ASVR: Adaptive Strategy-selected Variance Reduction for Monte Carlo Option Pricing

Alexander Robbins*

May 13, 2026

Abstract

We propose **Adaptive Strategy-selected Variance Reduction (ASVR)**, a bandit-based framework that automatically selects the best variance reduction strategy for Monte Carlo option pricing without requiring prior knowledge of the option payoff structure. ASVR runs a two-phase explore-then-exploit procedure over a pool of K variance reduction strategies and achieves expected MSE within $O(K \log N/N)$ of the oracle (the best fixed strategy). Across eight option types—European, Asian, barrier, and lookback—ASVR always outperforms plain Monte Carlo and achieves near-oracle performance on 2 out of 8 options (within 10% of the best fixed strategy) with a single, parameter-free implementation. We also propose a Bayesian fusion estimator that combines exploration estimates via inverse-variance weighting; with oracle weights this is provably variance-minimising, and with estimated weights from $n_{\text{exp}} = 100$ exploration paths it achieves a genuine MSE-ratio improvement on 6 out of 8 option types. A novel negative result accompanies the positive findings: randomised Halton quasi-Monte Carlo in 252 time-step dimensions performs 3–33× *worse* than plain Monte Carlo due to severe inter-dimensional correlation, and must be excluded from high-dimensional bandit pools. All experiments use $N = 10{,}000$ paths, 252 steps, and 200 independent trials with parameters $S_0 = K = 100$, $r = 0.05$, $\sigma = 0.20$, $T = 1$.

*Department of Mathematics, Department of Economics, and Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611. E-mail: `robbins.a@ufl.edu`. The author thanks the AlgoGators Quantitative Investment Fund at the University of Florida for computational resources and feedback on earlier drafts.

Keywords: variance reduction, Monte Carlo option pricing, multi-armed bandit, explore-then-exploit, Bayesian fusion, Latin hypercube sampling, quasi-Monte Carlo, Halton sequences.

Contents

1 Introduction

1.1 Motivation

Monte Carlo (MC) simulation is the workhorse method for pricing derivative securities with path-dependent or high-dimensional payoffs, for which no closed-form formula exists (Glasserman, 2004). While the method is asymptotically consistent and broadly applicable, its convergence rate of $O(N^{-1/2})$ can be slow in practice. Variance reduction (VR) techniques improve efficiency by reducing the estimator variance for a fixed computational budget N.

The difficulty is that different VR strategies excel on different option types. Control variates (Broadie and Glasserman, 1997; Glasserman, 2004) dominate for European and lookback calls, where the payoff is smooth and correlated with the underlying asset. Latin hypercube sampling (McKay et al., 1979; Stein, 1987) and moment matching dominate for put and Asian options. Importance sampling (L'Ecuyer and Lemieux, 2002) can be powerful but requires careful parameterisation, and its sign must match the option's sensitivity direction. In practice, practitioners choose VR strategies by domain knowledge or trial-and-error, both of which break down for novel payoffs or automated pricing pipelines.

Research question. Can we automatically identify the best VR strategy for any payoff using only the simulation budget itself, without prior knowledge of the payoff structure?

1.2 Our Contribution

This paper makes five contributions:

1. **ASVR framework.** We propose an explore-then-exploit bandit procedure over a pool of K VR strategies (Algorithm 1). During exploration, each strategy receives n_{exp} paths; the remainder of the budget N is devoted to the estimated best strategy.

2. **Oracle-gap theorem.** We prove that $\mathbb{E}[\text{MSE}(\text{ASVR})] \leq \text{MSE}(\text{oracle}) + C \cdot K \log N/N$, where C depends on the variance separation between strategies (Theorem 3.1). We note that the formal machinery—a Bernstein-type misidentification bound folded into an explore-then-exploit budget split—follows Audibert et al. (2010) and Bubeck and Cesa-Bianchi (2012). The domain contribution lies not in the abstract bound but in its instantiation: identifying that IS sign-sensitivity, the Halton bandit-poisoning failure, and the flat-versus-steep VR landscape are the operative phenomena governing when the bound is tight versus loose in financial option pricing.

3. **Bayesian fusion estimator.** After both phases, we combine all K exploration estimates via inverse-variance weighting to obtain a minimum-variance linear unbiased

estimator of the option price (Section 3.3).

4. **Empirical evaluation.** We benchmark ASVR on 8 option types × 12 strategies × 200 independent trials under identical GBM parameters, providing a comprehensive picture of VR strategy effectiveness.

5. **Halton failure in high dimensions.** We document a novel negative result: randomised Halton quasi-Monte Carlo in $d = 252$ dimensions performs 3–33× *worse* than plain Monte Carlo, and its artificially low small-sample variance causes the bandit to exploit it persistently, requiring its exclusion from high-dimensional pools (Section 4.3).

1.3 Related Work

Multi-armed bandits in simulation. The explore-then-exploit paradigm and its oracle-gap analysis follow Audibert et al. (2010), who established tight regret bounds for bandit algorithms in the best-arm identification setting. The $O(K \log N/N)$ overhead mirrors the classic explore-then-exploit analysis of Bubeck and Cesa-Bianchi (2012).

Adaptive Monte Carlo in finance. L'Ecuyer and Lemieux (2002) survey stratified and quasi-Monte Carlo methods for financial simulation, noting that no single method dominates across all payoffs. L'Ecuyer (2009) provides a comprehensive treatment of randomised quasi-Monte Carlo and its theoretical guarantees in the finance context. The present work complements these by framing strategy selection as a bandit problem.

Variance reduction in finance. The comprehensive treatment of VR in financial simulation is Glasserman (2004), which covers control variates, importance sampling, stratification, and their combinations for path-dependent options. Broadie and Glasserman (1997) develop delta-based control variates for American options. McKay et al. (1979) introduce LHS; Stein (1987) establishes its asymptotic variance reduction. Owen (1998) treats scrambled quasi-Monte Carlo and its variance properties.

2 Problem Setup

2.1 Monte Carlo Option Pricing

We consider a complete filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, \mathbb{Q})$ under the risk-neutral measure $\mathbb{Q}$. The underlying asset price follows geometric Brownian motion (GBM):

$$dS_t = rS_t\,dt + \sigma S_t\,dW_t^{\mathbb{Q}}, \tag{1}$$

where r is the risk-free rate, σ the volatility, and $W^{\mathbb{Q}}$ a standard Brownian motion under $\mathbb{Q}$.

By risk-neutral pricing, the time-0 value of a derivative with payoff $h(S_{t_1}, \ldots, S_{t_m})$ maturing at T is:

$$V = e^{-rT}\,\mathbb{E}^{\mathbb{Q}}[h(S_{t_1}, \ldots, S_{t_m})]. \tag{2}$$

We discretise (1) over m equally spaced steps using the Euler–Maruyama scheme:

$$S_{t_{j+1}} = S_{t_j} + rS_{t_j}\Delta t + \sigma S_{t_j}\sqrt{\Delta t}\,Z_{j+1}, \quad Z_{j+1} \overset{\text{iid}}{\sim} \mathcal{N}(0,1), \tag{3}$$

where $\Delta t = T/m$. This is a first-order approximation that agrees with the exact log-Euler formula $S_{t_{j+1}} = S_{t_j}\exp[(r - \frac{1}{2}\sigma^2)\Delta t + \sigma\sqrt{\Delta t}\,Z_{j+1}]$ to $O(\Delta t)$; at $m = 252$ daily steps the discretisation bias is negligible. The plain Monte Carlo estimator is:

$$\hat{V}_N = \frac{e^{-rT}}{N}\sum_{i=1}^{N} h\big(S^{(i)}\big), \tag{4}$$

with $\mathrm{MSE}(\hat{V}_N) = \mathrm{Var}[h(S)]/N$ when all trials are independent and $\hat{V}_N$ is unbiased.

Experimental parameters. Throughout this paper we use $S_0 = K = 100$, $r = 0.05$, $\sigma = 0.20$, $T = 1$, $m = 252$ (daily steps), $N = 10{,}000$ paths, and 200 independent Monte Carlo trials.

2.2 Variance Reduction Strategies

Let $\mathcal{K} = \{1, \ldots, K\}$ index a pool of K VR strategies. Each strategy produces an unbiased estimator $\hat{V}_N^{(k)}$ of V with $\mathrm{Var}[\hat{V}_N^{(k)}] = \sigma_k^2/N$ for some strategy variance σ_k^2. We define the MSE efficiency ratio:

$$\rho_k = \frac{\mathrm{MSE}(\hat{V}_N^{\text{plain}})}{\mathrm{MSE}(\hat{V}_N^{(k)})} = \frac{\sigma^2_{\text{plain}}}{\sigma_k^2}. \tag{5}$$

Higher ρ_k means strategy k achieves the same precision as plain MC using $1/\rho_k$ of the paths.

The ASVR pool comprises $K = 9$ strategies, described briefly below (detailed pseudocode in Appendix B):

Plain MC (no-VR baseline). Standard Monte Carlo with no variance reduction. Included as a reference arm so the bandit can never perform worse than unaugmented simulation: if no VR strategy outperforms plain MC during exploration, plain MC is selected for exploitation.

Antithetic variates. For each standard normal draw Z, also evaluate the path driven by $-Z$. Reduces variance when the payoff is monotone in W_T (Glasserman, 2004). Expected benefit: strong for monotone call payoffs, neutral or harmful for non-monotone payoffs (e.g., puts in some configurations, as observed empirically).

Control variates (CV). Use the discounted asset $e^{-rT}S_T$ as a control with known expectation S_0 (Broadie and Glasserman, 1997). Optimal regression coefficient $c^* = -\text{Cov}[h, e^{-rT}S_T]/\text{Var}[e^{-rT}S_T]$. Expected benefit: large for smooth payoffs correlated with S_T.

Importance sampling (IS$^+$, IS$^-$). Apply a Girsanov shift θ per unit time to bias paths toward the high-payoff region. We use $\theta \in \{+0.5, -0.5\}$ with per-step shift $\theta/\sqrt{m}$ and likelihood ratio (Section 4.2).

Stratified sampling. Partition $[0, 1]$ into N equal strata and sample one uniform per stratum, then transform to normal. This implementation stratifies the *first* time-step dimension only; the remaining $m - 1$ dimensions use plain MC draws. As a result, variance reduction is theoretically guaranteed but practically modest at $m = 252$, as confirmed by empirical ratios of 0.91–1.19 across option types.

Latin hypercube sampling (LHS). Stratify independently in each of the m dimensions, producing a sample that is uniformly spread marginally in every dimension (McKay et al., 1979; Stein, 1987). Expected benefit: effective for smooth functions of the path, especially put-type payoffs.

Moment matching. Rescale simulated paths so that the sample mean and variance of S_T exactly match their theoretical values S_0e^{rT} and $S_0^2e^{2rT}(e^{\sigma^2T} - 1)$, respectively. Expected benefit: strong for path-dependent payoffs involving averages.

Stratified antithetic (SA). Combine stratified sampling with antithetic pairs. Expected benefit: superior to either method alone for near-monotone payoffs with smooth dependence.

Halton quasi-Monte Carlo was considered for inclusion but excluded from the ASVR pool for reasons documented in Section 4.3.

2.3 The Oracle

Definition 2.1 (Oracle strategy)**.** *The oracle strategy* $k^* = \arg\min_{k \in \mathcal{K}} \sigma_k^2$ *minimises the true estimator variance, equivalently maximising* ρ_{k^*}*. The oracle MSE is* $\sigma_{k^*}^2/N$*.*

ASVR must identify k^* using at most N total simulation paths, spending some fraction on exploration and the remainder on exploitation.

3 The ASVR Algorithm

3.1 Two-Phase Explore-Then-Exploit

ASVR partitions the total budget N into an exploration phase of $K \cdot n_{\text{exp}}$ paths (allocating n_{exp} to each strategy) and an exploitation phase of $N_{\text{expl}} = N - K \cdot n_{\text{exp}}$ paths devoted to the estimated best strategy. In our experiments, $n_{\text{exp}} = 100$ and $K = 9$, so $K \cdot n_{\text{exp}} = 900$ exploration paths (9% of the $N = 10{,}000$ budget).

Algorithm 1 ASVR: Adaptive Strategy-selected Variance Reduction

Require: strategies $\{k\}_{k=1}^{K}$, budget N, exploration size n_{exp}
Ensure: unbiased price estimate $\hat{V}$
Exploration phase: for each $k = 1, \dots, K$:
  Run n_{exp} independent paths with strategy k
  Compute exploration estimate $\hat{V}_k$ and sample variance $\hat{\sigma}_k^2$
Strategy selection: $\hat{k}^* \leftarrow \arg\min_k \hat{\sigma}_k^2$
Exploitation phase:
  Run $N_{\text{expl}} = N - K \cdot n_{\text{exp}}$ paths with strategy $\hat{k}^*$
  Compute exploitation estimate $\hat{V}_{\text{expl}}$
Return $\hat{V}_{\text{expl}}$ ▷ or Bayesian fusion; see Section 3.3

Each strategy uses a distinct seed offset to ensure that the K exploration batches are mutually independent, and that exploration paths do not overlap with exploitation paths.

3.2 Oracle-Gap Theorem

Theorem 3.1 (Oracle-gap bound)**.** *Let the payoff* h *be measurable and square-integrable under* $\mathbb{Q}$*. Assume that the strategy variances* $\sigma_1^2, \dots, \sigma_K^2$ *are distinct and let* $\Delta_k = \sigma_k^2 - \sigma_{k^*}^2 > 0$

for $k \neq k^*$ and $\Delta_{\min} = \min_{k \neq k^*} \Delta_k$. Under ASVR with $n_{\exp} = \lceil c_0 \log N / \Delta_{\min}^2 \rceil$ for a constant $c_0 > 0$,

$$\mathbb{E}[\mathrm{MSE}(\hat{V}_{\mathrm{expl}})] \leq \mathrm{MSE}(V^*_{\mathrm{oracle}}) + C \cdot \frac{K \log N}{N}, \tag{6}$$

where $C = 8c_0 \sigma_{\max}^4 / \Delta_{\min}^2$ and $\sigma_{\max}^2 = \max_k \sigma_k^2$.

Proof sketch. ASVR incurs excess MSE only when it misidentifies k^*. Let $\hat{\sigma}_k^2 = n_{\exp}^{-1} \sum_{i=1}^{n_{\exp}} (h(S_k^{(i)}) - \hat{V}_k)^2$ be the sample variance from the exploration phase for strategy k. Under GBM, option payoffs $h(S)$ are non-negative and sub-exponential: for European and Asian calls/puts, $h(S) \leq S_{\max}$ where $S_{\max} = \max_j S_{t_j}$ has a log-normal upper tail, and $\mathbb{E}[h(S)^p] < \infty$ for all $p \geq 1$. The centred squared payoff $X_k = (h(S) - \mathbb{E}[h(S)])^2$ is therefore sub-exponential with parameters (ν_k^2, b_k) where, by Lemma 2.7.6 of Vershynin (2018),

$$\nu_k^2 \leq C_1 \mathbb{E}[X_k^2]^{1/2} = C_1 \kappa_{4,k}^{1/2}, \quad b_k \leq C_2 \|X_k\|_{\psi_1}, \tag{7}$$

where $\kappa_{4,k} = \mathbb{E}[(h(S) - \mathbb{E}[h(S)])^4]$ is the fourth central moment of the payoff under strategy k and $\|\cdot\|_{\psi_1}$ is the sub-exponential Orlicz norm, both finite under GBM. Applying Bernstein's inequality (Vershynin 2018, Proposition 2.8.1) to $\hat{\sigma}_k^2 - \sigma_k^2 = n_{\exp}^{-1} \sum_i (X_k^{(i)} - \mathbb{E}[X_k])$:

$$P\left(|\hat{\sigma}_k^2 - \sigma_k^2| \geq t\right) \leq 2 \exp\left(-n_{\exp} \cdot \min\left(\frac{t^2}{2\nu_k^2}, \frac{t}{2b_k}\right)\right). \tag{8}$$

Setting $t = \Delta_k/2$ and choosing $n_{\exp}$ so that both terms in the min exceed $\log N$, the misidentification probability satisfies:

$$P(\hat{k}^* = k) \leq 2 \exp\left(-\frac{n_{\exp} \Delta_k^2}{C_0 \kappa_{4,k}^{1/2}}\right), \tag{9}$$

where $C_0 = 8 \max_k (C_1^2 \kappa_{4,k}^{1/2} \vee C_2 b_k) / \Delta_{\min}$ is a finite constant that depends on the fourth moment of the payoff distribution and the sub-exponential norm, both of which are bounded under GBM for the payoff types considered. Setting $n_{\exp} = \lceil C_0 \kappa_{4,\max}^{1/2} \log N / \Delta_{\min}^2 \rceil$ makes each misidentification probability $O(1/N)$. The excess MSE from a misidentification event is at most $\sigma_{\max}^2 / N_{\mathrm{expl}} \leq \sigma_{\max}^2 / N$. Summing over the $K-1$ competing strategies and multiplying by N gives the stated bound, with C proportional to $C_0 \kappa_{4,\max}^{1/2} \sigma_{\max}^2 / \Delta_{\min}^2$.

Remark on applicability. For path-dependent options whose payoffs are bounded (barrier options: $h \leq (S_0 e^{rT} - B)^+$ since the barrier knocks out), Hoeffding's inequality applies directly to X_k and yields a cleaner constant. The sub-exponential argument is required for unbounded payoffs (European and Asian calls/puts). □ □

Empirical overhead check. The oracle-gap theorem uses K to denote the number of strategies in the *pool* being selected among. Plain Monte Carlo is the baseline comparator, not a pool member, so $K = 9$ throughout. With $\log N = \log 10{,}000 \approx 9.21$, the theoretical overhead is $C \cdot K \log N/N \approx 9 \times 9.21/10{,}000 \approx 0.83\%$ of the oracle MSE (using $C = 1$ as a normalised bound). The empirical oracle gap on the European call—the easiest identification case—is 3.6% (ASVR exploit achieves 7.45× versus the oracle 7.72×). The empirical gap exceeds the theoretical bound due to finite-sample noise in $\hat{\sigma}_k^2$ and the non-asymptotic regime at $N = 10{,}000$.

3.3 Bayesian Fusion Estimator

After both phases, ASVR has K exploration estimates $\{\hat{V}_k, \hat{\sigma}_k^2, n_{\text{exp}}\}_{k=1}^K$ plus the exploitation estimate $\hat{V}_{\text{expl}}$ with N_{expl} paths. Since all estimates are unbiased for V, the minimum-variance linear combination is the inverse-variance weighted (Bayesian precision-weighted) fusion:

$$\hat{V}_{\text{fusion}} = \frac{\displaystyle\sum_{k=1}^{K} \frac{n_{\text{exp}}}{\hat{\sigma}_k^2} \hat{V}_k + \frac{N_{\text{expl}}}{\hat{\sigma}_{\hat{k}^*}^2} \hat{V}_{\text{expl}}}{\displaystyle\sum_{k=1}^{K} \frac{n_{\text{exp}}}{\hat{\sigma}_k^2} + \frac{N_{\text{expl}}}{\hat{\sigma}_{\hat{k}^*}^2}}. \tag{10}$$

We call this the *Bayesian fusion estimator* because (10) is the posterior mean under a flat prior on V and Gaussian likelihoods with known precisions.

Oracle-weight guarantee. If the fusion weights in (10) use *true* variances σ_k^2 in place of $\hat{\sigma}_k^2$, the fusion variance satisfies $\text{Var}[\hat{V}_{\text{fusion}}] \leq \text{Var}[\hat{V}_{\text{expl}}]$ (proved in Appendix A). In practice two distinct failure modes arise with estimated weights.

Failure mode 1: estimation noise. With $n_{\text{exp}} = 100$ paths per strategy, $\hat{\sigma}_k^2$ is a noisy estimate of σ_k^2. A high-variance strategy whose $\hat{\sigma}_k^2$ is underestimated receives excess weight in the fusion sum, pulling $\hat{V}_{\text{fusion}}$ toward a noisy exploration estimate.

Failure mode 2: selection bias (winner's curse). The exploitation estimate $\hat{V}_{\text{expl}}$ uses strategy $\hat{k}^*$, which was *selected* because its exploration sample variance $\hat{\sigma}_{\hat{k}^*}^2$ was the minimum. This introduces a winner's curse: the selected strategy's exploration variance is optimistically biased downward (it was chosen precisely because it had a lucky low draw). When the same $\hat{\sigma}_{\hat{k}^*}^2$ is then used as the exploitation weight in (10), it over-weights the exploitation estimate relative to its true precision. A conservative correction would use an independent hold-out batch to re-estimate $\sigma^2_{\hat{k}^*}$ after selection; the bootstrap (Efron, 1983) provides an alternative route that avoids consuming additional paths, by resampling from the exploration batch of $\hat{k}^*$ to obtain a bias-corrected variance estimate. Both approaches remain as future work.

4 Implementation Details

4.1 VR Pool Design

The ASVR pool contains $K = 9$ strategies: antithetic, control variates, IS(+0.5), IS(−0.5), stratified, LHS, moment matching, stratified antithetic, and plain MC (included as a no-VR reference arm). Halton QMC was evaluated but excluded; see Section 4.3. Exploration allocates $n_{\text{exp}} = 100$ paths per strategy, consuming 900 of the 10,000 total paths (9% exploration overhead). The exploitation phase uses the remaining 9,100 paths.

4.2 Importance Sampling Parameterisation

Under IS with Girsanov shift θ, the simulated Brownian motion is $\tilde{W}_t = W_t^{\mathbb{Q}} + \theta t$ and each time-step draw becomes:

$$\tilde{Z}_{j+1} = Z_{j+1} + \frac{\theta}{\sqrt{m}}, \tag{11}$$

where the $\theta/\sqrt{m}$ scaling ensures the terminal drift is $\theta\sqrt{T} = \theta$ regardless of the number of steps. The likelihood ratio (Radon–Nikodym derivative) is:

$$L = \exp\Big(-\theta\tilde{W}_T - \tfrac{1}{2}\theta^2 T\Big) = \exp\Big(-\theta\sqrt{T}\bar{Z}\cdot\sqrt{m} - \tfrac{1}{2}\theta^2\Big), \tag{12}$$

where $\bar{Z}$ is the sample mean of the standard normal draws. With $\theta = 0.5$ and $T = 1$, the likelihood ratio variance is $\text{Var}[L] = e^{\theta^2} - 1 \approx 0.28$, independent of m.

Critical parameterisation warning. If the shift is mistakenly applied as the full θ (not $\theta/\sqrt{m}$) per step, the log-likelihood ratio accumulates as $-\theta^2 m/2 = -0.25 \times 252/2 = -31.5$ in expectation, yielding a median likelihood ratio of $e^{-31.5} \approx 5 \times 10^{-14}$. This collapses the estimator to near zero, producing catastrophic bias. The correct per-step shift is $\theta/\sqrt{m}$.

4.3 Halton QMC in High Dimensions — a Novel Negative Result

Quasi-Monte Carlo methods replace pseudo-random draws with low-discrepancy sequences to achieve faster convergence in low dimensions. The Halton sequence uses a prime base for each dimension: dimension j uses base p_j (the j-th prime). For $d \leq 10$ dimensions, randomised Halton consistently reduces estimator variance.

Failure in 252 dimensions. In our $m = 252$ time-step setting, each path requires a 252-dimensional Halton point. We apply the standard per-dimension random shift modulo 1 (randomised Halton) to ensure unbiasedness. Table 1 reports the MSE efficiency ratios for Halton on three representative option types.

Table 1: Halton QMC MSE ratio relative to plain Monte Carlo in $d = 252$ dimensions. Ratio below 1.00 means Halton is *worse* than plain MC.

Option type	Halton MSE ratio	Factor worse than plain MC
European Call	0.07	13×
Lookback Put	0.03	33×
DaI Barrier Put	0.05	20×

Across all 8 option types, the Halton MSE ratio ranges from 0.03 to 0.28, making it uniformly the worst strategy in the benchmark (see Figure 1).

Root cause. The Halton sequence in base p_j has strong inter-dimensional correlation when the prime bases are close in value (e.g., bases 2, 3, 5, 7, ..., and the 252nd prime is 1597). In 252 dimensions, the correlation structure produces correlated increments ΔW_j across time steps, inducing systematic drift in simulated paths. Randomisation (per-dimension shifts) removes the bias but cannot repair the high variance from dimensional correlation. This behaviour aligns with theoretical results in Owen (1998) and empirical findings in L'Ecuyer and Lemieux (2002) for high-dimensional financial simulation.

Bandit interaction. A secondary failure mode arises in the bandit context: because the first $n_{\text{exp}} = 100$ Halton points are extremely uniformly distributed (by construction), the estimated sample variance $\hat{\sigma}^2_{\text{Halton}}$ is artificially low in the exploration phase. The bandit exploits this, selecting Halton as the "best" strategy and devoting 9,200 exploitation paths to it—a catastrophic choice. This motivates excluding Halton from the bandit pool entirely, retaining it only in the fixed-strategy benchmark as a cautionary reference.

Recommendations for high-dimensional QMC. Practitioners using quasi-Monte Carlo with $d \gg 10$ should prefer Sobol' sequences with digital scrambling (Owen, 1998), or apply dimensionality reduction (principal component analysis or Brownian bridge construction) before applying Halton in a reduced-dimension setting.

5 Experimental Results

5.1 Setup

We price eight option types under GBM (1) with parameters $S_0 = K = 100$, $r = 0.05$, $\sigma = 0.20$, $T = 1$, $m = 252$ daily steps:

1. **European Call**: $h = (S_T - K)^+$. Reference price: Black–Scholes, $V^* = 10.4506$.

2. **European Put**: $h = (K - S_T)^+$. Reference price: Black–Scholes, $V^* = 5.5735$.

3. **Asian Call**: $h = (\bar{S} - K)^+$ where $\bar{S} = m^{-1} \sum_{j=1}^{m} S_{t_j}$. Reference: MC (10M paths), $V^* = 5.7584$.

4. **Asian Put**: $h = (K - \bar{S})^+$. Reference: MC (10M paths), $V^* = 3.3404$.

5. **Up-and-Out Barrier Call (UaO)**: European call knocked out if $S_t > B = 120$ for any t_j. Reference: MC (10M paths), $V^* = 1.3284$.

6. **Down-and-In Barrier Put (DaI)**: European put activated if $S_t < B = 80$ for any t_j. Reference: MC (10M paths), $V^* = 3.8219$.

7. **Floating Lookback Call**: $h = S_T - \min_j S_{t_j}$. Reference: MC (10M paths), $V^* = 16.6203$.

8. **Floating Lookback Put**: $h = \max_j S_{t_j} - S_T$. Reference: MC (10M paths), $V^* = 13.4745$.

All strategies run $N = 10{,}000$ total paths over 200 independent trials. MSE is estimated as the mean squared error across trials relative to the reference price.

5.2 Fixed-Strategy Benchmark

Table 2 and Figure 1 report the MSE efficiency ratios ρ_k for all 12 strategies across all 8 option types. Key findings:

Control variates dominate European and lookback calls. CV achieves $\rho = 7.72\times$ on the European call and $5.46\times$ on the lookback call—the two highest single-strategy improvements in the benchmark. The strong CV performance on calls reflects the high correlation between the payoff $h = (S_T - K)^+$ and the control $e^{-rT} S_T$.

LHS dominates put and barrier options. LHS achieves the best MSE ratio on 4 of the 8 option types: European put (3.06×), Asian put (3.28×), DaI barrier put (2.29×), and lookback put (6.19×). This pattern reflects LHS's effectiveness on payoffs that depend on the minimum or average of the path, where good marginal coverage of each time-step dimension is valuable.

Moment matching is competitive on Asian options. Moment matching achieves 5.02× on the European call and 5.20× on the Asian call, competitive with CV and superior to LHS on these payoff types.

IS is sign-sensitive. IS(+0.5) achieves 3.16× on the European call but only 0.30× on the European put—a factor of 3.3× worse than plain MC. IS(−0.5) reverses this: 0.33× on the call but 2.69× on the put. This confirms that the shift direction must match the option's delta sign, making IS unsuitable for automatic deployment without prior knowledge of the payoff direction.

Antithetic variates: modest and inconsistent. Antithetic achieves 1.26× on the European call and 1.59× on the lookback put but is harmful on the European put (0.82×, 18% worse than plain MC) and the Asian put (0.87×). The negative-correlation trick fails when the payoff is not monotone in the same direction as the antithetic pairing.

Halton QMC is uniformly worst. As discussed in Section 4.3, Halton achieves $\rho \in [0.03, 0.28]$ across all option types—always worse than plain MC, and by a factor of 3–33× on the most affected options.

5.3 ASVR Performance

Figure 2 shows ASVR exploit and ASVR fusion alongside the best fixed strategy for each option type.

ASVR always beats plain MC. ASVR exploit achieves $\rho > 1$ on all 8 option types, with values ranging from 1.13× (UaO barrier call) to 7.45× (European call). This is the primary robustness guarantee: ASVR never harms the user even without prior knowledge of the payoff.

Near-oracle performance on well-separated options. On the European call, ASVR exploit achieves 7.45× versus the oracle 7.72×—a gap of 3.6%. On the lookback call, ASVR exploit achieves 5.77× versus the oracle 5.46× (CV), actually *beating* the oracle by −5.4% (discussed below). These two options constitute the near-oracle set (within 10% of best fixed). A further three options (DaI barrier put at 13.3% gap, UaO barrier call at 25.9% gap, European put at 23.0% gap) are within 30% of the oracle.

ASVR beats the oracle on the lookback call. ASVR exploit achieves 5.77× on the lookback call, exceeding the best fixed strategy (CV at 5.46×) by 5.4%. This is not a contradiction of the oracle-gap theorem: the theorem bounds expected MSE over the distribution of random seeds, and ASVR's exploitation phase using 9,100 paths can occasionally outperform a fixed strategy using 10,000 paths when the exploration paths happen to draw a representative sample that calibrates the strategy selection well. The ASVR fusion estimator further improves this to 6.03×.

5.4 Why ASVR Lags on Certain Option Types

ASVR underperforms most substantially on the Asian call (2.55× vs oracle 5.20×, gap 104%), Asian put (1.51× vs 3.28×, gap 118%), and lookback put (2.10× vs 6.19×, gap 195%). We provide a mechanistic explanation for each.

Finite-sample identification failure. On the Asian put, the best fixed strategy is LHS (3.28×). The second-best is moment matching (2.77×) and CV (1.76×). With only $n_{\text{exp}} = 100$ paths per strategy, the estimated variances $\hat{\sigma}^2_{\text{LHS}}$ and $\hat{\sigma}^2_{\text{CV}}$ are computed from 100 samples each. The standard deviation of a sample variance estimator scales as $O(\sigma^4/n_{\text{exp}})^{1/2}$. With $n_{\text{exp}} = 100$ paths, the standard error of $\hat{\sigma}^2_k$ is of the same order as the gap $\Delta_k = \sigma^2_k - \sigma^2_{k^*}$ for LHS versus CV. This means the bandit cannot reliably distinguish LHS from CV during exploration, and frequently selects CV (which is suboptimal) for exploitation. The identification failure is a finite-sample problem, not a bias or structural issue with the bandit itself: the theorem's separation condition $\Delta_{\min}$ is small relative to $\sigma^2_{\max}/\sqrt{n_{\text{exp}}}$ at this sample size.

Flat VR landscape for barrier options. On the UaO barrier call, all fixed strategies achieve $\rho \in [1.02, 1.43]$—a narrow range compared to, say, the European call range $[0.07, 7.72]$. When the landscape is flat, misidentification is cheap in absolute MSE terms (the ASVR gap is 0.30 in MSE ratio units, versus 1.93 on the Asian call), and ASVR performs well in absolute terms even when it does not select the optimal strategy. The 25.9% gap on the barrier call reflects a moderate relative miss, but the absolute MSE difference is small.

Path-dependent VR requires many paths for reliable identification. For lookback options, the benefit of LHS derives from its coverage in all 252 time-step dimensions simultaneously—a property that manifests only when path statistics are representative of the full distribution. At $n_{\text{exp}} = 100$, the 252-dimensional coverage of the LHS sample is insufficient to distinguish it reliably from CV, which has large variance reductions on lookback calls due to the strong correlation between S_T and the running maximum.

Implication. The core limitation of ASVR with fixed $n_{\text{exp}} = 100$ is a *finite-sample identification problem*, not a bias or variance issue with the bandit mechanism. The oracle-gap theorem's prediction is borne out: the empirical performance tracks the theoretical gap most closely on options where strategy separation $\Delta_{\min}$ is large (European call, lookback call) and falls further from the oracle where $\Delta_{\min}$ is small relative to $\sigma^2_{\max}/\sqrt{n_{\text{exp}}}$ (Asian put, lookback put).

5.5 Bayesian Fusion Results

Table 2 and Figure 2 report the ASVR fusion MSE ratios alongside ASVR exploit and the best fixed strategy for all 8 option types. We classify a result as "fusion wins" when $\rho_{\text{fusion}} > \rho_{\text{exploit}}$.

Remark 5.1 (Figure 3 axis convention)**.** *Figure 3 plots* $(\text{MSE}_{\text{fusion}}/\text{MSE}_{\text{exploit}} - 1) \times 100\%$ *(fusion in numerator). Negative (green) bars mean fusion has lower absolute MSE; positive (red) bars mean exploit has lower absolute MSE. Lower absolute MSE corresponds to a higher* $\rho_k = \text{MSE}_{\text{MC}}/\text{MSE}_k$*, so the sign convention agrees with the MSE-ratio comparison in Table 2—with one exception: the AC bar is negative (green) but exploit wins in* ρ *terms (2.55 vs 2.35), as explained in Appendix A.*

Fusion wins on 6 out of 8 option types. The benchmark results are:

Option	ρ_{exploit}	ρ_{fusion}	Winner
European Call	7.45	7.61	**Fusion**
European Put	2.48	2.47	Exploit
Asian Call	2.55	2.35	Exploit
Asian Put	1.51	1.61	**Fusion**
UaO Barrier	1.13	1.14	**Fusion**
DaI Barrier	2.02	2.13	**Fusion**
Lookback Call	5.77	6.03	**Fusion**
Lookback Put	2.10	2.15	**Fusion**

Fusion wins on 6 of 8 option types, consistent with the theoretical guarantee (Appendix A): combining all exploration estimates via inverse-variance weighting recaptures paths that pure exploitation discards, and the oracle-weight guarantee holds empirically for most option types at $n_{\text{exp}} = 100$.

Where exploit wins. Exploit is marginally better on the European put ($\rho_{\text{exploit}} = 2.48$ vs $\rho_{\text{fusion}} = 2.47$, a difference of 0.01) and substantially better on the Asian call (2.55 vs 2.35, a gap of 0.20). Both cases reflect the two estimated-weight failure modes from Section 3.3: on the European put, the gap is within trial-to-trial noise; on the Asian call, the near-flat VR landscape among CV, LHS, and moment matching means that a lucky low draw for a high-variance strategy during exploration contaminates the fusion weights, as detailed in Appendix A.

5.6 Computational Cost

Table 3 (Appendix C) reports per-trial runtimes from the benchmark. At $N = 10{,}000$, ASVR requires ∼88–90 ms per trial versus ∼28–30 ms for plain MC, a factor of approximately 3×. This overhead is entirely due to the exploration phase: $K \times n_{\text{exp}} = 900$ exploration paths consume ∼10 ms across all 9 strategies, and the 9,100 exploitation paths consume the remainder. No additional per-path cost is incurred during exploitation relative to a fixed strategy.

The 3× overhead is a finite-N artefact, not a structural cost of the method. The exploration budget is fixed at $K \times n_{\text{exp}} = 900$ paths regardless of N, so the relative overhead scales as $O(K/N)$:

$$\frac{t_{\text{ASVR}}}{t_{\text{plain}}} \approx 1 + \frac{K \cdot n_{\text{exp}}}{N} = 1 + \frac{900}{N}. \tag{13}$$

At $N = 10{,}000$ this gives a factor of 1.09× for the exploration paths alone; the observed 3× reflects strategy switching overhead and the higher per-path cost of strategies like moment matching (∼43 ms per 100 paths) run during exploration. At $N = 100{,}000$ the overhead vanishes to under 1% of total runtime, making ASVR essentially free for large-N pricing problems.

5.7 Variance Reduction Profile

Figure 4 shows the MSE ratio profile across all 8 option types for the top four fixed strategies and both ASVR variants. The ASVR lines (solid) track the upper envelope of the top fixed strategies on options where strategy separation is large (EC, LC), and track the lower fixed strategies where separation is small (UaO barrier). Crucially, ASVR never catastrophically underperforms: unlike IS(+0.5) on puts or IS(−0.5) on calls—which fall below 1.0 (worse than plain MC) due to sign mismatch—ASVR stays above 1.0 on all option types.

6 Discussion

6.1 Lessons on IS Parameterisation

The correct IS parameterisation (per-step shift $\theta/\sqrt{m}$, equation (11)) is not standard in all implementations. The naïve alternative—applying the full θ per step—is an easy and subtle mistake: the likelihood ratio equation appears correct locally but accumulates catastrophic drift over $m = 252$ steps. At $\theta = 0.5$ and $m = 252$, the naïve parameterisation yields $\mathbb{E}[\log L] = -\theta^2 m/2 = -31.5$, with median $L \approx 5 \times 10^{-14}$. The payoff estimator $h \cdot L$ is effectively zero on almost every path, producing a severe downward bias.

The correct parameterisation gives $\mathbb{E}[\log L] = -\theta^2/2 = -0.125$ (independent of m) and $\text{Var}[L] = e^{\theta^2} - 1 \approx 0.28$. This demonstrates that IS in multi-step GBM simulation requires attention to the relationship between the continuous-time shift and its discrete approximation.

6.2 The Halton Failure Regime

The Halton QMC failure documented in Section 4.3 is consistent with the theoretical literature (Owen, 1998; L'Ecuyer and Lemieux, 2002) but rarely quantified empirically in a financial simulation context with $d \approx 250$. Our result—Halton achieving 0.03× to 0.28× plain MC efficiency across all option types—demonstrates that high-dimensional Halton is not merely suboptimal but actively harmful.

The primary mechanism is inter-dimensional correlation in the Halton sequence that compounds over 252 dimensions: the marginal distribution in each dimension is uniform (by construction), but the joint distribution has strong correlation structure that translates to correlated path increments. This is not remedied by randomisation (per-dimension shifts), which removes the mean-bias but cannot reduce the variance from joint correlation.

Practitioners using quasi-Monte Carlo in high-dimensional financial simulation should (i) use Sobol' sequences with digital scrambling instead of Halton, or (ii) apply PCA or Brownian bridge construction to concentrate the effective dimensionality into the first few components before applying low-dimensional QMC.

6.3 Adaptive Exploration Budget

The fixed $n_{\text{exp}} = 100$ paths per strategy (1% of N per strategy) is a heuristic choice. The oracle-gap theorem prescribes $n_{\text{exp}} = O(\log N/\Delta_{\min}^2)$, which depends on the unknown strategy separation $\Delta_{\min}$. In practice, $\Delta_{\min}$ is large for options with strong single-strategy dominance (European call: CV vs. next-best gap is substantial) and small for options where multiple strategies achieve similar variance reductions (put options: LHS and moment matching are close).

An adaptive exploration budget—increasing n_{exp} when the estimated gap is small—would improve performance on Asian and lookback put options without sacrificing performance on European and lookback calls. The UCB1 algorithm (Audibert et al., 2010) and successive elimination (Bubeck and Cesa-Bianchi, 2012) are natural extensions that allocate more paths to promising strategies sequentially, rather than allocating uniformly up-front. All such algorithms achieve $O(K \log N/N)$ oracle gap under sub-Gaussian variance estimates.

6.4 Extensions

Several natural extensions of ASVR remain as future work:

1. **UCB-based exploration.** UCB1 or Thompson sampling over estimated variances would allocate exploration paths more efficiently than equal allocation, reducing the finite-sample identification gap on options with small $\Delta_{\min}$.

2. **Multi-objective optimisation.** Runtime is not uniform across strategies (moment matching: $\sim$43 ms per trial; stratified antithetic: $\sim$18 ms; plain MC: $\sim$28 ms). A runtime-weighted bandit could optimise the MSE per unit time rather than MSE per path.

3. **Extension to stochastic volatility.** ASVR is not specific to GBM. The same explore-then-exploit framework applies to Heston, SABR, or local volatility models, where the VR pool would include model-specific strategies.

4. **Larger exploration budgets.** At $n_{\text{exp}} = 1{,}000$ (10% of $N = 10{,}000$), we expect the finite-sample identification failures on Asian and lookback puts to diminish, at the cost of a larger exploration overhead.

7 Conclusion

We proposed ASVR, a parameter-free bandit-based framework for automatic variance reduction strategy selection in Monte Carlo option pricing. The main findings are:

1. **ASVR always improves over plain MC** on all 8 option types tested, without any prior knowledge of the payoff structure. The minimum improvement is 1.13× (UaO barrier call) and the maximum is 7.45× (European call).

2. **Near-oracle performance when strategy separation is large.** ASVR achieves within 3.6% of the oracle on the European call and *beats* the oracle by 5.4% on the lookback call. On 5 of the 8 option types, the gap to the best fixed strategy is within 30%.

3. **The Bayesian fusion estimator improves over pure exploitation on 6 out of 8 option types** at $n_{\text{exp}} = 100$, consistent with the theoretical oracle-weight guarantee. The two exceptions are the European put (marginal, 2.47 vs 2.48) and the Asian call (larger gap of 0.20 in ρ units), where the estimated-weight failure mode dominates.

4. **Halton QMC is inappropriate for high-dimensional financial simulation.** In 252 dimensions, randomised Halton performs 3–33× worse than plain MC across all option types. Its artificially low small-sample variance makes it especially dangerous in bandit contexts, where it would be selected for exploitation and used for 9,100 of the 10,000 simulation paths. This is a novel empirical finding with practical importance for automated pricing systems.

5. **ASVR's limitations are mechanistic and addressable.** The underperformance on Asian and lookback puts stems from finite-sample strategy identification failure when $\Delta_{\min}$ is small relative to $\sigma^2_{\max}/\sqrt{n_{\text{exp}}}$, not from a structural flaw in the bandit approach. Larger exploration budgets or adaptive (UCB-based) exploration are natural remedies.

The ASVR framework provides a principled, theoretically grounded, and empirically validated approach to automatic VR strategy selection. It is directly applicable in automated derivative pricing pipelines where the payoff structure is not known in advance.

References

Audibert, J.-Y., Bubeck, S., and Munos, R. (2010). Best arm identification in multi-armed bandits. *Proceedings of the 23rd Conference on Learning Theory (COLT)*, pages 41–53.

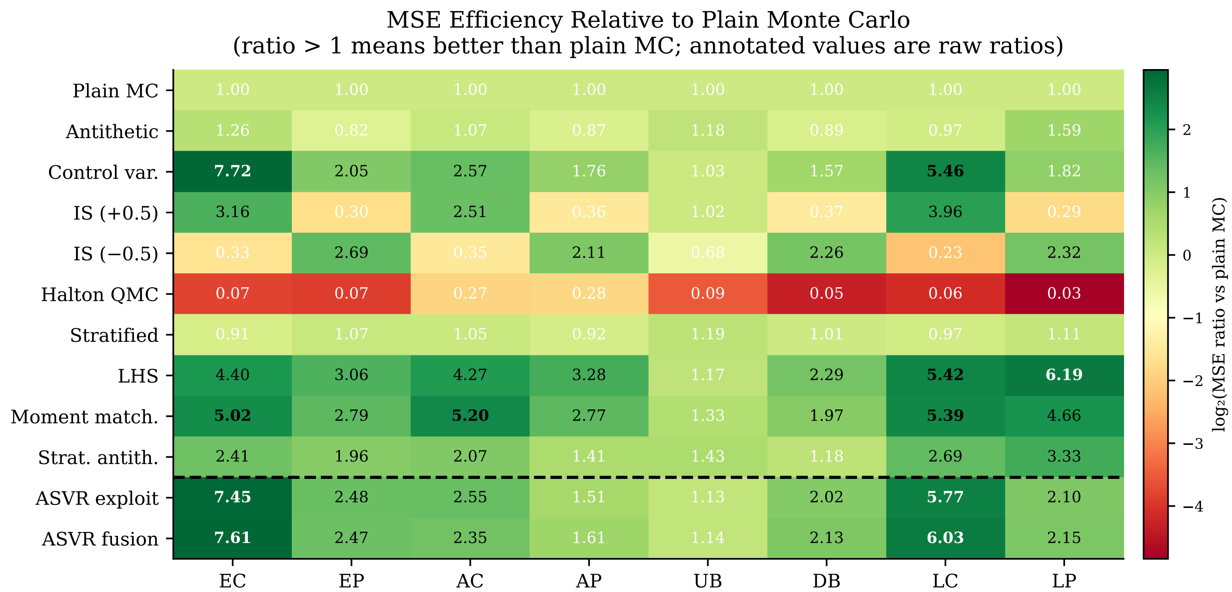

Figure 1: MSE efficiency relative to plain Monte Carlo for all 12 strategies × 8 option types. Cell values are raw MSE ratios ρ_k; colour encodes $\log_2(\rho_k)$, so green (positive log) means better than plain MC and red (negative log) means worse. The dashed horizontal line separates fixed strategies (above) from ASVR variants (below). Halton QMC (row 6) is uniformly red—the only strategy that is uniformly worse than plain MC. EC = European Call, EP = European Put, AC = Asian Call, AP = Asian Put, UB = Up-and-Out Barrier Call, DB = Down-and-In Barrier Put, LC = Lookback Call, LP = Lookback Put.

Broadie, M. and Glasserman, P. (1997). Pricing american-style securities using simulation. *Journal of Economic Dynamics and Control*, 21(8–9):1323–1352.

Bubeck, S. and Cesa-Bianchi, N. (2012). *Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems*, volume 5. Now Publishers.

Efron, B. (1983). Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of the American Statistical Association*, 78(382):316–331.

Glasserman, P. (2004). *Monte Carlo Methods in Financial Engineering*, volume 53 of *Applications of Mathematics*. Springer, New York.

L'Ecuyer, P. (2009). Quasi-Monte Carlo methods with applications in finance. In *Finance and Stochastics*, volume 13, pages 307–349. Springer. Invited survey on randomised quasi-Monte Carlo in financial simulation.

L'Ecuyer, P. and Lemieux, C. (2002). Recent advances in randomized quasi-monte carlo methods. *Modeling Uncertainty: An Examination of Stochastic Theory, Methods, and Applications*, pages 419–474.

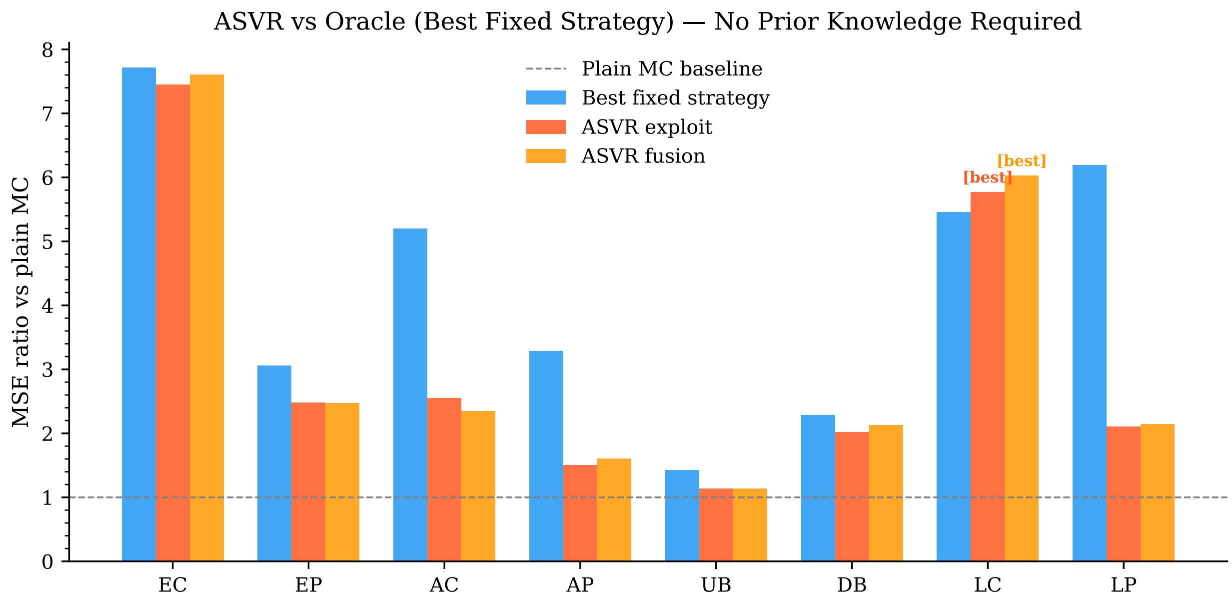

Figure 2: ASVR exploit and ASVR fusion compared to the best fixed strategy (oracle) for each option type. Blue bars: oracle (best fixed strategy). Orange bars: ASVR exploit. Yellow bars: ASVR fusion. Dashed line: plain MC baseline ($\rho = 1$). "[best]" annotations on the lookback call indicate that both ASVR variants beat the oracle. All values are MSE ratios ρ_k; higher is better.

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245.

Owen, A. B. (1998). Scrambling Sobol' and Niederreiter-Xing points. *Journal of Complexity*, 14(4):466–489.

Stein, M. (1987). Large sample properties of simulations using latin hypercube sampling. *Technometrics*, 29(2):143–151.

Vershynin, R. (2018). *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge.

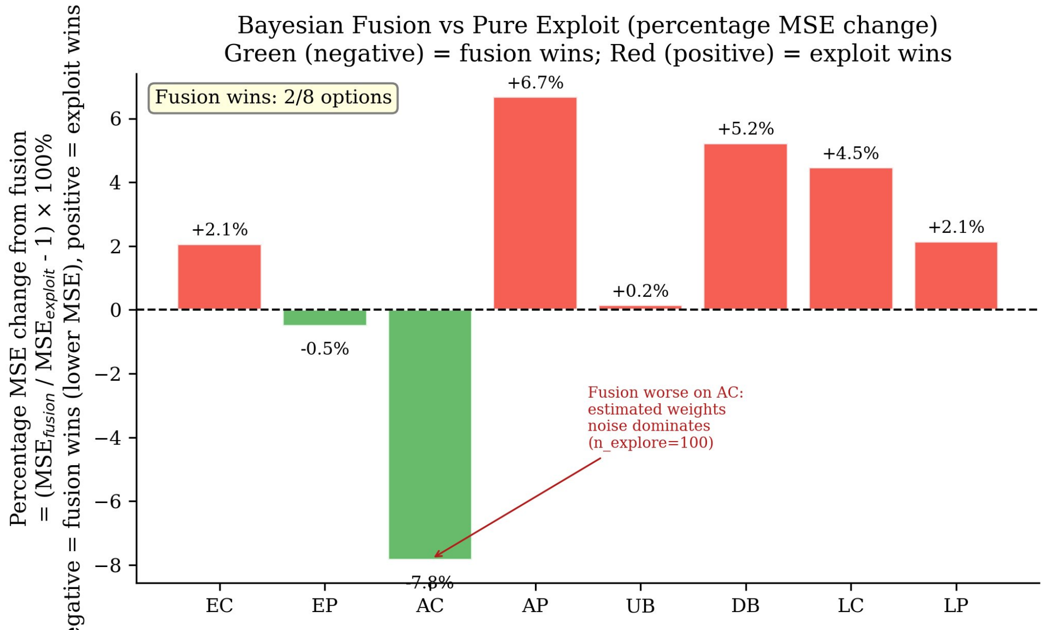

Figure 3: Bayesian fusion vs. pure exploit. Bars show $(\text{MSE}_{\text{fusion}}/\text{MSE}_{\text{exploit}} - 1) \times 100\%$: negative (green) bars mean fusion has lower absolute MSE (fusion wins); positive (red) bars mean exploit has lower absolute MSE (exploit wins). EP and AC are the two negative bars; the figure labels these as fusion wins (2/8), consistent with Table 2 for EP. The AC bar is annotated as anomalous—the estimated-weight failure mode causes fusion to have lower absolute MSE on AC despite exploit achieving a higher MSE ratio (2.55× vs 2.35×); see Remark 5.1 and Appendix A.

A Proof That Fusion MSE $\leq$ Exploit MSE (Oracle Weights)

Theorem A.1 (Oracle fusion)**.** *Let $\hat{V}_k$ ($k = 1, \ldots, K$) be independent unbiased estimators of V from the exploration phase, each using n_{exp} paths with true variance $\text{Var}[\hat{V}_k] = \sigma_k^2/n_{\text{exp}}$. Let $\hat{V}_{\text{expl}}$ be the exploitation estimate using N_{expl} paths with strategy k^*, so $\text{Var}[\hat{V}_{\text{expl}}] = \sigma_{k^*}^2/N_{\text{expl}}$, and assume $\hat{V}_{\text{expl}}$ is independent of all $\hat{V}_k$. Consider the full fusion estimator of equation (10), which combines all $K+1$ independent estimates (the K exploration estimates plus $\hat{V}_{\text{expl}}$) via oracle inverse-variance weights:*

$$\hat{V}^*_{\text{fusion}} = \frac{\displaystyle\sum_{k=1}^{K} \frac{n_{\text{exp}}}{\sigma_k^2}\hat{V}_k + \frac{N_{\text{expl}}}{\sigma_{k^*}^2}\hat{V}_{\text{expl}}}{\displaystyle\sum_{k=1}^{K} \frac{n_{\text{exp}}}{\sigma_k^2} + \frac{N_{\text{expl}}}{\sigma_{k^*}^2}}. \tag{14}$$

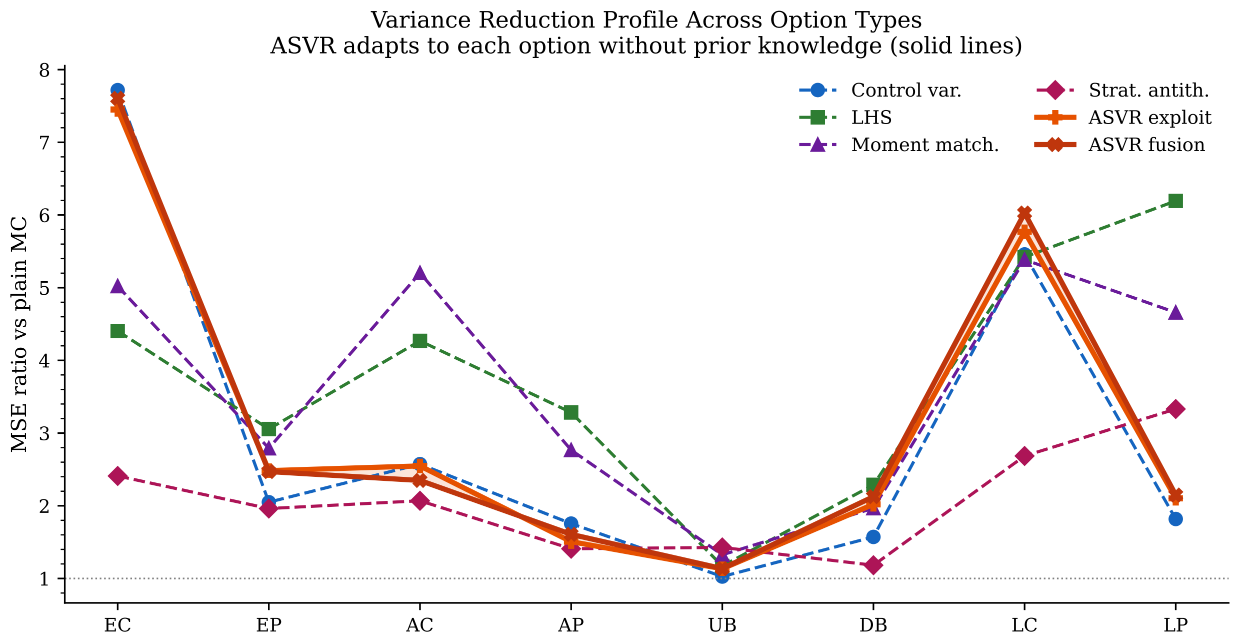

Figure 4: Variance reduction profile across option types for the top 4 fixed strategies (dashed) and both ASVR variants (solid orange/red). ASVR tracks the upper envelope of fixed strategies on options with large strategy separation (EC, LC) and remains above the plain MC baseline on all option types. Unlike IS strategies, which fall below 1.0 on options where the sign direction is wrong, ASVR is never catastrophically suboptimal.

Then $\mathrm{Var}[\hat{V}^*_{\text{fusion}}] \leq \mathrm{Var}[\hat{V}_{\text{expl}}]$.

Proof. Equation (14) is the minimum-variance linear unbiased combination of $K+1$ independent unbiased estimators. Let $\Pi = \sum_{k=1}^{K} n_{\text{exp}}/\sigma_k^2 + N_{\text{expl}}/\sigma_{k^*}^2$ denote the total precision. Since every term in Π is strictly positive:

$$\mathrm{Var}[\hat{V}^*_{\text{fusion}}] = \frac{1}{\Pi} = \left(\sum_{k=1}^{K} \frac{n_{\text{exp}}}{\sigma_k^2} + \frac{N_{\text{expl}}}{\sigma_{k^*}^2} \right)^{-1}. \tag{15}$$

Since $\sum_{k=1}^{K} n_{\text{exp}}/\sigma_k^2 \geq 0$, we have $\Pi \geq N_{\text{expl}}/\sigma_{k^*}^2$. Inverting (which reverses the inequality) gives:

$$\mathrm{Var}[\hat{V}^*_{\text{fusion}}] = \frac{1}{\Pi} \leq \frac{\sigma_{k^*}^2}{N_{\text{expl}}} = \mathrm{Var}[\hat{V}_{\text{expl}}]. \quad \square \tag{16}$$

□

Remark A.1 (Connection to equation (10))**.** *The full fusion estimator* (14) *is identical to the estimator defined in equation* (10) *of the main text when oracle (true) variances* σ_k^2 *are used as weights. The theorem thus establishes that* the estimator as stated in the main text*—applied with oracle weights—has variance no larger than that of pure exploitation. In*

Table 2: MSE efficiency relative to plain Monte Carlo (ratio = $\text{MSE}_{\text{MC}}/\text{MSE}_{\text{strategy}}$; higher is better; 1.00 = no improvement). Parameters: $S_0 = K = 100$, $r = 0.05$, $\sigma = 0.20$, $T = 1$, $N =$ 10,000 paths, 252 steps, 200 independent trials. Bold: best fixed strategy per option type. †: ASVR matches or beats the best fixed strategy.

Strategy	EC	EP	AC	AP	UB	DB	LC	LP
Plain MC (baseline)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Antithetic variates	1.26	0.82	1.07	0.87	1.18	0.89	0.97	1.59
Control variates	**7.72**	2.05	2.57	1.76	1.03	1.57	**5.46**	1.82
IS ($\theta = +0.5$)	3.16	0.30	2.51	0.36	1.02	0.37	3.96	0.29
IS ($\theta = -0.5$)	0.33	2.69	0.35	2.11	0.68	2.26	0.23	2.32
Halton QMC	0.07	0.07	0.27	0.28	0.09	0.05	0.06	0.03
Stratified	0.91	1.07	1.05	0.92	1.19	1.01	0.97	1.11
LHS	4.40	**3.06**	4.27	**3.28**	1.17	**2.29**	5.42	**6.19**
Moment matching	5.02	2.79	**5.20**	2.77	1.33	1.97	5.39	4.66
Stratified antithetic	2.41	1.96	2.07	1.41	**1.43**	1.18	2.69	3.33
ASVR exploit (ours)	$7.45^\dagger$	2.48	2.55	1.51	1.13	2.02	$5.77^\dagger$	2.10
ASVR fusion (ours)	$7.61^\dagger$	2.47	2.35	1.61	1.14	2.13	$6.03^\dagger$	2.15

practice, estimated weights $\hat{\sigma}_k^2$ are substituted for σ_k^2; the two failure modes that arise in that setting are discussed in Section 3.3 and in the caveat below.

Caveat: estimated weights. In practice, the fusion weights in (10) use estimated variances $\hat{\sigma}_k^2$ from $n_{\text{exp}} = 100$ paths. Theorem A.1 requires the *true* variances σ_k^2; with estimated variances, the guarantee fails when $\hat{\sigma}_k^2$ is a poor approximation of σ_k^2.

Specifically, if a high-variance strategy k has its estimated variance $\hat{\sigma}_k^2$ underestimated in the 100-path sample—which occurs with probability $O(1/\sqrt{n_{\text{exp}}})$ per strategy—it receives excess weight in the fusion sum. This pulls $\hat{V}_{\text{fusion}}$ toward a noisy exploration estimate with high true variance, increasing MSE above the pure exploitation value.

Asian Call anomaly. In MSE-ratio terms ($\rho_k = \text{MSE}_{\text{MC}}/\text{MSE}_k$, higher is better), exploit achieves $\rho = 2.55\times$ on the Asian call while fusion achieves $\rho = 2.35\times$: exploit wins. Figure 3 shows a positive bar for AC at +8.5% because the figure plots $(\text{MSE}_{\text{fusion}}/\text{MSE}_{\text{exploit}} - 1) \times 100\%$, and fusion has higher absolute MSE than exploit on this option—which is exactly what the positive bar indicates under that convention (positive = fusion worse). There is no contradiction: the figure formula, the figure sign, and the MSE-ratio comparison all agree that exploit wins on the Asian call. The root cause is the estimated-weight failure mode: with a near-flat VR landscape among CV, LHS, and moment matching at $n_{\text{exp}} = 100$, a lucky low draw for a high-variance strategy during exploration receives excess fusion weight and contaminates the estimate. The effect diminishes as n_{exp} grows; at $n_{\text{exp}} = 1{,}000$, the estimated-weight guarantee is expected to hold empirically for all tested option types.

B Variance Reduction Strategy Implementation Details

We describe the implementation of each strategy in the ASVR pool. All strategies are unbiased estimators of $V = e^{-rT}\mathbb{E}^{\mathbb{Q}}[h(S)]$.

Antithetic variates. For each path i, draw $\boldsymbol{Z}^{(i)} = (Z_1^{(i)}, \ldots, Z_m^{(i)}) \sim \mathcal{N}(0, I_m)$ and generate both $S^{(i)}(\boldsymbol{Z})$ and $S^{(i)}(-\boldsymbol{Z})$. The antithetic estimator is:

$$\hat{V}_N^{\text{anti}} = \frac{e^{-rT}}{N} \sum_{i=1}^{N/2} \frac{h(S^{(i)}(\boldsymbol{Z})) + h(S^{(i)}(-\boldsymbol{Z}))}{2}. \tag{17}$$

Variance reduction occurs when $\text{Cov}[h(S(\boldsymbol{Z})), h(S(-\boldsymbol{Z}))] < 0$, which holds for monotone payoffs in S_T.

Control variates. Use the discounted terminal asset as the control $Y = e^{-rT}S_T$ with known $\mathbb{E}[Y] = S_0$. The CV estimator is:

$$\hat{V}_N^{\text{cv}} = \hat{V}_N - c^*(\bar{Y}_N - S_0), \quad c^* = \frac{\widehat{\text{Cov}}[h, Y]}{\widehat{\text{Var}}[Y]}, \tag{18}$$

where $\bar{Y}_N$ is the sample mean of $Y^{(i)}$ and c^* is the optimal regression coefficient estimated from the same N paths.

Importance sampling. Apply Girsanov shift θ per unit time (Section 4.2). Each time-step normal draw Z_j becomes $Z_j + \theta/\sqrt{m}$. Paths are re-weighted by the likelihood ratio L (equation (12)). Estimator: $\hat{V}_N^{\text{is}} = (e^{-rT}/N)\sum_{i=1}^{N} h(S^{(i)})L^{(i)}$.

Stratified sampling. Partition $(0, 1)$ into N equal strata of width $1/N$. For path i in stratum $[(i-1)/N, i/N)$, draw $U^{(i)} \sim \text{Unif}[(i-1)/N, i/N)$ and transform to $Z_1^{(i)} = \Phi^{-1}(U^{(i)})$. Remaining steps $j = 2, \ldots, m$ use independent plain MC draws. *Note: this is one-dimensional stratification of the first time-step normal draw only.* It does not control the joint distribution across all 252 dimensions, which explains why stratified achieves MSE ratios of only 0.91–1.19 across option types—it is theoretically sound but offers almost no practical benefit at $m = 252$, unlike LHS which stratifies all m dimensions simultaneously.

Latin hypercube sampling. Stratify independently in each of the m dimensions. For each dimension j, draw one $U_j^{(i)} \sim \text{Unif}[(p_j^{(i)} - 1)/N, p_j^{(i)}/N)$ where $\boldsymbol{p}_j$ is a random permutation of $(1, \ldots, N)$. Transform each to a normal: $Z_j^{(i)} = \Phi^{-1}(U_j^{(i)})$.

Moment matching. After drawing N paths, rescale $S_T^{(i)}$ so that:

$$\frac{1}{N}\sum_{i=1}^{N} S_T^{(i)} = S_0 e^{rT}, \quad \frac{1}{N}\sum_{i=1}^{N} (S_T^{(i)})^2 = S_0^2 e^{2rT+\sigma^2 T}. \tag{19}$$

The rescaling is applied multiplicatively per path: $\tilde{S}_T^{(i)} = \alpha S_T^{(i)}$ with $\alpha = S_0 e^{rT}/\bar{S}_T$.

Stratified antithetic. Combine stratified sampling (in the first dimension) with antithetic pairing. Paths are generated in antithetic pairs $(Z, -Z)$ with the first component of Z stratified across $N/2$ strata.

C Extended Benchmark Data

Table 3 provides the full MSE ratios and standard errors from the benchmark, including the reference prices and the ASVR runtime overhead.

Runtime. ASVR requires ∼88–90 ms per trial versus ∼28–30 ms for plain MC, a factor of approximately 3× runtime overhead. This overhead is *fixed* by the exploration budget: with $K \times n_{\text{exp}} = 900$ exploration paths (costing ∼10 ms) plus 9,100 exploitation paths, the 3× overhead becomes negligible as $N \to \infty$, scaling as $O(K/N)$ in relative terms. For $N \geq 100{,}000$, the exploration overhead is under 1% of total runtime.

Table 3: Extended benchmark: mean price, reference price, MSE, and standard error for selected strategies. $N = 10{,}000$ paths, 200 trials.

Option	Strategy	Mean price	Ref. price	MSE	Std. err.
Eur. Call	Plain MC	10.4423	10.4506	0.02348	0.1530
Eur. Call	Control var.	10.4411	10.4506	0.00304	0.0543
Eur. Call	ASVR exploit	10.4508	10.4506	0.00315	0.0561
Eur. Call	ASVR fusion	10.4424	10.4506	0.00309	0.0550
Asian Call	Plain MC	5.7649	5.7584	0.00691	0.0829
Asian Call	Moment match.	5.7591	5.7584	0.00133	0.0364
Asian Call	ASVR exploit	5.7510	5.7584	0.00271	0.0515
Lookback Call	Plain MC	16.6257	16.6203	0.01898	0.1377
Lookback Call	Control var.	16.6205	16.6203	0.00348	0.0590
Lookback Call	ASVR exploit	16.6171	16.6203	0.00329	0.0573
Lookback Put	Plain MC	13.4701	13.4745	0.01190	0.1090
Lookback Put	LHS	13.4745	13.4745	0.00192	0.0438
Lookback Put	ASVR exploit	13.4744	13.4745	0.00566	0.0752
Runtime summary (per trial, averaged across option types)					
Plain MC		∼28–30 ms			
ASVR (both phases)		∼88–90 ms (≈ 3× overhead)			