Optimal randomized multilevel Monte Carlo for repeatedly nested expectations

Yasa Syed [1] **Guanyang Wang** [1]

Abstract

The estimation of repeatedly nested expectations is a challenging task that arises in many real-world systems. However, existing methods generally suffer from high computational costs when the number of nestings becomes large. Fix any non-negative integer D for the total number of nestings. Standard Monte Carlo methods typically cost at least $\mathcal{O}(\varepsilon^{-(2+D)})$ and sometimes $\mathcal{O}(\varepsilon^{-2(1+D)})$ to obtain an estimator up to ε-error. More advanced methods, such as multilevel Monte Carlo, currently only exist for $D = 1$. In this paper, we propose a novel Monte Carlo estimator called READ, which stands for "Recursive Estimator for Arbitrary Depth." Our estimator has an optimal computational cost of $\mathcal{O}(\varepsilon^{-2})$ for every fixed D under suitable assumptions, and a nearly optimal computational cost of $\mathcal{O}(\varepsilon^{-2(1+\delta)})$ for any $0 < \delta < \frac{1}{2}$ under much more general assumptions. Our estimator is also unbiased, which makes it easy to parallelize. The key ingredients in our construction are an observation of the problem's recursive structure and the recursive use of the randomized multilevel Monte Carlo method.

1. Introduction

Monte Carlo methods are a class of algorithms that use random sampling to estimate quantities of interest, such as integrals or expected values. When the estimand can be expressed as an expectation, for example $\mathbf{E}_\pi[g(X)]$, these methods work by generating independent random samples $X_1, \ldots, X_n$ from π, and using the average $\sum_{i=1}^n g(X_i)/n$ as an estimator. Monte Carlo estimators are unbiased and converge at a rate of $n^{-1/2}$, regardless of the dimension of the samples. This dimension-independent convergence rate makes Monte Carlo methods a powerful tool for approximating high-dimensional integrations, as they do not suffer from the curse of dimensionality that plagues deterministic numeric integration methods.

However, the above analysis implicitly assumes the integrand g can be pointwisely evaluated, which may not be possible in many situations. This can arise, for instance, when it is expressed as another integration over latent variables or when it involves solving a optimization problem. In this paper, we study the problem of estimating repeatedly nested expectations (RNE), which means the integrand depends on a sequence of other functions and conditional expectations. Specifically, fix any positive integer D for the total number of nestings, and $\{g_d\}_{d=0}^D$ for a family of real-valued functions which can be pointwisely evaluated. Let $(y^{(0)}, \ldots, y^{(D)})$ be a finite-time stochastic process with underlying joint distribution π, and let $y^{(0:d)}$ denote the vector $(y^{(0)}, \ldots, y^{(d)})$ for every $d \leq D$. The RNE, first formally formulated in (Rainforth et al., 2018), is defined as:

$$\gamma_0 = \mathbf{E}\left[g_0\left(y^{(0)}, \gamma_1\left(y^{(0)}\right)\right)\right], \tag{1}$$

where $\{\gamma_i\}_{i=1}^{D-1}$ is recursively defined as:

$$\gamma_d(y^{(0:d-1)}) = \mathbf{E}\left[g_d\left(y^{(0:d)}, \gamma_{d+1}\left(y^{(0:d)}\right)\right) \,\middle|\, y^{(0:d-1)}\right], \tag{2}$$

and

$$\gamma_D(y^{(0:D-1)}) = \mathbf{E}\left[g_D\left(y^{(0:D)}\right) \,\middle|\, y^{(0:D-1)}\right]. \tag{3}$$

The estimation of Resource-Optimal Nested Expectations (RNEs) is a significant challenge that encompasses various practical scenarios, where the desired outcome relies on multiple stages or decision points. Here, we provide several instances to illustrate this:

- In financial modeling, one crucial problem involves estimating RNEs when γ_0 represents the expected utility of an optimal strategy in a D-horizon optimal stopping problem. Here, $g_d(y^{(0:d)}, u)$ is defined as $\max y^{(d)}, u$ for $0 \leq d \leq D-1$, and $g_D(y^{(0:D)})$ is simply $y^{(D)}$.

[1]Department of Statistics, Rutgers University, New Brunswick, United States. Correspondence to: Guanyang Wang <guanyang.wang@rutgers.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

- When $D = 2$, a recent paper by (Giles et al., 2023) focuses on risk estimation for the credit valuation adjustment. In their analysis, the outermost function g_0 is a Heaviside function, while the inner functions g_1 and g_2 are smooth functions.
- When $D = 1$, RNE estimation finds extensive applications in Bayesian experimental design (Goda et al., 2022), portfolio risk management (Gordy & Juneja, 2010), stochastic and bilevel optimization (Hu et al., 2021), as well as variational Bayes (He et al., 2022).

In addition to the aforementioned examples, RNE estimation, sometimes also referred to nonlinear Monte Carlo, finds relevance in various fields including probabilistic programs (Rainforth, 2018), numerical partial differential equations (PDEs) (Beck et al., 2020), as well as physics and chemistry (Dauchet et al., 2018).

However, estimating RNEs is challenging. As shown in formulas (1) – (3), we are interested in the expectation of g_0, which depends on the random variable $y^{(0)}$ and $\gamma_1(y^{(0)})$ – a conditional expectation of g_1 given $y^{(0)}$. Then g_1 further depends on a random variable $y^{(1)}$ and $\gamma_2(y^{(0)}, y^{(1)})$ which is a conditional expectation of g_2 given $y^{(0)}$ and $y^{(1)}$. This procedure is recursively defined until it reaches the deepest depth, D. Since $\gamma_1(y^{(0)})$ (and also $\gamma_2, \gamma_3, \ldots$) cannot be directly evaluated in most practical cases, estimating RNEs cannot be handled by standard Monte Carlo methods.

The most natural way to estimate RNEs is by nesting Monte Carlo (NMC) estimators. In the $D = 1$ case, this method works by first sampling independent and identically distributed (i.i.d.) copies $y_1^{(0)}, \ldots, y_{N_0}^{(0)}$ according to the distribution of $y^{(0)}$. For each fixed $y_i^{(0)}$, one further samples N_1 i.i.d. $y_1^{(1)}, \ldots, y_{N_1}^{(1)}$ according to $\pi(y^{(1)} \mid y_i^{(0)})$, and uses the standard estimator $\hat{\gamma}_1(y_i^{(0)}) \coloneqq \sum_{i=j}^{N_1} g_1(y_i^{(0)}, y_j^{(1)})/N_1$ to estimate $\gamma_1(y_i^{(0)})$. The final estimator uses the estimated $\hat{\gamma}_1(y_i^{(0)})$ to replace the intractable $\gamma_1(y_i^{(0)})$, i.e.,

$$I_{N_0,N_1} = \frac{1}{N_0} \sum_{i=1}^{N_0} g_0(y_i^{(0)}, \hat{\gamma}_1(y_i^{(0)})).$$

This nested estimator can be easily extended to the general D case, albeit the notations become more complex. Roughly, one still samples N_0 i.i.d. copies according to $\pi(y^{(0)})$, and for each fixed trajectory $y^{(0:d-1)}$, the user generates N_d i.i.d. samples from $\pi(y^{(d)} \mid y^{(0:d-1)})$ all the way to depth D and then form the nested estimator from the deepest depth to the shallower depths. The construction details are referred to Section 3.2 of (Rainforth et al., 2018).

After suitably allocating the number of samples $(N_d)_{d=0}^D$ for each depth, the root-mean-square error (rMSE) of the NMC estimator converges to 0 at a rate of $N^{-1/(2D+2)}$ or $N^{-1/(D+2)}$ (Rainforth et al., 2018), depending on the regularity conditions of the functions $\{g_d\}_{d=0}^D$, where $N = \prod_{d=0}^D N_d$ is the total number of samples used to form a nested estimator. This convergence rate diminishes exponentially with D, meaning that NMC estimators do not have the same dimension-free convergence rate as standard Monte Carlo estimators. As a result, NMC methods require at least $\mathcal{O}(\varepsilon^{-(2+D)})$ and sometimes $\mathcal{O}(\varepsilon^{-2(1+D)})$ samples to get an estimator within ε-rMSE, while standard Monte Carlo estimators require only $O(\varepsilon^{-2})$ samples. Although there are a few cases mentioned in (Rainforth et al., 2018) where the canonical $\mathcal{O}(N^{-1/2})$ rate can be achieved, the problem of estimating RNEs with an optimal (or dimension-free) convergence rate remains largely open.

In the special case $D = 1$, more efficient methods have been proposed (Giles, 2018; Giles & Goda, 2019; Giles & Haji-Ali, 2019) based on the celebrated multilevel Monte Carlo (MLMC) methods (Heinrich, 2001; Giles, 2008). These estimators achieve up to ε-rMSE with cost $O(\varepsilon^{-2}\log(1/\varepsilon)^2)$ or $O(\varepsilon^{-2})$ under varying conditions, comparing favorably with the NMC estimator. However, existing methods cannot be directly generalized to solve the general D case. Meanwhile, implementing these methods requires users to prespecify the precision level ε and conduct preliminary experiments/calculations to carefully estimate/bound the parameters in the MLMC algorithm (see, e.g., Theorem 1 of (Giles & Goda, 2019)). Therefore, existing MLMC estimators seem to be harder to implement and less amendable to our original problem, which has a recursive structure.

In this work, we propose the READ, a novel Monte Carlo estimator for the RNE estimation with an arbitrary number of nestings D. Our construction is interesting in the following three aspects. Firstly, under suitable regularity conditions similar to those in (Rainforth et al., 2018), the rMSE of our estimator has an optimal convergence rate $N^{-1/2}$ regardless of D. Equivalently, our method costs in expectation $\mathcal{O}(\varepsilon^{-2})$ to get an estimator up to ε-rMSE. Under much more general assumptions, our method still achieves a nearly-optimal cost of $\mathcal{O}(\varepsilon^{-2(1+\delta)})$ for any $0 < \delta < \frac{1}{2}$ to get an estimator up to ε-mean-absolute-error (MAE).

It is worth mentioning that most of our effort is devoted to designing unbiased estimators of γ_0 in (1) with finite computational cost and finite variance (or finite (2-δ)-th moment under more general assumptions). After developing such an unbiased estimator, we can simulate independent copies of these estimators and average them. The $N^{-1/2}$ convergence rate and $\mathcal{O}(\varepsilon^{-2})$ cost are then immediate corollaries of the bias-variance decomposition formula, see Corollary 2.3.

Therefore, another appealing property of READ, in contrast to existing methods, is that it admits no estimation bias. Unbiasedness implies these estimators can be implemented in parallel processors without requiring any communication

between them. Designing unbiased estimators has recently attracted much interest in statistics, operations research, and machine learning communities for its potential for parallelization. Our methods add to the rich body of works of (Glynn & Rhee, 2014; Rhee & Glynn, 2015; Blanchet & Glynn, 2015; Jacob et al., 2020; Biswas et al., 2019; Wang et al., 2021; Wang & Wang, 2022; Kahale, 2022).

Finally, our algorithm for constructing READ relies on the randomized multilevel Monte Carlo (rMLMC) method (McLeish, 2011; Rhee & Glynn, 2015; Blanchet & Glynn, 2015), but it is significantly different from its previous applications. Many of the current applications of randomized Multilevel Monte Carlo (rMLMC) methods (Rhee & Glynn, 2015; Vihola, 2018; Goda et al., 2022) also have a deterministic version known as the original Multilevel Monte Carlo (MLMC) (Giles, 2008), which offers similar or even better guarantees in terms of computational cost. As a result, it is natural to speculated that every problem solved by rMLMC has a corresponding deterministic version. However, our findings indicate that this assumption may not always be accurate. The rMLMC framework is well-suited to the recursive structure of RNEs, and can be used as a subroutine in our method. In contrast, the non-randomized MLMC cannot be easily applied to the general case of $D > 1$. This suggests that the rMLMC framework may be more widely applicable than previously thought.

The rest of this paper is organized as follows: in the remainder of this section, we discuss related works, set up our notation, and introduce our technical assumptions. In Section 2, we introduce our algorithm and show that it attains the optimal and nearly-optimal computational cost under two different assumptions, respectively. In Section 3, we demonstrate the empirical performance of our method on several toy examples. We conclude this paper with a short discussion in Section 4. Proof and experiment details are deferred to the Appendix. An additional experiment is also included in Appendix F.

1.1. Related work

Our algorithm design strategy mainly follows the randomized multilevel Monte Carlo (rMLMC) framework (McLeish, 2011; Rhee & Glynn, 2015; Blanchet & Glynn, 2015). Our algorithm is inspired by the unbiased optimal stopping estimator (Zhou et al., 2022), which develops estimators for the optimal stopping problem by recursively calling the rMLMC algorithm. We extend the methodology in (Zhou et al., 2022) both in scope and depth. Our method works with a more general class of problems formulated by (Rainforth et al., 2018), which includes the optimal stopping problem as a special case, and provides more precise results under practical assumptions.

Throughout this paper, we will assume the functions $\{g_d\}_{d=0}^{D-1}$ are all continuous and the process π can be perfectly simulated. When $D = 1$ and g_0 is discontinuous, progress has been made by (Broadie et al., 2011) and (Giles & Haji-Ali, 2019; 2022). When the underlying distribution is itself challenging, users have to first use MCMC to approximately sample from π. The case of $D = 1$ and challenging π is considered in (Wang & Wang, 2022).

1.2. Notations

Now we introduce our notations. Many of our notations follow those used in the original definition (Rainforth et al., 2018), despite generalizing their setting to a multivariate underlying process. Throughout this paper, we preserve the letter D for the total number of nestings. We denote by π the underlying joint distribution of a finite-time, real-valued, M-dimensional stochastic process $(y^{(0)}, \dots, y^{(D)})$, i.e. $y^{(d)} \in \mathbf{R}^M$ for each $0 \le d \le D$.

For every $0 \le i \le j \le D$, we use the $y^{(i:j)}$ to denote the vector $(y^{(i)}, \dots, y^{(j)})$. The conditional distribution of $y^{(d:D)}$ given the value of $y^{(0:d-1)}$ is denoted by $\pi_{d:D}(\cdot \mid y^{(0:d-1)})$. The marginal distribution of $y^{(d)}$ conditioning on $y^{(0:d-1)}$ is denoted by $\pi_d(\cdot \mid y^{(0:d-1)})$. We adopt the convention that $y^{(0:-1)} = \varnothing$, and therefore π_0 stands for the (unconditioned) marginal distribution of $y^{(0)}$. Let Π be any probability distribution on some probability space, and Z be some random variable on the same space, then we use $\|Z\|_{\Pi,m}$ to denote the L^m–norm of Z under Π, i.e., $\left(\mathbf{E}_\Pi[|Z|^m]\right)^{1/m}$. The geometric distribution with parameter r is denoted by $\mathsf{Geo}(r)$. We also define $p_r(n) := \mathbf{P}[\mathsf{Geo}(r) = n] = r(1-r)^n$ for every $n \in \{0, 1, 2, \dots, \}$. For every $0 \le d \le D-1$, the function g_d introduced in (1) – (2) maps from $\mathbf{R}^{(d+1)M+1}$ to $\mathbf{R}$ since g_d takes as its first $d+1$ arguments M-dimensional vectors, and it takes only a scalar as its final argument. The function g_D in (3) maps from $\mathbf{R}^{(D+1)M}$ to $\mathbf{R}$ since g_D has all $D+1$ vectors in the M-dimensional process as its arguments. For random variables $X_1, \dots, X_n$, we denote their summation by $S_n := \sum_{i=1}^n X_i$. When n is even, we denote by $S_{n/2}^{\mathsf{O}} := \sum_{k=1}^{n/2} S_{2k-1}$ and $S_{n/2}^{\mathsf{E}} := \sum_{k=1}^{n/2} S_{2k}$ the summations of their odd and even terms, respectively.

1.3. Assumptions

Throughout this paper, we assume that we can access a simulator $\mathcal{S}$. The simulator can take any trajectory $y^{(0:d-1)}$ with $0 \le d \le D$ as input, and outputs $y^{(d)}$ which follows the distribution $\pi_d(\cdot \mid y^{(0:d-1)})$. In particular, $\mathcal{S}$ can take $\varnothing$ as input and simulates $y^{(0)} \sim \pi_0$. Calling $\mathcal{S}$ recursively for $D+1$ times generates one complete sample path. This assumption enables us to sample from any marginal or conditional distribution perfectly. This assumption is also standard and is posed explicitly or implicitly in nearly all the existing works concerning the estimation of nested

expectations, see (Giles & Goda, 2019; Goda et al., 2022; Zhou et al., 2022) for examples.

For $0 \leq d \leq D-1$, fix $g_d : \mathbf{R}^{(d+1)M+1} \to \mathbf{R}$. We say g_d satisfies the last-component bounded second derivative condition (LBS) if there exists a $K_d < \infty$ such that

$$\sup_{(y^{(0:d)},z)} \left|\partial^2_{(d+1)M+1} g_d(y^{(0:d)}, z)\right| < K_d. \tag{4}$$

We say g_d satisfies the last-component bounded Lipschitz condition (LBL) if there exists an $L_d < \infty$ such that for all $x, z \in \mathbf{R}$

$$\sup_{y^{(0:d)}} |g_d(y^{(0:d)}, x) - g_d(y^{(0:d)}, z) < L_d|x-z|. \tag{5}$$

These assumptions (and their variants) are also posed in related works such as (Rainforth, 2018; Blanchet & Glynn, 2015; Giles, 2018).

2. Algorithm, estimator, and theoretical results

Now we are ready to present our main results. As discussed in Section 1, we will be focusing on designing a Monte Carlo estimator which is unbiased, has a finite computational cost, and has finite variance or (2-δ)-th moment under different assumptions.

2.1. Preliminary analysis

One of the challenges in estimating the RNEs is the difficulty of estimating $\gamma_1(y^{(0)})$. Users typically first estimate $\gamma_1(y^{(0)})$ and then use these estimators to estimate γ_0. For the time being, we are temporarily adding the assumption that users can simulate unbiased estimators $\hat{\gamma}_1(y^{(0)})$ of $\gamma_1(y^{(0)})$ for every fixed $y^{(0)}$ with finite computational cost. This assumption will be removed in Section 2.2. It easily holds when $D = 1$, as users can repeatedly simulate $y_i^{(1)} \sim \pi_1(\cdot \mid y^{(0)})$ and it follows from the problem definition that each $g_1(y^{(0)}, y_i^{(1)})$ is unbiased for $\gamma_1(y^{(0)})$. In the general case of $D > 1$, this assumption is far from trivial, as $\gamma_1(y^{(0)})$ is itself a nested expectation with a nesting depth of $D-1$. Nevertheless, as we will see in Section 2.2, this assumption helps us to capture and reduce the intrinsic difficulty of the problem and, therefore, will guide us to design the general algorithm.

With this extra assumption, constructing unbiased estimators of (1) is equivalent to constructing unbiased estimators of $g_0(y^{(0)}, \gamma_1(y^{(0)}))$. Even with access to unbiased estimators of $\gamma_1(y^{(0)})$, the intuitive plug-in estimator $g_0\big(y^{(0)}, \hat{\gamma}_1(y^{(0)})\big)$ is still biased, as in general $\mathbf{E}[g_0\big(y^{(0)}, \hat{\gamma}_1(y^{(0)})\big) \mid y^{(0)}] \neq g_0(y^{(0)}, \mathbf{E}[\hat{\gamma}_1(y^{(0)}) \mid y^{(0)}])$. To eliminate this bias, we use the rMLMC method (Blanchet & Glynn, 2015), which is briefly reviewed below.

The rMLMC method uses the Law of Large Numbers (LLN) and rewrites g_0 as the following telescoping summation.

$$\begin{aligned} g_0(y^{(0)}, \gamma_1(y^{(0)})) &= \mathbf{E}\left[g_0\left(y^{(0)}, \lim_{k\to\infty} \frac{S_k}{k}\right) \middle| y^{(0)}\right] \\ &= \sum_{n=1}^{\infty} \mathbf{E}\left[g_0\left(y^{(0)}, \frac{S_{2^n}}{2^n}\right) \middle| y^{(0)}\right] \\ &\qquad - \mathbf{E}\left[g_0\left(y^{(0)}, \frac{S_{2^{n-1}}}{2^{n-1}}\right) \middle| y^{(0)}\right], \end{aligned}$$

where $S_k = \sum_{i=1}^k \hat{\gamma}_{1,i}(y^{(0)})$ is the summation of i.i.d. copies of $\hat{\gamma}_1(y^{(0)})$. To unbiasedly estimate the infinite sum, the rMLMC algorithm first samples $y^{(0)} \sim \pi_0$, then samples a random $N \sim \mathsf{Geo}(r)$, finally generates 2^N unbiased estimators $\{\hat{\gamma}_{1,i}(y^{(0)})\}_{i=1}^{2^N}$ of $\gamma_1(y^{(0)})$ and estimates γ_0 by $R_0 \coloneqq \Delta_N / p_r(N)$, where Δ_n is defined as:

$$\begin{aligned} \Delta_n \coloneqq g_0\left(y^{(0)}, \frac{S_{2^n}}{2^n}\right) - \frac{1}{2}\Bigg[& g_0\left(y^{(0)}, \frac{S^{\mathsf{E}}_{2^{n-1}}}{2^{n-1}}\right) \\ & + g_0\left(y^{(0)}, \frac{S^{\mathsf{O}}_{2^{n-1}}}{2^{n-1}}\right)\Bigg] \end{aligned}$$

for $n \geq 1$ and $\Delta_0 \coloneqq g_0(y^{(0)}, \hat{\gamma}_{1,1}(y^{(0)}))$.

The next theorem justifies the theoretical properties of R_0:

Theorem 2.1. *With all the notations as above, suppose $g_0 : \mathbf{R}^{M+1} \to \mathbf{R}$ satisfies LBS condition defined in* (4)*, and $\|\hat{\gamma}_1(y^{(0)})\|_{\pi,m} < \infty$ for some $m \geq 4$. Then for any $r \in (1/2, 3/4)$, the estimator $R_0 \coloneqq \Delta_N / p_r(N)$ has expectation γ_0, finite variance, and finite expected computational cost.*

Theorem 2.1 will be proved as a special case of our Theorem 2.2. For now, we use the following heuristic calculation to justify the unbiasedness of $\hat{\gamma}_0$:

$$\begin{aligned} \mathbf{E}[R_0 \mid y^{(0)}] &= \sum_{n=0}^{\infty} \mathbf{E}\left[\frac{\Delta_n}{p_r(n)} p_r(n) \middle| y^{(0)}\right] \\ &= \sum_{n=0}^{\infty} \mathbf{E}\left[g_0\left(y^{(0)}, \frac{S_{2^n}}{2^n}\right) - g_0\left(y^{(0)}, \frac{S_{2^{n-1}}}{2^{n-1}}\right) \middle| y^{(0)}\right] \\ &= g_0(y^{(0)}, \gamma_1(y^{(0)})). \end{aligned}$$

Therefore $\mathbf{E}[R_0] = \mathbf{E}[g_0(y^{(0)}, \gamma_1(y^{(0)}))] = \gamma_0$ by (1). More technical discussions such as the range of r, other possible regularity conditions on g_0, and the moment guarantees of γ_0 will all be deferred after Theorem 2.2.

2.2. Recursive rMLMC algorithm for general D

Theorem 2.1 is useful to solve our original problem (without extra assumptions) in two ways. First, Theorem 2.1 already solves the case where $D = 1$, as our extra assumption

automatically holds. It states that if g_0 has a bounded second derivative on its last component, and $g_1(y^{(0)}, y^{(1)})$ has at least finite fourth moment under π, then R_0 is unbiased, has finite variance, and finite expected computational cost. More importantly, Theorem 2.1 tells us that the original problem of estimating an RNE with a depth of D can be solved if we can unbiasedly estimate $\gamma_1(y^{(0)})$ for fixed $y^{(0)}$, which is another RNE with a depth of $D-1$. Therefore, we have successfully reduced the number of nestings by one. This observation motivates us to come up with an algorithm for the general D case, as explained below.

We first go one step further to illustrate the $D = 2$ case. When $D = 2$, estimating $\gamma_1(y^{(0)})$ reduces to the case we have analyzed in Section 2.1. To be precise, since $g_2(y^{(0:2)})$ is unbiased for $\gamma_2(y^{(0:1)})$ if $y^{(2)} \sim \pi_2(\cdot \mid y^{(0:1)})$, one can first sample $y^{(1)} \sim \pi_1(\cdot \mid y^{(0)})$, then simulate $N \sim \mathsf{Geo}(r)$ and 2^N samples $\{y_i^{(2)}\}_{i=1}^{2^N}$ from $\pi_2(\cdot \mid y^{(0:1)})$. Let $\hat{\gamma}_{2,i}(y^{(0:1)}) := g_2(y^{(0:1)}, y_i^{(2)})$, our estimator of $\gamma_1(y^{(0)})$ is then constructed in the same way as Section 2.1, i.e., $R_1(y^{(0)}) := \Delta_N / p_r(N)$ with

$$\Delta_n := g_1\left(y^{(0:1)}, \frac{S_{2^n}}{2^n}\right) - \frac{1}{2}\left[g_1\left(y^{(0:1)}, \frac{S^{\mathsf{E}}_{2^{n-1}}}{2^{n-1}}\right) + g_1\left(y^{(0:1)}, \frac{S^{\mathsf{O}}_{2^{n-1}}}{2^{n-1}}\right)\right],$$

where $S_{2^n}, S^{\mathsf{E}}_{2^{n-1}}, S^{\mathsf{O}}_{2^{n-1}}$ are the summation of every, even, and odd terms of $\{\hat{\gamma}_{2,i}(y^{(0:1)})\}$, respectively. The same procedure of simulating $R_1(y^{(0)})$ can be repeated independently. Therefore we can sample another geometrically distributed random variable $N' \sim \mathsf{Geo}(r')$, and generate $R_{1,i}(y^{(0)}) := \Delta_{N'}/p_r(N')$ independently. Since each $R_{1,i}(y^{(0)})$ is unbiased for $\gamma_1(y^{(0)})$, one can again use the method described in Section 2.1 to form our final estimator for γ_0. After checking $R_1(y^{(0)})$ satisfies the finite fourth-moment assumption, Theorem 2.1 can be applied which implies our estimator is unbiased, has finite variance and finite cost (for the $D = 2$ case).

The general case works in the same way. A key observation is that, due to the nested structure of the problem, Theorem 2.1 not only states that an unbiased estimator of γ_0 can be constructed if one can unbiasedly estimate $\gamma_1(y^{(0)})$ for every $y^{(0)}$, but also directly implies that an unbiased estimator of $\gamma_d(y^{(0:d-1)})$ can be constructed if one can unbiasedly estimate $\gamma_{d+1}(y^{(0:d)})$ for every $y^{(0:d)}$. Therefore, we can estimate γ_0 in a backward, inductive manner.

To begin, we consider the deepest depth of the problem, fixing any $y^{(0:D-1)}$. An unbiased estimator of $\gamma_D(y^{(0:D-1)})$ can be directly constructed as $g_D(y^{(0:D-1)}, y^{(D)})$, where $y^{(D)} \sim \pi_D(\cdot \mid y^{(0:D-1)})$. For $0 \le d \le D-1$, if we assume that users can generate unbiased estimators of $\gamma_{d+1}(y^{(0:d)})$ for every $y^{(0:d)}$, then we can obtain an unbiased estimator of $\gamma_d(y^{(0:d-1)})$ by sampling one $y^{(d)}$, generating $N_d \sim \mathsf{Geo}(r_d)$ and 2^{N_d} unbiased estimators of $\gamma_{d+1}(y^{(0:d)})$, and applying the method described in Section 2.1. This process continues until we reach $d = 0$, at which point we have an unbiased estimator of γ_0. The parameters $(r_0, r_1, \ldots, r_{D-1})$ will be carefully chosen and depend on the regularity assumptions of $(g_0, g_1, \ldots, g_{D-1})$. These choices will be discussed in more detail later.

Our algorithm is described in Algorithm 1. It is written as a recursive algorithm, though it could also be equivalently written in an iterative form with much more cumbersome notations. Algorithm 1 takes a depth index, a trajectory, a simulator, and parameters for the geometric distribution as inputs, and outputs an unbiased estimator of $\gamma_d(H)$. In particular, with inputs {depth = 0, trajectory = $\varnothing$, parameters = $(r_0, r_1, \ldots, r_{D-1})$}, it outputs READ – an unbiased estimator of the RNE defined in (1). The logic of Algorithm 1 is precisely the same as we just discussed. To estimate $\gamma_d(y^{(0:d-1)})$, the algorithm first checks the value of d. When $d = D$, the problem becomes straightforward. When $d < D$, the algorithm samples $y^{(d)}$, appends $y^{(d)}$ to the trajectory, samples N_d, and calls itself 2^{N_d} times with depth $d+1$ and new trajectory $\{y^{(0:d)}\}$ to get 2^{N_d} unbiased estimators of $\gamma_{d+1}(y^{(0:d)})$. Finally, we split these 2^{N_d} estimators into even and odd terms and apply the method described in Section 2.1. The algorithm is guaranteed to stop, as the depth will eventually reach the deepest depth D.

2.3. Theoretical guarantees

We now discuss the computational costs of Algorithm 1 and the statistical properties of READ. Our theoretical results depend on the smoothness conditions of $\{g_d\}_{d=0}^{D-1}$, so we will examine the LBS and LBL cases separately.

2.3.1. The LBS case

The following theorem shows, under the LBS assumption, the computational cost and the variance of READ can be controlled simultaneously.

Theorem 2.2. *Suppose for every $d \in \{0, 1, \ldots, D-1\}$, the function g_d satisfies the LBS assumption* (4)*, $r_d := 1 - 2^{-k_d}$ satisfies $k_d \in \left(1, \frac{2^{d+1}}{2^{d+1}-1}\right)$, and $\|g_D(y^{(0:D)})\|_{\pi, 2^{D+1}} < \infty$. Then for every $0 \le d \le D$, the output $R_d(y^{(0:d-1)})$ of Algorithm 1 with inputs {depth = d, trajectory = $y^{(0:d-1)}$, $\mathcal{S}$, parameters $(r_d, \ldots, r_{D-1})$} satisfies:*

- *For π-almost every fixed $y^{(0:d-1)}$,*

$$\mathbf{E}\left[R_d(y^{(0:d-1)}) \mid y^{(0:d-1)}\right] = \gamma_d(y^{(0:d-1)}).$$

Algorithm 1 A recursive rMLMC algorithm for RNEs

Input: Depth index $d \in \{0, ..., D\}$. Trajectory history $H = \{y^0, ..., y^{d-1}\}$ or $\varnothing$. A simulator $\mathcal{S}$. Parameters $r_d, ..., r_{D-1}$ determined by conditions on $\{g_d\}_{d=0}^{D-1}$ (see Theorem 2.2, 2.4).
Output: An unbiased estimator of $\gamma_d(H)$
if $d = D$ **then**
  Sample one $y^{(D)} \sim \pi_D(\cdot \mid H)$;
  Return: $R_D \coloneqq g_D\left(y^{(0:D)}\right)$.
else
  Sample $y^{(d)} \sim \pi_d(\cdot \mid H)$;
  Update the trajectory $H \leftarrow H \cup \{y^{(d)}\}$;
  Sample $N_d \sim \mathsf{Geo}(r_d)$;
  Call Algorithm 1 for 2^{N_d} times with inputs $\{d+1; H; \mathcal{S}; r_{d+1}, ..., r_{D-1}\}$, and label the observations as $R_{d+1}(y^{(0:d)})(1), ..., R_{d+1}(y^{(0:d)})\left(2^{N_d}\right)$;
  Calculate $S_{2^{N_d}}, S^{\mathsf{E}}_{2^{N_d-1}}, S^{\mathsf{O}}_{2^{N_d-1}}$ defined in Section 1.2;
  Calculate $\left(\text{note } \Delta_0 \coloneqq g_d\left(y^{(0:d)}, R_{d+1}(y^{(0:d)})(1)\right)\right)$:

$$\Delta_{N_d} = g_d\left(y^{(0:d)}, \frac{S_{2^{N_d}}}{2^{N_d}}\right) - \frac{1}{2}\left[g_d\left(y^{(0:d)}, \frac{S^{\mathsf{O}}_{2^{N_d-1}}}{2^{N_d-1}}\right) + g_d\left(y^{(0:d)}, \frac{S^{\mathsf{E}}_{2^{N_d-1}}}{2^{N_d-1}}\right)\right];$$

  Return: $R_d \coloneqq \Delta_{N_d}/p_{r_d}(N_d)$.
end if

- *The expected computational cost of R_d equals*

$$\prod_{k=d}^{D-1} \frac{r_k}{2r_k - 1} < \infty.$$

- *The output has finite 2^{d+1}-th moment, i.e.,*

$$\mathbf{E}_\pi\left[|R_d(y^{(0:d-1)})|^{2^{d+1}}\right] < \infty \text{ for } 0 \le d \le D.$$

Theorem 2.2 states for π-almost every $y^{(0:d-1)}$, the expectation of the output R_d conditioning on the input is unbiased for $\gamma_d(y^{(0:d-1)})$. The computational cost has a finite expectation, and the output has a finite 2^{d+1}-th moment [1]. The detailed proof of Theorem 2.2 will be provided in the Appendix. Here, we highlight two special cases. First, Theorem 2.2 shows that READ, the output R_0 of Algorithm 1 when given input {depth = 0, trajectory = $\varnothing$, $\mathcal{S}$, parameters = $(r_0, \ldots, r_{D-1})$}, has the desired properties. Specifically, it is an unbiased estimator for γ_0 with finite expected computational cost and finite variance. Second, Theorem 2.2 includes Theorem 2.1 as a special case when $D = 1$.

Let $R_{0,1}, R_{0,2}, \ldots$ be the i.i.d. outcomes by repeatedly implementing Algorithm 1. Since each $R_{0,i}$ is unbiased and has a finite variance, the standard Central Limit Theorem (CLT) implies that $\sqrt{n}(\sum_{i=1}^n R_{0,i}/n - \gamma_0) \to \mathbf{N}(0,1)$ in distribution. This means that the estimator $\sum_{i=1}^n R_{0,i}/n$ converges to γ_0 at a rate of $n^{-1/2}$ in rMSE (or equivalently, n^{-1} in MSE), which compares quite favorably with the rates obtained by NMC estimators in (Rainforth, 2018). This rate is optimal in the sense that it matches the minimax lower bound over all Monte Carlo methods (Theorem 2.1 of (Heinrich & Sindambiwe, 1999)). The next corollary shows that, by repeatedly implementing Algorithm 1, users can easily obtain an unbiased estimator for γ_0 with at most ε-rMSE within $O(\varepsilon^{-2})$ computational cost.

Corollary 2.3. *With all the assumptions the same as Theorem 2.2, for any $\varepsilon > 0$, we can construct an estimator R with expected computational cost $\mathcal{O}(\varepsilon^{-2})$ such that the rMSE $\sqrt{\mathbf{E}[(R-\gamma_0)^2]}$ is at most ϵ.*

Proof of Corollary 2.3. Calling Algorithm 1 independently for n times with {depth = 0, trajectory = $\varnothing$, $\mathcal{S}$, parameters = $(r_0, \ldots, r_{D-1})$} yield i.i.d. unbiased estimators $R_{0,1}, ..., R_{0,n}$ for γ_0. Let our estimator be $R \coloneqq \frac{1}{n}\sum_{i=1}^n R_{0,i}$. Then,

$$\mathbf{E}[(R-\gamma_0)^2] = \mathbf{E}\left[\left(\frac{1}{n}\sum_{i=1}^n R_{0,i} - \gamma_0\right)^2\right] = \frac{1}{n}\mathrm{Var}(R_0).$$

Thus noting that $\mathrm{Var}(R_0) < \infty$ by Theorem 2.2, taking $n = \mathrm{Var}(R_0)/\varepsilon^2$ samples ensures R has up to ε-rMSE. Finally, let $C \coloneqq C(D) < \infty$ be the expected computational cost for one call of Algorithm 1 . The expected computational cost for constructing R is then $C \cdot \mathrm{Var}(R_0)/\varepsilon^2 = \Theta(\varepsilon^{-2})$. □

We add two additional remarks regarding the above corollary. Firstly, while the above result demonstrates that our algorithm achieves optimal dependency on ϵ, it is important to highlight that we are operating within the context of the 'fixed D' regime, where the constant in our $\mathcal{O}$ notation depends on D. In fact, it is clear from Theorem 2.2 that each invocation of Algorithm 1 has a cost of at least $\Omega((1+\omega)^D)$ for some $\omega > 0$, indicating that our algorithm does not scale well with increasing nesting levels. Nevertheless, our algorithm remains practically relevant in scenarios where D is small or moderate, including the examples discussed in Section 1. Secondly, the ε-rMSE of R can be easily translated to other performance metrics via standard inequalities. For example, for any δ, Markov's inequality implies the absolute error $|R - \gamma_0|$ is less than $\varepsilon/\sqrt{\delta}$ with probability at least $1 - \delta$.

[1] Readers should notice that the expectation of $R_d(y^{(0:d-1)})$ is calculated under the conditional distribution $\pi_{d:D}(\cdot \mid y^{(0:d-1)})$. The computational cost and the 2^{d+1}-th moment are calculated under the joint distribution π. When the input depth $= 0$, these two underlying distributions coincide.

Next, we discuss the assumptions and proof strategies of Theorem 2.2. We require the first D functions $\{g_d\}_{d=0}^{D-1}$ all satisfy the LBS condition, and the final function g_D has finite 2^{D+1}-th moment under π. The LBS assumption also appears in the work of NMC estimators (see the second part of Theorem 3 in (Rainforth et al., 2018)). The moment assumption of g_D is not required in (Rainforth et al., 2018). Nevertheless, it is a mild assumption that holds in most practical applications. It covers all the cases where g_D is bounded or has a moment generating function (including the uniform, Gaussian, Poisson, or exponential distributions), which implies $\mathbf{E}[|g_D|^k] < \infty$ for every k. As we will see in our proofs, these assumptions help us to establish the moment guarantee of Theorem 2.2 in a backward inductive way. For example, the 2^{D+1}-th moment assumption on g_D and the LBS assumption on g_{D-1} implies R_{D-1} has finite 2^D-th moment. More generally, the finiteness of the 2^{d+1}-th moment of R_d follows from the LBS assumption on g_d and the 2^{d+2}-th moment of R_{d+1} (which is the conclusion of the previous inductive step). Eventually, we conclude R_0 has a finite variance. Finally, we want to emphasize our moment assumption on g_D is not 'trajectory-dependent'. We require g_D has finite 2^{D+1}-th moment under the joint distribution π of $y^{(0:D)}$, which is much weaker than g_D has a uniformly bounded finite 2^{D+1}-th moment under $\pi_D(\cdot \mid y^{(0:D-1)})$ for every fixed trajectory $y^{(0:D-1)}$.

Finally, the parameters $\{r_d\}_{d=0}^{D=1}$ reflect the trade-off between the variance and computation cost. Since 2^{N_d} calls are required for each d, standard calculation shows that $\mathbf{E}[2^{N_d}] = r_d/(2r_d - 1)$ when $r_d > 0.5$, and $+\infty$ if $r_d \leq 0.5$. Therefore, every r_d has to be strictly greater than 0.5 to ensure a finite expected computational cost. Meanwhile, we cannot guarantee finite variance or unbiasedness of READ when r_d becomes too large. Our range for r_d in Theorem 2.2 follows from a careful calculation in our proof to ensure unbiasedness, finite computational cost, and variance simultaneously.

2.3.2. The LBL case

The assumptions in Theorem 2.2 guarantee that READ enjoys an optimal convergence rate and computational cost. However, the second-order derivative assumption also rules out many functions of practical interest, such as $\max$ and $\min$. In this section, we study the theoretical properties of Algorithm 1 and READ under weaker smoothness and moment assumptions. Our result is summarized below:

Theorem 2.4. *Fix any $0 < \delta < 1/2$. Suppose for every $d \in \{0, 1, \ldots, D-1\}$, the function g_d satisfies the LBL assumption defined in* (5)*, and $r_d := 1 - 2^{-k_d}$ satisfies*

$$k_d \in \left(1, \left(\frac{2^{d+2} - 3\delta}{2^{d+3} - 3\delta}\right)\left(\frac{2^{d+1} - \delta}{2^d - \delta}\right)\right).$$

Moreover, suppose $\|g_D(y^{(0:D)})\|_{\pi,2} < \infty$. Then for every $0 \leq d \leq D$, the output $R_d(y^{(0:d-1)})$ of Algorithm 1 with inputs $\{$depth = d, trajectory = $y^{(0:d-1)}$, $\mathcal{S}$, parameters $(r_d, \ldots, r_{D-1})\}$ has the following properties:

- *For π-almost every fixed $y^{(0:d-1)}$,*
$$\mathbf{E}\left[R_d(y^{(0:d-1)}) \mid y^{(0:d-1)}\right] = \gamma_d(y^{(0:d-1)}).$$
- *The expected computational cost of R_d equals*
$$\prod_{k=d}^{D-1} \frac{r_k}{2r_k - 1} < \infty.$$
- *The output has finite $(2 - \delta/2^d)$-th moment, i.e.,*
$$\mathbf{E}_\pi\left[|R_d(y^{(0:d-1)})|^{(2-\delta/2^d)}\right] < \infty \text{ for } 0 \leq d \leq D.$$

Comparing Theorem 2.2, which requires the LBS assumption for $\{g_d\}_{d=0}^{D-1}$ and finite 2^{D+1}-th moment for g_D, with Theorem 2.4, which only requires the LBL assumption for $\{g_d\}_{d=1}^{D-1}$ and finite second moment for g_D, it is clear that Theorem 2.4 has more general assumptions. However, it does not guarantee that READ has a finite variance. Nevertheless, it remains unbiased and has a finite expected computational cost. To minimize the loss of moment guarantees, one can choose suitable parameters such that READ has finite $(2 - \delta)$-th moment for any small δ.

Again, let $R_{0,1}, R_{0,2}, \ldots,$ be the i.i.d. outcomes by repeatedly implementing Algorithm 1. There are more technical challenges when analyzing the convergence rate of $\sum_{i=1}^n R_{0,i}/n - \gamma_0$, as the CLT cannot be applied. Instead, we use the Marcinkiewicz-Zygmund generalized law of large numbers (see Theorem A.4 in Appendix A), which shows $n^{-1}\mathbf{E}[|\sum_{i=1}^n X_i|^p] \to 0$ if $\{X_i\}_{i=1}^n$ are i.i.d., centered random variables with finite p-th moment for $p \in [1, 2)$. Our result is the following:

Corollary 2.5. *With all the assumptions the same as Theorem 2.4, let $R_{0,1}, R_{0,2}, \ldots,$ be the i.i.d. outcomes by repeatedly implementing Algorithm 1, we have:*

- $\mathbf{E}[|\sum_{i=1}^n R_{0,i}/n - \gamma_0|] = o(n^{-1/(2(1+\delta))})$.
- *We can construct an estimator R with expected computational cost $\mathcal{O}(\varepsilon^{-2(1+\delta)})$ such that the mean absolute error $\mathbf{E}[|R - \gamma_0|] < \varepsilon$.*

Proof of Corollary 2.5. Applying Theorem A.4 with $p = 2 - \delta$, $X_i = R_{0,i} - \gamma_0$ and Jensen's inequality, we have:

$$\mathbf{E}\left[\left|\sum_{i=1}^n R_{0,i}/n - \gamma_0\right|\right] = n^{-1}\mathbf{E}\left[\left|\sum_{i=1}^n X_i\right|\right]$$
$$\leq n^{-1}\left(\mathbf{E}\left[\left|\sum_{i=1}^n X_i\right|^p\right]\right)^{1/p} = o(n^{-1+\frac{1}{p}}) = o(n^{\frac{-1}{2(1+\delta)}}),$$

which proves the first part. The last step follows from $1 - 1/(2-\delta) > 1/(2+2\delta)$ for $\delta \in (0, 1/2)$. Setting $n = \Omega(\varepsilon^{-2(1+\delta)})$ and the second part immediately follows. $\square$

Although we are not able to recover the optimal $n^{-1/2}$ convergence rate under this more general regime, our convergence rate is still near-optimal as it can be as close to $n^{-1/2}$ as we want. Still, the convergence rate does not depend on D, and, although we replace the MSE by MAE due to the moment constraint, one can still use Markov's inequality to show the absolute error $|R - \gamma_0|$ is less than ε/δ with probability at least $1 - \delta$.

As the $\max$ function satisfies the LBL assumption, our results here include the optimal stopping problem as a special case. Our results complement the work of (Zhou et al., 2022), where the authors use rMLMC to design an estimator with $\mathcal{O}(\varepsilon^{-2})$ computational cost under stronger assumptions (see their Assumption 4). We have a slightly worse cost of $\mathcal{O}(\varepsilon^{-2(1+\delta)})$ but under more general assumptions.

3. Numerical experiments

We test our algorithm on three examples. Some additional statistics and an extra experiment are provided in Appendix E and F. Our code is available at `https://github.com/guanyangwang/rMLMC_RNE`.

3.1. A toy example

We consider the following simple example with known ground-truth. Suppose the process $(y^{(0)}, y^{(1)}, y^{(2)})$ satisfies $y^{(0)} \sim \mathbf{N}(\pi/2, 1), y^{(1)} \sim \mathbf{N}(y^{(0)}, 1), y^{(2)} \sim \mathbf{N}(y^{(1)}, 1)$. Define $g_0(y^{(0)}, z) := \sin\big(y^{(0)} + z\big), g_1(y^{(0:1)}, z) := \sin\big(y^{(1)} - z\big)$, and $g_2(y^{(0:2)}) := y^{(2)}$. The target quantity γ_0 defined (1) is a nested expectation with $D = 2$. One can use the formula $\mathbf{E}_{Z\sim\mathbf{N}(\mu,\sigma^2)}[\sin(Z)] = \sin(\mu)\exp\big(-\sigma^2/2\big)$ to analytically calculate $\gamma_0 = \exp(-1/2) \approx 0.6065$. Now we compare our READ estimator with the NMC estimators in (Rainforth et al., 2018).

For the NMC estimator, users first specify N_0, N_1, N_2. Then we sample N_0 copies of $y^{(0)}$, N_1 copies of $y^{(1)}$ for each fixed $y^{(0)}$, and N_2 copies of $y^{(2)}$ for each fixed $y^{(0:1)}$, and use these samples to form the NMC estimator, details are explained in Appendix D. Following (Rainforth, 2018), we consider two ways of allocating (N_0, N_1, N_2). The first estimator NMC1 is to choose $N_0 = N_1 = N_2$, the second NMC2 is to choose $N_0 = N_1^2 = N_2^2$. For READ, since all assumptions in Theorem 2.2 are satisfied, therefore when $r_0 \in (1/2, 3/4)$ and $r_1 \in (1/2, 1 - 2^{-4/3})$, the READ estimator generated by Algorithm 1 is unbiased and of finite variance. Since the computational cost gets lower when each r_i gets larger, we choose $r_0 = 0.74$ and $r_1 = 0.6$ (close to the upper-end of their respective ranges above) to facilitate the computational efficiency. Therefore, implementing Algorithm 1 once has an expected sample size/computational cost $(r_1/(2r_1 - 1))\,(r_2/(2r_2 - 1)) \approx 4.625$.

(a) (b)

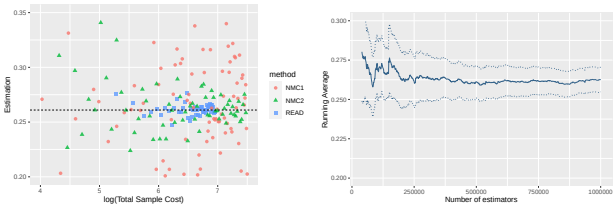
Figure 1. (a): The comparison on the empirical MSEs of estimating the RNE among READ (blue), NMC1 (red), and NMC2 (green). All the logarithms are of the base 10. Each method's empirical errors are calculated based on 20 independent repetitions. (b) The trace plot (solid blue curve) of the running averages of READ. The blue dotted curves are the 95% confidence intervals. The red dashed line is the ground truth $\exp(-1/2)$.

Our comparison result is summarized in Figure 1. Since the NMC methods and READ have different ways of generating estimators. To make a fair comparison, we compare the estimation errors with the total sample cost used by these three estimators. For NMC1 and NMC2, the total sample cost is $n = N_0 N_1 N_2$. For READ, the total sample cost is random, therefore we use its expected value, which equals $4.625\times$ Number of repetitions of Algorithm 1. The slopes of the blue, red, and green lines, which correspond to the empirical convergence rate of READ, NMC1, NMC2, equals $-0.97, -0.35, -0.47$, respectively. They match well with the theoretical predictions n^{-1} in Corollary 2.3 for READ, $n^{-1/3}$ for NMC1, and $n^{-1/2}$ for NMC2 in (Rainforth et al., 2018). It is clear from Figure 1(a) that READ has a significant advantage over NMC estimators, with both faster convergence rate and orders of magnitude lower errors.

We also call Algorithm 1 for 10^6 times and plot the running averages of our estimates in Figure 1(b). Our estimator becomes more accurate when we increase the number of repetitions. For each $k \in (1, 2, \ldots, 10^6)$, we also calculate the standard deviation (SD) of the first k repetitions and use Mean ± 1.96 SD to form the 95% confidence interval. It is also clear from Figure 1(b) that our confidence intervals always include the ground-truth, suggesting the high accuracy of our method. In contrast, constructing confidence intervals of NMC estimators are much more time-consuming.

3.2. Example with heavy-tail underlying distribution

All three estimators are also evaluated on the same set of functions using an independent, non-central t-distribution with 10 degrees of freedom and a noncentrality parameter of 0.5. The outcomes of these tests are illustrated in Figure



(a) (b)

Figure 2. (a): Scatterplot of the estimation of γ_0. Blue, red, green points correspond to READ, NMC1, NMC2 estimators respectively. (b) Trace plot (solid blue curve) of the running averages of READ. The blue dotted curves are the 95% confidence intervals.

2. Despite the fact that the t-distribution exhibits a significantly heavier tail compared to the Gaussian distribution, it is evident from Figure 2(a) that the convergence rate, as indicated by the speed at which each color converges to the black dotted line, is considerably faster for the READ method compared to the NMC estimators.

3.3. Pricing the Bermudan Options

Finally, we utilize our method to price high-dimensional Bermudan basket put options. Given that option pricing can be formulated as an optimal stopping problem, our estimator simplifies to the MUSE estimator in (Zhou et al., 2022). The underlying process $y^{(0:D)} = (S_0, S_{T/D}, S_{2T/D}, \ldots, S_T)$ where S_t is a M-dimensional geometric Brownion motion, each coordinate follows $dS_i(t) = (r-\delta)S_i(t)dt + \sigma S_i(t)dW_i(t)$. For $d \leq D-1$, our $g_d(y^{(0:d)}, z) := \max\{e^{-rT/D}, z\}$, where $U(x) := \max\{(K - \overline{x}), 0\}$. For $d = D$, $g_D(y^{(0:D)}) := U(y^{(D)})$. We also adopt the standard parameters in (Jain & Oosterlee, 2012; Bender et al., 2006; Zhou et al., 2022): $T = 3, M = 5, \sigma = 0.2, r = 0.05, K = y_i^{(0)} = 100$ for every i.

We follow previous works and set $D = 3$. We tested READ on a 500-core cluster, with each computer generating 10^4 estimators. Our estimator is obtained by averaging all the 5 million estimators. We also tested NMC1 and NMC2 on the same cluster. Each computer generates 10 estimators for both methods. NMC1 uses $N_0 = N_1 = \cdots = 80$, NMC2 uses $N_0 = N_1^2 = .. = 900$. The results are summarized in Table 1. After comparing with existing algorithms tailored for option pricing/optimal stopping, we observe that READ aligns closely with the results of previous works. In contrast, both NMC1 and NMC2 exhibit a significant overestimation of the target. This discrepancy arises from the convex nature of the max function, which introduces a systematic bias in the NMC estimators. Therefore, our method provides more reliable estimates with a much shorter completion time. Similarly, we test the case where $D = 4$. READ generates 5 million estimators over 500 processors. NMC1 generates 5000 estimators with $N_0 = 25$, NMC2 generates 5000 estimators with $N_0 = 225$. The outcomes of these tests are also presented in Table 1. Again, both NMC estimators overestimate the target, albeit a smaller standard error.

$D = 3$	READ	NMC1	NMC2
Cost	6.9×10^7	2.05×10^{11}	1.22×10^{11}
Time/s	$(14.6, 22.6, 87.3)$	$(116.1, 169.1, 215.9)$	$(316.3, 350.7, 377.8)$
Estimate (se)	$2.159(0.008)$	$2.169(0.005)$	$2.180(0.014)$
$D = 4$	READ	NMC1	NMC2
Cost	2.36×10^8	4.88×10^{10}	5.69×10^{10}
Time/s	$(19.8, 44.5, 299.8)$	$(66.6, 118.2, 189.1)$	$(316.3, 350.7, 377.8)$
Estimate (se)	$2.284(0.065)$	$2.357(0.008)$	$2.393(0.003)$

Table 1. Summary of results when $D = 3$. The three values in the "Time" row correspond to the minimum, average, and maximum completion times across 500 processors.

4. Further discussions

Here we provide some remarks for practical implementation and discuss some potential generalizations. The users need to specify the parameters $\{r_d\}_{d=0}^{D-1}$ when implementing Algorithm 1. Larger values of r_i lead to a shorter time for each implementation but potentially larger variance. When some r_i is not chosen according to Theorem 2.2 or 2.4, the algorithm can still be implemented, but the variance may be infinite. The trade-off between the values of $\{r_d\}$ and the fluctuations of the resulting estimator is problem-specific. In practice, knowing how many repetitions are sufficient is important to provide an accurate estimator. One possible way is to bound certain moments of READ and use Corollary 2.3 or 2.5 to choose a sufficiently large n. But this bound can be problem-specific and very conservative. Instead, we follow (Glynn & Rhee, 2014) and suggest the following adaptive stopping rule: users first specify a precision-level ε and a small $\delta\%$. When repeatedly implementing Algorithm 1, users calculate the empirical $(1-\delta\%)$ confidence interval $[L_\delta(k), U_\delta(k)]$ for first k repetitions in the same way as Section 3 for every k. Users can stop when the width of the confidence interval is less than 2ε. The validity of this stopping rule is proven in (Glynn & Whitt, 1992).

One potential direction for extension is as follows. Here we only consider the 'fix D' regime and construct estimators with optimality guarantees. However, the cost of Algorithm 1 scales exponentially with D. Therefore, although our algorithm is more efficient than the NMC estimator for every fixed D, both methods are not practical when D becomes too large. Indeed, the poor scaling with D is a common issue in related literature such as (Glasserman & Yu, 2004; Zanger, 2013) and seems unavoidable. An interesting direction would be to construct modifications of Algorithm 1 under extra practical assumptions for large or infinite D. For example, if we know that the 'influence' of γ_d on γ_0 decays exponentially or double-exponentially with d, it is then sufficient to truncate the depth to $\tilde{D} := \log(1/\varepsilon)$ or $\sqrt{\log(1/\varepsilon)}$. We hope to report progress in the future.

References

Beck, C., Jentzen, A., and Kruse, T. Nonlinear Monte Carlo methods with polynomial runtime for high-dimensional iterated nested expectations. *arXiv preprint arXiv:2009.13989*, 2020.

Bender, C., Kolodko, A., and Schoenmakers, J. Policy iteration for american options: overview. 2006.

Biswas, N., Jacob, P. E., and Vanetti, P. Estimating convergence of Markov chains with L-lag couplings. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Blanchet, J. H. and Glynn, P. W. Unbiased Monte Carlo for optimization and functions of expectations via multilevel randomization. *2015 Winter Simulation Conference (WSC)*, pp. 3656–3667, 2015.

Broadie, M., Du, Y., and Moallemi, C. C. Efficient risk estimation via nested sequential simulation. *Management Science*, 57(6):1172–1194, 2011.

Dauchet, J., Bezian, J.-J., Blanco, S., Caliot, C., Charon, J., Coustet, C., El Hafi, M., Eymet, V., Farges, O., Forest, V., et al. Addressing nonlinearities in Monte Carlo. *Scientific reports*, 8(1):1–11, 2018.

Giles, M. B. Multilevel monte carlo path simulation. *Operations research*, 56(3):607–617, 2008.

Giles, M. B. MLMC for nested expectations. In *Contemporary Computational Mathematics-A Celebration of the 80th Birthday of Ian Sloan*, pp. 425–442. Springer, 2018.

Giles, M. B. and Goda, T. Decision-making under uncertainty: using MLMC for efficient estimation of EVPPI. *Statistics and computing*, 29(4):739–751, 2019.

Giles, M. B. and Haji-Ali, A.-L. Multilevel nested simulation for efficient risk estimation. *SIAM/ASA Journal on Uncertainty Quantification*, 7(2):497–525, 2019.

Giles, M. B. and Haji-Ali, A.-L. Multilevel path branching for digital options. *arXiv preprint arXiv:2209.03017*, 2022.

Giles, M. B., Haji-Ali, A.-L., and Spence, J. Efficient risk estimation for the credit valuation adjustment. *arXiv preprint arXiv:2301.05886*, 2023.

Glasserman, P. and Yu, B. Number of paths versus number of basis functions in American option pricing. *The Annals of Applied Probability*, 14(4):2090–2119, 2004.

Glynn, P. W. and Rhee, C.-h. Exact estimation for Markov chain equilibrium expectations. *Journal of Applied Probability*, 51(A):377–389, 2014.

Glynn, P. W. and Whitt, W. The asymptotic validity of sequential stopping rules for stochastic simulations. *The Annals of Applied Probability*, 2(1):180–198, 1992.

Goda, T., Hironaka, T., Kitade, W., and Foster, A. Unbiased MLMC stochastic gradient-based optimization of bayesian experimental designs. *SIAM Journal on Scientific Computing*, 44(1):A286–A311, 2022.

Gordy, M. B. and Juneja, S. Nested simulation in portfolio risk measurement. *Management Science*, 56(10):1833–1848, 2010.

Gut, A. *Probability: A Graduate Course*. Springer, 2005.

He, Z., Xu, Z., and Wang, X. Unbiased MLMC-based variational bayes for likelihood-free inference. *SIAM Journal on Scientific Computing*, 44(4):A1884–A1910, 2022.

Heinrich, S. Multilevel Monte Carlo methods. In *International Conference on Large-Scale Scientific Computing*, pp. 58–67. Springer, 2001.

Heinrich, S. and Sindambiwe, E. Monte Carlo complexity of parametric integration. *Journal of Complexity*, 15(3): 317–341, 1999.

Hu, Y., Chen, X., and He, N. On the bias-variance-cost tradeoff of stochastic optimization. *Advances in Neural Information Processing Systems*, 34:22119–22131, 2021.

Jacob, P. E., O'Leary, J., and Atchadé, Y. F. Unbiased Markov chain Monte Carlo methods with couplings. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(3):543–600, 2020.

Jain, S. and Oosterlee, C. W. Pricing high-dimensional bermudan options using the stochastic grid method. *International Journal of Computer Mathematics*, 89(9):1186–1211, 2012.

Kahale, N. Unbiased time-average estimators for Markov chains. *arXiv preprint arXiv:2209.09581*, 2022.

McLeish, D. A general method for debiasing a Monte Carlo estimator. *Monte Carlo methods and applications*, 17(4): 301–315, 2011.

Rainforth, T. Nesting probabilistic programs. In *Conference on Uncertainty in Artificial Intelligence*, pp. 249–258, 2018.

Rainforth, T., Cornish, R., Yang, H., Warrington, A., and Wood, F. On nesting Monte Carlo estimators. In *International Conference on Machine Learning*, pp. 4267–4276. PMLR, 2018.

Rhee, C.-H. and Glynn, P. W. Unbiased estimation with square root convergence for SDE models. *Operations Research*, 63(5):1026–1043, 2015.

Vihola, M. Unbiased estimators and multilevel Monte Carlo. *Operations Research*, 66(2):448–462, 2018.

Wang, G. and Wang, T. Unbiased Multilevel Monte Carlo methods for intractable distributions: MLMC meets MCMC. *arXiv preprint arXiv:2204.04808*, 2022.

Wang, G., O'Leary, J., and Jacob, P. Maximal couplings of the metropolis-hastings algorithm. In *International Conference on Artificial Intelligence and Statistics*, pp. 1225–1233. PMLR, 2021.

Zanger, D. Z. Quantitative error estimates for a least-squares Monte Carlo algorithm for American option pricing. *Finance and Stochastics*, 17(3):503–534, 2013.

Zhou, Z., Wang, G., Blanchet, J. H., and Glynn, P. W. Unbiased optimal stopping via the MUSE. *Stochastic Processes and their Applications*, 2022.

A. Auxiliary results

The following theorem is from pg. 150 (Gut, 2005).

Theorem A.1. *Let $p \geq 1$. Suppose that $X_1, X_2, ...X_n$ are independent random variables, with mean 0 and $\mathbf{E}|X_k|^p < \infty$ for all k, and let $S_n := \sum_{i=1}^n X_i$ denote the partial sums. Then there exist constants A_p^*, B_p^* depending only on p such that*

$$A_p^* \mathbf{E}\left(\sum_{k=1}^n X_k^2\right)^{p/2} \leq \mathbf{E}|S_n|^p \leq B_p^* \mathbf{E}\left(\sum_{k=1}^n X_k^2\right)^{p/2}.$$

The following corollary to the above theorem is from pg. 151 (Gut, 2005), Corollary 8.2.

Corollary A.2. *Let $p \geq 1$. Suppose that $X_1, X_2, \ldots$ are i.i.d. random variables, with mean 0 and $\mathbf{E}|X_1|^p < \infty$, and let $S_n := \sum_{i=1}^n X_i$ denote the partial sums. Then there exists a constant B_p depending only on p, such that*

$$\mathbf{E}|S_n|^p \leq \begin{cases} B_p n^{p/2} \mathbf{E}|X_1|^p, & p > 2 \\ B_p n \mathbf{E}|X_1|^p, & 1 \leq p \leq 2. \end{cases}$$

The following lemma is instrumental in the proofs for the theoretical guarantees of our algorithm under both the LBS and LBL assumptions.

Lemma A.3. *Let (Z_1, Z_2) be a 2-stage stochastic process and there exists $p \geq 1$, such that $\mathbf{E}[|Z_2|^p] < \infty$. Conditioning on Z_1, sample $i.i.d.$ $Z_2(1), ..., Z_2(n)$. Then,*

$$\mathbf{E}\left[\left|\frac{1}{n}\sum_{i=1}^n Z_2(i) - \mathbf{E}[Z_2 \mid Z_1]\right|^p\right] \leq \begin{cases} B_p' \frac{\mathbf{E}[|Z_2|^p]}{n^{p/2}} & p > 2 \\ B_p' \frac{\mathbf{E}[|Z_2|^p]}{n^{p-1}} & 1 \leq p \leq 2 \end{cases}$$

Proof. Let $p > 2$. For arbitrary fixed $Z_1 = z_1$, define $\bar{Z}_2(i) := Z_2(i) - \mathbf{E}[Z_2(i) \mid Z_1 = z_1]$, and apply Corollary A.2 on the $i.i.d.$ mean 0 random variables $\bar{Z}_2(i)$ under the probability distribution $\pi(\cdot \mid Z_1 = z_1)$, we have

$$\begin{aligned}
\mathbf{E}\left[\left|\frac{1}{n}\sum_{i=1}^n Z_2(i) - \mathbf{E}[Z_2 \mid Z_1]\right|^p\right] &= \int_\Omega \mathbf{E}\left[\left|\frac{1}{n}\sum_{i=1}^n Z_2(i) - \mathbf{E}[Z_2 \mid Z_1]\right|^p \mid Z_1 = z_1\right] \pi_1(\mathrm{d}z_1) \\
&= \frac{1}{n^p}\mathbf{E}[\left|\sum_{i=1}^n \bar{Z}_2(i)\right|^p \mid Z_1 = z_1]\pi_1(\mathrm{d}z_1) \\
&\leq \frac{B_p}{n^{p/2}}\mathbf{E}[|\bar{Z}_2(1)|^p \mid Z_1 = z_1]\pi_1(\mathrm{d}z_1) \\
&\leq \frac{B_p'}{n^{p/2}}\mathbf{E}[|Z_2(1)|^p \mid Z_1 = z_1]\pi_1(\mathrm{d}z_1) \\
&= \frac{B_p'}{n^{p/2}}\mathbf{E}[|Z_2(1)|^p|].
\end{aligned}$$

The second inequality follows from the inequality $(a+b)^p \leq 2^{p-1}(|a|^p + |b|^p)$ and the monotonicity of a random variable's L^p norm :

$$\mathbf{E}[|X - \mathbf{E}[X]|^p] \leq 2^{p-1}(\mathbf{E}[|X|^p] + |\mathbf{E}[X]|^p) \leq 2^p \mathbf{E}[|X|^p]$$

For the case $1 \leq p \leq 2$, the calculation is identical to the above, except replace the $B_p'/n^{p/2}$ with B_p'/n^{p-1} from Corollary A.2. □

The following theorem is the Marcinkiewicz-Zygmund law of large numbers from pg. 311 (Gut, 2005), which gives us the $\mathcal{O}\left(\varepsilon^{-2(1+\delta)}\right)$ sampling complexity for the LBL case for $0 < \delta < 1/2$.

Theorem A.4. *Suppose that $X_1, X_2, ...$ are i.i.d. random variables., and set $S_n = \sum_{k=1}^n X_k, n \geq 1$. If $\mathbf{E}|X_1|^p < \infty$ and $\mathbf{E}[X_1] = 0$ when $1 \leq p < 2$, then*

$$\mathbf{E}\left|\frac{S_n}{n^{1/p}}\right|^p = \mathbf{E}\frac{|S_n|^p}{n} \overset{n\to\infty}{\longrightarrow} 0.$$

B. Proof of Theorem 2.2

Recall that we are considering an M-dimensional process $y^{(0:D)}$, i.e. $y^{(d)} \in \mathbf{R}^M$ for each $0 \le d \le D$. Explicitly, for $d \in \{0, ..., D-1\}$, $g_d : \mathbf{R}^{(d+1)M+1} \to \mathbf{R}$ and $g_D : \mathbf{R}^{(D+1)M} \to \mathbf{R}$.

Then, we say $\{g_d\}_{d=0}^{D-1}$ satisfy the last-component bounded second derivative condition (LBS) if for $z \in \mathbf{R}$:

$$\sup_{(y^{(0:d)},z)} \left| \partial^2_{(d+1)M+1} g_d(y^{(0:d)}, z) \right| < K_d.$$

Proof. **Case 1:** $d = D$

When $d = D$, Algorithm 1 samples one $y^{(D)} \sim \pi_D$ and outputs $R_D(y^{(0:D-1)}) := g_D(y^{(0:D)})$. We first prove our output $R_D(y^{(0:D-1)})$ has a finite expectation for almost every fixed $y^{(0:D-1)}$, then its expectation equals $\gamma_D(y^{(0:D-1)})$ follows directly from the algorithm design. To show the first point, notice that the expectation of $|R_D(y^{(0:D-1)})|$ given $y^{(0:D-1)}$ equals the conditional expectation $\mathbf{E}[|g_D(y^{(0:D)})| \mid y^{(0:D-1)}]$. Since $\mathbf{E}[|g_D|] < \infty$ by assumption, we have $\mathbf{E}[|g_D(y^{(0:D)})| \mid y^{(0:D-1)}] < \infty$, almost surely. Therefore Algorithm 1 is unbiased when $d = D$ for almost every input $y^{(0:D-1)}$. Furthermore, it has computational cost 1, and the output has finite 2^{D+1}-st moment.

Case 2: $0 \le d \le D-1$

Now that our base case is proven, we proceed via backwards induction. Suppose unbiasedness, finite 2^{d+2}-th moment, and finite expected computational cost are all satisfied for $d+1$ where $0 \le d \le D-1$. Conditioning on $y^{(0:d-1)}$, we sample $y^{(d)} \sim \pi_d$ and $N_d \sim \mathsf{Geo}(r_d)$. Algorithm 1 will call itself independently for 2^{N_d} times, each with input {Depth index: $d+1$, Trajectory History: $H = y^{(0:d)}$, Parameters: $r_{d+1}, \cdots, r_{D-1}$}. This gives us i.i.d. samples $R_{d+1}(y^{(0:d)})(1), ..., R_{d+1}(y^{(0:d)})(2^{N_d})$ which are used to compute the following:

$$\begin{aligned}
S_{2^{N_d}} &= R_{d+1}(y^{(0:d)})(1) + R_{d+1}(y^{(0:d)})(2) + \cdots + R_{d+1}(y^{(0:d)})(2^{N_d}),\\
S^{\mathsf{O}}_{2^{N_d-1}} &= R_{d+1}(y^{(0:d)})(1) + R_{d+1}(y^{(0:d)})(3) + \cdots + R_{d+1}(y^{(0:d)})(2^{N_d}-1),\\
S^{\mathsf{E}}_{2^{N_d-1}} &= R_{d+1}(y^{(0:d)})(2) + R_{d+1}(y^{(0:d)})(4) + \cdots + R_{d+1}(y^{(0:d)})(2^{N_d}).
\end{aligned}$$

Then Algorithm 1 returns as output $R_d = \Delta_{N_d}/p_{r_d}(N_d)$, where Δ_{N_d} is the antithetic quantity in Algorithm 1. By the inductive hypothesis on $d+1$, we have for almost every $y^{(0:d)}$:

$$\mathbf{E}_{\pi_{d+1:D}}[R_{d+1} \mid y^{(0:d)}] = \gamma_{d+1}(y^{(0:d)}), \quad \mathbf{E}_{\pi}\left[|R_{d+1}|^{2^{d+2}}\right] < \left(\prod_{i=d+1}^{D} \tilde{C}_i\right) \left\| g_D(y^{(0:D)}) \right\|^{2^{D+1}}_{\pi,2^{D+1}}.$$

We will start with showing R_d has a finite computational cost and a finite 2^{d+1}-th moment, and then show the unbiasedness.
<u>Finite cost:</u>
To show the finite expected computational cost, recall that implementing Algorithm 1 with input depth d requires 2^{N_d} calls of Algorithm 1 with input depth $d+1$. Since $N_d \sim \mathsf{Geo}(r_d)$ with $r_d > 0.5$, calling Algorithm 1 with input depth d has an expected cost:

$$\frac{r_d}{2r_d - 1} \times \text{the expected cost of Algorithm 1 with input depth } d+1,$$

where $\frac{r_d}{2r_d-1} = \mathbf{E}[2^{N_d}] < \infty$. By our inductive hypothesis, the second term in the above product is finite, therefore the expected cost of Algorithm 1 with input depth d is also finite.

<u>Finite 2^{d+1}-th moment:</u>

Next we show R_d has a finite 2^{d+1}-th moment. For every fixed positive integer n, doing a Taylor expansions for g_d at

$(y^{(0:d)}, \gamma_{d+1})$ with respect to the last component gives us:

$$\begin{aligned}
g_d\left(y^{(0:d)}, \frac{S_{2^n}}{2^n}\right) &= g_d(y^{(0:d)}, \gamma_{d+1}) + \partial_{(d+1)M+1} g_d(y^{(0:d)}, \gamma_{d+1})\left(\frac{S_{2^n}}{2^n} - \gamma_{d+1}\right) \\
&\quad + \frac{1}{2}\partial^2_{(d+1)M+1} g_d(y^{(0:d)}, \xi(n))\left(\frac{S_{2^n}}{2^n} - \gamma_{d+1}\right)^2 \\
g_d\left(y^{(0:d)}, \frac{S^{\mathsf{O}}_{2^{n-1}}}{2^{n-1}}\right) &= g_d(y^{(0:d)}, \gamma_{d+1}) + \partial_{(d+1)M+1} g_d(y^{(0:d)}, \gamma_{d+1})\left(\frac{S^{\mathsf{O}}_{2^{n-1}}}{2^{n-1}} - \gamma_{d+1}\right) \\
&\quad + \frac{1}{2}\partial^2_{(d+1)M+1} g_d(y^{(0:d)}, \xi^{\mathsf{O}}(n-1))\left(\frac{S^{\mathsf{O}}_{2^{n-1}}}{2^{n-1}} - \gamma_{d+1}\right)^2 \\
g_d\left(y^{(0:d)}, \frac{S^{\mathsf{E}}_{2^{n-1}}}{2^{n-1}}\right) &= g_d(y^{(0:d)}, \gamma_{d+1}) + \partial_{(d+1)M+1} g_d(y^{(0:d)}, \gamma_{d+1})\left(\frac{S^{\mathsf{E}}_{2^{n-1}}}{2^{n-1}} - \gamma_{d+1}\right) \\
&\quad + \frac{1}{2}\partial^2_{(d+1)M+1} g_d(y^{(0:d)}, \xi^{\mathsf{E}}(n-1))\left(\frac{S^{\mathsf{E}}_{2^{n-1}}}{2^{n-1}} - \gamma_{d+1}\right)^2,
\end{aligned}$$

with $\xi(n)$ between γ_{d+1} and $S_{2^n}/2^n$, $\xi^{\mathsf{O}}(n-1)$ between γ_{d+1} and $S^{\mathsf{O}}_{2^{n-1}}/2^{n-1}$, $\xi^{\mathsf{E}}(n-1)$ between γ_{d+1} and $S^{\mathsf{E}}_{2^{n-1}}/2^{n-1}$. Thus, we have:

$$\begin{aligned}
\Delta_n &= g_d\left(y^{(0:d)}, \frac{S_{2^n}}{2^n}\right) - \frac{1}{2}\left[g_d\left(y^{(0:d)}, \frac{S^{\mathsf{O}}_{2^{n-1}}}{2^{n-1}}\right) + g_d\left(y^{(0:d)}, \frac{S^{\mathsf{E}}_{2^{n-1}}}{2^{n-1}}\right)\right] \\
&= \partial_{(d+1)M+1} g_d(y^{(0:d)}, \gamma_{d+1})\left(\frac{S_{2^n}}{2^n} - \gamma_{d+1}\right) + \frac{1}{2}\partial^2_{(d+1)M+1} g_d(y^{(0:d)}, \xi(n))\left(\frac{S_{2^n}}{2^n} - \gamma_{d+1}\right)^2 \\
&\quad - \frac{1}{2}\Bigg[\partial_{(d+1)M+1} g_d(y^{(0:d)}, \gamma_{d+1})\left(\frac{S^{\mathsf{O}}_{2^{n-1}}}{2^{n-1}} - \gamma_{d+1}\right) + \frac{1}{2}\partial^2_{(d+1)M+1} g_d(y^{(0:d)}, \xi^{\mathsf{O}}(n-1))\left(\frac{S^{\mathsf{O}}_{2^{n-1}}}{2^{n-1}} - \gamma_{d+1}\right)^2 \\
&\qquad + \partial_{(d+1)M+1} g_d(y^{(0:d)}, \gamma_{d+1})\left(\frac{S^{\mathsf{E}}_{2^{n-1}}}{2^{n-1}} - \gamma_{d+1}\right) + \frac{1}{2}\partial^2_{(d+1)M+1} g_d(y^{(0:d)}, \xi^{\mathsf{E}}(n-1))\left(\frac{S^{\mathsf{E}}_{2^{n-1}}}{2^{n-1}} - \gamma_{d+1}\right)^2\Bigg] \\
&= \frac{1}{2}\partial^2_{(d+1)M+1} g_d(y^{(0:d)}, \xi(n))\left(\frac{S_{2^n}}{2^n} - \gamma_{d+1}\right)^2 - \frac{1}{2}\Bigg[\frac{1}{2}\partial^2_{(d+1)M+1} g_d(y^{(0:d)}, \xi^{\mathsf{O}}(n-1))\left(\frac{S^{\mathsf{O}}_{2^{n-1}}}{2^{n-1}} - \gamma_{d+1}\right)^2 \\
&\qquad + \frac{1}{2}\partial^2_{(d+1)M+1} g_d(y^{(0:d)}, \xi^{\mathsf{E}}(n-1))\left(\frac{S^{\mathsf{E}}_{2^{n-1}}}{2^{n-1}} - \gamma_{d+1}\right)^2\Bigg].
\end{aligned}$$

By the LBS assumption which assumes $|\partial^2_{(d+1)M+1} g_d(y^{(0:d)}, z))| < K_d$ for every $(y^{(0:d)}, z)$, and our inductive hypothesis: $\gamma_{d+1} = \mathbf{E}[R_{d+1}(y^{(0:d)}) \mid y^{(0:d)}]$ which allows us to use Lemma A.3 with $Z_1 = y^{(0:d)}$, $Z_2 = R_{d+1}(y^{(0:d)})$, we have:

$$\begin{aligned}
\|\Delta_n\|_{\pi,2^{d+1}} &\le K_d\left\|\left(\frac{S_{2^n}}{2^n} - \gamma_{d+1}\right)^2\right\|_{\pi,2^{d+1}} + \frac{K_d}{2}\left\|\left(\frac{S^{\mathsf{O}}_{2^{n-1}}}{2^{n-1}} - \gamma_{d+1}\right)^2\right\|_{\pi,2^{d+1}} + \frac{K_d}{2}\left\|\left(\frac{S^{\mathsf{E}}_{2^{n-1}}}{2^{n-1}} - \gamma_{d+1}\right)^2\right\|_{\pi,2^{d+1}} \\
&\le K_d B'_{2^{d+2}}\left(\frac{\mathbf{E}_\pi\left[|R_{d+1}(y^{(0:d)})|^{2^{d+2}}\right]}{2^{(2^{d+1}n)}}\right)^{1/2^{d+1}} + K_d B'_{2^{d+2}}\left(\frac{\mathbf{E}_\pi\left[|R_{d+1}(y^{(0:d)})|^{2^{d+2}}\right]}{2^{(2^{d+1}(n-1))}}\right)^{1/2^{d+1}} \\
&= K_d B'_{2^{d+2}}\left(\frac{\mathbf{E}_\pi\left[|R_{d+1}(y^{(0:d)})|^{2^{d+2}}\right]}{2^{(2^{d+1}n)}}\right)^{1/2^{d+1}} + 2K_d B'_{2^{d+2}}\left(\frac{\mathbf{E}_\pi\left[|R_{d+1}(y^{(0:d)})|^{2^{d+2}}\right]}{2^{(2^{d+1}n)}}\right)^{1/2^{d+1}} \\
&= 3K_d B'_{2^{d+2}}\left(\frac{\mathbf{E}_\pi\left[|R_{d+1}(y^{(0:d)})|^{2^{d+2}}\right]}{2^{(2^{d+1}n)}}\right)^{1/2^{d+1}}.
\end{aligned}$$

Therefore, in total:

$$\mathbf{E}_\pi\left[|\Delta_n|^{2^{d+1}}\right] \leq D_d \frac{\mathbf{E}_\pi\left[|R_{d+1}(y^{(0:d)})|^{2^{d+2}}\right]}{2^{(2^{d+1}n)}},$$

with $D_d = (3K_d B'_{2^{d+2}})^{(2^{d+1})}$.

The result claimed in (c) is obtained as follows. We have for $(1-r_d) = 2^{-k_d}$ for some $k_d \in \left(1, \frac{2^{d+1}}{2^{d+1}-1}\right)$:

$$\begin{aligned}
\mathbf{E}_\pi\left[|R_d(y^{(0:d-1)})|^{2^{d+1}}\right] &= \sum_{n=0}^{\infty} \frac{\mathbf{E}_\pi\left[|\Delta_n|^{2^{d+1}}\right]}{p_{r_d}(n)^{2^{d+1}-1}} \\
&\leq \frac{D_d \mathbf{E}_\pi\left[|R_{d+1}(y^{(0:d)})|^{2^{d+2}}\right]}{r_d^{2^{d+1}-1}} \sum_{n=0}^{\infty} \frac{1}{2^{2^{d+1}n}(1-r_d)^{(2^{d+1}-1)n}} \\
&\leq C_d \left(\prod_{i=d+1}^{D} \tilde{C}_i\right) \left\|g_D(y^{(0:D)})\right\|_{\pi,2^{D+1}}^{2^{D+1}} \sum_{n=0}^{\infty} \left(\frac{1}{2^{2^{d+1}-k_d(2^{d+1}-1)}}\right)^n \quad \text{inductive hypothesis} \\
&= C_d \left(\prod_{i=d+1}^{D} \tilde{C}_i\right) \left\|g_D(y^{(0:D)})\right\|_{\pi,2^{D+1}}^{2^{D+1}} \left(\frac{2^{(2^{d+1}-k_d(2^{d+1}-1))}}{2^{(2^{d+1}-k_d(2^{d+1}-1))}-1}\right) \\
&= \left(\prod_{i=d}^{D} \tilde{C}_i\right) \left\|g_D(y^{(0:D)})\right\|_{\pi,2^{D+1}}^{2^{D+1}}.
\end{aligned}$$

Here the choice $k_d \in \left(1, \frac{2^{d+1}}{2^{d+1}-1}\right)$ is crucial. It ensures $2^{(2^{d+1})}(1-r_d)^{(2^{d+1}-1)} > 1$, and in turn ensures the infinite summation of the above geometric series is finite.

<u>Unbiasedness:</u>
Now we show the unbiasedness of $R_d(y^{(0:d-1)})$. Firstly, since we have just shown $R_d(y^{(0:d-1)})$ has a finite 2^{d+1}-th moment under π, it directly implies $|R_d(y^{(0:d-1)})|$ has a finite first moment, which further implies $\mathbf{E}[|R_d(y^{(0:d-1)})| \mid y^{(0:d-1)}]$ is finite for π-almost surely $y^{(0:d-1)}$.

We fix $y^{(0:d-1)}$ from now on, and we will write $\mathbf{E}_{\pi_{d:D}}[\cdot]$ as a shorthand notation for $\mathbf{E}[\cdot \mid y^{(0:d-1)}]$. Recall that we have output $R_d = \Delta_{N_d}/p_{r_d}(N_d)$, with

$$\Delta_{N_d} = g_d\left(y^{(0:d)}, \frac{S_{2^{N_d}}}{2^{N_d}}\right) - \frac{1}{2}\left[g_d\left(y^{(0:d)}, \frac{S^{\mathsf{O}}_{2^{N_d-1}}}{2^{N_d-1}}\right) + g_d\left(y^{(0:d)}, \frac{S^{\mathsf{E}}_{2^{N_d-1}}}{2^{N_d-1}}\right)\right].$$

Then,

$$
\begin{aligned}
\mathbf{E}_{\pi_{d:D}}[R_d(y^{(0:d-1)})] &= \mathbf{E}_{\pi_{d:D}}\left[\mathbf{E}\left[\frac{\Delta_{N_d}}{p_{r_d}(N_d)} \mid N_d\right]\right] \\
&= \mathbf{E}_{\pi_{d:D}}\left[\sum_{n=0}^{\infty}\frac{\Delta_n}{p_{r_d}(n)}p_{r_d}(n)\right] \\
&= \mathbf{E}_{\pi_{d:D}}\left[\sum_{n=0}^{\infty}\Delta_n\right] \\
&\overset{(\star\star\star)}{=} \sum_{n=0}^{\infty}\mathbf{E}_{\pi_{d:D}}[\Delta_n] \\
&= \sum_{n=0}^{\infty}\mathbf{E}_{\pi_{d:D}}\left\{g_d\left(y^{(0:d)}, \frac{S_{2^n}}{2^n}\right) - \frac{1}{2}\left[g_d\left(y^{(0:d)}, \frac{S^{\mathsf{O}}_{2^{n-1}}}{2^{n-1}}\right) + g_d\left(y^{(0:d)}, \frac{S^{\mathsf{E}}_{2^{n-1}}}{2^{n-1}}\right)\right]\right\} \\
&= \sum_{n=1}^{\infty}\left\{\mathbf{E}_{\pi_{d:D}}\left[g_d\left(y^{(0:d)}, \frac{S_{2^n}}{2^n}\right)\right] - \mathbf{E}_{\pi_{d:D}}\left[g_d\left(y^{(0:d)}, \frac{S_{2^{n-1}}}{2^{n-1}}\right)\right]\right\} + \mathbf{E}_{\pi_{d:D}}[g_d(y^{(0:d)}, g_{d+1}(1))] \\
&= \mathbf{E}_{\pi_{d:D}}\left[g_d\left(y^{(0:d)}, \lim_{n\to\infty}\frac{S_{2^n}}{2^n}\right)\right] \\
&= \mathbf{E}_{\pi_{d:D}}\left[g_d\left(y^{(0:d)}, \gamma_{d+1}(y^{(0:d)})\right)\right] \\
&= \gamma_d(y^{(0:d-1)}).
\end{aligned}
$$

All the above calculations are straightforward except for $(\star\star\star)$, which swaps the order of expectation and summation. Therefore we complete this proof of unbiasedness by justifying the swap in $(\star\star\star)$. To justify the swap, it suffices to show $\sum_n \mathbf{E}[|\Delta_n|] < \infty$. Notice that $R_d(y^{(0:d)})$ can be equivalently written as $\sum_{n=1}^{\infty}\Delta_n \mathbf{I}(N_d = n)/p_{r_d}(n)$ where N_d independent with $\{\Delta_i\}$. Calculating $\mathbf{E}_{\pi_{d:D}}[|R_d(y^{(0:d)})|]$ yields:

$$
\begin{aligned}
\mathbf{E}_{\pi_{d:D}}[|R_d(y^{(0:d-1)})|] &= \mathbf{E}_{\pi_{d:D}}\left[\left|\sum_{n=1}^{\infty}\frac{\Delta_n \mathbf{I}(N_d = n)}{p_{r_d}(n)}\right|\right] \\
&= \mathbf{E}_{\pi_{d:D}}\left[\sum_{n=1}^{\infty}\left|\frac{\Delta_n \mathbf{I}(N_d = n)}{p_{r_d}(n)}\right|\right] && \text{only one term in the summation is non-zero} \\
&= \sum_{n=1}^{\infty}\mathbf{E}_{\pi_{d:D}}\left[\left|\frac{\Delta_n \mathbf{I}(N_d = n)}{p_{r_d}(n)}\right|\right] && \text{every term is non-negative} \\
&= \sum_{n=1}^{\infty}\mathbf{E}_{\pi_{d:D}}\left[\left|\frac{\Delta_n}{p_{r_d}(n)}\right|\right]\mathbf{E}\left[\mathbf{I}(N_d = n)\right] && \text{independence between } N \text{ and } \{\Delta_i\} \\
&= \sum_{n=1}^{\infty}\mathbf{E}_{\pi_{d:D}}\left[|\Delta_n|\right].
\end{aligned}
$$

Since we already know $\mathbf{E}_{\pi_{d:D}}[|R_d(y^{(0:d-1)})|] < \infty$, this justifies our swap $(\star\star\star)$.

□

C. Proof of Theorem 2.4

Recall that we say $\{g_d\}_{d=0}^{D-1}$ satisfy the last-component bounded Lipschitz condition (LBL) if for all $x, z \in \mathbf{R}$:

$$|g_d(y^{(0:d)}, x) - g_d(y^{(0:d)}, z)| < L_d|x - z|.$$

The proof strategy of Theorem 2.4 is very similar to Theorem 2.2. We start with a backward induction.

Proof. **Case 1:** $d = D$

When $d = D$, Algorithm 1 samples one $y^{(D)} \sim \pi_D$ and outputs $R_D(y^{(0:D-1)}) := g_D(y^{(0:D)})$. Again, we first prove our output $R_D(y^{(0:D-1)})$ has a finite expectation for almost every fixed $y^{(0:D-1)}$, then its expectation equals $\gamma_D(y^{(0:D-1)})$ follows directly from the algorithm design. To show the first point, notice that the expectation of $|R_D(y^{(0:D-1)})|$ equals the conditional expectation $\mathbf{E}[|g_D(y^{(0:D)})| \mid y^{(0:D-1)}]$. Since $\mathbf{E}[|g_D|] < \infty$ by assumption, we have

$$\mathbf{E}[|g_D(y^{(0:D)})| \mid y^{(0:D-1)}] < \infty$$

almost surely. Therefore Algorithm 1 is unbiased when $d = D$ for almost every input $y^{(0:D-1)}$. Furthermore, it has computational cost 1, and the output has finite $\left(2 - \frac{\delta}{2^D}\right)$-th moment.

Case 2: $0 \leq d \leq D - 1$

Now that our base case is proven, we proceed via backwards induction. Let $\delta_d := \delta/2^d$ for every $d \in \{0, 1, \ldots, D\}$. Suppose unbiasedness, finite $(2 - \delta_{d+1})$-th moment, and finite expected computational cost are all satisfied for $d + 1$ where $0 \leq d \leq D-1$. Then Algorithm 1 will call itself independently for 2^{N_d} times, each with input {Depth index: $d+1$, Trajectory History: $H = y^{(0:d)}$, Parameters: $r_{d+1}, \cdots, r_{D-1}$}. This gives us i.i.d. samples $R_{d+1}(y^{(0:d)})(1), ..., R_{d+1}(y^{(0:d)})(2^{N_d})$ which are used to compute the following:

$$\begin{aligned}
S_{2^{N_d}} &= R_{d+1}(y^{(0:d)})(1) + R_{d+1}(y^{(0:d)})(2) + \cdots + R_{d+1}(y^{(0:d)})(2^{N_d}),\\
S^{\mathsf{O}}_{2^{N_d-1}} &= R_{d+1}(y^{(0:d)})(1) + R_{d+1}(y^{(0:d)})(3) + \cdots + R_{d+1}(y^{(0:d)})(2^{N_d} - 1),\\
S^{\mathsf{E}}_{2^{N_d-1}} &= R_{d+1}(y^{(0:d)})(2) + R_{d+1}(y^{(0:d)})(4) + \cdots + R_{d+1}(y^{(0:d)})(2^{N_d}).
\end{aligned}$$

Then Algorithm 1 returns as output $R_d(y^{(0:d)}) = \Delta_{N_d}/p_{r_d}(N_d)$, where Δ_{N_d} is defined in Algorithm 1. By the inductive hypothesis on $d + 1$, we have for almost every $y^{(0:d)}$:

$$\mathbf{E}[R_{d+1}(y^{(0:d)}) \mid y^{(0:d)}] = \gamma_{d+1}(y^{(0:d)})$$

and

$$\mathbf{E}_\pi\left[|R_{d+1}(y^{(0:d)})|^{2-\delta_{d+1}}\right] < \left(\prod_{i=d}^{D} \tilde{C}_i\right) \left\|g_D(y^{(0:D)})\right\|_{\pi,2}^{2-\delta_{d+1}}.$$

We will start with showing $R_d(y^{(0:d-1)})$ has a finite computational cost and a finite $(2 - \delta_d)$-th moment, and then show the unbiasedness.

<u>Finite cost:</u>

To show the computational cost, recall that implementing Algorithm 1 with input depth d requires 2^{N_d} calls of Algorithm 1 with input depth $d + 1$. It suffices to check $r_d > 0.5$, which reduces to check the upper bound for k_d (defined in Theorem 2.4) satisfies

$$\left(\frac{2^{d+2} - 3\delta}{2^{d+3} - 3\delta}\right)\left(\frac{2^{d+1} - \delta}{2^d - \delta}\right) > 1.$$

Let $t := 2^d$ and the above product becomes:

$$\frac{4t - 3\delta}{8t - 3\delta}\frac{2t - \delta}{t - \delta} = \frac{8t^2 + 3\delta^2 - 10\delta}{8t^2 + 3\delta^2 - 11\delta} > 1.$$

Since $N_d \sim \mathsf{Geo}(r_d)$ with $r_d > 0.5$, calling Algorithm 1 with input depth d has an expected cost:

$$\frac{r_d}{2r_d - 1} \times \text{the expected cost of Algorithm 1 with input depth } d+1,$$

where $\frac{r_d}{2r_d-1} = \mathbf{E}[2^{N_d}] < \infty$. By our inductive hypothesis, the second term in the above product is finite, therefore the expected cost of Algorithm 1 with input depth d is also finite.

Finite $(2-\delta_d)$-th moment:
Next we show R_d has a finite $(2-\delta_d)$-th moment. By the uniform L_d-Lipschitz property of g_d:

$$\begin{aligned}|\Delta_n| &\le \frac{1}{2}\left|g_d\left(y^{(0:d)}, \frac{S_{2^n}}{2^n}\right) - g_d\left(y^{(0:d)}, \frac{S^{\mathsf{O}}_{2^{n-1}}}{2^{n-1}}\right)\right| + \frac{1}{2}\left|g_d\left(y^{(0:d)}, \frac{S_{2^n}}{2^n}\right) - g_d\left(y^{(0:d)}, \frac{S^{\mathsf{E}}_{2^{n-1}}}{2^{n-1}}\right)\right| \\ &\le \frac{L_d}{2}\left|\frac{S^{\mathsf{O}}_{2^{n-1}}}{2^{n-1}} - \frac{S^{\mathsf{E}}_{2^{n-1}}}{2^{n-1}}\right|.\end{aligned}$$

Therefore, for any fixed $1 \le p < 2$, applying the triangle inequality under the norm $\|\cdot\|_{\pi,p}$, and applying Lemma A.3 with $Z_1 = y^{(0:d)}$, $Z_2 = R_{d+1}(y^{(0:d)})$, we have:

$$\begin{aligned}\|\Delta_n\|_{\pi,p} &\le \frac{L_d}{2}\left\|\frac{S^{\mathsf{O}}_{2^{n-1}}}{2^{n-1}} - \frac{S^{\mathsf{E}}_{2^{n-1}}}{2^{n-1}}\right\|_{\pi,p} \\ &\le \frac{L_d}{2}\left\|\frac{S^{\mathsf{O}}_{2^{n-1}}}{2^{n-1}} - \gamma_{d+1}\right\|_{\pi,p} + \frac{L_d}{2}\left\|\gamma_{d+1} - \frac{S^{\mathsf{E}}_{2^{n-1}}}{2^{n-1}}\right\|_{\pi,p} \\ &\le L_d\left(\frac{B_p \mathbf{E}_\pi[|R_{d+1}(y^{(0:d)})|^p]}{2^{(n-1)(p-1)}}\right)^{1/p},\end{aligned}$$

exponentiating both sides by p yields,

$$\mathbf{E}_\pi[|\Delta_n|^p] \le \frac{L_d^p B_p \mathbf{E}_\pi[|R_{d+1}(y^{(0:d)})|^p]}{2^{(n-1)(p-1)}} \le \frac{C(d,p)\mathbf{E}_\pi[|R_{d+1}(y^{(0:d)})|^p]}{2^{(p-1)n}},$$

where $C(d,p) = L_d^p B_p 2^{1-p}$.

Recall that $\delta_d = \delta/2^d$, let us choose $q_d = 2 - (\delta_d + \delta_{d+1})/2$. Since

$$\left(1, \left(\frac{q_d - 1}{q_d}\right)\left(\frac{2-\delta_d}{1-\delta_d}\right)\right) = \left(1, \left(\frac{2^{d+2} - 3\delta}{2^{d+3} - 3\delta}\right)\left(\frac{2^{d+1} - \delta}{2^d - \delta}\right)\right),$$

by definition of k_d in the Theorem statement we have

$$k_d < \left(\frac{q_d - 1}{q_d}\right)\left(\frac{2-\delta_d}{1-\delta_d}\right).$$

Now we estimate the $(2-\delta_d)$-th moment of R_d. An important trick in the calculation below is that we are not going to use the above estimate of $\mathbf{E}_\pi[|\Delta_n|^p]$ directly on $p = 2-\delta_d$. Instead, we will first use Hölder's inequality, and then bound $\mathbf{E}_\pi[|\Delta_n|^{q_d}]^{(2-\delta_d)/q_d}$ via the above estimate. It turns out the first way gives us an order of $2^{-n(1-\delta_d)}$, while the latter is of order $2^{-n(q_d-1)(2-\delta_d)/q_d}$. Since the function $(x-1)(2-\delta_d)/x$ is increasing with x when $x > 1$, and equals $1-\delta_d$ when $x = 2-\delta_d$, we gain an extra factor $2^{-\Omega(1)n}$ by choosing $q_d > 2-\delta_d$ and use Hölder's inequality, which is important for establishing our main result. The detailed calculation is below:

$$
\begin{aligned}
\mathbf{E}_\pi[|R_d(y^{(0:d-1)})|^{2-\delta_d}] &\le \sum_{n=0}^{\infty} \frac{\mathbf{E}_\pi[|\Delta_n|^{2-\delta_d}]}{p_{r_d}(n)^{1-\delta_d}} \\
&\le \sum_{n=0}^{\infty} \frac{\left(\mathbf{E}_\pi\left[|\Delta_n|^{2-\delta_d \cdot \frac{q_d}{2-\delta_d}}\right]\right)^{\frac{2-\delta_d}{q_d}}}{p_{r_d}(n)^{1-\delta_d}} && \text{Hölder's inequality} \\
&\le \frac{1}{r_d^{1-\delta_d}} \sum_{n=0}^{\infty} \left(\frac{C(d,q_d)\mathbf{E}_\pi[|R_{d+1}(y^{(0:d)})|^{q_d}]}{2^{(q_d-1)n}}\right)^{\frac{2-\delta_d}{q_d}} \frac{1}{(1-r_d)^{(1-\delta_d)n}} && \text{estimate of } \mathbf{E}_\pi[|\Delta_n|^p] \text{ with } p = q_d \\
&= C'(d)\left\|R_{d+1}(y^{(0:d)})\right\|_{\pi,q_d}^{2-\delta_d} \sum_{n=0}^{\infty} \left(\frac{1}{2^{\frac{(q_d-1)}{q_d}(2-\delta_d)-k_d(1-\delta_d)}}\right)^n && \text{here } C'(d) = \frac{C(d,q_d)^{(2-\delta_d)/q_d}}{r_d^{1-\delta_d}} \\
&\le C'(d)\left\|R_{d+1}(y^{(0:d)})\right\|_{\pi,2-\delta_{d+1}}^{2-\delta_d} \sum_{n=0}^{\infty} \left(\frac{1}{2^{\frac{(q_d-1)}{q_d}(2-\delta_d)-k_d(1-\delta_d)}}\right)^n && \text{since } q_d < 2-\delta_{d+1} \\
&\le C'(d)\left(\prod_{i=d+1}^{D} \tilde{C}_i\right)\left\|g_D(y^{(0:D)})\right\|_{\pi,2}^{2-\delta_d} \left(\frac{2^{\frac{(q_d-1)}{q_d}(2-\delta_d)-k_d(1-\delta_d)}}{2^{\frac{(q_d-1)}{q_d}(2-\delta_d)-k_d(1-\delta_d)}-1}\right) && \text{inductive hypothesis} \\
&= \left(\prod_{i=d}^{D} \tilde{C}_i\right)\left\|g_D(y^{(0:D)})\right\|_{\pi,2}^{2-\delta_d},
\end{aligned}
$$

and note the RHS is still finite given the assumption of our theorem on g_D. Again, as we can see in the proof, the choice of k_d and q_d is crucial for our calculation. It ensures $\frac{(q_d-1)}{q_d}(2-\delta_d) - k_d(1-\delta_d) > 0$, and in turn ensures the above summation of the geometric series converges.

Unbiasedness:

The proof of unbiasedness of our estimator in this case is identical to the LBS case, however we still require a justification of the existence of a finite conditional expectation of $R_d(y^{(0:d-1)})$. By what we have just proven,

$$\mathbf{E}_\pi[|R_d(y^{(0:d-1)})|] \le \left(\mathbf{E}_\pi\left[|R_d(y^{(0:d-1)})|^{2-\frac{\delta}{2^d}}\right]\right)^{1/(2-\frac{\delta}{2^d})} < \infty.$$

Given $\mathbf{E}_\pi[|R_d(y^{(0:d-1)})|] < \infty$, we immediately have $\mathbf{E}_{\pi_{d:D}}[R_d(y^{(0:d-1)})]$ exists for almost every $y^{(0:d-1)}$.

□

D. Construction of the NMC estimator

The construction of the NMC estimator is described in (Rainforth et al., 2018). For concreteness, we explain the construction details here for the $D = 2$ case, which we use in Section 3.

Fix positive integers N_0, N_1, N_2, users first simulate N_0 i.i.d. $\{y_i^{(0)}\}_{i=1}^{N_0} \sim \pi_0$. For each fixed $y_i^{(0)}$, users sample N_1 i.i.d. $\{y_{i,j}^{(1)}\}_{j=1}^{N_1}$ from $\pi_1(\cdot \mid y_i^{(0)})$. Then for each fixed trajectory $(y_i^{(0)}, y_{i,j}^{(1)})$, users sample N_2 i.i.d. $\{y_{i,j,k}^{(2)}\}_{k=1}^{N_2}$ from $\pi_2(\cdot \mid y_i^{(0)}, y_{i,j}^{(1)})$. After getting all these samples, we use the standard estimator:

$$\hat{\gamma}_2(y_i^{(0)}, y_{i,j}^{(1)}) := \frac{1}{N_2}\sum_{k=1}^{N_2} g_2\left(y_i^{(0)}, y_{i,j}^{(1)}, y_{i,j,k}^{(2)}\right)$$

to estimate $\gamma_2(y_i^{(0)}, y_{i,j}^{(1)})$. Then using the plug-in estimator

$$\hat{\gamma}_1(y_i^{(0)}) := \frac{1}{N_1}\sum_{j=1}^{N_1} g_1\left(y_i^{(0)}, y_{i,j}^{(1)}, \hat{\gamma}_2(y_i^{(0)}, y_{i,j}^{(1)})\right)$$

to estimate $\gamma_1(y_i^{(0)})$. Finally, plugging in these N_0 estimators $\{\hat{\gamma}_1(y_i^{(0)})\}_{i=1}^{N_0}$ to form the NMC estimator

$$\hat{\gamma}_0 := \frac{1}{N_0}\sum_{i=1}^{N_0} g_0\left(y_i^{(0)}, \hat{\gamma}_1(y_i^{(0)})\right)$$

for γ_0.

It is proven in (Rainforth et al., 2018) that when all N_0, N_1, N_2 go to ∞, $\hat{\gamma}_0$ converges to γ_0. It remains crucial to allocate N_0, N_1, N_2 to maximize the convergence rate with respect to the total sample size $n = N_0 N_1 N_2$. The choice $N_0 = N_1^2 = N_2^2$ is suggested in (Rainforth et al., 2018), which has a $\mathcal{O}(N^{-1/4})$ convergence rate for the rMSE, or a $\mathcal{O}(N^{-1/2})$ rate for the MSE.

E. Additional statistics of Section 3

Figure 1 in Section 3 compares the errors between READ, NMC1, and NMC2 in terms of the cost of the total sample size. Here we also compare their estimation errors in terms of the wall-clock time. For READ, we call Algorithm 1 repeatedly for 10^5 times. For NMC1, we choose $N_0 = N_1 = N_2 = 400$. For NMC2, we choose $N_0 = N_1^2 = N_2^2 = 10^4$. Their estimation errors and the corresponding wall-clock time costs are summarized in Table 2. It is clear that READ is both faster (in wall-clock time) and more accurate. We also calculate the time-normalized squared error, defined as the product between the time cost and the squared error (Glynn & Whitt, 1992). From the normalized squared error, READ is more than 130 times more efficient than NMC2, and more than 407 times more efficient than NMC1.

Method	Setting	Total Sample Cost	Time/Seconds	Squared Error	Time-normalized Squared Error
READ	10^5 repetitions	4.625×10^5	11.86	3.186×10^{-6}	3.78×10^{-5}
NMC1	$N_0 = N_1 = N_2 = 400$	6.4×10^7	43.79	3.51×10^{-4}	1.54×10^{-2}
NMC2	$N_0 = N_1^2 = N_2^2 = 10^4$	10^8	72.6	6.8×10^{-5}	4.93×10^{-3}

Table 2. Cost comparison between different methods

F. Extra experiments

We consider an extra experiment with unknown ground truth. Let $\sigma(x) := e^x/(1+e^x)$ be the sigmoid function. Suppose the process $(y^{(0)}, y^{(1)}, y^{(2)})$ satisfies $y^{(0)} \sim \mathbf{N}(0,1), y^{(1)} \sim \mathbf{N}(y^{(0)}, 1), y^{(2)} \sim \mathbf{N}(y^{(1)}, 1)$. Define $g_0(y^{(0)}, z) := \sigma(y^{(0)} + z), g_1(y^{(0:1)}, z) := \sigma(y^{(1)} + z)$, and $g_2(y^{(0:2)}) := \sigma(y^{(2)})$. The target quantity γ_0 defined (1) is again a nested expectation with $D = 2$. Although we can not analytically calculate out γ_0, we still implement our READ estimator with the NMC estimators in (Rainforth et al., 2018) and compare their performance. The parameters of READ are the same as Section 3, the allocation of N_0, N_1, N_2 of the NMC estimators also follows the same way as Section 3.

The scatter plot of the estimation results is shown in Figure 3. Although no ground truth is available, the trend for the estimation is clear. All three methods eventually get close to 0.612, represented by the dotted black line in Figure 3. It is also clear from the plot that READ always stays very close to the black line. In contrast, both NMC1 and NMC2 are significantly more fluctuated than READ, where NMC1 appears to be the most unstable estimator. This again matches with the theoretical predictions in our paper and (Rainforth, 2018) that READ converges the fastest while NMC1 converges the slowest.

Next we let the parameters (r_0, r_1) in Algorithm 1 vary and investigate the proper choice of the parameters in this experiment. Theorem 2.2 shows any $(r_0, r_1) \in (0.5, 0.75) \times (0.5, 1 - 2^{-4/3})$ guarantees READ has finite variance and finite cost. Therefore we choose r_0 on the lattice $\{0.6, 0.614, \ldots, 0.74\}$ and r_1 on the lattice $\{0.55, 0.555, \ldots, 0.6\}$. For each pair of (r_0, r_1), we repeat Algorithm 1 for 10^6 times, record the results and calculate their empirical standard deviation. The heatmap is shown in the left plot of Figure 4. The pattern suggests the standard deviation depends crucially on the choice of r_0, but less on r_1. The standard deviation decreases when r_0 increases.

Since the expected sample cost of Algorithm 1 equals $(r_1/(2r_1 - 1))\,(r_2/(2r_2 - 1))$. We also plot the 'work-normalized standard deviation', which is defined as $\sqrt{\text{Expected Sample Cost}} \times \text{Standard deviation}$ in (Glynn & Whitt, 1992) to measure

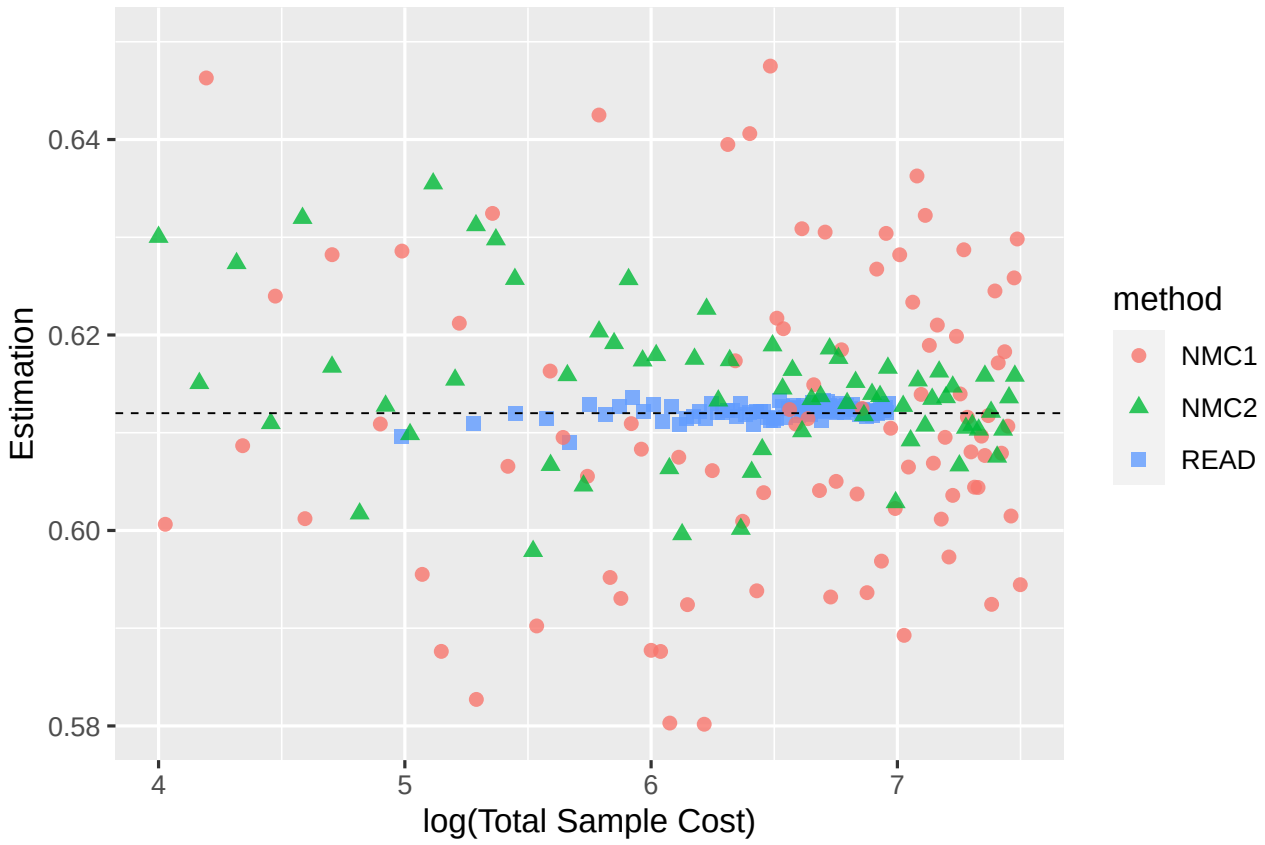
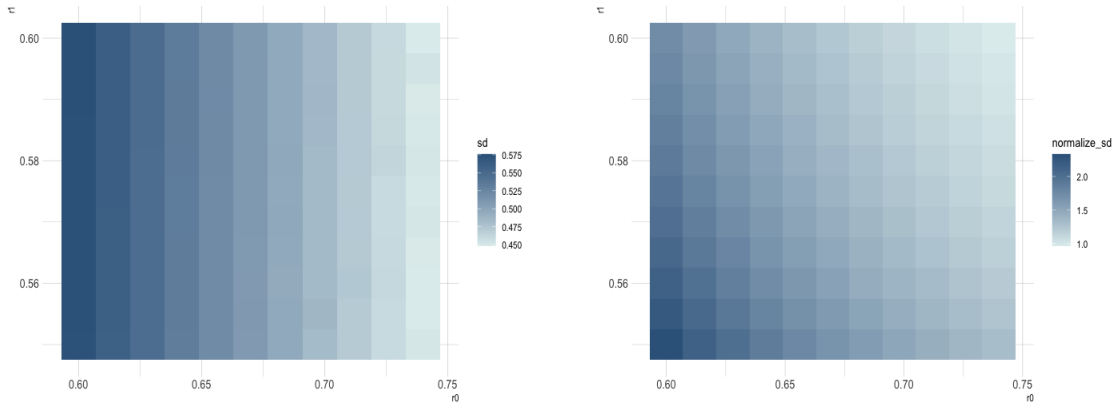


Figure 3. Scatterplot of the estimation of γ_0 as a function of $\log_{10}(\text{Total Sample Cost})$. Blue, red, green points correspond to READ, NMC1, NMC2 estimators respectively.

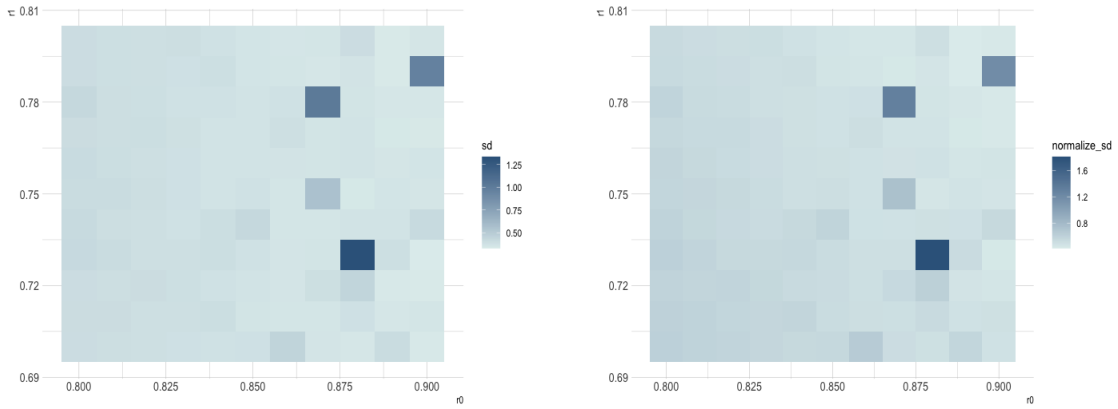


(a) (b)

Figure 4. (a): Heatmap of the (empirical) standard deviation of READ. (b): Heatmap of the work-normalized standard deviation of READ. Here $r_0 \in (0.6, 0.74)$, $r_1 \in (0.55, 0.6)$. Each standard deviation is estimated based on 10^6 repetitions of Algorithm 1.

the efficiency of difference choices of (r_0, r_1). The heatmap is shown in the right subplot of Figure 4. Our result suggests users should choose larger values of (r_0, r_1) to maximize the efficiency, at least in this example.

Finally we test our results when r_0, r_1 are both beyond the range given by Theorem 2.2. Algorithm 1 can still be implemented, though there is no guarantees on the finite variance. Nevertheless, we choose $r_0 \in \{0.8, 0.81, \ldots, 0.9\}$ and $r_1 \in \{0.7, 0.71, \ldots, 0.8\}$ and report the heatmaps of the standard deviations/work-normalized standard deviations in Figure s5. The estimates become significantly less stable, as some pairs of (r_0, r_1) have much larger standard deviation than their neighborhoods. This suggests the actual standard deviation maybe already infinity (though the empirical standard deviation will always be finite), and therefore our result is less reliable. In conclusion, although larger values of (r_0, r_1) can reduce the average cost of each implementation, users should not choose them too large as it may sacrifice the finite variance. Users can choose the parameters closer to the upper end of the ranges in Theorem 2.2, but not exceed these ranges.



(a) (b)

Figure 5. (a): Heatmap of the (empirical) standard deviation of READ. (b): Heatmap of the work-normalized standard deviation of READ. Here $r_0 \in (0.8, 0.9)$, $r_1 \in (0.7, 0.8)$. Each standard deviation is estimated based on 10^6 repetitions of Algorithm 1.