SIAM J. NUMER. ANAL.
Vol. 61, No. 6, pp. 2744–2774

MONTE CARLO GRADIENT IN OPTIMIZATION CONSTRAINED BY RADIATIVE TRANSPORT EQUATION*

QIN LI†, LI WANG‡, AND YUNAN YANG§

Abstract. Can Monte Carlo (MC) solvers be directly used in gradient-based methods for PDE-constrained optimization problems? In these problems, a gradient of the loss function is typically presented as a product of two PDE solutions, one for the forward equation and the other for the adjoint. When MC solvers are used, the numerical solutions are Dirac measures. As such, one immediately faces the difficulty in explaining the multiplication of two measures. This suggests that MC solvers are naturally incompatible with gradient-based optimization under PDE constraints. In this paper, we study two different strategies to overcome the difficulty. One is to adopt the discretize-then-optimize technique and conduct the full optimization on the algebraic system, avoiding the Dirac measures. The second strategy stays within the optimize-then-discretize framework. We propose a correlated simulation where, instead of using MC solvers separately for both forward and adjoint problems, we recycle the samples in the forward simulation in the adjoint solver. This frames the adjoint solution as a test function and hence allows a rigorous convergence analysis. The investigation is presented through the lens of the radiative transfer equation, either in the inverse setting from optical imaging or in the optimal control framework. We detail the algorithm development, convergence analysis, and complexity cost. Numerical evidence is also presented to demonstrate the claims.

Key words. Monte Carlo gradient, particle method, radiative transport equation, adjoint-state method

MSC codes. 65C05, 65K10, 65M75, 82C70

DOI. 10.1137/22M1524515

1. Introduction.

The Monte Carlo (MC) method consists of a class of computational strategies that use random samples/particles to represent the underlying distributions. It is immensely popular in numerical problems with high dimensions due to its practical and theoretical advantages. It is simple to code up and converges robustly with a convergence rate independent of the dimension [12] in certain metrics. One class of such high-dimensional problems is introduced by kinetic equations whose state variables reside in seven-dimensional phase space. Parallel to developing mesh-based methods, direct simulation Monte Carlo (DSMC) and its extensions [8, 33, 9, 34, 26, 35, 23] have garnered a great amount of interest.

*Received by the editors September 26, 2022; accepted for publication (in revised form) July 6, 2023; published electronically November 17, 2023.

https://doi.org/10.1137/22M1524515

Funding: The first author was supported in part by NSF grant DMS-1750488 and WARF-Wisconsin. The second author was partially supported by NSF grant DMS-1846854. The third author received support from Dr. Max Rössler, the Walter Haefner Foundation, and the ETH Zürich Foundation. This work was also partially supported by a grant from the Simons Foundation.

†Department of Mathematics, University of Wisconsin-Madison, Madison, WI 53705 USA (qinli@math.wisc.edu).

‡Department of Mathematics, University of Minnesota Twin Cities, Minneapolis, MN 55455 USA (wang8818@umn.edu).

§Department of Mathematics, Cornell University, Ithaca, NY 14853 USA (yunan.yang@cornell.edu).

PDE-constrained optimization is a classical formulation to solve inverse and control problems. In these problems, the parameters are adjusted accordingly to either fit the reference data or achieve a certain desired property. In the updating process, gradients of the loss function are computed in each iteration. This usually translates to computing two PDEs: one is the original forward equation and the other is the adjoint PDE. When these PDEs are presented on high dimensions, a prohibitive computational cost may incur.

To save numerical cost, we naturally face the option of utilizing MC solvers for gradient-based PDE-constrained optimizations. Then a natural question arises: can they be applied directly? We investigate this problem using kinetic equations as the underlying PDEs. An overarching message we would like to deliver in this paper is that, at least in the case of kinetic equations,

> *MC solvers are not immediately compatible with gradient-based optimization strategies.*

At the core of the difficulty is the incompatibility of the MC perspective and the computation of the gradients. Indeed, the MC solver views the PDE solution as a linear combination of Dirac delta functions, each representing one particle. The gradients derived for the PDE-constrained optimization problems, however, typically constitute a quadratic form that requires the multiplication of two PDE solutions (one forward and one adjoint). Mathematically, it is challenging to make sense of a multiplication of two delta measures. Physically, this incompatibility simply comes from the fact that two independent particles, formulated from forward and adjoint procedures, do not share trajectories, and there is an ambiguity in tracing the information. As such, a change of perspective is needed to integrate MC PDE solvers into the PDE-constrained optimization. Since MC solvers only provide convergence to the PDEs in the weak sense, we have to evaluate the convergence through the dual perspective, and all the rigorous proof must be done with exceedingly careful calculation in the observable space [2].

This shift of perspective will be demonstrated in this paper through the lens of the radiative transfer equation (RTE), a classical kinetic equation [14]. Kinetic equations are a class of equations describing the dynamics of many interacting particles. Kinetic equations are widely used in statistical mechanics to characterize the evolution of the distribution function of particles before it achieves the equilibrium state. Kinetic equations sit between thermo/fluid dynamics that describe macroscopic evolution and molecular dynamics that focus on fine-scale motions. Kinetic equations are practically useful in engineering and physics, and they also carry some unique mathematical features. Computationally, MC methods become a natural class of candidates due to the many-body nature of the system. Samples are drawn from the initial distribution and then moved around according to the physical law prescribed by the equation to simulate the evolution of the whole distribution. Monte Carlo strategy has been successfully applied to the study of the RTE (for photons), the Boltzmann equation (for rarefied gas), and the Vlasov–Fokker–Planck equation (for plasma), among many others [8, 12, 21, 18]. In the optimization setting, various MC solvers are designed in [22, 37]. In our paper, we choose RTE because the equation is linear with a clear collisional spectrum, and the well-posedness of the associated inverse problem forms a well-founded base for us to focus on the algorithmic aspect [30, 25, 24, 5, 18, 16, 15].

Our proposal consists of two strategies. One is the optimize-then-discretize (OTD) approach. It starts with the original optimization problem and derives the gradient of the loss function. This gradient can usually be written as a product of two PDE solutions. To resolve the incompatibility between MC solvers and the gradient com-

putation, instead of directly using MC solvers to compute both forward and adjoint equations separately, we propose to simulate only the forward equation using MC and record the trajectory for propagating the adjoint information backward in time. A similar strategy was applied to the Boltzmann equation constrained optimization in [11]. Such algorithmic design is effortless to execute and will be proved to be theoretically sound, in the sense that it still honors the law of large numbers. This way respects the physical intuition for the role of the adjoint solver (propagating information back to the origin of change) and also avoids the artificial mathematical difficulty in understanding the product of two Dirac delta measures. These features are utilized in the rigorous justification of the computation method.

The other approach falls in the discretize-then-optimize (DTO) framework [7]. It discretizes the forward PDE first and presents the PDE solution using MC particles. The objective function and the constraints are represented only by these particles. Therefore, the full system is already discrete, and the adjoint variables are particles with a one-to-one correspondence to the MC particles from the forward PDE particle solution [11]. The entirely discrete optimization problem removes the incompatibility issue mentioned above since the derived numerical scheme for the adjoint variables is always consistent with the forward discretization within this DTO framework. This MC gradient technique can be further applied to rejection sampling [32, 31, 41], which also applies to the MC method for the forward RTE. We will leave rigorous proof of this approach for future research.

Despite the differences, the OTD and DTO approaches designed in our work for computing the RTE gradient share some similar traits. First, both of them only need to simulate one set of particles for the forward RTE while the solution to the adjoint equation is obtained for free. Consequently, they have the same memory requirement. Second, the most expensive part of obtaining the gradient is computing the discrete integrals using disordered particles. These two approaches are equivalent under proper discretization schemes [19, 10, 11].

We now quickly summarize the equation and the setup in subsection 1.1. In section 2, we employ the OTD approach. We will review an MC solver for the RTE, present the failure of its direct extension to compute the gradient, and propose our fix. The associated rigorous numerical analysis is presented in section 3, including both the convergence of the MC solver for the forward RTE (see Algorithm 2.1) and the convergence of the gradient (see Algorithm 2.2). Section 4 is dedicated to the DTO approach, and Algorithm 4.1 will be developed for computing the RTE-constrained optimization gradient within the DTO framework. Numerical evidence is presented in section 5.

1.1. Equation and setup. RTE is a model problem for simulating light propagation in an optical environment [14]. For exposition simplicity, we restrict ourselves to the time-dependent RTE with no boundary effect,

$$\begin{cases} \partial_t f + v \cdot \nabla_x f &= \sigma(x)\mathcal{L}[f], \\ f(t=0,x,v) &= f_{\text{in}}(x,v), \end{cases} \qquad x \in \mathbb{R}^{d_x},\ v \in \Omega. \tag{1.1}$$

Here, $f(t,x,v)$ is the distribution function of photon particles at time t on the phase space (x,v). The left-hand side of the equation describes the photon moving in a straight line in x with velocity v, whereas the right-hand side characterizes the photon particles' interaction with the media characterized by the function $\sigma(x)$. The term $\mathcal{L}[f]$ is written as

$$\mathcal{L}[f] = \frac{1}{|\Omega|}\langle f\rangle_v - f, \quad \text{with} \quad \langle f\rangle_v = \int_\Omega f(x,v)\mathrm{d}v. \tag{1.2}$$

The term $\sigma(x)\mathcal{L}[f]$ represents that particles at location x have a probability proportional to $\sigma(x)$ to be scattered, into a new direction uniformly chosen in the v space. As a result, the distribution function in the phase space exhibits a gain on $|\Omega|^{-1}\langle f\rangle_v(x)$ and a loss on $f(x,v)$. Throughout the paper, we use $\langle\cdot\rangle_*$ to denote the integration with respect to the variable $*$, e.g., $\langle\cdot\rangle_v$, $\langle\cdot\rangle_{xv}$, and $\langle\cdot\rangle_{txv}$ with the Lebesgue measure.

The forward problem (1.1) has a unique solution under very mild conditions on both $\sigma(x)$ and the initial data f_{in}; see, for instance, [17, 4]. Moreover, the equation preserves mass because $\rho = \langle f\rangle_{xv}$ is a constant in time. Without loss of generality, we set $\rho = 1$ throughout the paper.

RTE is widely used as the forward model in inverse/control problems. One example of an inverse problem sits under the general umbrella of optical imaging [24, 37, 20]. In an experiment, one shines light into a domain with unknown optical properties and measures the light intensity that comes out of the domain. The measurements are then used to infer the optical property of the domain interior. This inverse problem is widely used in medical imaging and remote sensing, albeit the frequency of light is adjusted accordingly [39]. Another example in control theory using RTE as the forward model is lens design [29]. By adjusting the optical properties of the manufactured lens, light can be bent in the desired manner. In both cases, it is the optical parameter in the RTE that needs to be determined. We now examine the corresponding computational methods to solve such problems.

We frame our problem using the approach of PDE-constrained optimization. The objective function can be the mismatch of the simulated light intensity and the measured data, or the desired light output. The constraint comes from the fact that the solution needs to satisfy a forward RTE. Without loss of generality, we assume the measurement is a measurement operator $\mathcal{M}$ acting on the final-time solution of the RTE, and this generated data should be close to true reading:

$$d(x,v) \approx \mathcal{M}\left(f(t=T,x,v)\right).$$

Then the corresponding PDE-constrained optimization reads as

$$\min_\sigma J(\sigma), \quad \text{with } J(\sigma) = \frac{1}{2}\Big\langle \left|\mathcal{M}\left(f_\sigma(t=T,x,v)\right) - d(x,v)\right|^2\Big\rangle_{xv}, \tag{1.3}$$

where $f_\sigma(t,x,v)$ is the simulated data that solves (1.1) with the given initial condition and absorbing parameter $\sigma(x)$. We will write $f_\sigma(t,x,v)$ as $f(t,x,v)$ hereafter for simplicity of notation. The goal is to adjust σ so that the simulated data is as close to the true measured data $d(x,v)$ as possible. Here we use the standard L^2 norm to measure "closeness," but it shall be generalized to other metrics. In real-life problems, it is typical that the measurement is "intensity" ($\langle f\rangle_v$) instead of the solution profile f. Additionally, it is very typical to add a regularization term to mitigate the effects of noise and improve the convexity of the optimization problem. The derivation below can be extended easily to deal with all these variations.

The integration of MC methods and this constrained optimization problem will be presented from two perspectives in the rest of the paper. On the one hand, we can follow the OTD approach and derive the Fréchet derivative of J with respect to σ. This will come down to simulating two PDEs (one forward and one adjoint RTE), and the MC solver needs to be properly applied. On the other hand, we also take the DTO approach and start by reformulating (1.3) into a discrete form based on the MC solver. The gradient and optimization are then conducted entirely on this algebraic

system. These two pathways will be presented in sections 2 and 3, and section 4, respectively.

2. Optimize-then-discretize framework. The OTD framework is the most straightforward numerical strategy for solving PDE-constrained minimization problems. This strategy derives a Lagrangian to eliminate the constraints and adopts a minimization method, per the user's choice, to handle the resulting unconstrained optimization problem. This minimization strategy is performed directly on the PDE level, and the formulation is written in a continuous setting. When gradient-based optimization methods are used, one typically needs to compute the Fréchet derivatives for updates. Discretization is then performed to simulate the PDE and approximate the Fréchet derivative for the final execution of the algorithm. We detail the procedure below.

We first apply the method of Lagrange multipliers and transfer the original constrained optimization formulation (1.3) into an unconstrained one:

$$\min_{\sigma,f,g,\lambda} \mathfrak{L}(f,g,\lambda,\sigma),$$

where

$$\mathfrak{L}(f,g,\lambda,\sigma) = J(f) + \langle g, \partial_t f + v\cdot\nabla_x f - \sigma(x)\mathcal{L}[f]\rangle_{txv} + \langle \lambda, f(t=0,x,v) - f_{\text{in}}(x,v)\rangle_{xv}.$$

Here, functions $g(t,x,v)$ and $\lambda(x,v)$ are Lagrange multipliers with respect to the RTE solution $f(t,x,v)$ for $t>0$ and the initial condition $f(t=0,x,v)$, respectively. It is common to refer to $f(t,x,v)$ as the state variable and $g(t,x,v)$ as the adjoint variable in PDE-constrained optimizations. With integration by parts, we rewrite the Lagrangian as

$$\begin{aligned}\mathfrak{L}(f,g,\lambda,\sigma) = J(f) &+ \langle f, -\partial_t g - v\cdot\nabla_x g - \sigma(x)\mathcal{L}[g]\rangle_{txv} + \langle \lambda, f(t=0,x,v) - f_{\text{in}}(x,v)\rangle_{xv} \\ &+ \langle g(t=T,x,v), f(t=T,x,v)\rangle_{xv} - \langle g(t=0,x,v), f(t=0,x,v)\rangle_{xv}.\end{aligned}$$

Setting the variation of $\mathfrak{L}$ with respect to f to be 0, we obtain the adjoint equation

$$\begin{cases} -\partial_t g - v\cdot\nabla_x g &= \sigma(x)\mathcal{L}[g], \\ g(T,x,v) &= -\dfrac{\delta J}{\delta f(T,x,v)}, \end{cases} \tag{2.1}$$

where the final condition at T comes from the form of J given in (1.3). Then the Fréchet derivative of $\mathfrak{L}$ with respect to the function σ has the form

$$\begin{aligned}\mathfrak{G} &:= \frac{\mathrm{d}\mathfrak{L}}{\mathrm{d}\sigma} = \frac{\delta\mathfrak{L}}{\delta f}\frac{\mathrm{d}f}{\mathrm{d}\sigma} + \frac{\delta\mathfrak{L}}{\delta g}\frac{\mathrm{d}g}{\mathrm{d}\sigma} + \frac{\delta\mathfrak{L}}{\delta\lambda}\frac{\mathrm{d}\lambda}{\mathrm{d}\sigma} + \frac{\delta\mathfrak{L}}{\delta\sigma} \\ &= \frac{\delta\mathfrak{L}}{\delta\sigma}(x) = -\int_{v,t} f\mathcal{L}[g]\mathrm{d}v\mathrm{d}t = \int_t \langle fg\rangle_v \mathrm{d}t - \frac{1}{|\Omega|}\int_t \langle f\rangle_v\langle g\rangle_v \mathrm{d}t =: \mathfrak{G}_1 - |\Omega|^{-1}\mathfrak{G}_2,\end{aligned} \tag{2.2}$$

where the terms $\frac{\delta\mathfrak{L}}{\delta f}$, $\frac{\delta\mathfrak{L}}{\delta g}$, and $\frac{\delta\mathfrak{L}}{\delta\lambda}$ vanish since f and g solve (1.1) and (2.1), respectively. Here, $\mathfrak{G}_1 = \int_t \langle fg\rangle_v \mathrm{d}t$ and $\mathfrak{G}_2 = \int_t \langle f\rangle_v\langle g\rangle_v \mathrm{d}t$.

The derivation above is carried out completely on the function space in the continuous setting, as done in the OTD framework. Here, we endow the functional space for the parameter $\sigma(x)$ with the L^2 inner product, so the derivative $\frac{\delta\mathfrak{L}}{\delta\sigma}(x)$ is indeed the L^2 gradient. As such, we use "gradient" and "derivative" interchangeably. Upon obtaining (2.2), the next step is to find a discrete approximation to it. In particular, we represent the unknown parameter function $\sigma(x)$ using a finite-dimensional object,

such as a piecewise polynomial function on a compact domain, and replace f and g by their associated numerical representations. As a result, the numerical error purely comes from the computation of (1.1), (2.1), and the derivative-assembling (2.2).

A handful of numerical strategies can be used to solve the constraint PDE (1.1), such as finite difference, finite element, and spectral methods [27, 1, 28]. In this paper, we focus on the MC method and investigate the possibility of using it to compute the gradient (2.2).

2.1. Monte Carlo method for f. In this subsection, we first review the MC method [34, 18] used in solving the forward problem (1.1). It serves as the base of our computation. Though intuitive, the proof has not been thoroughly presented in the literature. We give a theoretical guarantee for the convergence of this method in this section.

Using MC to solve (1.1) amounts to representing the solution f as an ensemble of many particles:

$$f(t,x,v) \approx \frac{1}{N}\sum_{n=1}^{N} \delta\left(x - x_n(t)\right)\delta\left(v - v_n(t)\right) =: \mathsf{f}_N(t,x,v), \tag{2.3}$$

where $(x_n(t), v_n(t))$ denote the nth particle's location and velocity, respectively. Since (1.1) has a natural interpretation of particle interaction, it is straightforward to single out the particle dynamics. More precisely, we have the following:

- Term $v\cdot\nabla_x f$: This is a transport term suggesting that particles should move with velocity v, and therefore we set
$$\dot{x}_n = v_n.$$
- Term $\sigma(x)\mathcal{L}[f]$: This term indicates that, with intensity $\sigma(x)$, particles interact with the media and adopt a new velocity, which is uniformly chosen from the velocity domain Ω. This is reminiscent of the Poisson process in that for all $s-t<p\sim\mathrm{Pois}(e^{-\sigma(x_n)})$, $v_n(s)=v_n(t)$, and when $s=t+p$, v_n switches to
$$v_n(t+p)=\eta\sim\mathcal{U}(\Omega),$$
where $\mathcal{U}$ stands for a uniform distribution.

These understandings prompt the following formulation for Algorithm 2.1; see also [18].

Algorithm 2.1 Monte Carlo method for solving the forward RTE (1.1).

Preparation: N pairs of samples $\{(x_n^0, v_n^0)\}_{n=1}^N$, sampled from the initial distribution $f_{\mathrm{in}}(x,v)$; the total time steps M and the time interval Δt so that $T=M\Delta t$; and the parameter function $\sigma(x)$.
for $m=0$ to $M-1$ **do**
  Given $\{(x_n^m, v_n^m)\}_{n=1}^N$, set $x_n^{m+1}=x_n^m+\Delta t\, v_n^m$, $n=1,\dots,N$.
  Draw random numbers $\{p_n^{m+1}\}_{n=1}^N$ from the uniform distribution $\mathcal{U}([0,1])$.
  if $p_n^{m+1}\geq\alpha_n^{m+1}=\exp(-\sigma(x_n^{m+1})\Delta t)$ **then**
    Set $v_n^{m+1}=\eta_n^{m+1}$ where $\eta_n^{m+1}\sim\mathcal{U}(\Omega)$.
  else
    Set $v_n^{m+1}=v_n^m$.
  end if
end for

As written, all particles (x_n^m, v_n^m) are independent of each other, so we drop the subindex n, and only keep m representing the time step. From each time step to the next, two random variables are involved. One is the rejection sampling conducted through p^{m+1}. The other is the uniform sampling of η^{m+1}. We denote the expectation with respect to these two variables as $\mathbb{E}_{p^{m+1}}$ and $\mathbb{E}_{\eta^{m+1}}$, respectively.

We will prove that Algorithm 2.1 provides an accurate solver for (1.1) in the weak sense. In particular, consider the test function set

$$\Phi = C_c^\infty(\mathbb{R}^{d_x} \times \Omega). \tag{2.4}$$

We will show that the error, when tested against any $\phi \in \Phi$, is well controlled in the sense of both the expectation and the law of large numbers (LLN). More specifically, let

$$\mathsf{e}_{N,\phi}^m := \langle \mathsf{f}^m - f(t^m, x, v), \phi \rangle_{xv} = \frac{1}{N} \sum_{n=1}^N \phi(x_n^m, v_n^m) - \int f(t^m, x, v)\phi(x, v) \mathrm{d}x\mathrm{d}v \tag{2.5}$$

denote the error term. It can be viewed as a dual norm associated with the function space Φ, which metricizes the weak convergence. We will use the short-hand notation $f(t^m)$ to denote $f(t^m, x, v)$ hereafter. The quantity defined in (2.5) is related to the flat norm defined in [36, Example 8.8]. In principle, the fact that $\mathsf{e}_{N,\phi}^m$ converges to zero for all $\phi \in \Phi$ as $N \to \infty$ is equivalent to the weak convergence of f^m to $f(t^m)$.

For different realizations of $\{x_n^m, v_n^m\}$, the value $\mathsf{e}_{N,\phi}^m$ changes accordingly, and thus is a stochastic process on a probability space spanned by random variables $\{p^m, \eta^m\}_{m=1}^M$ and hence naturally generates a filtration. To stress the m dependence, we sometimes denote

$$\mathbb{E}^m = \begin{cases} \mathbb{E}_{(x^0, v^0)}, & m = 0, \\ \mathbb{E}_{(x^0, v^0)} \mathbb{E}_{\eta^1} \mathbb{E}_{p^1} \cdots \mathbb{E}_{\eta^m} \mathbb{E}_{p^m}, & m \geq 1, \end{cases}$$

as taking the expectation up to the filtration at time $t^m = m\Delta t$. In what follows, $\mathbb{E}^m[\mathsf{e}_{N,\phi}^m]$ and $\mathbb{E}[\mathsf{e}_{N,\phi}^m]$ are used interchangeably. We will prove the following theorem:

THEOREM 2.1. *Let* f *be produced by Algorithm* 2.1 *with* $T > 0$, $M \in \mathbb{N}$, *and* $\Delta t = T/M$. *Then for any* $\phi \in \Phi$, *the error* $\mathsf{e}_{N,\phi}^M$

- *is first order in time in expectation:*

$$\mathbb{E}\left|\mathsf{e}_{N,\phi}^M\right| = \mathcal{O}(\Delta t); \tag{2.6}$$

- *has an exponential concentration bound in* N. *Namely, for all* $\epsilon > 0$,

$$\mathbb{P}\left(|\mathsf{e}_{N,\phi}^M| \geq \epsilon + \Delta t\right) \lesssim \exp\left(-N\epsilon^2\right). \tag{2.7}$$

We have a few comments on Theorem 2.1. First, the notations $\mathcal{O}$ and $\lesssim$ hide a constant dependence. As expected, this constant depends on T, initial condition f_{in}, and ϕ, but we stress that it does not depend on M and Δt. We will make this constant dependence clear in the proofs of Propositions 3.2 and 3.5 in section 3, which are dedicated to explaining the two bullet points in Theorem 2.1 separately. Second, the conclusion above presents a strong contrast against traditional numerical methods such as finite difference or finite volume, where the Lax theorem requires both consistency and stability for the convergence, and the stability requirement usually poses conditions on Δt. The statement in our theorem holds true with Δt not experiencing

the extra stability issue. Third, the error is first order in time and $\mathcal{O}(1/\sqrt{N})$ in the number of particles. Note that the exponential form in (2.7) is consistent with the prediction of the LLN. Indeed, for a small δ,

$$\exp\left(-N\epsilon^2\right) \le \delta \quad \Longleftrightarrow \qquad \epsilon > \sqrt{|\log\delta|/N}.$$

Thus, the lower bound of ϵ honors the celebrated $\mathcal{O}(1/\sqrt{N})$ convergence rate for MC methods.

We stress that the error is quantified when tested against a test function ϕ. This is inevitable since f is a summation of delta measures, and the convergence has to be presented in the weak form.

2.2. Pitfall of a direct Monte Carlo for g. Due to the similarity between the forward (1.1) and adjoint (2.1) equations, it is natural to use the same MC method to obtain g. Indeed, letting $\tau = T - t$, (2.1) rewrites as

$$\partial_\tau g - v \cdot \nabla_x g = \sigma \mathcal{L}[g], \tag{2.8}$$

which is precisely the same as (1.1) except for the flip of the sign in velocity. Therefore, if we denote $\{x_n, v_n\}_{n=1}^N$ to be a list of N particles, then the same particle motion described in subsection 2.1 can be used here to produce a consistent algorithm. As before, we can define

$$\mathsf{g}_N(\tau, x, v) = \frac{1}{N}\sum_{n=1}^{N} \delta(x - x_n(\tau))\delta(v - v_n(\tau)).$$

As with f_N, we expect that g_N approximates $g(\tau, x, v)$ in the same way as in Theorem 2.1, namely, $\langle g - \mathsf{g}_N, \phi\rangle \sim 0$ in the $\Delta t \to 0$ and $N \to \infty$ limit.

Together, we have $\mathsf{g}_N \approx g$ and $\mathsf{f}_N \approx f$. It is tempting to compute $\mathfrak{G}$ by replacing f and g with their numerical approximations f_N and g_N, respectively. The discrete version of (2.2) may take the form

$$\mathfrak{G}_1 = \int \langle f\rangle_v \langle g\rangle_v \mathrm{d}t \approx \Delta t \sum_{m=1}^{M} \langle \mathsf{f}_N^m\rangle_v \langle \mathsf{g}_N^m\rangle_v, \quad \mathfrak{G}_2 = \int_t \langle fg\rangle_v \mathrm{d}t \approx \Delta t \sum_{m=1}^{M} \langle \mathsf{f}_N^m \mathsf{g}_N^m\rangle_v.$$

However, this immediately yields a problem since both f_N and g_N are defined as delta measures. The definitions of both $\langle \mathsf{f}_N^m\rangle_v \langle \mathsf{g}_N^m\rangle_v$ and $\langle \mathsf{f}_N^m \mathsf{g}_N^m\rangle_v$ represent multiplications of two delta measures and thus cannot be justified mathematically. Indeed, according to Theorem 2.1, the convergence of $\mathsf{f}_N \to f$ and similarly $\mathsf{g}_N \to g$ is achieved only in the weak sense. Then the product of two weak limits loses its precise definition. This suggests that naively computing the forward and adjoint equations using the standard MC solvers for assembling the gradient in (2.2) cannot produce an accurate numerical approximation to the gradient.

We address the fact that this incompatibility of MC solvers with the Fréchet derivative computation is not a mathematical artifact but is rooted in the physical meaning of forward and adjoint equations. At the core of computing the Fréchet derivative, the forward solver delivers the initial data to the final time by picking up the media information along the evolution. The Fréchet derivative captures the dependence of the final data on the media $\sigma(x)$ where the forward trajectories have visited. A small perturbation in $\sigma(x)$ will change the final data accordingly. Adjoint solvers allow us to trace back this change in the final data to the perturbation in media. The inconsistency problem of the MC solver comes from the fact that the forward

solver and the adjoint solver use independent trajectories. While the forward solver delivers the change of media along certain trajectories to the change of the final data, the adjoint solver traces back such change along a totally different set of trajectories. There is a chance to conduct computation if the two sets of trajectories are close, but these events occur with an extremely small probability due to the independent nature of the trajectory generation process.

We believe there are numerical strategies to resolve this issue, for example, approximating Dirac delta functions using Gaussian kernels with small support. In this specific context, this amounts to running MC for both the forward RTE f and the adjoint RTE g and then performing density estimation using basis functions such as polynomial functions or Gaussian kernels to smooth them out for the integration. By doing so, the computational cost mainly comes from density estimation and numerical integration. We should point out that even though numerical integration only requires a linear order complexity, the density estimation can be computationally prohibitive. For piecewise polynomial interpolation, for example, the sample complexity is $N \sim \mathsf{N}d/\epsilon^2$ and the computational complexity is $\mathsf{poly}(\mathsf{N}, d, 1/\epsilon)$, where N and d are the number of grids and the degree of polynomial, respectively [13]. Given that the grid number exponentially depends on the dimension, this cost is substantially higher than what a classical MC solver requires. Moreover, according to the derivative above, such density estimation needs to be done at *every time step*. In a nutshell, the approach of integrating MC with a grid-based method, if not deployed well, garners the worse sides of both: it loses accuracy due to the implementation of MC, and it suffers from the curse of dimensionality due to the grid requirements; see reduced variance techniques or sparse grid methods employed [40, 3] on this front. Such cost can be completely avoided if we sample g at the final time in a smart way, and this leads us to the discussion in the upcoming section.

2.3. Gradient calculation revisited. In this work, we develop a new method for (2.1) to avoid ambiguity in defining the product of two delta measures. To begin with, we examine the relation between the adjoint and forward variables. One observation is that the adjoint variable is designed to have certain quantities preserved in time:

$$\partial_t \langle fg \rangle_{xv} \equiv 0, \tag{2.9}$$

which can be justified, in our particular setting, by comparing (1.1) multiplied by g and (2.1) by f.

This relation, when replacing f by f_N, becomes

$$\sum_{n=1}^{N} g(t^m, x_n^m, v_n^m) = \sum_{n=1}^{N} g(t^{m+1}, x_n^{m+1}, v_n^{m+1}),$$

which can be easily satisfied if we require

$$g(t^m, x_n^m, v_n^m) = g(t^{m+1}, x_n^{m+1}, v_n^{m+1}), \quad m = M-1, \dots, 0,$$

with the final condition set to be

$$g(T, x_n^M, v_n^M) = \psi(x_n^M, v_n^M) = -\frac{\delta J}{\delta f}(T, x_n^M, v_n^M). \tag{2.10}$$

As a result, along every particle's trajectory of $\{(x_n^m, v_n^m)\}_{m=0}^M$, g takes a constant value. In other words, if we denote $\mathsf{g}_N^m(x, v)$ as the numerical solution approximating

$g(t^m, x, v)$ using N particles, then we have scattered value of $\mathsf{g}_N^m(x,v)$ at the particle locations along the dynamics:

$$\mathsf{g}_N^m(x,v)\big|_{(x_n^m, v_n^m)} = \mathsf{g}_N^m(x_n^m, v_n^m) = \mathsf{g}_n := \psi(x_n^M, v_n^M) \tag{2.11}$$

for all $0 \le m \le M$ and $1 \le n \le N$. For $(x,v) \notin \{(x_n^M, v_n^M)\}_{n=1}^N$, we may define the value of $\mathsf{g}_N^m(x,v)$ through interpolation from $\{\mathsf{g}_N^m(x_n^m, v_n^m)\}_{n=1}^N = \{\mathsf{g}_n\}_{n=1}^N$.

We call this approach a *correlated* approach, due to the fact that g and f are not fully independent, as discussed in subsection 2.2.

Remark 1. We should emphasize that the algorithm presented in (2.11) is derived merely from the relation (2.9), which is *not* equivalent to the original adjoint equation (2.1). Indeed, starting from (2.9), one can only tell that g satisfies

$$\langle \partial_t g - v \cdot \nabla_x g - \sigma \mathcal{L} g, f \rangle_{xv} = 0.$$

That is, the adjoint equation holds only when projected on f, the solution manifold to the forward equation (1.1). This is different from saying that g is a weak solution to (2.1), which requires the above equation to hold for any test function in Φ instead of f alone. Nevertheless, the solver (2.11) indeed provides an approximation to (2.1), and the proof is omitted from the current paper.

With the numerical solution g in hand, we now proceed to find numerical approximation to the Fréchet derivative (2.2). Based on the weak formulation of f and that f is presented discrete-in-time, the Fréchet derivative is written in its weak semidiscrete-in-time form as well. After testing (2.2) against a test function $\phi(x)$ and conducting the simple Riemann sum in time, we have

$$\begin{aligned} \mathfrak{G}_\phi := \left\langle \frac{\delta \mathfrak{L}}{\delta \sigma}, \phi \right\rangle_x &= \int \langle f(t^m,\cdot,\cdot) g(t^m,\cdot,\cdot),\, \phi(\cdot)\rangle_{xv} \mathrm{d}t \\ &\quad - |\Omega|^{-1} \int \left\langle \langle f(t^m,\cdot,\cdot)\rangle_v \langle g(t^m,\cdot,\cdot)\rangle_v,\, \phi(\cdot) \right\rangle_x \mathrm{d}t \\ &\approx \Delta t \sum_{m=1}^M \left[\mathfrak{G}_{1,\phi}^m - |\Omega|^{-1} \mathfrak{G}_{2,\phi}^m \right], \end{aligned} \tag{2.12}$$

where we used the notation

$$\mathfrak{G}_{1,\phi}^m := \langle f(t^m,\cdot,\cdot) g(t^m,\cdot,\cdot),\, \phi(\cdot)\rangle_{xv}, \qquad \mathfrak{G}_{2,\phi}^m := \left\langle \langle f(t^m,\cdot,\cdot)\rangle_v \langle g(t^m,\cdot,\cdot)\rangle_v,\, \phi(\cdot) \right\rangle_x. \tag{2.13}$$

At every discrete time t^m, according to Algorithm 2.2, f_N^m approximates $f(t^m,\cdot,\cdot)$ using N particles, and g_N^m records the value of $g(t^m,\cdot,\cdot)$ on these particle trajectories. As a consequence, the discrete version of (2.13) is written as

$$\mathfrak{G}_{N,1,\phi}^m = \frac{1}{N} \sum_{n=1}^N \phi(x_n^m)\, \mathsf{g}_n, \quad \mathfrak{G}_{N,2,\phi}^m = \frac{1}{N} \sum_{n=1}^N \phi(x_n^m)\, \langle \mathsf{g} \rangle_v(x_n^m), \tag{2.14}$$

and thereby, the final discrete Fréchet derivative, when tested on ϕ, takes the following form as the N-particle approximation to (2.12):

$$\mathfrak{G}_{N,\phi} := \left\langle \frac{\delta \mathfrak{L}^N}{\delta \sigma}, \phi \right\rangle_x = \Delta t \sum_{m=1}^M \left(\mathfrak{G}_{N,1,\phi}^m - |\Omega|^{-1} \mathfrak{G}_{N,2,\phi}^m \right). \tag{2.15}$$

In the implementation, we want the derivative to be evaluated on a given mesh. To do so, for every $\bar{x}$ on the mesh, we denote by Δx the mesh size and by $Q(\bar{x}, \Delta x)$ the hypercube centered at $\bar{x}$ with side length Δx, and set $\phi = \frac{1}{|\Delta x|^d}\mathbb{1}_{Q(\bar{x},\Delta x)}$, the corresponding indicator function. The approximated derivative at a particular mesh point $\bar{x}$ then becomes

$$\mathfrak{G}_N(\bar{x}) = \frac{\delta \mathfrak{L}^N}{\delta \sigma}(\bar{x}) = \Delta t \sum_{m=1}^{N} \Big(\langle \mathsf{f}_N^m \mathsf{g}_N^m \rangle_v(\bar{x}) - |\Omega|^{-1} \langle \mathsf{f}_N^m \rangle_v(\bar{x}) \langle \mathsf{g}_N^m \rangle_v(\bar{x}) \Big), \tag{2.16}$$

with each term computed as

$$\langle \mathsf{f}_N^m \rangle_v(\bar{x}) = \frac{1}{|\Delta x|^d} \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}_{x_n^m \in Q(\bar{x};\Delta x)}, \tag{2.17}$$

$$\langle \mathsf{g}_N^m \rangle_v(\bar{x}) = \sum_{x_{n_1}^m \in Q(\bar{x};\Delta x)} w_{n_1}^m \, \mathsf{g}_{n_1}, \tag{2.18}$$

$$\langle \mathsf{f}_N^m \mathsf{g}_N^m \rangle_v(\bar{x}) = \frac{1}{|\Delta x|^d} \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}_{x_n^m \in Q(\bar{x};\Delta x)} \, \mathsf{g}_n. \tag{2.19}$$

Here, $\{w_{n_1}^m\}$ are the quadrature weights in the v domain such that

$$\begin{aligned} \sum_{x_{n_1}^m \in Q(\bar{x};\Delta x)} w_{n_1}^m \, \mathsf{g}_{n_1} &= \sum_{x_{n_1}^m \in Q(\bar{x};\Delta x)} w_{n_1}^m \mathsf{g}_N^n(x_{n_1}^m, v_{n_1}^m) \\ &\approx \sum_{x_{n_1}^m \in Q(\bar{x};\Delta x)} w_{n_1}^m g(t^m, \bar{x}, v_{n_1}^m) \approx \int g(t^m, \bar{x}, v) \mathrm{d}v. \end{aligned}$$

In particular, out of three calculations listed in (2.17)–(2.19), (2.18) is the most expensive part. The reason is that, at each time step, one needs to loop over space to sort the particles, which results in $\mathcal{O}(NM(\log N))$ total complexity with N being the total number of particles and M the total time steps.

Remark 2. Above, we compute the gradient by convolving f with a test function ϕ centered at $\bar{x}$, the point of gradient evaluation. We choose ϕ as a characteristic function, so that the resulting gradient $\mathfrak{G}_N(\bar{x})$ is piecewise constant with respect to $\bar{x}$. However, if we know a priori that either $\sigma(x)$ or the true gradient has better regularity, we can use a test function ϕ with the same regularity to enforce it in the computed gradient $\mathfrak{G}_N(\bar{x})$ as a function of $\bar{x}$.

In Algorithm 2.2, we summarize the steps of computing the gradient using the correlated approach.

We note that the approximation to the Fréchet derivative (2.16) contains four layers of error:

- Time discretization. $\Delta t \sum_{m=1}^{M}$ is used as a replacement of $\int \mathrm{d}t$. This is the simplest Riemann sum for the time integration, and we expect the error at the order of Δt. Throughout our analysis, this part of the error is omitted as we regard the following semidiscrete in-time Fréchet derivative as the ground truth:

$$\frac{\delta \mathfrak{L}}{\delta \sigma}(x) = \Delta t \sum_{m=1}^{M} \Big(\langle f g(t^m, x, \cdot) \rangle_v - |\Omega|^{-1} \langle f(t^m, x, \cdot) \rangle_v \langle g(t^m, x, \cdot) \rangle_v \Big). \tag{2.20}$$

  It is the strong form of (2.12).

Algorithm 2.2 A correlated approach for gradient computation.

1: Given cell centers $\{\bar{x}_j\}$ where we want to evaluate the gradient (2.2).
2: Implement Algorithm 2.1 to solve the forward RTE (1.1) and store the trajectories $\{(x_n^m, v_n^m)\}$ for all $n = 0, \dots, M$ and $n = 1, \dots, N$.
3: Use (1.3) and (2.10) to determine the final condition for g and set g_n following (2.11) for all $1 \le n \le N$.
4: **for** $m = 1$ to M **do**
5: Use the stored $\{(x_n^m, v_n^m)\}_{n=1}^N$ and $\{\mathsf{g}_n\}_{n=1}^N$ to compute $\langle \mathsf{f}_N^m \rangle_v(\bar{x}_j)$, $\langle \mathsf{g}_N^m \rangle_v(\bar{x}_j)$, and $\langle \mathsf{f}_N^m \mathsf{g}_N^m \rangle_v(\bar{x}_j)$ following (2.17), (2.18), and (2.19), respectively, for every element in $\{\bar{x}_j\}$.
6: **end for**
7: The gradient evaluated at $\bar{x}_j$ is computed following (2.16).

- Spatial discretization. This is used in the final assembling of the gradient (2.16) where for every fixed $\bar{x}$, a small hypercube of volume Δx^d is drawn, and values are averaged out inside the cube. It naturally brings a smoothing effect. Such smoothing effects are heavily studied in the literature, and we only cite [38] for details.
- Quadrature error in the velocity domain. This enters through the term (2.18) to approximate the velocity domain integration $\langle \cdot \rangle_v$. The error from this term highly depends on the quadrature rule selected for defining $\{w_{n_1}^m\}$. With the simplest trapezoidal rule, we expect the error to be of order $\mathcal{O}(\Delta v^2)$, where Δv is the largest discrepancy between two sampled particles in velocity in the same $\bar{x}$-neighborhood. Throughout the paper, we denote this part of the error by e_v. We comment that this term inherits the randomness from MC sampling and requires some delicate analysis. This goes beyond the current scope of the paper and will be left to future study.
- Monte Carlo error. This comes from the fact that random particles represent the PDE solution. Theorem 2.1 states that f_N^m is a good approximation, with LLN ($\mathcal{O}(1/\sqrt{N})$) convergence rate, to the ground truth $f(t^m)$. A statement of similar flavor shall be provided for the gradient.

Among these four types of errors, the second and third kinds start to matter only when the error is presented in the strong form. If the error is studied in the weak form, one only needs to analyze the first and fourth kinds. Since the time discretization is a standard Riemann sum error, the focus of the paper is placed on the fourth: the MC error. The estimates we obtain justify that in the weak sense, Algorithm 2.2 indeed provides, with high confidence, an accurate approximation to the semidiscrete-in-time ground-truth gradient. The rigorous statement is summarized as follows:

THEOREM 2.2. *For all $\phi \in \Phi$, $\mathfrak{G}_{N,\phi}$ defined in* (2.15) *approximates the true derivative $\mathfrak{G}_\phi$ in its semidiscrete form* (2.12) *with high probability. Namely, for any $\epsilon > 0$,*

$$\mathbb{P}(|\mathfrak{G}_{N,\phi} - \mathfrak{G}_\phi| > \mathcal{O}(\epsilon + \Delta t + |e_v|)) \le 2 \exp\left(-\frac{N\epsilon^2}{3C_3}\right), \tag{2.21}$$

with C_3 only depending on the regularity of the initial condition in (1.1)*, the final condition in* (2.1)*, and the test function; see* (3.22).

This theorem is the theoretical justification of Algorithm 2.2. It states that the numerical Fréchet derivative, defined using a summation of delta measures, approximates the true gradient in the weak sense, with a probability that depends on the number of particles in the algorithm. Involving more particles guarantees a higher probability of capturing the gradient with better precision. The formula (2.21) gives a precise quantification of the decay of possibilities, in N, of getting outliers. Similar to Theorem 2.1, this convergence rate is in line with the LLN prediction and obtains a $\frac{1}{\sqrt{N}}$ decay. Since this is the rate of the MC solver for the forward problem (see Theorem 2.1), it is the best one can hope for in the computation of the gradient.

3. Numerical analysis. We hereby provide the justification to Theorems 2.1 and 2.2 regarding the computation of f in (1.1) and the Fréchet derivative (2.2) calculation.

3.1. Convergence of the Monte Carlo method for the forward RTE. In Algorithm 2.1, we have presented the MC solver for computing (1.1). From each time step to the next, two random variables are involved. One performs the rejection sampling, and the other is the uniform sampling for the new velocity direction. The current section is dedicated to proving Theorem 2.1 that gives both an expectation and concentration bound for the error $\mathsf{e}^m_{N,\phi}$.

Based on the definition in (2.5), $\mathsf{e}^m_{N,\phi} = \langle \mathsf{f}^m_N - f(t^m), \phi\rangle_{x,v}$ where $\mathsf{f}_N = \frac{1}{N}\sum_n \delta(x - x^m_n)\delta(v - v^m_n)$. Since $\{(x^m_n, v^m_n)\}_{n=1}^N$ are independent of each other for any fixed m, we have

$$\mathsf{e}^m_{N,\phi} = \frac{1}{N}\sum_{n=1}^{N} \mathsf{e}^m_{1,\phi}(n), \tag{3.1}$$

where $\mathsf{e}^m_{1,\phi}(n)$ is the nth realization of $\mathsf{e}^m_{1,\phi}$. This then implies that at the expectation level, $\mathbb{E}[\mathsf{e}^m_{1,\phi}] = \mathbb{E}[\mathsf{e}^m_{N,\phi}]$. At the variance level, the concentration inequality shall be applied. These arguments clearly suggest the following roadmap:

- We will first prove that each particle, in expectation, solves the RTE (1.1). This is to say that, if $N = 1$ and we accordingly define

  $$\mathsf{f}^m_1(x,v) := \delta(x - x^m)\delta(v - v^m), \tag{3.2}$$

  then $\mathsf{f}^m_1(x,v)$ numerically solves (1.1) in the weak sense in expectation, up to a discretization error in Δt. More precisely, for any $\phi \in \Phi$,

  $$\mathbb{E}^m[\mathsf{e}^m_{1,\phi}] = \mathbb{E}^m\left[\langle \mathsf{f}^m_1 - f(t^m), \phi\rangle_{x,v}\right] = \mathcal{O}(m\Delta t^2).$$

  The analysis is presented in subsection 3.1.1, and it concludes the first part of Theorem 2.1.
- We then level the calculation up to the N-particle system using (3.1), calling the Bernstein inequality. To directly utilize the concentration bound, we will control the variance of $\mathsf{e}^m_{1,\phi}$. This is to be discussed in subsection 3.1.2 and it concludes the second part of Theorem 2.1.

3.1.1. Single particle. We prove that each particle sampled according to Algorithm 2.1, in expectation, traces the evolution of f in the weak sense. For any $\phi \in \Phi$, we recall the definition of the error in (2.5) and denote the expected value to be

$$e^m_{1,\phi} = \mathbb{E}^m\left[\mathsf{e}^m_{1,\phi}\right] = \mathbb{E}^m\left[\langle \mathsf{f}^m_1 - f(t^m), \phi\rangle_{x,v}\right], \tag{3.3}$$

where the subindex 1 in e and e reflects that there is only one particle. This index will be omitted in the remainder of section 3.

We will track the growth of e_ϕ^m with respect to m and prove that in each time step, the growth is controlled by $\mathcal{O}(\Delta t^2)$. Thus, the whole scheme is first-order accurate in time. In the proof of Lemma 3.1, we use several positive constants denoted by C_i, $i = 1, 2, \ldots$. We will make explicit the constants' dependence on different parameters but do not spell out the specific dependence. Their exact values may change from line to line.

LEMMA 3.1. *Let $\sigma(x) \in W^{1,\infty}(\mathbb{R}^{d_x})$ and Ω be a bounded velocity domain. Denote by $\{(x^m, v^m)\}$ the solution of the particle trajectory through Algorithm* 2.1. *Then f_1 defined in* (3.2) *approximates f, the solution to* (1.1), *in the expectation sense. More specifically, for any $\phi \in \Phi$ and any $m \geq 0$, we have*

$$e_\phi^{m+1} \leq e^m_{(I+\Delta t\mathcal{P})\phi} + C_1 \Delta t^2, \quad \mathcal{P}\phi := v \cdot \nabla_x \phi + \sigma \mathcal{L}[\phi], \tag{3.4}$$

where the positive constant

$$C_1 = C_1(\|f_{\text{in}}\|_{W^{2,\infty}_x L^\infty_v}, \|v\|_{L^\infty(\Omega)}, \|\sigma\|_{W^{1,\infty}_x}, \|\phi\|_{W^{2,\infty}_{x,v}}, T). \tag{3.5}$$

Moreover, $e_\phi^0 = 0$, and we have

$$|e_\phi^m| \leq m C_1 \Delta t^2.$$

Therefore, $e_\phi^m \to 0$ in the limit of $\Delta t \to 0$ for every fixed $1 \leq m \leq M$.

Proof. According to the definition,

$$\mathsf{e}_\phi^m = \langle \mathsf{f}_1^m - f(t^m), \phi \rangle_{xv} = \phi(x^m, v^m) - \langle f(t^m), \phi \rangle_{xv}.$$

To study e_ϕ^m, we need to track the evolution of $\mathsf{e}_\phi^{m+1} - \mathsf{e}_\phi^m$, which amounts to evaluating $\phi(x^{m+1}, v^{m+1}) - \phi(x^m, v^m)$ and $\langle f(t^{m+1}) - f(t^m), \phi \rangle_{xv}$.

The term $\phi(x^{m+1}, v^{m+1}) - \phi(x^m, v^m)$ can be expanded as

$$\begin{aligned}
&\phi(x^{m+1}, v^{m+1}) - \phi(x^m, v^m) \\
&\quad = \phi(x^{m+1}, v^{m+1}) - \phi(x^{m+1}, v^m) + \phi(x^{m+1}, v^m) - \phi(x^m, v^m) \\
&\quad = \phi(x^{m+1}, v^{m+1}) - \phi(x^{m+1}, v^m) + \Delta t\, v^m \cdot \nabla_x \phi(x^m, v^m) + \frac{\Delta t^2}{2} (v^m)^\top \nabla_x^2 \phi(\xi_1) v^m,
\end{aligned} \tag{3.6}$$

where $\nabla_x^2 \phi$ is the Hessian of ϕ with respect to the x direction, and ξ_1 is between x^m and x^{m+1}. Note that the event $v^{m+1} = \eta \neq v^m$ occurs with probability $1 - e^{-\sigma(x^{m+1})\Delta t}$. Thus,

$$\begin{aligned}
&\mathbb{E}_{\eta^{m+1}} \mathbb{E}_{p^{m+1}} \left[\phi\left(x^{m+1}, v^{m+1}\right) - \phi\left(x^{m+1}, v^m\right) \right] \\
&\quad = \left(1 - e^{-\sigma(x^{m+1})\Delta t}\right) \mathcal{L}[\phi]\left(x^{m+1}, v^m\right) \\
&\quad = \Delta t \left(\sigma \mathcal{L}[\phi]\right)\big|_{(x^m, v^m)} + \Delta t^2 \left(\nabla_x \left(\sigma \mathcal{L}[\phi]\right) \cdot v\right)\big|_{(\xi_2, v^m)},
\end{aligned} \tag{3.7}$$

where ξ_2 is again between x^m and x^{m+1}.

Consequently, the above equation (3.7), along with (3.6), leads to

$$\begin{aligned}
&\mathbb{E}^{m+1} \left[\phi(x^{m+1}, v^{m+1}) - \phi(x^m, v^m)\right] \\
&\quad \leq \Delta t\, \mathbb{E}^m \left[\mathcal{P}\phi(x^m, v^m)\right] + \Delta t^2 \left(\mathfrak{c}_1 \|\nabla_x \left(\sigma \mathcal{L}[\phi]\right)\|_\infty + \mathfrak{c}_2 \|\nabla_x^2 \phi\|_\infty\right),
\end{aligned} \tag{3.8}$$

where $\mathfrak{c}_1 = 2\|v\|_{L^\infty(\Omega)}$, $\mathfrak{c}_2 = \|v\|^2_{L^\infty(\Omega)}$, and the operator $\mathcal{P}$ is defined in (3.4).

To study $\langle f(t^{m+1}) - f(t^m), \phi\rangle_{xv}$, we employ (1.1):

$$\frac{f(t^{m+1}) - f(t^m)}{\Delta t} = \sigma\,\mathcal{L}[f](t^m) + v\cdot\nabla_x f(t^m) + \frac{\Delta t}{2}\partial_{tt} f(\xi_t) \qquad \text{for} \quad \xi_t \in (t^m, t^{m+1}).$$

Since $\partial_{tt} f$ solves the same equation (1.1) with initial condition

$$\partial_{tt} f(0,x,v) = \sigma\mathcal{L}[\sigma\mathcal{L}[f_{\text{in}}] - v\cdot\nabla_x f_{\text{in}}] - v\cdot\nabla_x(\sigma\mathcal{L}[f_{\text{in}}] - v\cdot\nabla_x f_{\text{in}}),$$

it follows that

$$\|\partial_{tt} f(t,\cdot,\cdot)\|_{L^1_{x,v}} \le C\|f_{\text{in}}\|_{W^{2,\infty}_x L^\infty_v} := \mathfrak{c}_3, \tag{3.9}$$

where C and $\mathfrak{c}_3$ are constants. Thus, we have

$$\langle f(t^{m+1}), \phi\rangle_{xv} \ge \langle f(t^m), \phi\rangle_{xv} + \Delta t\,\langle f(t^m), \mathcal{P}\phi\rangle_{xv} - \Delta t^2\,\mathfrak{c}_3\|\phi\|_\infty. \tag{3.10}$$

Combining (3.10) with (3.8), we obtain that for any $n \ge 0$,

$$e_\phi^{m+1} \le \quad e^m_{(I+\Delta t\mathcal{P})\phi} + \Delta t^2\left(\mathfrak{c}_1\|\mathcal{T}_1\phi\|_\infty + \mathfrak{c}_2\|\mathcal{T}_2\phi\|_\infty + \mathfrak{c}_3\|\mathcal{T}_3\phi\|_\infty\right), \tag{3.11}$$

where the operators $\{\mathcal{T}_i\}$ are defined as

$$\mathcal{T}_1\phi := \nabla_x\left(\sigma\mathcal{L}[\phi]\right), \quad \mathcal{T}_2\phi := \nabla_x^2\phi, \quad \mathcal{T}_3\phi := \phi. \tag{3.12}$$

A key observation is that $\mathcal{T}_i(\mathcal{P}\phi) = \mathcal{P}(\mathcal{T}_i\phi)$, $1 \le i \le 3$.

Since (3.11) applies to any $m = 0, 1, 2, \ldots, M-1$, we have that

$$\begin{aligned} e_\phi^{m+1} &\le e^m_{(I+\Delta t\mathcal{P})\phi} + \Delta t^2 \sum_{i=1}^3 \mathfrak{c}_i\|\mathcal{T}_i\phi\|_\infty \\ &\le e^{m-1}_{(I+\Delta t\mathcal{P})^2\phi} + \Delta t^2 \sum_{i=1}^3 \mathfrak{c}_i\left(\|\mathcal{T}_i\phi\|_\infty + \|(I+\Delta t\mathcal{P})(\mathcal{T}_i\phi)\|_\infty\right) \\ &\le e^{m+1-k}_{(I+\Delta t\mathcal{P})^k\phi} + \Delta t^2 \sum_{i=1}^3 \mathfrak{c}_i \sum_{q=0}^{k-1}\|(I+\Delta t\mathcal{P})^q(\mathcal{T}_i\phi)\|_\infty \qquad \forall k = 1,\ldots,m+1. \end{aligned} \tag{3.13}$$

It is straightforward to prove (3.13) by induction. For $0 \le q \le m+1$, we have

$$e^0_{(I+\Delta t\mathcal{P})^q\phi} = \mathbb{E}^0[((I+\Delta t\mathcal{P})^q\phi)(x^0,v^0)] - \int f(0,x,v)\,((I+\Delta t\mathcal{P})^q\phi)(x,v)\mathrm{d}x\mathrm{d}v = 0.$$

Moreover, when Δt is small,

$$(I+\Delta t\mathcal{P})^q(\mathcal{T}_i\phi) \approx \psi_i(t = q\Delta t),$$

where ψ_i solves

$$\begin{cases} \partial_t\psi_i(t,x,v) &= \mathcal{P}\psi(x,v), \\ \psi_i(t=0,x,v) &= \mathcal{T}_i\phi(x,v) \end{cases} \tag{3.14}$$

for $i = 1,2,3$. Note that (3.14) is the same as (2.8), which is the regular RTE with the velocity sign reversed. Thus, (3.14) enjoys all the properties of the forward RTE (1.1). We then have the maximum principle [6], and thus

$$\|(I+\Delta t\mathcal{P})^q(\mathcal{T}_i\phi)\|_\infty \le \|\mathcal{T}_i\phi\|_\infty \quad \forall q \ge 0,$$

and hence

$$\Delta t^2 \sum_{i=1}^{3} \mathfrak{c}_i \sum_{q=0}^{k-1} \|(I+\Delta t\mathcal{P})^q(\mathcal{T}_i\phi)\|_\infty \le k\Delta t^2 \sum_{i=1}^{3} \mathfrak{c}_i \|\mathcal{T}_i\phi\|_\infty.$$

Therefore, using (3.13) for $k=m+1$, we have

$$e_\phi^{m+1} \le (m+1)\Delta t^2 \sum_{i=1}^{3} \mathfrak{c}_i \|\mathcal{T}_i\phi\|_\infty \le (m+1)C_1\Delta t^2,$$

where C_1 is defined in (3.5). The same analysis applies to $e_{-\phi}^{m+1}$ with precisely the same upper bound $(m+1)C_1\Delta t^2$. Since $e_{-\phi}^{m+1} = -e_\phi^{m+1}$, we have $|e_\phi^{m+1}| < (m+1)C_1\Delta t^2$. ∎

Inductively, Lemma 3.1 allows us to arrive at Proposition 3.2. It is essentially the first bullet point of Theorem 2.1, and we spell out the constant dependence.

PROPOSITION 3.2. *Given a fixed time $T \ge 0$, $M \in \mathbb{N}$, and $\Delta t = T/M$, for any $\phi \in \Phi$, we have $\left|e_\phi^M\right| \le C_1\Delta t$, where e_ϕ^M is defined in* (3.3) *and C_1 is defined in* (3.5).

Proof. Recall that $\mathbb{E}^M$ includes all the randomness along the trajectory to obtain the final-time particle (x^M, v^M). Based on Lemma 3.1, we obtain an upper bound for e_ϕ^M by setting $m=M$ and $T=M\Delta t$. ∎

We should note that though Proposition 3.2 is formulated for a one-particle system, due to the fact that $\mathsf{e}_{N,\phi}^M$ is the simple N-average of $\mathsf{e}_{1,\phi}^M$, the mean is preserved, and the extension to the many-particle system is trivial. This concludes (2.6) in Theorem 2.1.

3.1.2. Many particles. Using formula (3.1), we view $\mathsf{e}_{N,\phi}^m$ as the average of many independent and identically distributed (i.i.d.) samples drawn according to the distribution of $\mathsf{e}_{1,\phi}^m$. Thus, we can directly apply the concentration tail bound. To do so, we first cite the famous Bernstein inequality.

THEOREM 3.3 (Bernstein inequality). *Let $x_1, \dots, x_N$ be i.i.d. real-valued samples of a random variable X, whose expectation and variance are $\mu = \mathbb{E}[X]$ and $\sigma^2 = \mathrm{Var}[X]$. Assume that there exists b such that $|X-\mu| \le b$ almost surely. Then for any $t>0$, we have*

$$\mathbb{P}\left(\frac{1}{N}\sum_{n=1}^{N} x_n - \mu \ge t\right) \le \exp\left(-\frac{Nt^2}{2\left(\sigma^2 + \frac{1}{3}bt\right)}\right). \tag{3.15}$$

According to (3.15), to control the tail of $\mathsf{e}_{N,\phi}^m$, we need to give an estimate on the variance (σ^2 above) and the range (b above). These bounds are presented in the following lemma.

LEMMA 3.4. *Let $\mathsf{e}_{1,\phi}^m$ be defined in* (2.5) *with f_1^m computed through Algorithm* 2.1. *Then $\mathsf{e}_{1,\phi}^m$ has a bounded range and bounded variance. In particular, we have the following, for all $m \le M$ and $\phi \in \Phi$:*

- *The range of the error term is bounded:*

$$|\mathsf{e}_{1,\phi}^m| \le 2\|\phi\|_{L^\infty_{x,v}}. \tag{3.16}$$

- *The variance is bounded:*

$$\mathrm{Var}[\mathsf{e}_{1,\phi}^m] \le C_2, \tag{3.17}$$

where C_2 depends on $\|v\|_{L^\infty(\Omega)}$, $\|\sigma\|_{L^\infty}$, T, and $\|\phi\|_{W^{1,\infty}_{x,v}}$.

Proof. To show (3.16), recall

$$\mathsf{e}_{1,\phi}^m = \phi(x^m, v^m) - \int f(t^m)\phi(x,v)\mathrm{d}x\mathrm{d}v.$$

Then we have

$$\begin{aligned}\left|\mathsf{e}_{1,\phi}^m\right| &\le \left|\phi(x^m,v^m)\right| + \left|\int f(t^m)\phi(x,v)\mathrm{d}x\mathrm{d}v\right| \\ &\le \|\phi\|_{L^\infty_{x,v}} + \|\phi\|_{L^\infty_{x,v}}\int f(t^m,x,v)\mathrm{d}x\mathrm{d}v = 2\|\phi\|_{L^\infty_{x,v}},\end{aligned}$$

where we have used the fact that $f(t^m)$ is nonnegative and integrates to one thanks to the mass conservation.

Since we have just proved (3.16) which indicates $|\mathbb{E}[\mathsf{e}_{1,\phi}^m]|^2 \le 4\|\phi\|^2_{L^\infty_{x,v}}$, we will show the boundedness of the second-order moment $V_\phi^m := \mathbb{E}[|\mathsf{e}_{1,\phi}^m|^2]$ to demonstrate that the variance is bounded. Note that

$$V_\phi^m = \mathbb{E}\left[|\langle \mathsf{f}_1^m - f(t^m), \phi\rangle_{xv}|^2\right] = \mathbb{E}[|\phi(x^m,v^m) - \langle f(t^m),\phi\rangle_{xv}|^2].$$

Then the proof follows from induction. We begin with the following estimate:

(3.18)
$$\begin{aligned}V_\phi^0 &= \mathbb{E}^0\left[|\phi(x^0,v^0) - \langle f_{\mathrm{in}},\phi\rangle_{xv}|^2\right] = \mathbb{E}^0\left[\phi(x^0,v^0)^2 - 2\phi(x^0,v^0)\langle f_{\mathrm{in}},\phi\rangle_{xv} + \langle f_{\mathrm{in}},\phi\rangle_{xv}^2\right] \\ &\le \|\phi\|^2_{L^\infty_{x,v}} + 2\|\phi\|^2_{L^\infty_{x,v}}\int f_{\mathrm{in}}\mathrm{d}x\mathrm{d}v + \|\phi\|^2_{L^\infty_{x,v}}\left(\int f_{\mathrm{in}}\mathrm{d}x\mathrm{d}v\right)^2 = 4\|\phi\|^2_{L^\infty_{x,v}},\end{aligned}$$

where we use the nonnegativity of f_{in} and the fact that its mass equals one. We also show below that V_ϕ^m does not grow fast in m. For any $m \ge 0$,

$$\begin{aligned}&V_\phi^{m+1} \\ &\quad= \mathbb{E}^{m+1}\left[|\phi(x^{m+1},v^{m+1}) - \langle f(t^{m+1}),\phi\rangle_{xv}|^2\right] \\ &\quad= \mathbb{E}^{m+1}\Bigg[|\underbrace{\phi(x^m,v^m) - \langle f(t^m),\phi\rangle_{xv}}_{A_1} \\ &\qquad+ \underbrace{\phi(x^{m+1},v^{m+1}) - \phi(x^m,v^m) + \langle f(t^m) - f(t^{m+1}),\phi\rangle_{xv}}_{A_2}|^2\Bigg] \\ &\quad= \mathbb{E}^{m+1}\left[|A_1|^2\right] + \mathbb{E}^{m+1}\left[|A_2|^2\right] + 2\mathbb{E}^{m+1}\left[A_1A_2\right] \\ &\quad= V_\phi^m + \mathbb{E}^m\left[\mathbb{E}_{\eta^{m+1}}\mathbb{E}_{p^{m+1}}\left[|A_2|^2\right]\right] + 2\mathbb{E}^m\left[A_1\mathbb{E}_{\eta^{m+1}}\mathbb{E}_{p^{m+1}}[A_2]\right].\end{aligned} \tag{3.19}$$

First, using the mean-value theorem, we see that

$$\begin{aligned}\mathbb{E}_{\eta^{m+1}}\mathbb{E}_{p^{m+1}}[A_2] &\le \Delta t\|v\cdot\nabla_x\phi + \sigma\mathcal{L}[\phi]\|_\infty\|f(t=\xi,x,v)\|_1 \\ &\quad+ \left(1 - e^{-\sigma(x^{m+1})\Delta t}\right)\mathcal{L}[\phi](x^{m+1}) \\ &\quad+ \Delta t\,\nabla_x\phi(\xi_x,v^m)\cdot v^m, \\ &\le \Delta t\, C_2(\|v\|_{L^\infty(\Omega)}, \|\sigma\|_{L^\infty}, \|\phi\|_{W^{1,\infty}_{x,v}}),\end{aligned}$$

where $t^m \le \xi \le t^{m+1}$ and ξ_x is between x^m and x^{m+1}. Thus,

$$2\mathbb{E}^m\left[A_1\mathbb{E}_{\eta^{m+1}}\mathbb{E}_{p^{m+1}}[A_2]\right] \le \Delta t C_2\,\mathbb{E}^m\left[A_1\right] \le C_2\Delta t,$$

where we have used (3.16). Next, we focus on $\mathbb{E}_{\eta^{m+1}}\mathbb{E}_{p^{m+1}}\left[|A_2|^2\right]$:

$$\begin{aligned}
&\mathbb{E}_{\eta^{m+1}}\mathbb{E}_{p^{m+1}}\left[|A_2|^2\right] \\
&\quad= \left(1-e^{-\sigma(x^{m+1})\Delta t}\right)\left||\Omega|^{-1}\langle\phi\rangle_v(x^{m+1})-\phi(x^m,v^m)+\Delta t\langle\mathcal{P}\phi,f(t=\xi)\rangle_{xv}\right|^2 \\
&\qquad+\Delta t^2 e^{-\sigma(x^{m+1})\Delta t}\left|\nabla_x\phi(\xi_x,v^m)\cdot v^m+\langle\mathcal{P}\phi,f(t=\xi)\rangle_{xv}\right|^2 \\
&\quad\le \Delta t\, C_2(\|v\|_{L^\infty(\Omega)},\|\sigma\|_{L^\infty},\|\phi\|_{W^{1,\infty}_{x,v}}).
\end{aligned}$$

As a result,

$$\mathbb{E}^{m+1}\left[|A_2|^2\right]\le \Delta t\, C_2\left(\|v\|_{L^\infty(\Omega)},\|\sigma\|_{L^\infty},\|\phi\|_{W^{1,\infty}_{x,v}}\right). \tag{3.20}$$

Combining the above two inequalities together, we have

$$V_\phi^{m+1}\le V_\phi^m+C_2\,\Delta t. \tag{3.21}$$

Noting that $\mathrm{Var}[\mathsf{e}_{1,\phi}^m]=V_\phi^m-|\mathbb{E}[\mathsf{e}_{1,\phi}^m]|^2$, (3.17) is obtained by using the boundedness of both terms from (3.16) and (3.21), with the constant depending on T and on the arguments that are listed in (3.20). We abuse the notation and still call it C_2. ∎

We are now ready to show (2.7) in Theorem 2.1 through the following proposition.

PROPOSITION 3.5. *Let f_N^m be the solution from running Algorithm* 2.1 *with MC samples $\{(x_n^m,v_n^m)\}_{m=0}^M$. We claim it approximates f with high probability. Namely, for small enough $\epsilon,\Delta t\ll 1$, and any $\phi\in\Phi$, the chance of the weak error* (2.5) *being bigger than $\mathcal{O}(\epsilon+\Delta t)$ is exponentially small in N:*

$$\mathbb{P}\left(|\mathsf{e}_{N,\phi}^m|\ge\epsilon+C_1\Delta t\right)\le 2\exp\left(-\frac{N\epsilon^2}{3C_1}\right).$$

Proof. Calling Proposition 3.2 and Lemma 3.4, we have shown that for any $\phi\in\Phi$,

$$\mathbb{E}\left[|\mathsf{e}_{1,\phi}^m|\right]\le C_1\Delta t,\quad \mathrm{Var}\left[\mathsf{e}_{1,\phi}^m\right]\le C_2,\quad\text{and}\quad |\mathsf{e}_{1,\phi}^m|<2\|\phi\|_{L^\infty_{x,v}}.$$

Noting (3.1), we apply the Bernstein inequality (3.15) to have, for any $\epsilon>0$,

$$\begin{aligned}
\mathbb{P}\left(|\mathsf{e}_{N,\phi}^m|\ge\epsilon+C_1\Delta t\right)&\le\mathbb{P}\left(|\mathsf{e}_{N,\phi}^m-e_{1,\phi}^m|\ge\epsilon\right)\\
&\le 2\exp\left(-\frac{N\epsilon^2}{2C_2+2\epsilon(2\|\phi\|_{L^\infty_{x,v}}+C_1\Delta t)/3}\right).
\end{aligned}$$

We conclude the proof by absorbing the constants into C_1 and setting $\epsilon,\Delta t$ to be sufficiently small. ∎

We have completed the proof of Theorem 2.1 with the constants' dependence explicitly spelled out.

Remark 3. In the next section, we will directly show that the computation of the Fréchet derivative is accurate. The proof for the solver for g, as expressed in (2.11), is not directly needed, but we nevertheless make some comments here. As stated in Remark 1, the derivation for the algorithm only comes from the conservation constraint (2.9), but this constraint cannot fully represent the entire dynamical information of the adjoint equation (2.1). To show that the solver is consistent with the adjoint solver, we need to return to the equation and test $\mathsf{g}-g$ on a given smooth function

ϕ and trace the evolution of the error. This comes down to proving that the error is not enlarged by much in every iteration:

$$\langle \mathsf{g}_1^{m+1} - g(t^{m+1}), \phi\rangle_{xv} - \langle \mathsf{g}_1^m - g(t^m), \phi\rangle_{xv} \sim 0.$$

While $\langle \mathsf{g}_1^{m+1} - \mathsf{g}_1^m, \phi\rangle_{xv} = \phi(x^{m+1}, v^{m+1}) - \phi(x^m, v^m)$, which translates to the algorithmic relation between (x^{m+1}, v^{m+1}) and (x^m, v^m), the term $\langle g(t^{m+1}) - g(t^m), \phi\rangle$ calls for the Taylor expansion in time for the PDE. Following the same strategy as shown in Lemma 3.1, one can show that the same trajectory $\{(x^m, v^m)\}$ represents the g dynamics backward in time. We should also note that unlike the MC solver for f where each particle is i.i.d. sampled, initiated from the initial distribution to represent f_{in}, this solver for g encodes a nonuniform weight for each particle. The particle takes on the value for $g(T)$. This difference is also recognized in different particle methods; see [38].

3.2. Convergence of the gradient. This section is dedicated to Theorem 2.2, which shows that Algorithm 2.2 provides an accurate numerical approximation to the true numerical gradient (2.20) in its semidiscrete-in-time form. The proof is conducted on the weak formulation to eliminate the complication from the delta function. Namely, we will examine the difference between $\mathfrak{G}_\phi$ and $\mathfrak{G}_{N,\phi}$, separately defined in (2.12) and (2.15). Since they both consist of two terms, we show below that $\mathfrak{G}^m_{N,i,\phi}$ defined in (2.14) approximates $\mathfrak{G}^m_{i,\phi}$ defined in (2.13) for $i = 1, 2$ and for all $1 \le m \le M$. The results are encapsulated in the following two propositions.

PROPOSITION 3.6. *If the initial condition f_{in} and measurement $d(x, v)$ are sufficiently regular such that they both are $C_c^\infty(\mathbb{R}^{d_x} \times \Omega)$, then for small ϵ and Δt,*

$$\mathbb{P}\left(|\mathfrak{G}^m_{N,1,\phi} - \mathfrak{G}^m_{1,\phi}| \ge \epsilon + C_3\Delta t\right) \le 2\exp\left(-\frac{N\epsilon^2}{3C_3}\right), \qquad 1 \le m \le M,$$

where

$$C_3 = C_3(\|f_{\text{in}}\|_{W_x^{2,\infty}L_v^\infty}, \|\sigma\|_{W_x^{1,\infty}}, \|\phi\|_{W_{x,v}^{2,\infty}}, \|d\|_{W_x^{2,\infty}L_v^\infty}, T). \tag{3.22}$$

PROPOSITION 3.7. *Under the same assumptions with Proposition* 3.6*, we have*

$$\mathbb{P}\left(|\mathfrak{G}^m_{N,2,\phi} - \mathfrak{G}^m_{2,\phi}| \ge \epsilon + C_3\Delta t + |e_v|\right) \le 2\exp\left(-\frac{N\epsilon^2}{3C_3}\right), \qquad 1 \le m \le M,$$

where e_v is the quadrature error in computing $\langle \mathsf{g}_n\rangle_v$.

Theorem 2.2 is then a direct corollary of the two propositions.

Remark 4. One strong requirement in Proposition 3.6 is the regularity assumption on $d(x, v)$. This assumption comes from the proof strategy where we view $g\phi \in \Phi$ as the test function and directly invoke the application of Theorem 2.1. One can relax the assumption on the test function space Φ in the previous theorem, and the relaxed condition would carry over. It is an interesting question to study the minimum bound for the regularity.

We now present the proofs for Propositions 3.6 and 3.7. Note that the two terms $\mathfrak{G}^m_{N,1,\phi}$ (resp., $\mathfrak{G}^m_{N,2,\phi}$) and $\mathfrak{G}^m_{1,\phi}$ (resp., $\mathfrak{G}^m_{2,\phi}$) share the same format, so in the following, we will only present details for $\mathfrak{G}^m_{N,1,\phi}$ and $\mathfrak{G}^m_{1,\phi}$. Consider an auxiliary formulation

$$\tilde{\mathfrak{G}}^m_{N,1,\phi} = \frac{1}{N}\sum_{n=1}^N \phi(x_n^m) g(t^m, x_n^m, v_n^m). \tag{3.23}$$

Then by the triangle inequality,

$$|\mathfrak{G}^m_{N,1,\phi} - \mathfrak{G}^m_{1,\phi}| \le |\mathfrak{G}^m_{N,1,\phi} - \tilde{\mathfrak{G}}^m_{N,1,\phi}| + |\tilde{\mathfrak{G}}^m_{N,1,\phi} - \mathfrak{G}^m_{1,\phi}|. \tag{3.24}$$

Proposition 3.6 is a direct corollary of Lemmas 3.8 and 3.9 that give control over the two terms on the right-hand side of (3.24), respectively.

The second term $\tilde{\mathfrak{G}}^m_{N,1,\phi} - \mathfrak{G}^m_{1,\phi}$ can be easily controlled since $\phi(x)g(t^m, x, v) \in \Phi$. We state Lemma 3.8 without proof.

LEMMA 3.8. *Under the same assumptions as in Proposition* 3.6*, we have*

$$\mathbb{P}\left(|\tilde{\mathfrak{G}}^m_{N,1,\phi} - \mathfrak{G}^m_{1,\phi}| \ge \epsilon + C_3\Delta t\right) \le 2\exp\left(-\frac{N\epsilon^2}{3C_3}\right), \qquad 1 \le m \le M, \tag{3.25}$$

where C_3 *is defined in Proposition* 3.6.

The first term in (3.24) reads

$$\mathfrak{G}^m_{N,1,\phi} - \tilde{\mathfrak{G}}^m_{N,1,\phi} = \frac{1}{N}\sum_{n=1}^{N} \phi(x^m_n)\left(\mathfrak{g}_n - g(t^m, x^m_n, v^m_n)\right), \tag{3.26}$$

and we control it as follows.

LEMMA 3.9. *Under the same assumptions as in Proposition* 3.6*, we have*

$$\mathbb{P}\left(|\mathfrak{G}^m_{N,1,\phi} - \tilde{\mathfrak{G}}^m_{N,1,\phi}| \ge \epsilon + C_3\Delta t\right) \le 2\exp\left(-\frac{N\epsilon^2}{3C_3}\right), \qquad 1 \le m \le M.$$

Proof. First note that due to (2.9), we always have

$$\int_{xv} f(t^M, x, v)g(t^M, x, v)\mathrm{d}x\mathrm{d}v = \int_{xv} f(t^m, x, v)g(t^m, x, v)\mathrm{d}x\mathrm{d}v, \qquad 1 \le m \le M.$$

Again, we assume that g, as a solution of the adjoint RTE, is sufficiently smooth such that $g(t, \cdot, \cdot) \in \Phi$. Then, using a variant of Proposition 3.2 (i.e., following the proof of Proposition 3.2 but starting at time t^m instead of t_0), we have

$$\tilde{\mathbb{E}}^m[g(t^M, x^M_n, v^M_n)|(x^m_n, v^m_n)] = g(t^m, x^m_n, v^m_n) + e,$$

with $|e| \le C_3\Delta t$, and $\tilde{\mathbb{E}}^m$ defined as

$$\tilde{\mathbb{E}}^m[\,\cdot\,|(x^m_n, v^m_n)] = \mathbb{E}_{\eta^{m+1}}\mathbb{E}_{p^{m+1}}\cdots\mathbb{E}_{\eta^M}\mathbb{E}_{p^M}[\,\cdot\,|(x^m_n, v^m_n)]. \tag{3.27}$$

Recall the definition of $\mathfrak{g}_n$ in (2.11). We then have

$$\tilde{\mathbb{E}}^m[\mathfrak{g}_n] = g(t^m, x^m_n, v^m_n) + e.$$

With the same argument, we have that

$$\tilde{\mathbb{E}}^m[\phi(x^m_n)\mathfrak{g}_n] = \phi(x^m_n)g(t^m, x^m_n, v^m_n) + e\,\phi(x^m_n). \tag{3.28}$$

Therefore, (3.26) becomes

$$\mathfrak{G}^m_{N,1,\phi} - \tilde{\mathfrak{G}}^m_{N,1,\phi} = \frac{1}{N}\sum_{n=1}^{N}\left(\phi(x^m_n)\mathfrak{g}_n - \tilde{\mathbb{E}}^m[\phi(x^m_n)\mathfrak{g}_n] + e\,\phi(x^m_n)\right), \quad |e\,\phi(x^m_n)| \le C_3\Delta t.$$

Note further from Lemma 3.4 (here we again alter the lemma by considering the randomness starting at time t^{m+1}, and the result remains the same) we have

$$\mathrm{Var}[\phi(x_n^m)\mathsf{g}_n] = \tilde{\mathbb{E}}^m\left[\phi(x_n^m)\mathsf{g}_n - \tilde{\mathbb{E}}^m\left[\phi(x_n^m)\mathsf{g}_n\right]\right]^2 = \phi(x_n^m)^2\tilde{\mathbb{E}}^m\left[\mathsf{g}_n - \tilde{\mathbb{E}}^m\left[\mathsf{g}_n\right]\right]^2 \le C_3.$$

Then the final result follows from the Bernstein inequality (3.15). ∎

The proof for Proposition 3.7 is almost identical. We present the details below.

Proof of Proposition 3.7. Parallel to (3.23), we define the auxiliary function

$$\tilde{\mathfrak{G}}_{N,2,\phi}^m = \frac{1}{N}\sum_{n=1}^N \phi(x_n^m)\langle g(t^m, x_n^m, \cdot)\rangle_v$$

and thereby write

$$\mathfrak{G}_{N,2,\phi}^m - \mathfrak{G}_{2,\phi}^m = \mathfrak{G}_{N,2,\phi}^m - \tilde{\mathfrak{G}}_{N,2,\phi}^m + \tilde{\mathfrak{G}}_{N,2,\phi}^m - \mathfrak{G}_{2,\phi}^m.$$

Here, the term $\mathfrak{G}_{N,2,\phi}^m - \tilde{\mathfrak{G}}_{N,2,\phi}^m$ has exactly the same estimate as in (3.25). For $\tilde{\mathfrak{G}}_{N,2,\phi}^m - \mathfrak{G}_{2,\phi}^m$, in addition to the error between g_n and $g(t^m, x_n^m, v_n^m)$, there is another error when computing the integral in v. More precisely, if $\mathsf{g}_n^m = g(t^m, x_n, v_n)$, then $\langle \mathsf{g}^m\rangle_v(\bar{x})$ is obtained via quadrature formula (2.18), which approximates the real integral $\int g(t^m, \bar{x}, v)\mathrm{d}v$ with a quadrature error denoted as e_v. Now, we follow the same proof as in Lemma 3.9 until (3.28), where we need to include this additional quadrature error to get

$$\tilde{\mathbb{E}}^m\left[\phi(x_n^m)\langle \mathsf{g}_n\rangle_v\right] = \phi(x_n^m)\langle g(t^m, x_n^m, \cdot)\rangle_v + e\,\phi(x_n^m) + e_v.$$

Therefore,

$$\mathfrak{G}_{N,2,\phi}^m - \tilde{\mathfrak{G}}_{N,2,\phi}^m = \frac{1}{N}\sum_{n=1}^N \left(\phi(x_n^m)\langle \mathsf{g}_n\rangle_v - \tilde{\mathbb{E}}^m\left[\phi(x_n^m)\langle \mathsf{g}_n\rangle_v\right] + e\,\phi(x_n^m) + e_v\right).$$

The same arguments as in Lemma 3.9 show that $\mathrm{Var}[\phi(x_n^m)\langle \mathsf{g}_n\rangle_v] \le C_3$, and the boundedness of $\phi(x_n^m)\langle \mathsf{g}_n\rangle_v$ is a direct consequence of the fact that both ϕ and g are bounded. Then by the Bernstein inequality, we have

$$\mathbb{P}\left(|\mathfrak{G}_{N,2,\phi}^m - \tilde{\mathfrak{G}}_{N,2,\phi}^m - C_3\Delta t - e_v| \ge \epsilon\right) \le 2\exp\left(-\frac{N\epsilon^2}{2C_3 + 2\epsilon(2\|\phi\|_{L_{x,v}^\infty} + C_3\Delta t)/3}\right).$$

Then if ϵ and Δt are small enough such that $2\epsilon(2\|\phi\|_{L_{x,v}^\infty} + C_3\Delta t)/3 < C_3$, we have

$$\begin{aligned}\mathbb{P}\left(|\mathfrak{G}_{N,2,\phi}^m - \tilde{\mathfrak{G}}_{N,2,\phi}^m| \ge \epsilon + C_3\Delta t + |e_v|\right) &\le \mathbb{P}\left(|\mathfrak{G}_{N,2,\phi}^m - \tilde{\mathfrak{G}}_{N,2,\phi}^m - C_3\Delta t - e_v| \ge \epsilon\right)\\ &\le 2\exp\left(-\frac{N\epsilon^2}{3C_3}\right).\end{aligned}$$

∎

4. Discretize-then-optimize framework. In this section, we consider the so-called discretize-then-optimize (DTO) framework to compute the gradient of an RTE-constrained optimization problem. We regard the MC method in Algorithm 2.1 as our discretization of the forward RTE (1.1).

Consider the objective functional

$$J = \iint r(x,v) f(T,x,v) \mathrm{d}x\mathrm{d}v, \tag{4.1}$$

where $J = \mathbb{E}_{f(T,x,v)}[r(x,v)]$ denotes the expectation of r with respect to the final-time RTE solution $f(T,x,v)$. The choice of J is only for notational convenience, and the derivations shall easily apply to general objective functionals; see examples shown in subsection 5.1. To carry on with the notation in subsection 3.1, the value in (4.1) could be approximated by the MC quadrature

$$J \approx \frac{1}{N}\sum_{n=1}^{N} r(x_n^M, v_n^M) =: \mathcal{J}. \tag{4.2}$$

Since we have already established the convergence theory of the MC method for RTE in subsection 3.1, here we assume that (2.3) holds and

$$(x_n^m, v_n^m) \sim f(t^m, x, v) \quad \forall n = 1, \dots, N, \quad m = 0, \dots M.$$

Given any test function $\phi(v)$ in the admissible set (2.4), we have

$$\begin{aligned} &\mathbb{E}_{\eta^{m+1}} \mathbb{E}_{p^{m+1}}[\phi(v)|(x_n^m, v_n^m)] \\ &\quad = \alpha_n^{m+1}\phi(v_n^m) + (1-\alpha_n^{m+1})\frac{1}{|\Omega|}\int_\Omega \phi(\eta) d\eta, \quad m = 0, \dots, M-1, \end{aligned} \tag{4.3}$$

where $\alpha_n^{m+1} = \exp(-\sigma(x_n^{m+1})\Delta t) = \exp(-\sigma(x_n^m + \Delta t v_n^m)\Delta t)$. As a result, the final objective functional can be expressed as sums of conditional expectations. We refer the reader to [41] for more details regarding this technique. Next, we use the $\tilde{\mathbb{E}}^m$ notation from (3.27) for $m = 0, \dots, M-1$, and (4.1) can be expressed as

$$J = \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{(x_n^m, v_n^m)\sim f(t^m,x,v)}\left[\tilde{\mathbb{E}}^m[r(x_n^M, v_n^M)]\right], \quad 0 \le m \le M-1, \tag{4.4}$$

as a result of the law of total expectations. Note that the above formula holds for any m. We can further write J into $J = \sum_{n=1}^N J_n$ following (4.4), where

$$J_n = \frac{1}{N}\mathbb{E}_{(x_n^m, v_n^m)\sim f(t^m,x,v)}\left[\mathcal{R}_n^m(x_n^m, v_n^m)\right], \tag{4.5}$$

$$\mathcal{R}_n^m(x_n^m, v_n^m) = \tilde{\mathbb{E}}^m[r(x_n^M, v_n^M)]. \tag{4.6}$$

The dependence of $\mathcal{R}_n^m(x_n^m, v_n^m)$ on the coefficient function $\sigma(x)$ is through the evaluations of $\{\sigma(x_n^{m+1})\}$ where $x_n^{m+1} = x_n^m + \Delta t\, v_n^m$, which are used in the acceptance-rejection probabilities in each $\mathbb{E}_{p^{m+1}}$; see (4.3). Note that $\mathcal{R}_n^m$ is conditioned on (x_n^m, v_n^m), so it can be seen as a function of (x_n^m, v_n^m) and consequently a function of $\sigma(x_n^{m+1})$ for a given function $\sigma(x)$. Thus, using the score function [32, 31, 41], we can express the derivative of $\mathcal{R}_n^m$ with respect to each $\sigma(x_n^{m+1})$ as

$$\frac{\partial \mathcal{R}_n^m}{\partial \sigma(x_n^{m+1})} = \tilde{\mathbb{E}}^m\left[\frac{\partial \log \kappa(x_n^{m+1})}{\partial \sigma(x_n^{m+1})} r(x_n^M, v_n^M)\right], \tag{4.7}$$

where the probability for the rejection sampling and its score function are

$$\kappa(x_n^{m+1}) = \begin{cases} \alpha_n^{m+1} & \text{if } v_n^{m+1} = v_n^m, \\ 1-\alpha_n^{m+1} & \text{otherwise,} \end{cases}$$

$$\frac{\partial \log \kappa(x_n^{m+1})}{\partial \sigma(x_n^{m+1})} = \begin{cases} -\Delta t & \text{if } v_n^{m+1} = v_n^m, \\ \Delta t \frac{\alpha_n^{m+1}}{1-\alpha_n^{m+1}} & \text{otherwise.} \end{cases}$$

Using the same trajectories from the MC solver for the forward RTE (see Algorithm 2.1), we obtain samples $\{\widehat{\mathcal{R}}_n^m\}_{m=0}^{M-1}$ based on (4.6), and the Monte Carlo gradient $\{\widehat{\mathcal{G}}_n^m\}_{m=0}^{M-1}$ based on (4.7), respectively, where

$$\widehat{\mathcal{R}}_n^m = r(x_n^M, v_n^M), \qquad \widehat{\mathcal{G}}_n^m = \begin{cases} -r(x_n^M, v_n^M)\Delta t & \text{if } v_n^{m+1} = v_n^m, \\ r(x_n^M, v_n^M)\dfrac{\alpha_n^{m+1}\Delta t}{1-\alpha_n^{m+1}} & \text{otherwise} \end{cases} \tag{4.8}$$

for all $m = 0, \ldots, M-1$.

Based on (4.5) and (4.6), we can also treat $\{\frac{1}{N}\widehat{\mathcal{R}}_n^m\}_{m=0}^{M-1}$ as samples of J_n. Given the same RTE particle trajectories obtained from Algorithm 2.1, $\{(x_n^m, v_n^m)\}_{m=0}^M$, the following holds for any $m = 0, \ldots, M-1$:

$$J = \sum_{n=1}^N J_n \overset{\text{sample (4.5)}}{\approx} \frac{1}{N}\sum_{n=1}^N \mathcal{R}_n^m \overset{\text{sample (4.6)}}{\approx} \frac{1}{N}\sum_{n=1}^N \widehat{\mathcal{R}}_n^m,$$
$$\overset{\text{based on (4.2)}}{\approx} \mathcal{J} = \frac{1}{N}\sum_{n=1}^N r(x_n^M, v_n^M).$$

Thus, the Monte Carlo gradient of $\mathcal{J}$ with respect to $\sigma(x_n^{m+1})$ can be approximated by

$$\frac{\partial \mathcal{J}}{\partial \sigma(x_n^{m+1})} \approx \frac{1}{N}\sum_{n=1}^N \frac{\partial \mathcal{R}_n^m}{\partial \sigma(x_n^{m+1})} = \frac{1}{N}\frac{\partial \mathcal{R}_n^m}{\partial \sigma(x_n^{m+1})} \approx \frac{1}{N}\widehat{\mathcal{G}}_n^m \quad \forall m, \tag{4.9}$$

since $\mathcal{R}_i^m$ does not depend on $\sigma(x_n^{m+1})$ if $i \neq n$ $\forall m$. Combining (4.9) with (4.8), we obtain a Monte Carlo gradient formula for $\frac{\partial \mathcal{J}}{\partial \sigma(x_n^{m+1})}$.

However, it is worth noting that

$$\frac{\partial \mathcal{J}}{\partial \sigma(x_n^{m+1})} \neq \frac{\delta J}{\delta \sigma}(x_n^{m+1}). \tag{4.10}$$

The left-hand side gradient treats $\sigma(x_n^{m+1})$ as a single parameter, and therefore the effective parameters are a collection of MN number of scalars, $\{\sigma(x_n^{m+1})\}$, where $n = 0, \ldots, N$ and $m = 0, \ldots M-1$. On the other hand, the right-hand side is a functional derivative with respect to the parameter function $\sigma(x)$ (the same with the OTD derivative (2.2)) evaluated at $x = x_n^{m+1}$. We use the following example to show how to relate both sides of (4.10).

Consider a separate mesh grid $\{\bar{x}_j\}$ in the spatial domain, and Q_j is a small neighborhood of $\bar{x}_j$ for each j. Consider a perturbed parameter function $\sigma(x) + \delta\sigma(x)$, where $\delta\sigma(x) = \mathbb{1}_{x \in Q_j}\epsilon$ for some small constant ϵ. We then have

$$J(\sigma + \delta\sigma) - J(\sigma) \approx \int_x \frac{\delta J}{\delta \sigma}(x)\delta\sigma(x)\mathrm{d}x = \epsilon \int_{Q_j} \frac{\delta J}{\delta \sigma}(x)\mathrm{d}x. \tag{4.11}$$

Algorithm 4.1 The particle-based DTO approach for gradient computation.

1: Given cell centers $\{\bar{x}_j\}$ where we want to evaluate the gradient (2.2).
2: Implement Algorithm 2.1 to solve the forward RTE (1.1) and store the trajectories $\{(x_n^m, v_n^m)\}$ in memory where $m = 0, \ldots, M$ and $n = 1, \ldots, N$.
3: Compute the objective function (4.2) based on f^M and evaluate $\widehat{\mathcal{G}}_n^m$ following (4.8) for all m and n.
4: Evaluate the gradient at $x = \bar{x}_j$ following (4.13).

On the other hand, with $\mathcal{J}(\sigma)$ denoting the value (4.2) calculated with a given parameter function σ, we have the following based on (4.9):

(4.12)
$$\mathcal{J}(\sigma + \delta\sigma) - \mathcal{J}(\sigma) \approx \sum_{n=1}^{N} \sum_{m=0}^{M-1} \frac{\partial \mathcal{J}}{\partial \sigma(x_n^{m+1})} \delta\sigma(x_n^{m+1}) \approx \epsilon \frac{1}{N} \sum_{n=1}^{N} \sum_{m=0}^{M-1} \mathbb{1}_{x_n^{m+1} \in Q_j} \widehat{\mathcal{G}}_n^m.$$

Combining the last terms in (4.11)–(4.12) and assuming the approximated gradient function $\frac{\delta J}{\delta \sigma}$ is piecewise constant on $\{Q_j\}$, we have

$$\frac{\delta J}{\delta \sigma}(\bar{x}_j)\,|Q_j| \approx \int_{Q_j} \frac{\delta J}{\delta \sigma}(x)\mathrm{d}x \approx \frac{1}{N} \sum_{n=1}^{N} \sum_{m=0}^{M-1} \mathbb{1}_{x_n^{m+1} \in Q_j} \widehat{\mathcal{G}}_i^n.$$

Finally, we may approximate the gradient using values of (4.8) by

(4.13)
$$\begin{aligned} \frac{\delta J}{\delta \sigma}(\bar{x}_j) &\approx \frac{1}{|Q_j|} \frac{1}{N} \sum_{n=1}^{N} \sum_{m=0}^{M-1} \mathbb{1}_{x_n^{m+1} \in Q_j} \widehat{\mathcal{G}}_n^m \\ &= \frac{1}{|Q_j|} \frac{\Delta t}{N} \sum_{n=1}^{N} \sum_{m=1}^{M} \mathbb{1}_{x_n^m \in Q_j}\, r(x_n^M, v_n^M)\, \xi_n^m, \quad \text{where } \xi_n^m = \begin{cases} -1 & \text{if } v_n^m = v_n^{m-1}, \\ \frac{\alpha_n^m}{1-\alpha_n^m} & \text{otherwise.} \end{cases} \end{aligned}$$

Similar to the particle-based OTD approach presented in Algorithm 2.2, sorting is also needed in computing (4.13) for all particles in the long time horizon and leads to $\mathcal{O}(NM(\log N + \log M))$ complexity. Note that this complexity can be reduced to $\mathcal{O}(NM(\log N))$ if one sorts the particles at each time step instead of doing it all at once in the end, which reduces it to the same complexity as for the OTD approach.

We summarize steps of this particle-based DTO approach for gradient calculation in Algorithm 4.1.

5. Numerical examples. In this section, we present a few numerical tests to illustrate the gradient computed by particle methods following the OTD and DTO approaches presented in sections 2 and 4. We will refer to these methods as P-OTD and P-DTO in this section. As a reference, we will use a forward Euler scheme along with an upwind spatial discretization for both forward and adjoint equations. Details are provided in Appendix A. We remark that the gradient calculation based on the finite-volume method (FVM) also belongs to the OTD approach for which we use a consistent finite-volume upwind scheme (adjoint with respect to the FVM for solving (1.1)) to discretize the adjoint equation (2.1). Thus, it also coincides with what one would get from the DTO approach, with the discretization being the FVM.

5.1. Inverse problem. First, we consider a setup based on inverse data matching problems similar to the one described in subsection 1.1. We measure the spatial-domain density at the final time T, and the reference probability density function is $d(x)$. For simplicity, we choose the L^2-based objective functional

$$J_1(\sigma) = \frac{1}{2}\int_D |\rho_T(x) - d(x)|^2 \mathrm{d}x, \quad \rho_T(x) = \frac{1}{|\Omega|}\int_\Omega f(T,x,v)\mathrm{d}v, \tag{5.1}$$

where other proper metrics and divergence for the probability space could also be considered.

It is worth noting that (5.1) is different from (1.3). Recall in section 2 where we use the method of Lagrange multipliers to derive the equation for g shown in (2.1). The final condition for g will change with respect to the functional J evaluated at the final-time RTE solution $f(T,x,v)$, while the back-propagation rule for g (i.e., the PDE itself) is independent of the choice of J. Hence, the final-time condition of g for the objective function J_1 should be

$$g(T,x,v) = -\frac{\delta J_1}{\delta f(T,x,v)} = |\Omega|^{-1}\left(d(x) - \rho_T(x)\right),$$

which is constant in v.

In a one-dimensional (1D) setting, we consider the spatial domain $D = [-2,2]$ and the velocity domain $\Omega = [-1,1]$, with the periodic boundary condition on D. We run the forward RTE (1.1) on the time interval $[0,T]$ where $T = 2$. The step size $\Delta t = 0.01$ in both FVM and P-OTD. The initial distribution is

$$f_{\mathrm{in}}(x,v) = 2\pi^{-\frac{1}{2}}\exp(-4x^2),$$

which is constant in v, and the measurement density is

$$d(x) = \sqrt{5}\pi^{-\frac{1}{2}}\exp(-5|x-0.6|^2).$$

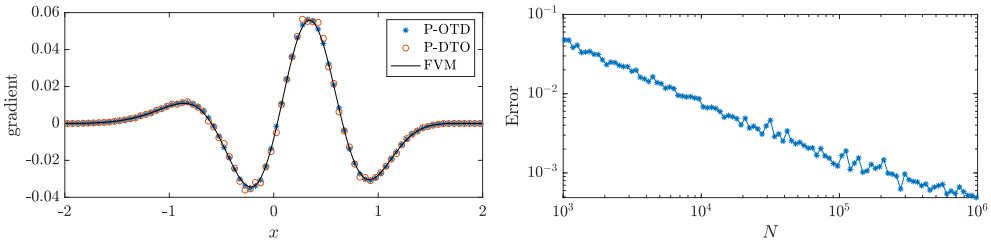
Note that $\iint f_{\mathrm{in}}(x,v)dxdv = 2$ and $\int d(x)dx = 1$. We evaluate the gradient at $\sigma(x) = 2$ for any $x \in D$. We use the test function $\phi(x) = |\Delta x|^{-d}\mathbb{1}_{Q(\bar{x},\Delta x)}(x)$ for $\bar{x}$ at grid points where the FVM gradient is evaluated. We define the weak formulation of the true gradient

$$\mathfrak{G}(\bar{x}) := \left\langle \frac{\delta \mathfrak{L}}{\delta \sigma}, \frac{1}{|\Delta x|^d}\mathbb{1}_{Q(\bar{x},\Delta x)} \right\rangle_x, \tag{5.2}$$

which is approximated by the computed FVM gradient.

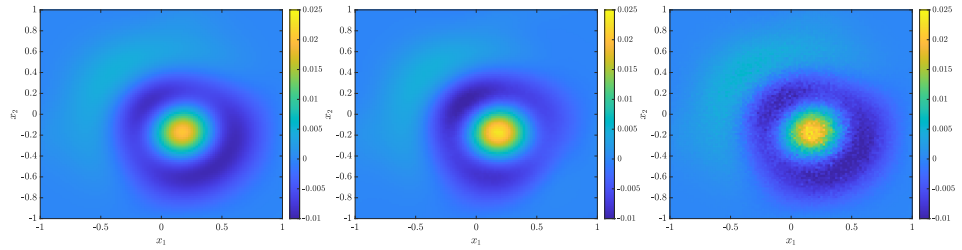
In Figure 1(a), we illustrate the gradients computed by FVM, P-OTD (using $N = 10^6$ particles), and P-DTO (using $N = 10^6$ particles), evaluated at different grid points $\{\bar{x}_j\}$. The P-OTD gradient is computed following (2.16), while the P-DTO gradient is computed by adapting the formula (4.13) for the particular objective function (5.1). We define $E(\bar{x}) = \mathfrak{G}_N(\bar{x}) - \mathfrak{G}(\bar{x})$, and in Figure 1(b), we show a log-log plot of $\|E(\bar{x})\|_2$ with respect to the variable $\bar{x}$ as the number of particles N used in P-OTD increases. It reflects the difference in the weak form of the gradient between the reference and the one computed from P-OTD. The error decay demonstrates the expected MC error of $\mathcal{O}(1/\sqrt{N})$, as proved in Theorem 2.2.

In a two-dimensional (2D) setting, we consider spatial domain $D = [-1,1]^2$ and velocity domain $\Omega = \mathbb{S}^1$, again with the periodic boundary condition on D. We



(a) P-OTD and FVM gradients (b) The 2-norm of $\mathfrak{G}_N(\bar{x}) - \mathfrak{G}(\bar{x})$

FIG. 1. *Comparison among the gradients calculated by FVM, P-OTD, and P-DTO for the objective function* (5.1) *in a* 1*D setting.* (a) *Illustration of the three gradients using* $N = 10^6$ *particles in P-OTD and P-DTO.* (b) *The* 2*-norm error of* $E(\bar{x}) := \mathfrak{G}_N(\bar{x}) - \mathfrak{G}(\bar{x})$ *as a function of* $\bar{x}$ *with respect to the number of particles* N *in P-OTD, where* $\mathfrak{G}_N(\bar{x})$ *is defined in* (2.16) *and the true (weak-form) gradient* $\mathfrak{G}(\bar{x})$ *given in* (5.2) *is approximated by the FVM gradient.*



(a) FVM gradient (b) P-OTD gradient (c) P-DTO gradient

FIG. 2. *Comparison between the gradients calculated by* (a) *FVM,* (b) *P-OTD, and* (c) *P-DTO methods for the objective function* (5.1) *in a* 2*D setting described in subsection* 5.1. *The number of particles in P-OTD and P-DTO is* $N = 10^6$.

parameterize the velocity using the polar coordinate, $v = [\cos\theta, \sin\theta]^\top$, $\theta \in [-\pi, \pi]$. The initial distribution is

$$f_{\text{in}}(x, v) = 4\pi^{-1} \exp(-4|x|^2),$$

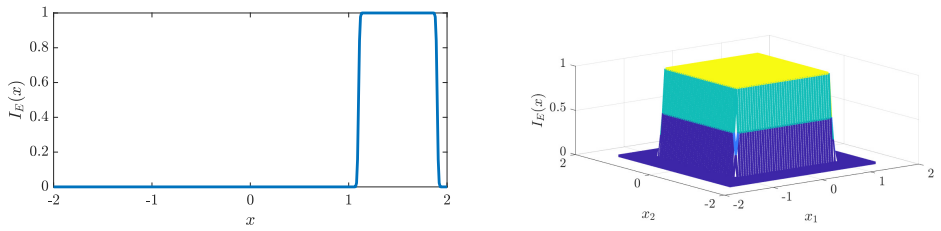
which is again constant in v, and the measurement density is

$$d(x) = 5\pi^{-1} \exp(-5|x_1 - 0.3|^2 - 5|x_2 + 0.3|^2), \quad x = [x_1, x_2]^\top.$$

Note that $\iint f_{\text{in}}(x, v)dxdv = 2\pi$ and $\int d(x)dx = 1$. We evaluate the gradient at $\sigma(x) = 2\ \forall x \in D$. The time step $\Delta t = 0.01$ and the final time $T = 0.5$. The gradients computed by the three methods are shown in Figure 2. We used $N = 10^6$ particles in the P-OTD and P-DTO methods and plotted the averaged value from 100 i.i.d. runs in Figures 2(a) and 2(b) to further reduce the random error by a factor of 10. We observe that the P-DTO method introduces more variance in the gradient calculation for this example.

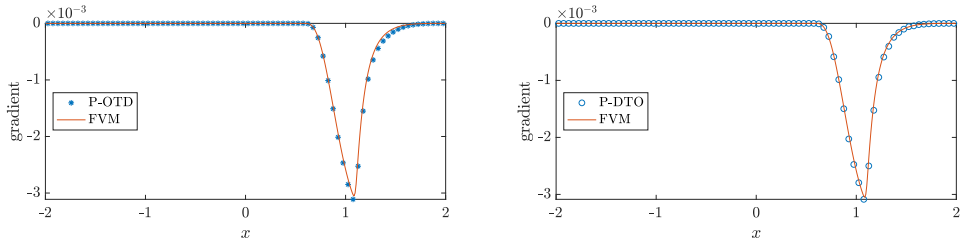
5.2. Optimal control. In this subsection, we focus on a different objective functional (4.1) which we will refer to as J_2. It is often used in optimal control or optimal design applications. Here, we measure the macroscopic quantity at the final time T for function

$$r(x, v) = s(v)I_E(x),$$



(a) 1D setting (b) 2D setting

FIG. 3. *Illustration of approximated indicator function $I_E(x)$ in 1D* (a) *and 2D* (b) *settings.*



(a) FVM vs. P-OTD (b) FVM vs. P-DTO

FIG. 4. *Gradient comparisons among FVM, P-OTD, and P-DTO methods in a 1D setting for the objective function J_2. The number of particles in both particle methods is $N = 10^6$.*

where $I_E(x)$ is a smooth approximation to the indicator function $\mathbb{1}_E(x) = \mathbb{1}_{x\in E}$ for a chosen measurement set $E \subset D$; see Figure 3 for illustrations of $I_E(x)$ in 1D and 2D settings. Based on J_2, the final condition of the adjoint variable g should be

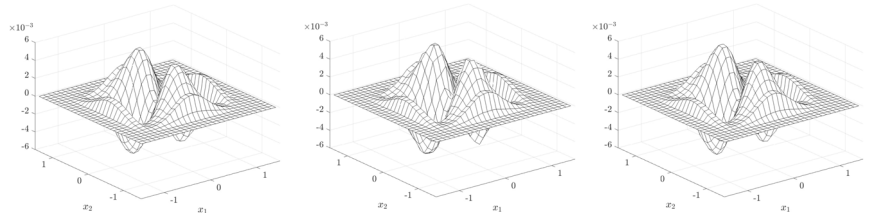
$$g(T,x,v) = -\frac{\delta J_2}{\delta f(T,x,v)} = -r(x,v).$$

The initial conditions for the RTE remain the same as in the examples in subsection 5.1.

We use three methods to compute the gradient of J_2 at $\sigma(x) = 2$: FVM, P-OTD, and P-DTO. The first two methods follow the OTD approach since they discretize the forward RTE (1.1) and the continuous adjoint equation (2.1), whose solutions are plugged into (2.16) for gradient calculation. On the other hand, the P-DTO method derived in section 4 follows the DTO approach, and it does not solve the adjoint equation (2.1). Using the solution based on Algorithm 2.1 to the forward RTE (1.1) and the history of rejection samplings therein, the gradient can be approximated by formula (4.13).

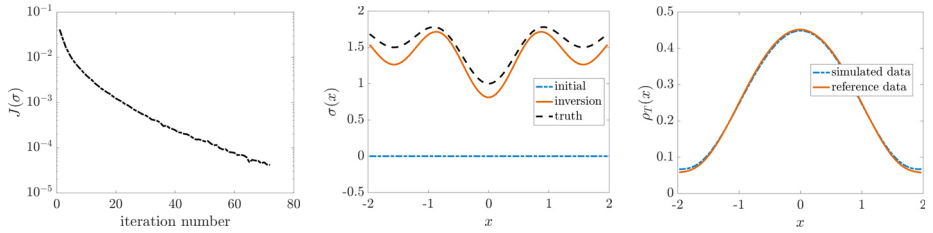
In one dimension, we set $s(v) = v^2$, $T = 0.5$, and $\Delta t = 0.005$. Figure 4 illustrates the comparison among the three methods. In the 2D case, we consider the spatial domain $D = [-1.5, 1.5]^2$ and the velocity domain $\Omega = \mathbb{S}^1$, and set $s(v) = |v_1|^2$, where $v = [v_1, v_2]^\top$. The total simulation time $T = 0.2$, while $\Delta t = 0.005$. We show the gradients calculated from the three methods in Figure 5. We further reduce the variances in the P-OTD gradient by taking its averaged value after 100 i.i.d. runs. We also average the P-DTO gradient based on 300 i.i.d. runs.

5.3. An inversion example. In this subsection, we present an inversion example using the gradient compute from the OTD-based approach presented in Algorithm 2.2. The experimental setup is based on the 1D case in subsection 5.1, where we



(a) P-OTD (b) P-OTD (c) FVM

FIG. 5. *Gradient comparisons among FVM, P-OTD, and P-DTO methods in a 2D setting for the objective function J_2. The number of particles in both particle methods is $N = 10^6$.*



(a) convergence history (b) reconstructed $\sigma(x)$ (c) simulated and reference data

FIG. 6. (a) *The convergence history during the inversion process;* (b) *the comparison among the initial guess, the inverted scattering coefficient, and the ground truth;* (c) *the simulated density using the inverted coefficient and the reference data with respect to the true coefficient.*

further parameterize the scattering coefficient $\sigma(x) = c_0 + c_1 \sin(x) + c_2 \sin(2x)$. We employ this low-dimensional parameterization to reduce the number of local minima in the optimization landscape for problem (5.1), not due to concerns from the computational cost of the gradient calculation or optimization.

Using the gradient computed from the P-OTD method, we use the gradient descent method with back-tracking line search to ensure that the objective function monotonically decays along the iterations; see Figure 6(a) where the objective function is reduced to 0.1% of its initial value. In Figure 6, we show the initial guess, the inverted scattering coefficient, and the ground truth. Even though the reconstruction result is slightly different from the truth, their corresponding densities, i.e., ρ_T generated from the inversion and the ground truth, are very close, as shown in Figure 6(c). It is a sign that the inverse problem under the current experimental setup is unstable. More data, rather than simply a final-time density ρ_T, could help improve the resolution of the reconstruction. It is also worth noting that when the reconstructed σ is close to the truth, the random noise from the forward RTE MC solver starts to affect the objective function as we see small fluctuations after 60 iterations. However, this also means one does not have to worry about the random noise in the objective function or the gradient computed from the MC methods when the reconstructed coefficient is far from the truth since the data mismatch at that stage dominates the objective function.

Appendix A. The finite volume scheme. We summarize the finite volume scheme used to compute the reference solution. As an illustration, we only consider the 1D case. Denote

$$f_{i,j}^m \approx f(t^m, x_i, v_j), \qquad 0 \le m \le M, \quad 1 \le j \le N_v,$$

as the numerical approximation, and let Δx, Δt, and Δv be the corresponding mesh size in x, t, and v, respectively. We consider the time domain $[0,T]$, the spatial domain $D=[-2,2]$, and the velocity domain $\Omega=[-1,1]$. Then $M\Delta t = T$, and $\Delta v N_v = |\Omega| = 2$, where T is the final time. We have the following discretization for (1.1):

$$f_{i,j}^{m+1} - f_{i,j}^m + \frac{\Delta t}{\Delta x} v_j^+ (f_{i,j}^m - f_{i-1,j}^m) + \frac{\Delta t}{\Delta x} v_j^- (f_{i+1,j}^m - f_{i,j}^m) = \sigma_i \Delta t \left(|\Omega|^{-1} \langle f_{i,j}^m \rangle - f_{i,j}^m \right)$$

for $0 \le m \le M-1$ and $1 \le j \le N_v$, with initial condition

$$f_{i,j}^0 = f_{\text{in}}(x_i, v_j).$$

Here, $v_j = -1 + (j-1/2)\Delta v$ representing the cell center, and $v^+ = \max\{v,0\}$, $v^- = \min\{v,0\}$. The average in v is computed by a simple midpoint rule: $\langle f_{i,j}^n \rangle = \sum_{j=1}^{N_v} f_{i,j}^n \Delta v$.

Writing down the discrete version of the objective function (taking (5.1) as an example),

$$J = \frac{\Delta x}{2} \sum_i \left(d_i - |\Omega|^{-1} \langle f_{i,j}^M \rangle \right)^2 + \Delta x \Delta v \sum_i \sum_{m=1}^{M-1} \sum_{j=1}^{N_v} g_{i,j}^{m+1}$$
$$\left(f_{i,j}^{m+1} - f_{i,j}^m + \frac{\Delta t}{\Delta x} v_j^+ (f_{i,j}^m - f_{i-1,j}^m) + \frac{\Delta t}{\Delta x} v_j^- (f_{i+1,j}^m - f_{i,j}^m) - \sigma_i \Delta t \left(\frac{\langle f_{i,j}^m \rangle}{|\Omega|} - f_{i,j}^m \right) \right),$$

the optimality condition leads to the following discretization for (2.1):

$$g_{i,j}^m - g_{i,j}^{m+1} + \frac{\Delta t}{\Delta x} v_j^+ (g_{i,j}^m - g_{i+1,j}^m) + \frac{\Delta t}{\Delta x} v_j^- (g_{i-1,j}^m - g_{i,j}^m) = \sigma_i \Delta t (|\Omega|^{-1} \langle g_{i,j}^{m+1} \rangle - g_{i,j}^{m+1})$$

for $M-1 \ge m \ge 0, 1 \le j \le N_v$, with final condition depending on the objective function

$$g_{i,j}^M = |\Omega|^{-1} \left(d_i - |\Omega|^{-1} \langle f_{i,j}^M \rangle \right),$$

where $|\Omega|$ is the Lebesgue measure of the velocity domain. We use the periodic boundary condition in x throughout the whole calculation. Then it is straightforward to calculate the gradient as

$$\frac{\delta J}{\delta \sigma_i} = \Delta x \Delta v \Delta t \sum_{m=1}^{M-1} \sum_{j=1}^{N_v} f_{i,j}^m \left(g_{i,j}^{m+1} - |\Omega|^{-1} \langle g_{i,j}^{m+1} \rangle \right).$$

Similar to (4.10), $\frac{\delta J}{\delta \sigma_i}$ above is not the same as $\frac{\delta J}{\delta \sigma}(x_i)$, but we can relate these two by following procedures similar to those in (4.11)–(4.12). Assuming that the gradient $\frac{\delta J}{\delta \sigma}(x)$ is piecewise constant over the spatial cells $\{[x_i - \Delta x/2, x_i + \Delta x/2)\}$, we have that

$$\frac{\delta J}{\delta \sigma}(x_i) \approx \frac{1}{\Delta x} \frac{\delta J}{\delta \sigma_i} = \Delta v \Delta t \sum_{m=1}^{M-1} \sum_{j=1}^{N_v} f_{i,j}^m \left(g_{i,j}^{m+1} - |\Omega|^{-1} \langle g_{i,j}^{m+1} \rangle \right).$$

Acknowledgments. All three authors participated in "Geometric Methods in Optimization and Sampling," held at the Simons Institute for the Theory of Computing in Fall 2021, and "Frontiers in kinetic theory: connecting microscopic to macroscopic scale" held at the Isaac Newton Institute (UK) in Spring 2022, where this work was initiated.

REFERENCES

[1] G. W. Alldredge, C. D. Hauck, and A. L. Tits, *High-order entropy-based closures for linear transport in slab geometry* II*: A computational study of the optimization problem*, SIAM J. Sci. Comput., 34 (2012), pp. B361–B391, https://doi.org/10.1137/11084772X.

[2] H. Babovsky and R. Illner, *A convergence proof for Nanbu's simulation method for the full Boltzmann equation*, SIAM J. Numer. Anal., 26 (1989), pp. 45–65, https://doi.org/10.1137/0726004.

[3] L. L. Baker and N. G. Hadjiconstantinou, *Variance reduction for Monte Carlo solutions of the Boltzmann equation*, Phys. Fluids, 17 (2005) 051703, https://doi.org/10.1063/1.1899210.

[4] G. Bal, *Radiative transfer equations with varying refractive index: A mathematical perspective*, J. Opt. Soc. Amer. A, 23 (2006), pp. 1639–1644.

[5] G. Bal, *Inverse transport theory and applications*, Inverse Problems, 25 (2009), 053001.

[6] C. Bardos, F. Golse, B. Perthame, and R. Sentis, *The nonaccretive radiative transfer equations: Existence of solutions and Rosseland approximation*, J. Funct. Anal., 77 (1988), pp. 434–460.

[7] J. T. Betts and S. L. Campbell, *Discretize then optimize*, in Mathematics for Industry: Challenges and Frontiers, SIAM, Philadelphia, 2005, pp. 140–157.

[8] G. Bird, *Direct simulation and the Boltzmann equation*, Phys. Fluids, 13 (1970), pp. 2676–2681.

[9] A. Bobylev and K. Nanbu, *Theory of collision algorithms for gases and plasmas based on the Boltzmann equation and the Landau–Fokker–Planck equation*, Phys. Rev. E, 61 (2000), pp. 4576–4586.

[10] J. Burkardt, M. Gunzburger, and J. Peterson, *Insensitive functionals, inconsistent gradients, spurious minima, and regularized functionals in flow optimization problems*, Int. J. Comput. Fluid Dyn., 16 (2002), pp. 171–185.

[11] R. Caflisch, D. Silantyev, and Y. Yang, *Adjoint DSMC for nonlinear Boltzmann equation constrained optimization*, J. Comput. Phys., 439 (2021), 110404.

[12] R. E. Caflisch, *Monte Carlo and quasi-Monte Carlo methods*, Acta Numer., 7 (1998), pp. 1–49.

[13] S.-O. Chan, I. Diakonikolas, R. A. Servedio, and X. Sun, *Efficient density estimation via piecewise polynomial approximation*, in Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing, STOC '14, Association for Computing Machinery, New York, 2014, pp. 604–613, https://doi.org/10.1145/2591796.2591848.

[14] S. Chandrasekhar, *Radiative Transfer*, Courier Corporation, 2013.

[15] K. Chen, Q. Li, and J.-G. Liu, *Online learning in optical tomography: A stochastic approach*, Inverse Problems, 34 (2018), 075010.

[16] K. Chen, Q. Li, and L. Wang, *Stability of stationary inverse transport equation in diffusion scaling*, Inverse Problems, 34 (2018), 025004.

[17] R. Dautray and J.-L. Lions, *Mathematical Analysis and Numerical Methods for Science and Technology*, Vol. 6, Springer Science & Business Media, 1993.

[18] G. Dimarco, L. Pareschi, and G. Samaey, *Asymptotic-Preserving Monte Carlo methods for transport equations in the diffusive limit*, SIAM J. Sci. Comput., 40 (2018), pp. A504–A528, https://doi.org/10.1137/17M1140741.

[19] W. W. Hager, *Runge–Kutta methods in optimal control and the transformed adjoint system*, Numer. Math., 87 (2000), pp. 247–282.

[20] M. Herty, R. Pinnau, and M. Seaïd, *Optimal control in radiative transfer*, Optim. Methods Softw., 22 (2007), pp. 917–936.

[21] E. Hirvijoki, A. Brizard, A. Snicker, and T. Kurki-Suonio, *Monte Carlo implementation of a guiding-center Fokker–Planck kinetic equation*, Phys. Plasmas, 20 (2013), 092505.

[22] R. Hochuli, S. Powell, S. R. Arridge, and B. Cox, *Forward and adjoint radiance Monte Carlo models for quantitative photoacoustic imaging*, in Photons Plus Ultrasound: Imaging and Sensing 2015, Vol. 9323, SPIE, 2015, pp. 245–254.

[23] H. Huang, J.-G. Liu, and P. Pickl, *On the mean-field limit for the Vlasov–Poisson–Fokker–Planck system*, J. Stat. Phys., 181 (2020), pp. 1915–1965.
[24] A. D. Klose and A. H. Hielscher, *Optical tomography using the time-independent equation of radiative transfer—Part 2: Inverse model*, J. Quant. Spectrosc. Radiat. Transf., 72 (2002), pp. 715–732.
[25] A. D. Klose, U. Netz, J. Beuthan, and A. H. Hielscher, *Optical tomography using the time-independent equation of radiative transfer—Part 1: Forward model*, J. Quant. Spectrosc. Radiat. Transf., 72 (2002), pp. 691–713.
[26] P. Kratzer, *Monte Carlo and kinetic Monte Carlo methods—A Tutorial*, in Multiscale Simulation Methods in Molecular Sciences Lecture Notes, Vol. 42, Institute for Advanced Simulation, Forschungszentrum Jülich, 2009, pp. 51–76.
[27] E. E. Lewis and W. F. Miller, *Computational Methods of Neutron Transport*, John Wiley and Sons, New York, 1984.
[28] Q. Li and L. Wang, *Implicit asymptotic preserving method for linear transport equations*, Commun. Comput. Phys., 22 (2017), pp. 157–181.
[29] H. Mashaal, D. Feuermann, and J. M. Gordon, *Aplanatic lenses revisited: The full landscape*, Appl. Optics, 55 (2016), pp. 2537–2542.
[30] N. McCormick, *Inverse radiative transfer problems: A review*, Nuclear Sci. Eng., 112 (1992), pp. 185–198.
[31] S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih, *Monte Carlo gradient estimation in machine learning*, J. Mach. Learn. Res., 21 (2020), pp. 1–62.
[32] C. Naesseth, F. Ruiz, S. Linderman, and D. Blei, *Reparameterization gradients through acceptance-rejection sampling algorithms*, in Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), 2017.
[33] K. Nanbu, *Direct simulation scheme derived from the Boltzmann equation.* I. *Monocomponent gases*, J. Phys. Soc. Japan, 49 (1980), pp. 2042–2049.
[34] L. Pareschi and G. Russo, *An introduction to Monte Carlo method for the Boltzmann equation*, ESAIM Proc., 10 (2001), pp. 35–75.
[35] L. Pareschi and G. Toscani, *Interacting Multiagent Systems: Kinetic Equations and Monte Carlo Methods*, Oxford University Press, Oxford, 2013.
[36] G. Peyré and M. Cuturi, *Computational optimal transport: With applications to data science*, Found. Trends Mach. Learn., 11 (2019), pp. 355–607.
[37] S. Powell, R. Hochuli, and S. R. Arridge, *Radiance Monte-Carlo for application of the radiative transport equation in the inverse problem of diffuse optical tomography*, in Optical Tomography and Spectroscopy of Tissue XII, Vol. 10059, SPIE, 2017, pp. 84–96.
[38] P.-A. Raviart, *An analysis of particle methods*, in Numerical Methods in Fluid Dynamics, Springer, 1985, pp. 243–324.
[39] K. Ren, G. Bal, and A. H. Hielscher, *Frequency domain optical tomography based on the equation of radiative transfer*, SIAM J. Sci. Comput., 28 (2006), pp. 1463–1489, https://doi.org/10.1137/040619193.
[40] L. F. Ricketson and A. J. Cerfon, *Sparse grid techniques for particle-in-cell schemes*, Plasma Phys. Control Fusion, 59 (2016), 024002.
[41] Y. Yang, D. Silantyev, and R. Caflisch, *Adjoint DSMC for nonlinear spatially-homogeneous Boltzmann equation with a general collision model*, J. Comput. Phys., (2023), 112247.