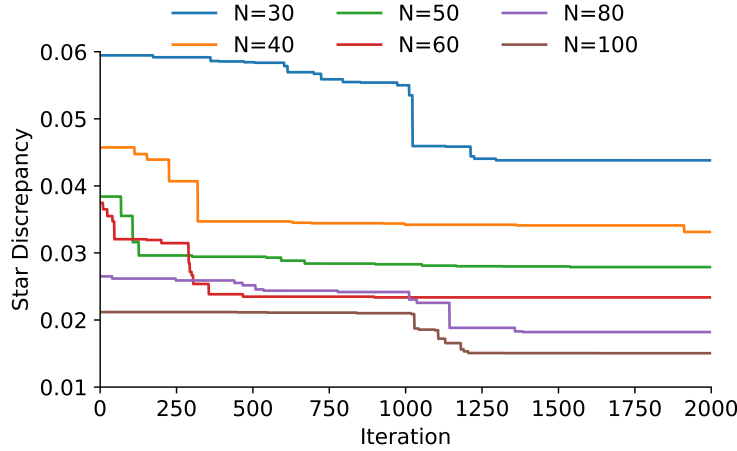


Figure 4: Best-so-far star discrepancy in 2D as a function of iteration for $N \in \{30, 40, 50, 60, 80, 100\}$. Each curve corresponds to a single evolutionary run; the value at iteration t is the minimum discrepancy observed up to that point.

the number of samples dominated by $\mathbf{y}$ (component-wise $\leq$) and the same count with points on any upper face removed. The local discrepancy is the maximum of the two absolute differences from the volume, capturing the half-open convention on anchored boxes. Inputs are clipped to $[0, 1]^d$, grids are built from unique coordinates (plus the terminal 1.0).

B.2 NEW BENCHMARKS IN THREE DIMENSIONS

N	LLM Discrepancy (D_N^*)
9	0.1758
10	0.1652
11	0.1551
12	0.1483
13	0.1402
14	0.1337
15	0.1275
16	0.1207

Table 1: The Star Discrepancy Values for $N > 8$ 3D point sets found via LLM evolutionary search.

We report high-quality constructions for N from 9 to 16 in three dimensions (Table 1).

B.3 OPTIMIZATION TRAJECTORIES IN 2D

To better understand how the evolutionary loop improves the point sets over time, we record the best-so-far star discrepancy after each iteration for several values of N in 2D. Figure 4 shows the resulting trajectories for $N \in \{30, 40, 50, 60, 80, 100\}$ over the full 2000-iteration budget.

Across all N , we observe characteristic two-phase behavior. First, there is a rapid improvement during the early iterations (roughly the first 200–400 steps), where the star discrepancy drops sharply from the initial constructions. After this initial phase, all curves exhibit a much slower regime of incremental refinement: the trajectories continue to decrease but with smaller step sizes and less frequent improvements. Occasional late “jumps” (e.g., $N = 80$ around iteration ~ 1100 or $N = 30$ around iteration ~ 1200) show that useful innovations can still be discovered well into the run, but the marginal gains are modest compared to those in the early iterations.

N	Halton	Sobol'	Hammersley	Fibonacci	MPMC	LLM
140	0.03686	0.02794	0.02701	0.01870	0.01373	0.01151
180	0.03200	0.02300	0.02283	0.01454	0.01147	0.01058
220	0.02323	0.01924	0.01868	0.01190	0.00975	0.00891
260	0.02062	0.01546	0.01581	0.01063	0.00843	0.00831
300	0.01994	0.01341	0.01424	0.00972	0.00752	0.00741
340	0.01950	0.01353	0.01266	0.00858	0.00710	0.00696
380	0.01659	0.01167	0.01170	0.00768	0.00695	0.00638
420	0.01617	0.01194	0.01058	0.00694	0.00584	0.00563
460	0.01279	0.00972	0.00966	0.00634	0.00540	0.00471
500	0.01172	0.00848	0.00889	0.00583	0.00518	0.00509
540	0.01101	0.00967	0.00823	0.00540	0.00488	0.00407
580	0.01261	0.00956	0.00766	0.00503	0.00476	0.00377
620	0.01140	0.00782	0.00744	0.00470	0.00454	0.00386
660	0.01074	0.00956	0.00699	0.00463	0.00437	0.00360
700	0.00933	0.00865	0.00682	0.00437	0.00416	0.00379
740	0.00891	0.00709	0.00646	0.00435	0.00397	0.00346
780	0.00870	0.00662	0.00612	0.00412	0.00376	0.00310
820	0.00881	0.00678	0.00583	0.00392	0.00360	0.00294
860	0.00914	0.00604	0.00556	0.00374	0.00356	0.00316
900	0.00751	0.00561	0.00531	0.00357	0.00319	0.00296
940	0.00785	0.00481	0.00508	0.00342	0.00321	0.00289
980	0.00768	0.00558	0.00487	0.00328	0.00308	0.00280
1020	0.00777	0.00395	0.00468	0.00315	0.00303	0.00245

Table 2: 2D Star Discrepancy Comparison for $N \geq 140$. This table compares the performance of LLM evolutionary search against classical low-discrepancy sequences and the state-of-the-art MPMC method for larger point sets where optimal solutions are not known. Lower values indicate better uniformity. The best result in each row is **bolded**.

B.4 EXTENDED PERFORMANCE IN TWO DIMENSIONS

In two dimensions, while the problem is better understood than in 3D, optimal solutions for larger point sets ($N > 21$) are not known. We provide a detailed comparison of the 2D star discrepancy values achieved by LLM evolutionary optimization against several key baselines for N ranging from 140 to 1020 (Table 2).

- **Consistent Outperformance:** Across every single value of N tested, the point sets discovered by LLM evolutionary search achieve a lower star discrepancy than all other methods, including classical sequences (Halton, Sobol', Hammersley, Fibonacci) and the recent state-of-the-art MPMC method.
- **Significant Improvement Margin:** The performance gap is not trivial. For example, at $N = 140$, the LLM discrepancy of 0.01151 is approximately 16% lower than the next-best method (MPMC at 0.01373) and over 57% lower than the widely used Halton sequence. At $N = 1020$, the LLM discrepancy of 0.00245 is nearly 20% better than MPMC's 0.00303.
- **Superior Scaling:** The results demonstrate that the LLM evolutionary search's ability to find superior configurations is not limited to a specific range of N , but holds consistently as the number of points increases. This suggests that the two-phase discovery strategy (direct construction followed by iterative optimization) is effective at navigating the increasingly complex search space associated with larger point sets.

B.5 PHASE-II LOCAL OPTIMIZATION

To determine whether Phase II improvements could be attributable solely to the use of a standard local optimizer, we constructed a suite of baselines in 2D that used purely local optimization, specifically sequential least squares programming (SLSQP) which is the optimizer predominantly used by the LLM constructed programs. For each $N \in \{2, \dots, 20, 30, 40, 50, 60, 100\}$ we consider three initial point sets: the Sobol' sequence, a simple Fibonacci construction, and the best point set returned by the LLM's Phase I constructive search. From each of these initializations, we then run a

N	Sobol	Sobol+SLSQP	Fibonacci	Fibonacci+SLSQP	Phase I	Phase I+SLSQP	LLM
2	0.7500	0.3660	0.5365	0.3660	0.4375	0.3660	0.3660
3	0.6250	0.3333	0.4850	0.3333	0.3375	0.3333	0.2847
4	0.4375	0.2500	0.3637	0.2500	0.2856	0.2500	0.2500
5	0.4375	0.2000	0.2910	0.2000	0.2500	0.2500	0.2000
6	0.2917	0.1833	0.2836	0.1795	0.2078	0.1771	0.1667
7	0.3393	0.1571	0.2431	0.1587	0.1870	0.1518	0.1500
8	0.3125	0.1495	0.2127	0.1389	0.1697	0.1429	0.1328
9	0.3264	0.1318	0.1891	0.1270	0.1551	0.1270	0.1235
10	0.2906	0.1318	0.1702	0.1143	0.1375	0.1172	0.1111
11	0.2088	0.1099	0.1547	0.1074	0.1255	0.1095	0.1039
12	0.2240	0.1303	0.1418	0.0985	0.1258	0.0979	0.0960
13	0.2404	0.1176	0.1309	0.0909	0.1161	0.0918	0.0892
14	0.1970	0.1092	0.1295	0.0850	0.1075	0.0857	0.0844
15	0.2094	0.1020	0.1209	0.0793	0.1012	0.0804	0.0791
16	0.1719	0.0945	0.1264	0.0743	0.0962	0.0757	0.0745
17	0.1976	0.0937	0.1190	0.0732	0.0903	0.0735	0.0712
18	0.1649	0.0980	0.1124	0.0691	0.0885	0.0676	0.0676
19	0.1357	0.0969	0.1065	0.0673	0.0850	0.0665	0.0654
20	0.1313	0.0824	0.1011	0.0642	0.0818	0.0633	0.0611
30	0.0690	0.0613	0.0674	0.0442	0.0595	0.0440	0.0438
40	0.0836	0.0481	0.0579	0.0356	0.0457	0.0365	0.0331
50	0.0745	0.0379	0.0463	0.0281	0.0355	0.0278	0.0278
60	0.0484	0.0349	0.0386	0.0236	0.0320	0.0253	0.0234
100	0.0423	0.0227	0.0245	0.0153	0.0212	0.0161	0.0150

Table 3: 2D star discrepancy comparison for $N \in \{2, \dots, 20, 30, 40, 50, 60, 100\}$ between Sobol, Fibonacci, Phase I (with and without SLSQP refinement), and LLM search. Best values are **bolded**.

single SLSQP local optimization directly on the point coordinates, using the same star-discrepancy objective as in the main experiments and box constraints $[0, 1]^2$ for all points. The resulting sets are denoted *Sobol+SLSQP*, *Fibonacci+SLSQP*, and *Phase I+SLSQP*. Importantly, these baselines use the same objective and a comparable iteration budget to Phase II, but without any LLM-designed program structure (no LLM-chosen restarts, parameterizations, or heuristics). For comparison, we also report the LLM-evolved point sets from Phase II (“LLM”).

Table 3 reports the resulting star discrepancies. As expected, adding SLSQP on top of classical constructions already yields a large improvement over the Sobol, Fibonacci, and Phase I sets, confirming that local optimization is a strong baseline for this problem. However, the LLM-evolved Phase II sets consistently achieve even lower star discrepancies. In 22 out of the 24 values of N in Table 3, the LLM achieved strictly lower star discrepancy than every SLSQP baseline (Sobol+SLSQP, Fibonacci+SLSQP, Phase I+SLSQP). The largest relative improvements occur at small and moderate N (e.g., $0.3333 \rightarrow 0.2847$ at $N = 3$, a reduction of about 15%, and $0.0356 \rightarrow 0.0331$ at $N = 40$, roughly 7%). In the remaining two cases ($N = 16$ and $N = 50$), the best SciPy baseline is better by less than 2×10^{-4} in absolute discrepancy.

These results show that while local optimization is essential for obtaining strong point sets, the LLM-guided Phase II search still provides a measurable advantage over just local optimization starting from strong human-designed seeds.

B.6 PROMPTING VS EVOLUTIONARY OPTIMIZATION AND LLM SIZE

We next isolate the contributions of (i) the evolutionary search loop versus plain iterative prompting, and (ii) the choice of LLM. At $N = 100$ in 2D, we ran 16 independent seeds for four variants:

- **singleturn**: a single one-shot call to the main LLM (no iteration, no evolution);
- **multiturn**: repeated prompting of the LLM for up to 2000 calls, but without evolutionary operators (no population, no selection, no mutation); this is an iterative prompting baseline;

Method	Mean	Std. dev.	Min	Max
singleturn	0.02117	0.00012	0.02090	0.02135
multiturn	0.01852	0.00254	0.01479	0.02098
LLM	0.01541	0.00065	0.01492	0.01756
LLM-lite	0.01540	0.00064	0.01492	0.01690

Table 4: Ablation at $N = 100$ (2D): star discrepancy over 16 seeds for single-shot prompting (singleturn), multi-turn prompting without evolution (multiturn), and evolutionary search with the Gemini Flash model (LLM) or the smaller Flash-Lite model (LLM-lite).

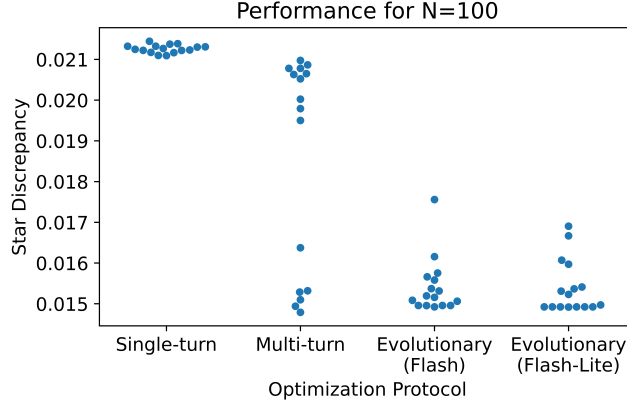


Figure 5: Per-seed star discrepancy at $N = 100$ in 2D for single-turn and multi-turn prompting, and evolutionary search with Gemini Flash and Flash-Lite. Each point is one run with a different seed.

- **LLM**: the full evolutionary pipeline with the main Gemini Flash model, including population-based evolutionary search (selection, mutation, and archiving);
- **LLM-lite**: the same evolutionary pipeline as LLM, but using a smaller Flash-Lite model as the LLM.

All variants optimize the same 2D star-discrepancy objective as in the main experiments. Table 4 summarizes the distribution of the best star discrepancy for $N = 100$ points over the 16 seeds for each method, and Figure 5 visualizes the per-seed outcomes.

First, the single-shot baseline is insufficient to reach the low-discrepancy regime we target. Second, simply calling the LLM multiple times without an evolutionary mechanism (multiturn) substantially improves over single-turn prompting (mean 0.0185 vs. 0.0212) and occasionally discovers very strong solutions (best seed 0.01479). However, this baseline exhibits high variability across seeds (standard deviation $\approx 2.5 \times 10^{-3}$), with some runs remaining close to the single-turn performance. This variability is clearly visible in Figure 5.

Third, the full evolutionary search variants (LLM and LLM-lite) achieve consistently lower discrepancies on average (≈ 0.0154) with much smaller variance ($\approx 6.5 \times 10^{-4}$). Their best runs (0.01492) are within 1.3×10^{-4} of the best multiturn run, but the evolutionary runs are far more robust: almost all seeds converge to strong solutions, as seen in the tight clusters on the right of Figure 5. The comparison between LLM and LLM-lite further indicates that the overall evolutionary search pipeline remains effective even when replacing the main LLM with a smaller Flash-Lite model; their means and variances are nearly identical.

C GENERATED SOBOLOV DIRECTION NUMBERS

We provide the direction numbers discovered by the LLM evolutionary search routine. Only the parameters for dimensions 4, 5, and 6 were updated (Table 5).

D	S	A	M_i
4	3	1	1 3 5
5	3	2	1 3 7
6	4	1	1 1 3 7

Table 5: Sobol’ direction number parameters updated by LLM evolutionary search. D is the dimension, S is the polynomial degree, A is the polynomial’s coefficients, and M_i are the initial direction numbers. All other dimensions remained unchanged.

C.1 COMPARISON WITH LATNETBUILDER DIGITAL NETS

We compare Sobol’ direction numbers found via evolutionary LM search against digital nets constructed with LATNETBUILDER, a state-of-the-art QMC design tool. We use the recommended random-CBC search (2000 iterations) in LatNetBuilder with a projection-dependent t -value based criterion in $d = 32$ dimensions for 2^N points:

```
latnetbuilder -t net -c sobol -s 2^N -d 32 \
  -e random-CBC:2000 -f projdep:t-value -q inf \
  -w order-dependent:0:0,1,1
```

For each resulting Sobol digital net and each number of points $N \in \{32, 64, \dots, 8192\}$, we construct randomized QMC estimators for the six 32-dimensional Asian-style option-pricing payoffs considered in the training example, out-of-the-money, at-the-money, in-the-money, high volatility, and low volatility. Table 6 reports, for each payoff and N , the empirical MSE for LatNetBuilder vs. our implementation (LLM), together with a p -value from a one-sided Wilcoxon signed-rank test with false discovery rate correction across payoffs for each N ; significant p -values ($p < 0.05$) are bolded.

For small to moderate sample sizes ($N \leq 1024$), the LLM discovered Sobol’s sequence consistently and substantially outperforms the LatNetBuilder Sobol’ digital nets, often reducing MSE by factors between $2\times$ and $10\times$, with extremely small p -values across all scenarios. As N increases further, the MSEs of the two constructions converge. In this regime differences are small in absolute terms and typically not statistically significant. Overall, the Sobol’ direction numbers found by our implementation is competitive with, and usually improves upon, Sobol’ direction numbers optimized by a dedicated QMC design tool over a wide range of sample sizes.

C.2 ROLE OF LLM SIZE AND INITIALIZATION FOR SOBOL’ DIRECTION NUMBERS

To better understand the role of LLM size and the importance of initialization in our direction-number search, we compare three LLM variants against the standard Joe–Kuo Sobol’ sequence on the six 32D Asian-style options used in our main randomized QMC experiments. For each variant, we run 16 independent evolutionary searches with different random seeds and report the mean $\pm$ standard deviation of the resulting randomized QMC MSE. The direction numbers used for evaluation are exactly those optimized on the training example; they are not retuned for the other payoffs. The results are summarized in Table 7.

The first two variants differ only in the underlying LLM: LLM uses Gemini Flash, while LLM-lite replaces it with Gemini’s smaller Flash-Lite model. Across all scenarios and sample sizes N , their MSEs are very similar, with differences typically well within one standard deviation. On the training example, for instance, both variants have slightly worse MSE than Joe–Kuo for small N but achieve lower MSE for larger $N \geq 2048$. A similar pattern holds for the out-of-the-money, at-the-money, and volatility-shifted payoffs: for $N \gtrsim 2,000$, both LLM-based variants match or modestly improve on the Joe–Kuo baseline across almost all scenarios, despite being optimized only on the training payoff. This suggests that the overall evolutionary pipeline is not overly sensitive to LLM size; a smaller model can still discover high-quality direction numbers. Furthermore, both LLM and LLM-lite have relatively low standard deviations (typically less than 25% of the mean for large N), reflecting their stability across evolutionary runs.

The third variant, LLM (random init), replaces the Joe-Kuo direction numbers with a random initialization in the same parameter space, keeping the rest of the OpenEvolve pipeline unchanged. This variant is consistently worse than standard Sobol' at small and moderate N (often by factors of $2\times-5\times$ in MSE), but approaches baseline performance at the largest N (e.g., $N = 8192$). In several scenarios, such as at-the-money and high volatility, it remains worse than Joe-Kuo. These results indicate that, under a realistic compute budget, evolutionary search alone is not sufficient to recover high-quality Sobol' parameters from scratch; a strong initialization such as Joe-Kuo is a crucial inductive bias.

D OPTION SCENARIOS

To ensure a rigorous evaluation of the Sobol' direction numbers discovered, we designed a comprehensive suite of benchmark scenarios. This suite includes the primary optimization target (the Asian option, see Table 8) as well as a diverse set of exotic options known to be challenging for Quasi-Monte Carlo integration. The purpose of this suite is twofold: first, to confirm superior performance on the target problem class, and second, to test for generalizability and ensure that the evolved parameters were not merely overfitted to the specific payoff structure of the Asian option.

The configuration parameters for all tested scenarios are consolidated in (Table 9). For all options, the time to expiration (T) was set to 1.0 year and the risk-free interest rate (r) was 0.05. The underlying asset prices are assumed to follow a geometric Brownian motion.

D.1 ASIAN OPTIONS

An Asian option is a path-dependent exotic option whose payoff is determined by the average price of the underlying asset over a pre-set period of time. This is in contrast to a standard European option, which only depends on the asset price at expiration. The averaging feature reduces volatility and makes the option generally cheaper than its European counterpart. Because its value depends on the entire price path, pricing it requires simulating all 32 time steps, making it an excellent candidate for QMC methods.

D.2 LOOKBACK OPTIONS

A lookback option is another path-dependent option whose payoff is determined by the maximum or minimum price of the underlying asset over the option's life. The scenarios tested here are for a floating strike lookback call option, whose payoff at expiration is the difference between the final asset price and the minimum price achieved ($S_T - S_{min}$).

D.3 BARRIER OPTIONS

A barrier option is a path-dependent option that is either activated ("knocks-in") or extinguished ("knocks-out") if the underlying asset price crosses a predetermined "barrier" level. This feature introduces a significant discontinuity in the payoff function.

D.4 BASKET OPTIONS

A basket option's payoff depends on the value of a portfolio or "basket" of multiple underlying assets. It is an inherently high-dimensional problem where the correlation (ρ) between the assets is a critical parameter. For these tests, we assume a uniform initial price of $S_0 = 100.00$ for all 32 assets in the basket.

D.5 BERMUDAN OPTIONS

A Bermudan option is a hybrid between a European option (exercisable only at expiration) and an American option (exercisable at any time). It can be exercised on a discrete set of pre-specified dates. Pricing a Bermudan option is a highly complex problem that requires solving a dynamic programming problem to determine the optimal exercise strategy.

E RQMC INTEGRATION RESULTS

This appendix provides the detailed results for the primary benchmark (Asian option) and the generalizability tests (exotic options). Each table compares the performance of the standard Sobol' sequence (Joe & Kuo) against the direction numbers discovered by LLM evolutionary search. We report the Mean Squared Error (MSE) and its constituent parts, Squared Bias and Variance, for an increasing number of points (N). We consistently found lower variance and MSE for all option types for all $N \geq 512$ with two exceptions: close barrier option ($N = 2048$) and a high correlation basket option ($N = 2048$).

Scenario	N	LatNet MSE	LLM MSE	p-value
Training Example	32	3.472e-01	2.523e-01	1.97e-44
	64	1.308e-01	6.649e-02	1.55e-177
	128	7.057e-02	1.922e-02	0.00e+00
	256	1.809e-02	6.108e-03	0.00e+00
	512	1.384e-02	1.614e-03	0.00e+00
	1024	9.989e-03	5.244e-04	0.00e+00
	2048	2.753e-04	2.254e-04	4.15e-15
	4096	1.213e-04	9.346e-05	9.37e-27
	8192	4.286e-05	4.104e-05	1.63e-01
Out-of-the-Money	32	2.532e-01	1.979e-01	7.18e-25
	64	1.002e-01	6.309e-02	6.55e-77
	128	5.247e-02	2.313e-02	8.85e-261
	256	1.597e-02	8.893e-03	1.46e-148
	512	1.087e-02	3.311e-03	0.00e+00
	1024	6.971e-03	1.313e-03	0.00e+00
	2048	5.971e-04	5.353e-04	3.29e-07
	4096	2.459e-04	2.432e-04	1.77e-01
	8192	8.641e-05	1.066e-04	1.00e+00
At-the-Money	32	4.503e-01	3.346e-01	1.46e-42
	64	1.691e-01	9.079e-02	1.74e-152
	128	9.224e-02	2.822e-02	0.00e+00
	256	2.413e-02	9.815e-03	0.00e+00
	512	1.794e-02	3.106e-03	0.00e+00
	1024	1.269e-02	1.168e-03	0.00e+00
	2048	5.418e-04	5.212e-04	3.04e-01
	4096	2.355e-04	2.261e-04	1.18e-01
	8192	9.411e-05	9.523e-05	1.00e+00
In-the-Money	32	1.925e-01	1.428e-01	1.95e-37
	64	7.452e-02	3.818e-02	2.27e-168
	128	3.849e-02	1.110e-02	0.00e+00
	256	1.037e-02	3.373e-03	0.00e+00
	512	7.949e-03	8.593e-04	0.00e+00
	1024	5.697e-03	2.601e-04	0.00e+00
	2048	1.315e-04	1.000e-04	2.91e-33
	4096	6.003e-05	3.949e-05	2.16e-72
	8192	1.693e-05	1.683e-05	6.02e-01
High Volatility	32	2.902e+00	2.149e+00	1.25e-40
	64	1.133e+00	5.958e-01	1.74e-152
	128	6.053e-01	1.877e-01	0.00e+00
	256	1.647e-01	6.425e-02	0.00e+00
	512	1.235e-01	1.959e-02	0.00e+00
	1024	8.672e-02	7.060e-03	0.00e+00
	2048	3.225e-03	2.867e-03	1.75e-04
	4096	1.365e-03	1.233e-03	4.04e-05
	8192	4.255e-04	5.321e-04	1.00e+00
Low Volatility	32	3.253e-02	2.455e-02	2.93e-38
	64	1.200e-02	6.833e-03	2.93e-129
	128	6.584e-03	2.205e-03	0.00e+00
	256	1.726e-03	8.026e-04	6.07e-252
	512	1.251e-03	2.747e-04	0.00e+00
	1024	8.675e-04	1.088e-04	0.00e+00
	2048	5.024e-05	4.961e-05	5.67e-01
	4096	2.207e-05	2.196e-05	6.46e-01
	8192	9.317e-06	9.300e-06	1.00e+00

Table 6: Mean-squared error (MSE) of randomized QMC estimators for six 32-dimensional Asian-style option payoffs, comparing digital nets constructed with LATNETBUILDER to those optimized by LLM evolutionary search, for $N \in \{32, 64, \dots, 8192\}$. Lowest MSE in each row is **bolded**. The last column reports FDR-corrected one-sided Wilcoxon p-values; significant differences ($p < 0.05$) are **bolded**.

Scenario	N	Sobol MSE	LLM MSE	LLM-lite MSE	LLM (random init) MSE
Training Example	32	2.48e-01	3.12e-01 $\pm$ 1.48e-01	3.17e-01 $\pm$ 9.02e-02	6.20e-01 $\pm$ 1.81e-01
	64	6.42e-02	1.05e-01 $\pm$ 8.77e-02	1.11e-01 $\pm$ 5.53e-02	2.85e-01 $\pm$ 1.04e-01
	128	1.83e-02	3.39e-02 $\pm$ 3.80e-02	3.75e-02 $\pm$ 2.53e-02	1.34e-01 $\pm$ 5.56e-02
	256	5.55e-03	1.26e-02 $\pm$ 1.85e-02	1.32e-02 $\pm$ 8.09e-03	5.90e-02 $\pm$ 2.24e-02
	512	1.65e-03	4.22e-03 $\pm$ 6.52e-03	2.64e-03 $\pm$ 1.63e-03	1.95e-02 $\pm$ 9.69e-03
	1024	5.42e-04	1.66e-03 $\pm$ 3.39e-03	7.97e-04 $\pm$ 6.39e-04	7.28e-03 $\pm$ 4.31e-03
	2048	2.33e-04	2.26e-04 $\pm$ 3.91e-06	2.43e-04 $\pm$ 1.65e-05	2.52e-03 $\pm$ 2.75e-03
	4096	9.88e-05	9.72e-05 $\pm$ 3.02e-06	1.01e-04 $\pm$ 7.70e-06	3.49e-04 $\pm$ 4.03e-04
Out-of-the-Money	8192	4.52e-05	4.18e-05 $\pm$ 1.14e-06	4.25e-05 $\pm$ 1.65e-06	4.53e-05 $\pm$ 2.57e-06
	32	1.92e-01	2.28e-01 $\pm$ 8.22e-02	2.32e-01 $\pm$ 4.92e-02	3.95e-01 $\pm$ 1.01e-01
	64	6.05e-02	8.39e-02 $\pm$ 4.89e-02	8.83e-02 $\pm$ 3.15e-02	1.83e-01 $\pm$ 5.91e-02
	128	2.20e-02	3.06e-02 $\pm$ 2.07e-02	3.33e-02 $\pm$ 1.42e-02	8.46e-02 $\pm$ 3.04e-02
	256	8.60e-03	1.25e-02 $\pm$ 1.05e-02	1.32e-02 $\pm$ 4.40e-03	3.75e-02 $\pm$ 1.22e-02
	512	3.47e-03	4.93e-03 $\pm$ 3.82e-03	4.07e-03 $\pm$ 8.52e-04	1.27e-02 $\pm$ 4.99e-03
	1024	1.46e-03	2.02e-03 $\pm$ 1.94e-03	1.62e-03 $\pm$ 3.36e-04	4.83e-03 $\pm$ 2.24e-03
	2048	6.19e-04	5.44e-04 $\pm$ 1.42e-05	5.93e-04 $\pm$ 4.38e-05	1.74e-03 $\pm$ 1.43e-03
At-the-Money	4096	2.58e-04	2.48e-04 $\pm$ 6.55e-06	2.57e-04 $\pm$ 2.43e-05	3.59e-04 $\pm$ 1.84e-04
	8192	1.17e-04	1.12e-04 $\pm$ 4.33e-06	1.10e-04 $\pm$ 1.18e-05	1.10e-04 $\pm$ 1.07e-05
	32	3.25e-01	4.04e-01 $\pm$ 1.82e-01	4.12e-01 $\pm$ 1.12e-01	7.76e-01 $\pm$ 2.20e-01
	64	8.73e-02	1.38e-01 $\pm$ 1.08e-01	1.47e-01 $\pm$ 6.94e-02	3.56e-01 $\pm$ 1.27e-01
	128	2.77e-02	4.56e-02 $\pm$ 4.53e-02	5.11e-02 $\pm$ 3.10e-02	1.64e-01 $\pm$ 6.51e-02
	256	9.23e-03	1.77e-02 $\pm$ 2.26e-02	1.87e-02 $\pm$ 9.72e-03	7.20e-02 $\pm$ 2.65e-02
	512	3.25e-03	6.36e-03 $\pm$ 8.09e-03	4.36e-03 $\pm$ 1.79e-03	2.33e-02 $\pm$ 1.10e-02
	1024	1.23e-03	2.54e-03 $\pm$ 4.09e-03	1.51e-03 $\pm$ 6.94e-04	8.69e-03 $\pm$ 4.87e-03
In-the-Money	2048	5.50e-04	5.14e-04 $\pm$ 1.52e-05	5.36e-04 $\pm$ 2.67e-05	3.00e-03 $\pm$ 3.08e-03
	4096	2.40e-04	2.30e-04 $\pm$ 7.48e-06	2.33e-04 $\pm$ 1.03e-05	4.80e-04 $\pm$ 4.07e-04
	8192	1.02e-04	9.95e-05 $\pm$ 3.97e-06	9.50e-05 $\pm$ 6.18e-06	1.03e-04 $\pm$ 7.57e-06
	32	1.40e-01	1.74e-01 $\pm$ 8.01e-02	1.76e-01 $\pm$ 4.77e-02	3.45e-01 $\pm$ 1.02e-01
	64	3.67e-02	5.85e-02 $\pm$ 4.74e-02	6.12e-02 $\pm$ 2.95e-02	1.61e-01 $\pm$ 6.00e-02
	128	1.06e-02	1.93e-02 $\pm$ 2.16e-02	2.07e-02 $\pm$ 1.40e-02	7.87e-02 $\pm$ 3.34e-02
	256	3.08e-03	6.98e-03 $\pm$ 1.03e-02	7.14e-03 $\pm$ 4.70e-03	3.47e-02 $\pm$ 1.33e-02
	512	8.68e-04	2.28e-03 $\pm$ 3.56e-03	1.45e-03 $\pm$ 1.06e-03	1.21e-02 $\pm$ 6.13e-03
High Volatility	1024	2.61e-04	8.97e-04 $\pm$ 1.94e-03	4.25e-04 $\pm$ 4.24e-04	4.61e-03 $\pm$ 2.77e-03
	2048	1.01e-04	1.01e-04 $\pm$ 1.70e-06	1.10e-04 $\pm$ 1.02e-05	1.64e-03 $\pm$ 1.77e-03
	4096	4.04e-05	4.18e-05 $\pm$ 1.35e-06	4.36e-05 $\pm$ 4.54e-06	2.25e-04 $\pm$ 2.93e-04
	8192	1.77e-05	1.71e-05 $\pm$ 4.83e-07	1.72e-05 $\pm$ 8.41e-07	1.84e-05 $\pm$ 1.43e-06
	32	2.08e+00	2.60e+00 $\pm$ 1.19e+00	2.64e+00 $\pm$ 7.19e-01	5.08e+00 $\pm$ 1.50e+00
	64	5.71e-01	9.00e-01 $\pm$ 7.05e-01	9.54e-01 $\pm$ 4.52e-01	2.38e+00 $\pm$ 8.77e-01
	128	1.81e-01	3.05e-01 $\pm$ 3.09e-01	3.36e-01 $\pm$ 2.07e-01	1.14e+00 $\pm$ 4.65e-01
	256	5.98e-02	1.17e-01 $\pm$ 1.51e-01	1.23e-01 $\pm$ 6.66e-02	5.02e-01 $\pm$ 1.86e-01
Low Volatility	512	2.05e-02	4.14e-02 $\pm$ 5.37e-02	2.86e-02 $\pm$ 1.37e-02	1.70e-01 $\pm$ 8.15e-02
	1024	7.56e-03	1.66e-02 $\pm$ 2.81e-02	9.77e-03 $\pm$ 5.41e-03	6.43e-02 $\pm$ 3.65e-02
	2048	3.15e-03	2.87e-03 $\pm$ 7.08e-05	3.14e-03 $\pm$ 2.15e-04	2.27e-02 $\pm$ 2.33e-02
	4096	1.29e-03	1.26e-03 $\pm$ 2.95e-05	1.31e-03 $\pm$ 1.14e-04	3.43e-03 $\pm$ 3.47e-03
	8192	5.66e-04	5.59e-04 $\pm$ 2.86e-05	5.38e-04 $\pm$ 7.12e-05	5.58e-04 $\pm$ 6.19e-05
	32	2.38e-02	2.92e-02 $\pm$ 1.24e-02	2.99e-02 $\pm$ 7.67e-03	5.41e-02 $\pm$ 1.46e-02
	64	6.57e-03	1.00e-02 $\pm$ 7.30e-03	1.07e-02 $\pm$ 4.73e-03	2.44e-02 $\pm$ 8.32e-03
	128	2.17e-03	3.34e-03 $\pm$ 2.98e-03	3.77e-03 $\pm$ 2.08e-03	1.10e-02 $\pm$ 4.18e-03
	256	7.56e-04	1.32e-03 $\pm$ 1.51e-03	1.41e-03 $\pm$ 6.36e-04	4.81e-03 $\pm$ 1.72e-03
	512	2.85e-04	4.95e-04 $\pm$ 5.44e-04	3.59e-04 $\pm$ 1.07e-04	1.51e-03 $\pm$ 6.71e-04
	1024	1.13e-04	1.99e-04 $\pm$ 2.66e-04	1.32e-04 $\pm$ 4.12e-05	5.60e-04 $\pm$ 2.97e-04
	2048	5.21e-05	4.87e-05 $\pm$ 1.38e-06	5.03e-05 $\pm$ 2.17e-06	1.92e-04 $\pm$ 1.85e-04
	4096	2.32e-05	2.22e-05 $\pm$ 7.64e-07	2.23e-05 $\pm$ 8.73e-07	3.53e-05 $\pm$ 2.18e-05
	8192	1.01e-05	9.66e-06 $\pm$ 3.41e-07	9.40e-06 $\pm$ 4.93e-07	1.00e-05 $\pm$ 6.33e-07

Table 7: The table compares the standard Sobol’ sequence (Joe & Kuo) against three variants: our implementation (LLM), our implementation with the flash-lite model (LLM-lite), and our implementation starting from random initialization (LLM random init.) across all size Asian option scenarios. We report the Mean Squared Error (MSE).

Scenario	S_0	K	σ	S_{true}
Training Example	50.00	45.00	0.3	7.06
Out-of-the-Money	50.00	60.00	0.3	1.02
At-the-Money	50.00	52.50	0.3	2.98
In-the-Money	50.00	40.00	0.3	11.02
High Volatility	50.00	52.50	0.6	6.43
Low Volatility	50.00	52.50	0.1	0.69

Table 8: Asian option scenarios used for testing. The Training Example was used in the evaluation routine of the evolutionary search. All options have $T = 1.0$ and $r = 0.05$.

Option Type	Scenario	Initial Price (S_0)	Strike Price (K)	Volatility (σ)	Other Parameters
Asian	Training Example	50.00	45.00	0.3	—
	Out-of-the-Money	50.00	60.00	0.3	—
	At-the-Money	50.00	52.50	0.3	—
	In-the-Money	50.00	40.00	0.3	—
	High Volatility	50.00	52.50	0.6	—
	Low Volatility	50.00	52.50	0.1	—
Lookback	Base	100.00	—	0.2	—
	High Volatility	100.00	—	0.4	—
Barrier	Base	100.00	100.00	0.2	Barrier Level: 85.00
	Close Barrier	100.00	100.00	0.2	Barrier Level: 95.00
Basket (32D)	Low Correlation	100.00	100.00	0.2	ρ : 0.1
	High Correlation	100.00	100.00	0.2	ρ : 0.8
	Mixed Volatility	100.00	100.00	$U(0.15, 0.4)$	ρ : 0.5
	Out-of-the-Money	100.00	110.00	0.2	ρ : 0.1
Bermudan	At-the-Money	100.00	100.00	0.2	Exercise Dates: 4
	In-the-Money	90.00	100.00	0.2	Exercise Dates: 4

Table 9: Configuration Parameters for All Tested Option Scenarios. This table details the parameters for the 32-dimensional options used in the primary benchmark and generalizability tests. For all scenarios, the time to expiration (T) is 1.0 year and the risk-free interest rate (r) is 0.05. An em-dash (—) indicates a parameter is not applicable to that option type.

Scenario	N	Squared Bias		Variance		MSE		p-value
		Sobol	LLM	Sobol	LLM	Sobol	LLM	
Training Example	32	2.09e-05	5.59e-07	0.2484	0.2523	0.2484	0.2523	0.9757
	64	5.37e-06	9.58e-06	0.0642	0.0665	0.0642	0.0665	0.9980
	128	5.10e-06	4.96e-06	0.0183	0.0192	0.0183	0.0192	0.9999
	256	2.87e-07	1.43e-07	0.0056	0.0061	0.0056	0.0061	1.0000
	512	6.09e-08	3.51e-09	0.001646	0.001614	0.001646	0.001614	0.2841
	1024	7.89e-08	1.33e-08	0.000542	0.000524	0.000542	0.000524	0.0548
	2048	2.60e-08	2.30e-12	0.000233	0.000225	0.000233	0.000225	0.0199
	4096	9.11e-10	3.46e-09	9.876e-05	9.346e-05	9.876e-05	9.346e-05	0.0058
	8192	1.55e-09	1.93e-10	4.523e-05	4.104e-05	4.523e-05	4.104e-05	7.03e-06
Out-of-the-Money	32	3.73e-05	6.41e-07	0.1920	0.1979	0.1920	0.1979	0.9757
	64	1.48e-06	9.78e-09	0.0605	0.0631	0.0605	0.0631	0.9980
	128	3.74e-07	1.66e-06	0.0220	0.0231	0.0220	0.0231	0.9999
	256	3.07e-07	2.60e-07	0.0086	0.0089	0.0086	0.0089	1.0000
	512	4.47e-08	4.05e-11	0.003469	0.003311	0.003469	0.003311	0.0208
	1024	1.29e-08	4.45e-08	0.001457	0.001313	0.001457	0.001313	3.45e-08
	2048	2.92e-11	3.36e-08	0.000619	0.000535	0.000619	0.000535	1.45e-14
	4096	5.09e-08	1.17e-09	0.000258	0.000243	0.000258	0.000243	4.51e-04
	8192	3.71e-09	3.24e-09	0.000117	0.000107	0.000117	0.000107	1.06e-06
At-the-Money	32	5.90e-05	1.35e-09	0.3246	0.3346	0.3246	0.3346	0.9757
	64	1.35e-05	1.17e-05	0.0873	0.0908	0.0873	0.0908	0.9980
	128	9.72e-06	1.45e-06	0.0277	0.0282	0.0277	0.0282	0.9999
	256	1.12e-06	2.06e-08	0.0092	0.0098	0.0092	0.0098	1.0000
	512	1.83e-08	5.57e-10	0.003248	0.003106	0.003248	0.003106	0.0208
	1024	3.60e-08	4.58e-09	0.001228	0.001168	0.001228	0.001168	0.0304
	2048	7.85e-11	1.82e-08	0.000550	0.000521	0.000550	0.000521	0.0131
	4096	3.16e-10	8.98e-10	0.000240	0.000226	0.000240	0.000226	0.0019
	8192	7.47e-09	3.31e-09	0.000102	9.522e-05	0.000102	9.523e-05	0.0100
In-the-Money	32	1.93e-05	3.11e-09	0.1397	0.1428	0.1398	0.1428	0.9757
	64	2.30e-06	2.86e-06	0.0367	0.0382	0.0367	0.0382	0.9980
	128	1.67e-06	1.74e-06	0.0106	0.0111	0.0106	0.0111	0.9999
	256	1.08e-09	3.54e-08	0.003083	0.003373	0.003083	0.003373	1.0000
	512	3.07e-10	4.26e-08	0.000868	0.000859	0.000868	0.000859	0.2841
	1024	1.53e-09	6.44e-10	0.000261	0.000260	0.000261	0.000260	0.2581
	2048	3.54e-10	1.28e-08	0.000101	0.000100	0.000101	0.000100	0.3878
	4096	2.15e-11	3.38e-09	4.037e-05	3.949e-05	4.037e-05	3.949e-05	0.2180
	8192	2.77e-10	3.70e-09	1.766e-05	1.682e-05	1.766e-05	1.683e-05	0.0047
High Volatility	32	0.000333	2.83e-07	2.0813	2.1493	2.0816	2.1493	0.9757
	64	5.87e-05	4.32e-05	0.5713	0.5958	0.5714	0.5958	0.9980
	128	4.41e-05	5.25e-06	0.1810	0.1877	0.1811	0.1877	0.9999
	256	1.78e-06	1.90e-06	0.0598	0.0643	0.0598	0.0643	1.0000
	512	2.34e-09	1.49e-08	0.0205	0.0196	0.0205	0.0196	0.0073
	1024	1.21e-07	2.19e-07	0.007565	0.007060	0.007565	0.007060	3.38e-04
	2048	6.34e-08	1.22e-07	0.003145	0.002867	0.003145	0.002867	1.75e-06
	4096	2.93e-10	6.09e-09	0.001293	0.001233	0.001293	0.001233	0.0022
	8192	3.59e-08	4.35e-08	0.000566	0.000532	0.000566	0.000532	0.0156
Low Volatility	32	5.75e-06	1.26e-08	0.0238	0.0246	0.0238	0.0246	0.9757
	64	1.28e-06	6.48e-07	0.0066	0.0068	0.0066	0.0068	0.9980
	128	7.71e-07	2.71e-08	0.002173	0.002205	0.002174	0.002205	0.9999
	256	1.08e-07	1.12e-08	0.000756	0.000803	0.000756	0.000803	1.0000
	512	6.51e-09	1.55e-09	0.000285	0.000275	0.000285	0.000275	0.0208
	1024	5.18e-09	1.20e-09	0.000113	0.000109	0.000113	0.000109	0.0717
	2048	2.66e-10	2.17e-09	5.214e-05	4.961e-05	5.214e-05	4.961e-05	0.0056
	4096	1.26e-10	1.39e-10	2.323e-05	2.196e-05	2.323e-05	2.196e-05	5.68e-04
	8192	3.43e-10	3.09e-10	1.008e-05	9.300e-06	1.008e-05	9.300e-06	1.27e-04

Table 10: Integration results for the Asian option across all six scenarios. The table compares the standard Sobol’ sequence (Joe & Kuo) against the direction numbers discovered by LLM evolutionary search. We report the Mean Squared Error (MSE), its constituent parts (Squared Bias and Variance), and the FDR-corrected p-value from a one-sided Wilcoxon signed-rank test. P-values below 0.05 and the best methods are **bolded**.

Scenario	N	Squared Bias		Variance		MSE		p-value
		Sobol	LLM	Sobol	LLM	Sobol	LLM	
Base	32	3.90e-06	8.56e-05	1.2138	1.2443	1.2138	1.2444	0.9968
	64	4.16e-06	3.01e-05	0.4725	0.4839	0.4725	0.4840	1.0000
	128	3.14e-06	2.35e-07	0.1272	0.1352	0.1272	0.1352	1.0000
	256	1.61e-06	4.90e-06	0.0352	0.0381	0.0352	0.0381	1.0000
	512	8.86e-07	1.10e-06	0.0121	0.0120	0.0121	0.0120	0.4675
	1024	9.64e-07	1.43e-06	0.005021	0.004905	0.005022	0.004907	0.0342
	2048	6.88e-08	1.29e-07	0.001834	0.001744	0.001834	0.001744	4.13e-04
	4096	1.32e-07	5.59e-08	0.000807	0.000768	0.000807	0.000768	2.38e-04
	8192	5.21e-08	3.61e-08	0.000407	0.000377	0.000407	0.000377	9.84e-09
High Volatility	32	2.13e-04	4.44e-04	8.5591	8.7869	8.5593	8.7873	0.9968
	64	6.04e-08	1.16e-04	3.3964	3.5134	3.3964	3.5136	1.0000
	128	2.87e-05	2.57e-06	0.8854	0.9982	0.8854	0.9982	1.0000
	256	3.14e-06	1.46e-05	0.2403	0.2663	0.2403	0.2663	1.0000
	512	7.96e-07	1.11e-07	0.0809	0.0792	0.0809	0.0792	0.3836
	1024	2.57e-06	7.04e-06	0.0325	0.0312	0.0325	0.0312	6.24e-04
	2048	3.37e-07	4.92e-08	0.012003	0.011234	0.012003	0.011234	2.26e-06
	4096	1.47e-07	2.32e-08	0.004862	0.004558	0.004862	0.004558	2.65e-06
	8192	2.12e-10	6.25e-09	0.002335	0.002057	0.002335	0.002057	9.79e-17

Table 11: Integration results for the Lookback Option across two scenarios. We report FDR-corrected P-values from a one-sided Wilcoxon signed-rank test. P-values below 0.05 and the best methods are **bolded**.

Scenario	N	Squared Bias		Variance		MSE		p-value
		Sobol	LLM	Sobol	LLM	Sobol	LLM	
Base	32	1.13e-04	2.87e-05	2.2540	2.3339	2.2541	2.3339	0.9968
	64	2.17e-05	4.56e-06	0.8386	0.8894	0.8386	0.8894	1.0000
	128	2.32e-05	2.37e-05	0.2533	0.3101	0.2534	0.3101	1.0000
	256	1.97e-06	7.40e-06	0.0707	0.0834	0.0707	0.0834	1.0000
	512	5.72e-07	4.84e-06	0.0245	0.0241	0.0245	0.0241	0.4675
	1024	7.63e-07	7.40e-07	0.009948	0.009819	0.009948	0.009819	0.3313
	2048	4.03e-07	1.12e-06	0.004277	0.004230	0.004277	0.004231	0.3025
	4096	8.61e-08	1.00e-07	0.002019	0.001992	0.002019	0.001992	0.3863
	8192	1.82e-08	5.70e-08	0.000985	0.000955	0.000985	0.000955	0.0859
Close Barrier	32	9.55e-04	1.27e-04	3.2150	3.2754	3.2160	3.2755	0.9968
	64	6.55e-05	1.44e-05	1.1229	1.2054	1.1229	1.2054	1.0000
	128	5.52e-05	1.14e-05	0.4541	0.5183	0.4542	0.5183	1.0000
	256	3.62e-05	8.39e-07	0.1733	0.1775	0.1733	0.1775	1.0000
	512	3.76e-05	7.16e-07	0.0723	0.0690	0.0723	0.0690	0.2364
	1024	2.27e-05	2.62e-06	0.0328	0.0326	0.0328	0.0326	0.4459
	2048	1.52e-06	1.13e-09	0.01594	0.01648	0.01594	0.01648	0.9126
	4096	3.83e-07	4.25e-07	0.007027	0.006727	0.007028	0.006728	0.0574
	8192	3.48e-09	3.79e-07	0.003379	0.003266	0.003379	0.003267	0.1613

Table 12: Integration results for the Barrier Option across two scenarios. We report FDR-corrected p-values from a one-sided Wilcoxon signed-rank test. P-values below 0.05 and the best methods are **bolded**.

Scenario	N	Squared Bias		Variance		MSE		p-value
		Sobol	LLM	Sobol	LLM	Sobol	LLM	
Low Correlation	32	3.82e-06	5.13e-07	0.1152	0.1182	0.1152	0.1182	0.9968
	64	6.60e-07	1.99e-06	0.0342	0.0358	0.0342	0.0358	1.0000
	128	1.18e-06	1.38e-06	0.0109	0.0126	0.0109	0.0126	1.0000
	256	5.16e-08	1.38e-08	0.0036	0.0039	0.0036	0.0039	1.0000
	512	3.04e-07	1.15e-08	0.001259	0.001231	0.001259	0.001231	0.2381
	1024	2.00e-07	1.79e-08	0.000504	0.000473	0.000504	0.000473	0.0051
	2048	5.00e-08	6.46e-11	0.000229	0.000214	0.000229	0.000214	3.79e-04
	4096	1.12e-10	1.32e-09	9.886e-05	9.038e-05	9.886e-05	9.038e-05	1.68e-04
High Correlation	8192	1.39e-09	1.72e-10	4.740e-05	4.157e-05	4.740e-05	4.157e-05	1.92e-11
	32	9.26e-05	9.80e-05	0.2483	0.2557	0.2484	0.2558	0.9968
	64	7.77e-06	1.10e-05	0.0602	0.0570	0.0602	0.0571	2.61e-05
	128	6.94e-06	2.65e-06	0.0163	0.0168	0.0163	0.0168	1.0000
	256	7.96e-07	8.54e-07	0.004907	0.004944	0.004908	0.004945	1.0000
	512	6.53e-08	8.95e-08	0.001527	0.001518	0.001527	0.001518	0.4675
	1024	3.80e-10	1.16e-08	0.000463	0.000447	0.000463	0.000447	0.0784
	2048	3.45e-09	1.57e-09	0.0001750	0.0001779	0.0001750	0.0001779	0.7419
Mixed Volatility	4096	8.68e-10	7.39e-09	7.132e-05	6.609e-05	7.132e-05	6.610e-05	8.20e-04
	8192	6.90e-09	1.17e-10	2.598e-05	2.448e-05	2.599e-05	2.448e-05	5.32e-03
	32	2.19e-04	9.65e-05	0.7753	0.7993	0.7755	0.7994	0.9968
	64	9.78e-06	1.28e-05	0.2098	0.2071	0.2099	0.2071	0.5396
	128	1.95e-05	2.19e-06	0.0647	0.0702	0.0647	0.0702	1.0000
	256	4.42e-06	1.52e-06	0.0211	0.0216	0.0211	0.0216	1.0000
	512	6.51e-07	6.07e-07	0.006202	0.006112	0.006202	0.006112	0.6395
	1024	1.17e-08	5.20e-09	0.002089	0.002026	0.002089	0.002026	0.0784
Out-of-the-Money	2048	1.09e-08	4.65e-11	0.000947	0.000932	0.000947	0.000932	0.0672
	4096	3.40e-09	2.79e-08	0.000447	0.000393	0.000447	0.000393	1.45e-10
	8192	1.80e-08	2.19e-12	0.000156	0.000143	0.000156	0.000143	1.48e-05
	32	2.43e-05	2.29e-09	0.1165	0.1205	0.1166	0.1205	0.9968
	64	2.81e-06	4.90e-09	0.0367	0.0387	0.0367	0.0387	1.0000
	128	2.55e-06	2.01e-07	0.0127	0.0143	0.0127	0.0143	1.0000
	256	5.97e-07	4.26e-10	0.004370	0.004627	0.004371	0.004627	1.0000
	512	2.65e-07	2.67e-08	0.001602	0.001553	0.001602	0.001553	0.2364
	1024	3.70e-08	2.12e-09	0.000652	0.000623	0.000652	0.000623	0.0651
	2048	5.68e-09	3.00e-11	0.000292	0.000281	0.000292	0.000281	0.0856
	4096	1.01e-08	5.67e-09	0.000125	0.000120	0.000125	0.000120	0.0027
	8192	1.58e-11	1.66e-09	5.949e-05	5.392e-05	5.949e-05	5.392e-05	2.53e-06

Table 13: Integration results for the 32-dimensional Basket Option across four scenarios. We report FDR-corrected p-values from a one-sided Wilcoxon signed-rank test. P-values below 0.05 and the best methods are **bolded**.

Scenario	N	Squared Bias		Variance		MSE		p-value
		Sobol	LLM	Sobol	LLM	Sobol	LLM	
At-the-Money	32	0.8165	0.8272	0.5744	0.5826	1.3909	1.4098	0.9968
	64	0.2544	0.2627	0.2403	0.2372	0.4948	0.4999	1.0000
	128	0.0698	0.0725	0.0941	0.0951	0.1639	0.1677	1.0000
	256	0.0182	0.0192	0.0441	0.0445	0.0624	0.0637	1.0000
	512	0.0044	0.0041	0.0218	0.0221	0.02621	0.02622	0.4675
	1024	9.87e-04	9.33e-04	0.0118	0.0112	0.0128	0.0122	0.0115
	2048	2.77e-04	2.46e-04	0.005957	0.005713	0.006234	0.005960	0.0018
	4096	5.23e-05	4.25e-05	0.002696	0.002676	0.002749	0.002718	0.3863
	8192	1.10e-05	8.45e-06	0.001324	0.001273	0.001335	0.001281	0.0457
In-the-Money	32	0.8888	0.8979	0.5629	0.5549	1.4518	1.4528	0.9968
	64	0.2226	0.2296	0.2507	0.2509	0.4733	0.4805	1.0000
	128	0.0478	0.0489	0.1183	0.1169	0.1662	0.1658	1.0000
	256	0.0092	0.0093	0.0564	0.0563	0.0656	0.0656	1.0000
	512	0.0020	0.0019	0.0284	0.0278	0.0304	0.0297	0.4675
	1024	3.75e-04	4.09e-04	0.0145	0.0135	0.0149	0.0139	0.0142
	2048	6.75e-05	7.12e-05	0.007354	0.006584	0.007422	0.006655	1.22e-05
	4096	1.17e-05	9.36e-06	0.003369	0.003128	0.003381	0.003137	7.39e-04
	8192	3.56e-06	3.86e-06	0.001660	0.001523	0.001664	0.001527	1.62e-04

Table 14: Integration results for the Bermudan Option across two scenarios. We report FDR-corrected p-values from a one-sided Wilcoxon signed-rank test. P-values below 0.05 and the best methods are **bolded**.