



Title	Quantum algorithm for the Vlasov simulation of the large-scale structure formation with massive neutrinos
Author(s)	Miyamoto, Koichi; Yamazaki, Soichiro; Uchida, Fumio et al.
Citation	Physical Review Research. 2024, 6(1), p. 013200
Version Type	VoR
URL	https://hdl.handle.net/11094/94885
rights	This article is licensed under a Creative Commons Attribution 4.0 International License.
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

Quantum algorithm for the Vlasov simulation of the large-scale structure formation with massive neutrinos

Koichi Miyamoto^{1,*}, Soichiro Yamazaki², Fumio Uchida^{2,3}, Kotaro Fujisawa^{4,3} and Naoki Yoshida^{2,5}

¹Center for Quantum Information and Quantum Biology, Osaka University, Toyonaka, Osaka 560-0043, Japan

²Department of Physics, Graduate School of Science, The University of Tokyo, Bunkyo-ku, Tokyo 113-0033, Japan

³Research Center for the Early Universe, Graduate School of Science, The University of Tokyo, Bunkyo-ku, Tokyo 113-0033, Japan

⁴Department of Liberal Arts, Tokyo University of Technology, Ota-ku, Tokyo 144-0051, Japan

⁵Kavli IPMU (WPI), The University of Tokyo Institutes for Advanced Study (UTIAS), The University of Tokyo, Chiba 277-8583, Japan



(Received 11 October 2023; accepted 28 January 2024; published 26 February 2024)

Investigating the cosmological implication of the fact that neutrino has finite mass is of importance for fundamental physics. In particular, massive neutrino affects the formation of the large-scale structure (LSS) of the universe, and, conversely, observations of the LSS can give constraints on the neutrino mass. Numerical simulations of the LSS formation including massive neutrino along with conventional cold dark matter is thus an important task. For this, calculating the neutrino distribution in the phase space by solving the Vlasov equation is a suitable approach, but it requires solving the PDE in the $(6 + 1)$ -dimensional space and is thus computationally demanding: Configuring n_{gr} grid points in each coordinate and n_t time grid points leads to $O(n_{\text{gr}}^6)$ memory space and $O(n_t n_{\text{gr}}^6)$ queries to the coefficients in the discretized PDE. We propose a quantum algorithm for this task. Linearizing the Vlasov equation by neglecting the relatively weak self-gravity of the neutrino, we perform the Hamiltonian simulation to produce quantum states that encode the phase-space distribution of neutrino. We also propose a way to extract the power spectrum of the neutrino density perturbations as classical data from the quantum state by quantum amplitude estimation with accuracy ϵ and query complexity of order $\tilde{O}[(n_{\text{gr}} + n_t)/\epsilon]$. Our method also reduces the space complexity to $O[\text{polylog}(n_{\text{gr}}/\epsilon)]$ in terms of the qubit number, while using quantum random access memories with $O(n_{\text{gr}}^3)$ entries. As far as we know, this is the first quantum algorithm for the LSS simulation that outputs the quantity of practical interest with guaranteed accuracy.

DOI: [10.1103/PhysRevResearch.6.013200](https://doi.org/10.1103/PhysRevResearch.6.013200)

I. INTRODUCTION

A. Background

Quantum computing is an emerging technology and has the potential to speed up numerical tasks intractable by classical computers, today's ordinary computers including supercomputers. Witnessing the recent rapid advance of quantum computing, people are now trying to find use cases of quantum computers with the quantum advantage in various fields. In this paper, we focus on the simulation of the *large-scale structure* (LSS) of the universe with *massive neutrino*, an important task in cosmology.

In the standard cosmological model, all the rich structures of the present-day universe formed through gravitational instability of tiny density fluctuations seeded in the early universe [1]. The structure at the largest scale probed by cosmological observations is called the LSS. The evolution and the resultant LSS have been shaped by the nature of the mysterious constituents of the universe. Interestingly, the

major components of the universe are largely unidentified. In terms of the energy fraction, about 69% is contributed by the so-called *dark energy*, perhaps some unknown form of energy different from matter, and about 26% is *dark matter* (DM), which is often thought to be unknown elementary particles that are not predicted to exist in the Standard Model (SM) of particle physics [2]. Observations of the cosmic LSS can shed light on the nature of such dark components and, eventually, provide an important clue on physics beyond the SM.

An important issue toward a better understanding of particle physics through the LSS is that neutrinos have finite masses. Neutrinos are massless particles in the SM, but the detection of neutrino flavor oscillation [3] has now established that they have nonzero mass. The current constraint from the neutrino oscillation experiments is given as the lower bound on the neutrino mass. Although the estimated mass is much smaller than other SM particles such as electron with 0.51 MeV, the nonzero neutrino mass is definite evidence that there is physics beyond the SM. Intriguingly, astronomical observations of the cosmic LSS provide independent constraints on the neutrino mass.¹ Neutrinos produced in the early universe

*miyamoto.kouichi.qiqb@osaka-u.ac.jp

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

¹We also note that there are some researches for constraining the neutrino mass through other types of particle physics experiments or astronomical observations [4–7].

exist even today as *relics* and constitute a part of matter component, along with *cold dark matter* (CDM), the conventional picture of DM as particles with much larger mass, e.g., TeV order for supersymmetric particles. In the LSS formation, neutrino behaves differently from CDM and ordinary matter (baryons), which we hereafter call CDM collectively since they behave similarly under the action of mutual gravity. Neutrinos have nonrelativistic but extremely large velocities and thus stream almost freely in the gravitational potential of CDM. Hence neutrinos used to be thought as *hot dark matter* (HDM) distinguishing from CDM. The distribution and gravitational dynamics of neutrinos affect the formation of LSS differently from the conventional picture with CDM only. This gives hope that, by comparing the theoretical prediction on the LSS with massive neutrino to the results of cosmological observations, one can derive constraints on the neutrino mass. We also note that some particle physics models beyond the SM predict other light elementary particles such as axion [8–12], and they can behave as HDM and be constrained from LSS observations, too [13].

This background gives us a strong motivation for the numerical simulation of the LSS with massive neutrino. However, it is a computationally demanding task. N -body simulations are often chosen as a powerful method for the study of LSS, where we employ a large number of particles and follow their gravitational dynamics. Massive neutrinos have been also incorporated into this method, by either adopting approximate corrections or by directly represented by “light” particles [14–23]. However, there remains a concern that such N -body simulations with massive neutrino may lead to imprecise results because, unlike the conventional matter components, CDM, and baryons, neutrinos have typically a much larger velocity dispersion, and the so-called shot noise can be significant unless an extremely large number of simulation particles are employed.

An alternative approach is the Vlasov simulation, that is, solving directly the collisionless Boltzmann equation, also known as the Vlasov equation²:

$$\frac{\partial f}{\partial t}(t, \mathbf{x}, \mathbf{v}) + \mathbf{v} \cdot \frac{\partial f}{\partial \mathbf{x}}(t, \mathbf{x}, \mathbf{v}) + \mathbf{F}(t, \mathbf{x}) \cdot \frac{\partial f}{\partial \mathbf{v}}(t, \mathbf{x}, \mathbf{v}) = 0, \quad (1)$$

where $f(t, \mathbf{x}, \mathbf{v})$ is the neutrino’s distribution function in the six-dimensional (6D) phase space consisting of the 3D position \mathbf{x} and the 3D velocity \mathbf{v} at time t and $\mathbf{F}(t, \mathbf{x})$ is the gravitational force per unit mass on a neutrino at position \mathbf{x} at time t . However, solving this is also a challenging task. Since there is no exact analytic solution of Eq. (1) in realistic settings, we need to resort to some numerical method. A straightforward way to solve a partial differential equation (PDE) is discretization: Setting grid points in the phase space, we can convert the PDE to a linear system of ordinary differential equations (ODEs) and then apply some algorithm

for solving ODE. This approach, though, has limited feasibility because of the so-called *curse of dimensionality*. If we take n_{gr} grid points in one direction, then the total grid number in the 6D phase space becomes n_{gr}^6 and so does the dimension of the ODE. Then, in the time integration of the ODE, the memory space used is of order $O(n_{\text{gr}}^6)$, and if we take n_t time steps in the integration, then $O(n_{\text{gr}}^6 n_t)$ queries to the coefficients in the ODE are made in total. This sixth-order scaling makes the Vlasov simulation heavier than the N -body simulation, where we deal with equations of motion in the 3D space. Although there are some studies that try to solve Eq. (1) by supercomputers [24–26], the room to increase the grid number to improve accuracy is limited.

B. Our contribution

Motivated by the above things, in this paper, we propose the quantum algorithm for the Vlasov simulation of neutrino run on a fault-tolerant quantum computer (FTQC).

Our first key observation is that we can reduce Eq. (1) to the form to which *Hamiltonian simulation* [27–36] can be applied. At first glance, applying a quantum algorithm to Eq. (1) seems difficult, since it is a nonlinear PDE. \mathbf{F} is composed of not only the gravity from CDM but also the self-gravity of neutrino and thus depends on f , which makes Eq. (1) nonlinear. Because of the unitarity of quantum operations, using quantum computing for nonlinear problems is not straightforward. In fact, for solving differential equations, most existing quantum algorithms are dedicated to linear ones [37–55], and those for nonlinear ones have application conditions such as the nonlinear term needing to be small in some sense [50,56–62]. Nevertheless, we can approximately transform Eq. (1) into a linear PDE as follows. Since neutrino accounts for a much smaller fraction than CDM, we can neglect the gravity from neutrino to itself and CDM unless the neutrino density is extremely nonuniform. Then, we obtain the gravitational field $\mathbf{F}_{\text{CDM}}(t, \mathbf{x})$ by CDM using, e.g., the N -body simulation in advance and approximate $\mathbf{F}(t, \mathbf{x})$ with $\mathbf{F}_{\text{CDM}}(t, \mathbf{x})$. The resulting linear PDE can be transformed into the ODE by discretization. Importantly, the ODE has an antisymmetric coefficient matrix A and thus is considered as a Schrödinger equation with the Hermitian iA being the Hamiltonian. We then apply the Hamiltonian simulation, a methodology to generate a quantum state evolved under a given Hamiltonian, and yield the quantum state $|\mathbf{f}(T)\rangle$ encoding the value of f on the grid points at the terminal time T in the amplitudes. As seen later, this takes the $\tilde{O}(n_{\text{gr}} + n_t)^3$ complexity in terms of the number of queries to the oracle to access the entries in A , which indicates a large speedup from $O(n_{\text{gr}}^6 n_t)$.

It should be noted that extracting some quantities of interest from the quantum state encoding f in the amplitudes is another issue [63]. Then we also present how to obtain a typical target quantity in the LSS simulation, the power spectrum $P_v(k)$ of the neutrino density perturbation, which indicates the magnitude of the perturbation at the specified scale k , from

²Here we do not explicitly take into account cosmic expansion and hereafter we consider integration over a sufficiently short time so that the expansion parameter does not vary significantly. We leave the formulation including cosmic expansion as our future work.

³ $\tilde{O}(\dots)$ hides the logarithmic factors in the Landau’s big-O notation.

the quantum state. This is done by some additional unitary operations on $|\mathbf{f}(T)\rangle$ and quantum amplitude estimation (QAE) [64]. In total, the proposed method outputs an estimate on the power spectrum of accuracy ϵ with $\tilde{O}[(n_{\text{gr}} + n_t)/\epsilon]$ query complexity.

We also note that the proposed quantum algorithm can provide the advantage on space complexity, too. In the above task for calculating $P_v(k)$, our algorithm uses $O(\log^{5/2}(n_{\text{gr}}/\epsilon))$ qubits in total, which is exponentially smaller than the $O(n_{\text{gr}}^6)$ memory space in the aforementioned classical method. However, our method uses some quantum random access memories (QRAMs) [65] with $O(n_{\text{gr}}^3)$ entries to store the precomputed CDM gravitational field $\mathbf{F}_{\text{CDM}}(t, \mathbf{x})$ on the grid points.

In addition, the proposed method can cope with the following complication in the practical problem setting. Reflecting the stochastic nature of the initial value of the perturbation, $P_v(k)$ is defined as an ensemble average. In the classical LSS simulation, it is estimated through multiple runs of the N -body or Vlasov simulation with different initial conditions. In our quantum method, we do not need multiple runs: We can generate the quantum state that encodes the results from different initial conditions in superposition and estimate $P_v(k)$ with the single quantum state.

C. Comparison to previous studies

Quantum algorithms for solving the Vlasov equation have been considered in previous studies. Most of them give discussions in the context of plasma physics, and there has been no study focusing on the gravitational LSS simulation with massive neutrino as far as we know. Also in the technical aspect, the existing studies are in directions different from ours.

References [66–68] presented FTQC algorithms to solve the Vlasov-Poisson equation, in which the force induced by the particles themselves is considered, in the context of plasma physics. They considered the linearized Vlasov-Poisson equation, which describes the perturbative solution on the zeroth-order analytical solution. Compared to this, our approach for linearization, which approximates the force field by only that from CDM based on the small neutrino mass fraction, is different in the following points. First, the CDM gravity is externally given by, e.g., N -body simulation, in which the nonlinear dynamics is incorporated, and it is reflected to the neutrino dynamics solved in the current approach. Second, the condition for our approximation to be valid is that neutrino density nonuniformity is not so large that the neutrino self-gravity is negligible compared to the CDM gravity. This is different from the condition for the perturbative approach to be valid, the perturbation being smaller than the zeroth-order solution, which is the neutrino background density in this case. The fractional mass (and energy) density of neutrino is less than 1% of that of CDM, as given in Eq. (3) later. Also, recent fully nonlinear simulations show the maximum local over-density of neutrino is of the order unity [24,69]. These values depend slightly on the neutrino mass but remain extremely small compared to those of CDM whose maximum over-densities reach 500–1000 in nonlinear

“halos.” Hence the local gravitational potential is dominated by CDM in most of the regime of interest.

As a difference from the other aspect, although Refs. [66–68] also used the Hamiltonian simulation, they worked in the Fourier space instead of working in the position space \mathbf{x} like us. This does not fit our setting that the position-wise CDM gravity $\mathbf{F}_{\text{CDM}}(t, \mathbf{x})$ are given.

There are also studies on quantum algorithms to solve the Vlasov equation in the nonlinear form. Reference [70] considers approaches via some linearization methods such as Carleman linearization [71]. However, like the existing quantum solvers for nonlinear differential equations [50,56–62], the method in Ref. [70] has some application conditions such as weakness of the nonlinearity. If such conditions are satisfied, then quantum nonlinear differential equation solvers based on Carleman linearization provide a solution with space and query complexities of the same order as linear ones except for some logarithmic overheads [50]. Reference [72] summarizes the prospect of quantum algorithms to solve plasma dynamics in both linearized and nonlinear settings and also mentions the variational quantum algorithms (VQAs) [73]. They might be able to run on noisy intermediate-scale quantum devices in the near term, but they are genuinely heuristic.

When it comes to the simulation of self-gravitating systems, in which LSS simulation is included, Ref. [74] proposes a VQA to solve the nonlinear governing equation. Reference [75] also presents a VQA, with the fuzzy DM [76], a specific scenario for DM, in mind. Although their numerical experiment shows a promising result in the proof-of-concept problem, it is unclear whether their methods scale to the larger problem. Reference [77] proposes an FTQC algorithm for the Vlasov equation based on the reservoir method in the context of fluid dynamics, and Ref. [78] proposes a similar algorithm for self-gravitating systems. By their method, the Vlasov equation is simulated by appropriately arranging quantum circuits performing increment and decrement operations. While classically controlling the arrangement of the quantum circuits, it allows the delegation of six-dimensional operations to quantum computation. As a result, their method can reduce the complexity scaling on n_{gr} to $O(n_{\text{gr}}^3)$, and thus our method achieves larger speedup.

We also comment that the method to extract the power spectrum from the quantum state has not been proposed to our knowledge.

D. Organization

The rest of this paper is organized as follows. In Sec. II, we will explain some basics of the LSS simulation and quantum algorithms used in this paper. Section III is the main part, where we will explain each part of our method one by one: discretizing the Vlasov equation, obtaining the solution-encoding quantum state by Hamiltonian simulation, extracting the power spectrum from the quantum state by QAE, coping with the ensemble average, and so on. To illustrate our algorithm, we present a demonstrative numerical experiment on the Hamiltonian simulation-based time evolution of $f(t, \mathbf{x}, \mathbf{v})$ in Sec. IV. Section V summarizes this paper.

II. PRELIMINARIES

A. Notation

We use I to denote an identity operator. To avoid ambiguity, we may write it as I_n with $n \in \mathbb{N}$ when its size is $n \times n$. We define \mathbb{R}_+ by the set of all positive real numbers. For every $n \in \mathbb{N}$, we define $[n]_0 := \{0, 1, \dots, n-1\}$. If a matrix A has at most $s \in \mathbb{N}$ nonzero elements in each row and each column, then we say that A is s -sparse and the sparsity of A is s .

For a vector $\mathbf{x} \in \mathbb{C}^n$, $\|\mathbf{x}\|$ denotes its Euclidean norm. For an (unnormalized) quantum state $|\psi\rangle$ on a multiqubit system, $\|\psi\rangle\|$ denotes the Euclidean norm of its state vector. For a matrix $A \in \mathbb{C}^{m \times n}$, $\|A\|$ denotes its spectral norm, and $\|A\|_{\max}$ denotes its max norm, the maximum of the absolute values of its entries.

For $\epsilon \in \mathbb{R}_+$, we say that $x' \in \mathbb{R}$ is an ϵ approximation of $x \in \mathbb{R}$ if $|x' - x| \leq \epsilon$ holds. We use \log and \lg for the natural and binary logarithms, respectively.

We label the position of an entry in a vector and the row and column in a matrix with an integer starting from 0. For example, we write a vector $\mathbf{v} \in \mathbb{C}^n$ entrywise as $\mathbf{v} = (v_0, v_1, \dots, v_{n-1})$ and call v_i the i th entry, and a matrix $A \in \mathbb{C}^{m \times n}$ as $A = (a_{ij})_{i \in [m]_0, j \in [n]_0}$ and call a_{ij} the (i, j) -th entry.

We denote by $\mathbf{1}_C$ the indicator function, which takes 1 if the condition C is satisfied and 0 otherwise.

B. Simulation of large-scale structure formation with massive neutrinos

Here we briefly explain some basics of the LSS simulation and review recent developments of simulations that include massive neutrinos.

1. N -body simulation

N -body simulations are often employed to simulate the LSS formation. In this approach, the mass distribution is represented by a collection of a large but tractable number of *superparticles*. By populating the simulation volume with N_p superparticles, we numerically solve the time-evolution equation for them:

$$\frac{d^2}{dt^2} \mathbf{x}_i(t) = \mathbf{F}_i(\{\mathbf{x}_j(t)\}), \quad (2)$$

where $\mathbf{x}_i(t)$ is the space coordinate of the i th superparticle and \mathbf{F}_i is the gravitational force per unit mass on it caused by the other ones. Then the important statistical quantities such as the power spectrum of the density perturbation are estimated from the distribution of the superparticles. When simulating the LSS that consists of multiple species of particles, e.g., CDM and neutrino, one may naively attempt to use different sets of superparticles for each component.

We need to distribute the particles sufficiently densely in the 3D space in order to estimate the quantities of interest accurately. A common setting on N_p is $N_p = n_p^3$ with large n_p , say, $O(10^2)$, which means N_p can be of order of million. Thus, the N -body simulation is a rather heavy calculation, which may need a supercomputer in classical computing. Nevertheless, it is tractable compared to the Vlasov simulation, which deals with the 6D space-velocity phase space. In particular, it is commonly considered that if the velocity dispersion of the

particles is small as is the case for CDM, then the N -body simulation yields accurate result. On the other hand, for particles with large velocity dispersion such as massive neutrino, it is difficult to use a sufficiently large number of superparticles so that the distribution in the velocity space is represented, and often the N -body simulation may lead to imprecise results.

2. Vlasov simulation for massive neutrinos

In light of the above issue, a more desirable approach for LSS simulation with massive neutrinos is solving the Vlasov equation (1) directly. In particular, for neutrino, we can simplify Eq. (1) using the fact that it accounts for only a small part of the matter: The ratio of the neutrino energy fraction Ω_ν to Ω_m that of the whole of the matter is [24,79]

$$\frac{\Omega_\nu}{\Omega_m} = \frac{M_\nu/93.14 \text{ eV}}{\Omega_m h^2} \simeq 7.6 \times 10^{-3} \times \frac{M_\nu}{0.1 \text{ eV}}, \quad (3)$$

where h is the present value of the Hubble parameter in units of 100 km/s and we use $\Omega_m h^2 \simeq 0.14$ [2]. Based on this fact, we hereafter neglect the self-gravity of neutrino. Then, Eq. (1) approximately boils down to the linear equation,

$$\frac{\partial f}{\partial t}(t, \mathbf{x}, \mathbf{v}) + \mathbf{v} \cdot \frac{\partial f}{\partial \mathbf{x}}(t, \mathbf{x}, \mathbf{v}) + \mathbf{F}_{\text{CDM}}(t, \mathbf{x}) \cdot \frac{\partial f}{\partial \mathbf{v}}(t, \mathbf{x}, \mathbf{v}) = 0, \quad (4)$$

where

$$\mathbf{F}_{\text{CDM}}(t, \mathbf{x}) = (F_{\text{CDM},x}(t, \mathbf{x}), F_{\text{CDM},y}(t, \mathbf{x}), F_{\text{CDM},z}(t, \mathbf{x})) \quad (5)$$

is the gravitational force per unit mass on a neutrino particle at position \mathbf{x} and time t exerted by CDM. This type of approximation can be found also in previous studies [80]. We also neglect the gravitational back reaction from neutrino to CDM and then obtain \mathbf{F}_{CDM} from the dynamics of CDM only by, e.g., the N -body simulation. In other words, we consider the neutrino dynamics given the gravity by CDM as an external force field. This leads to the difference between the equation we try to solve and those in the previous studies on the Vlasov-Poisson equation in plasma physics and LSS simulation [66–68,70,72,74]. Although both consider the Vlasov equation (1) for some particles, in the latter, the force field \mathbf{F} is generated by the particles themselves and given through the Poisson equation, and thus the system of the Vlasov and Poisson equations is solved. On the other hand, in our setting, \mathbf{F} is explicitly given as a known function \mathbf{F}_{CDM} , and only Eq. (4) is solved.

Solving Eq. (4) numerically is still a difficult task. As explained in Introduction, taking n_{gr} grid points in each of the six directions in the phase space leads to the total number of the grid points being n_{gr}^6 . The total grid number rapidly increases with n_{gr} , and so do the query and space complexity in this approach. Although there are some studies in this direction⁴ that use supercomputers and take hundreds to thousands of grids in each position direction and tens of grids in each

⁴In Refs. [24–26], taking into account the gravity by neutrino to itself and CDM, the authors performed the N -body simulation for CDM and the Vlasov simulation for neutrino in combination, unlike this paper neglecting the neutrino gravity.

velocity direction [24–26], increasing the grid point number largely is hard to desire in classical computing.

3. Neutrino power spectrum

Once we get the neutrino distribution function $f(T, \mathbf{x}, \mathbf{v})$ at the terminal time T of the simulation, we can derive the quantities of interest from it as follows.

A quantity we are typically interested in is the *neutrino power spectrum* $P_v(k)$. It is defined as

$$\langle \tilde{\delta}_v(T, \mathbf{k}) \tilde{\delta}_v^*(T, \mathbf{k}') \rangle = (2\pi)^3 P_v(k) \delta^{(3)}(\mathbf{k} - \mathbf{k}'). \quad (6)$$

Here $\tilde{\delta}_v(t, \mathbf{k}) := \int \delta_v(t, \mathbf{x}) e^{-i\mathbf{k} \cdot \mathbf{x}} d^3\mathbf{x}$ is the Fourier component of the neutrino density perturbation

$$\delta_v(t, \mathbf{x}) := \frac{\rho_v(t, \mathbf{x}) - \bar{\rho}_v(t)}{\bar{\rho}_v(t)} \quad (7)$$

at time t , where

$$\rho_v(t, \mathbf{x}) := \int f(t, \mathbf{x}, \mathbf{v}) d^3\mathbf{v} \quad (8)$$

is the neutrino density and $\bar{\rho}_v(t)$ is its average with respect to position. $\langle \cdot \rangle$ denotes the ensemble average with respect to the randomness of the inflationary perturbation, which is assumed to be homogeneous. $\delta^{(3)}(\cdot)$ denotes the 3D Dirac's delta function. Because of the assumption that the universe is isotropic, P_v depends on \mathbf{k} only through $k := \|\mathbf{k}\|$, the norm of the wave-number vector.

Although neutrino and CDM behave as nonrelativistic matters in the present universe, the fluctuations of the neutrino density and the CDM density leave different imprints in cosmological observations. It is implied that the galaxy number density traces the mass density which is mostly contributed by CDM [19,20,81–86]. The effect of neutrino is imprinted in LSS over a wide range of length scales [1]. Combining the results from, for instance, galaxy surveys and gravitational lensing observations, we can obtain the information on neutrino distribution and on the shape of P_v . Comparing this to the result of the Vlasov simulation, we eventually obtain constraints on the neutrino mass.

C. Quantum algorithms

We now introduce some basics of quantum computing and quantum algorithms used as the building blocks in this paper. For more general basics on quantum computing, we refer to Ref. [87].

1. Arithmetic circuits

In this paper, we consider computation on the system with multiple quantum registers. We use the fixed-point binary representation for real numbers and, for each $x \in \mathbb{R}$, we denote by $|x\rangle$ the computational basis state on a quantum register where the bit string corresponds to the binary representation of x . We assume that every register has a sufficient number of qubits and thus neglect errors by finite-precision representation. For $z = x + iy \in \mathbb{C}$, we define $|z\rangle := |x\rangle |y\rangle$, and the real and imaginary parts are held on separate registers.

We can perform arithmetic operations on numbers represented on qubits. For example, we can implement

quantum circuits for four basic arithmetic operations: addition $O_{\text{add}} : |a\rangle |b\rangle |0\rangle \mapsto |a\rangle |b\rangle |a+b\rangle$; subtraction $O_{\text{sub}} : |a\rangle |b\rangle |0\rangle \mapsto |a\rangle |b\rangle |a-b\rangle$; multiplication $O_{\text{mul}} : |a\rangle |b\rangle |0\rangle \mapsto |a\rangle |b\rangle |ab\rangle$; and division $O_{\text{div}} : |a\rangle |b\rangle |0\rangle |0\rangle \mapsto |a\rangle |b\rangle |q\rangle |r\rangle$, where $a, b \in \mathbb{Z}$ and q and r are the quotient and remainder of a/b . For concrete implementations, see Ref. [88] and the references therein. In the finite-precision binary representation, these operations are immediately extended to those for real numbers and then complex numbers. Hereafter, we collectively call these circuits arithmetic circuits.

2. Block encoding

Block encoding, which was first introduced in Ref. [36], is a technique to encode a nonunitary matrix as an upper-left block of a unitary matrix. Overcoming the restriction of the unitarity of quantum circuits, this block-encoding technique makes various types of matrix computation efficiently implementable and thus is widely used in various quantum algorithms as a fundamental building block. We will give the rigorous definition of the block encoding in Appendix A 1. In some situations, there are some ways to implement the block-encoding unitary. For example, if we have sparse-access to a matrix A , that is, if A is sparse and we have access to oracles that provide positions and values of nonzero entries in A , then we can construct the block encoding of A , whose implementation cost is shown as Theorem 3 in Appendix A 1. The Vlasov simulation applies to this case, since, as we will see later, A in this case corresponds to the sparse matrix resulting from discretizing the Vlasov equation and we can calculate its entries for given $\mathbf{F}_{\text{CDM}}(t, \mathbf{x})$.

3. Hamiltonian simulation

With H a Hamiltonian of a multiqubit system and an initial state $|\psi_0\rangle$, the Hamiltonian simulation is the task to generate the quantum state $|\psi_t\rangle$ after the evolution under H for time t . In other words, we aim to implement the time-evolution operator $\exp(-iHt)$ as a quantum circuit. This task has been investigated as a core quantum algorithm for a long time [27,28,31], and recently it has been shown that, given a block encoding of H , we can implement a quantum circuit for the simulation of H with the optimal query complexity [35,89–91]. Roughly speaking, we can construct a block encoding of $\exp(-iHt)$ with $O(\|H\|t)$ queries to a block encoding of H . The rigorous statement is given as Theorem 4 in Appendix A 1.

4. Quantum amplitude estimation

Even if we obtain some quantum state by a quantum algorithm such as Hamiltonian simulation, extracting information of interest from the state is another issue [63]. There is no general prescription, unfortunately, and we need to devise a way to extract the quantity we want on a case-by-case basis. In some cases, the necessary quantity is encoded in the quantum state $|\Phi\rangle$ as the amplitude of a specific basis state, and then we can use the quantum algorithm called QAE [64] to estimate the amplitude. Roughly speaking, we can estimate the amplitude with accuracy ϵ , using the quantum circuit to generate $|\Phi\rangle$ $O(1/\epsilon)$ times. The rigorous statement

on the query complexity of QAE is given as Theorem 5 in Appendix A 1.

5. Quantum random access memory

As we will see later, in the algorithm we propose, we need to load the data obtained outside the quantum algorithm onto the quantum register. A QRAM [65] is a device for such an aim. Specifically, we assume that, given a set of N real numbers $\mathcal{X} = \{x_0, \dots, x_{N-1}\}$, we can implement the unitary operation on a two-register system like

$$U_{\mathcal{X}} \sum_{i=0}^{N-1} \alpha_i |i\rangle |0\rangle = \sum_{i=0}^{N-1} \alpha_i |i\rangle |x_i\rangle \quad (9)$$

for any $\alpha_0, \dots, \alpha_{N-1} \in \mathbb{C}$ such that $\sum_{i=0}^{N-1} |\alpha_i|^2 = 1$.

Originally, a QRAM was proposed as a dedicated device for data loading, consisting of “atoms” that emit “photons” and enabling the parallel data loading in superposition in time $O(\log N)$ [65]. However, some issues on the implementability of such a device have been pointed out, for example, the feasibility of error correction [92,93]. On the other hand, there are some proposals on quantum circuit-based QRAM [94–98], into which we can incorporate error correction. However, proposed circuit-based implementations need $O(N)$ gates and qubits, which reduces the feasibility for large N , although the circuit depth can be logarithmic in N . In any case, implementing a QRAM is highly challenging, and in this paper, we simply assume the availability of the operation (9).

III. QUANTUM ALGORITHM FOR VLASOV SIMULATION OF MASSIVE NEUTRINOS

Now we present the quantum algorithm for the neutrino Vlasov simulation, including the procedure to estimate the neutrino power spectrum.

A. Discretizing the Vlasov equation

Let us start from discretization, which transforms Eq. (4) to a set of ODEs.

We set the range $[0, L]$ for each of the position coordinates $\mathbf{x} = (x, y, z)$ and the range $[-V, V]$ for each of the velocity coordinates $\mathbf{v} = (u, v, w)$ with sufficiently large $L, V \in \mathbb{R}_+$. The region in the phase space we consider is thus $\mathcal{V} := [0, L] \times [0, L] \times [0, L] \times [-V, V] \times [-V, V] \times [-V, V]$. As a common boundary condition, we impose the periodic condition for each position coordinate, and the Dirichlet condition $f = 0$ for each velocity coordinate. Then, we introduce n_{gr} grid points in each coordinate: We set

$$x_{i_x} = i_x \Delta_x, \quad \Delta_x := \frac{L}{n_{\text{gr}}}, \quad i_x \in [n_{\text{gr}}]_0 \quad (10)$$

for x , and similarly for y and z , and

$$u_{i_u} = -V + (i_u + 1)\Delta_v, \quad \Delta_v := \frac{2V}{n_{\text{gr}} + 1}, \quad i_u \in [n_{\text{gr}}]_0 \quad (11)$$

for u , and similarly for v and w . The grid points in the 6D phase space are

$$(x_{i_x}, y_{i_y}, z_{i_z}, u_{i_u}, v_{i_v}, w_{i_w}), \quad i_x, i_y, i_z, i_u, i_v, i_w \in [n_{\text{gr}}]_0. \quad (12)$$

We hereafter label the grid points with vectors of indexes:

$$\mathbf{i} = (i_x, i_y, i_z, i_u, i_v, i_w) \in \mathcal{I}_6 := \underbrace{[n_{\text{gr}}]_0 \times \dots \times [n_{\text{gr}}]_0}_6. \quad (13)$$

Sometimes, we also use the label $i \in [N_{\text{gr}}]_0$, which is related to \mathbf{i} as

$$i = \sigma(\mathbf{i}) := i_x + i_y \times n_{\text{gr}} + i_z \times n_{\text{gr}}^2 + i_u \times n_{\text{gr}}^3 + i_v \times n_{\text{gr}}^4 + i_w \times n_{\text{gr}}^5. \quad (14)$$

Here $N_{\text{gr}} := n_{\text{gr}}^6$ is the total number of the grid points in the phase space. For the later convenience, we assume that $n_{\text{gr}} = 2^{m_{\text{gr}}}$ with some $m_{\text{gr}} \in \mathbb{N}$.

Although we take the same number of grid points in each of position and velocity coordinates just for simplicity, we can take different numbers and such a generalization is straightforward. In fact, in simulations in Refs. [24–26], the grid number in each position coordinate is different from that in each velocity coordinate.

Using this grid, we approximate the partial derivative with the central difference, that is,

$$\frac{\partial}{\partial x} f(t, \mathbf{x}, \mathbf{v}) \simeq \frac{f(t, \mathbf{x} + \Delta_x \mathbf{e}_x, \mathbf{v}) - f(t, \mathbf{x} - \Delta_x \mathbf{e}_x, \mathbf{v})}{2\Delta_x}, \quad (15)$$

where $\mathbf{e}_x = (1, 0, 0)$, and similarly for the derivatives with respect to y, z, u, v and w . Then Eq. (4) is transformed into

$$\frac{d}{dt} \mathbf{f}(t) = A(t) \mathbf{f}(t). \quad (16)$$

$\mathbf{f}(t)$ is a vector in $\mathbb{R}^{N_{\text{gr}}}$, which can be seen as $\underbrace{\mathbb{R}^{n_{\text{gr}}} \times \dots \times \mathbb{R}^{n_{\text{gr}}}}_6$.

In light of this, we label each entry in $\mathbf{f}(t)$ with $\mathbf{i} \in \mathcal{I}_6$, as well as $i \in [N_{\text{gr}}]_0$ in Eq. (14). Although we define $\mathbf{f}(t)$ as a solution of Eq. (16) with some initial value, we expect that the i th entry of $\mathbf{f}(t)$ approximates the value of f at time t on the i th grid point:

$$f_i(t) \simeq f(t, x_{i_x}, y_{i_y}, z_{i_z}, u_{i_u}, v_{i_v}, w_{i_w}). \quad (17)$$

$A(t)$ is a $N_{\text{gr}} \times N_{\text{gr}}$ matrix and we again label its rows and columns with both $\mathbf{i} \in \mathcal{I}_6$ and $i \in [N_{\text{gr}}]_0$. $A(t)$ is written as

$$A(t) = A_x + A_y + A_z + A_u(t) + A_v(t) + A_w(t). \quad (18)$$

A_x is defined as

$$A_x = -D_{\text{per}} \otimes I_{n_{\text{gr}}} \otimes I_{n_{\text{gr}}} \otimes E_u \otimes I_{n_{\text{gr}}} \otimes I_{n_{\text{gr}}}, \quad (19)$$

where D_{per} and E_u are $n_{\text{gr}} \times n_{\text{gr}}$ matrices defined as

$$D_{\text{per}} := \begin{pmatrix} 0 & 1 & & & & -1 \\ -1 & 0 & 1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & -1 & 0 & 1 & \\ 1 & & & -1 & 0 & \end{pmatrix} \quad (20)$$

and

$$E_u := \begin{pmatrix} \frac{u_0}{2\Delta_x} & & & & & \\ & \ddots & & & & \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & \frac{u_{n_{\text{gr}}-1}}{2\Delta_x} & \end{pmatrix}, \quad (21)$$

respectively. A_y and A_z are defined similarly. $A_u(t)$ is a time-varying $N_{\text{gr}} \times N_{\text{gr}}$ matrix and its (\mathbf{i}, \mathbf{j}) entry is

$$(A_u(t))_{\mathbf{i}, \mathbf{j}} = \begin{cases} -\frac{F_{\text{CDM},x}(t, x_{i_x}, y_{i_y}, z_{i_z})}{2\Delta_v}; & \text{if } \mathbf{j} = \mathbf{i} + \mathbf{e}_u \\ \frac{F_{\text{CDM},x}(t, x_{i_x}, y_{i_y}, z_{i_z})}{2\Delta_v}; & \text{if } \mathbf{j} = \mathbf{i} - \mathbf{e}_u, \\ 0; & \text{otherwise} \end{cases} \quad (22)$$

where $\mathbf{e}_u := (0, 0, 0, 1, 0, 0)$; $A_v(t)$ and $A_w(t)$ are defined similarly.

We now comment on what are conserved in the time evolution of the ODE system (16). First, $\|\mathbf{f}(t)\|$ the norm of the solution is constant in time, which is seen as follows. A is antisymmetric, as is easily checked from the definition. Thus,

$$H(t) := iA(t) \quad (23)$$

is Hermitian, and we can regard Eq. (16) as the Schrödinger equation with the Hamiltonian H . As a property of the Schrödinger equation, $\|\mathbf{f}(t)\|$ is conserved. This means that in the ODE system (16), there is no instability such that the solution explodes. This is a merit of the current discretization as Eq. (15), which leads to the antisymmetry of A .

Second, $f_{\text{sum}}(t) := \sum_{\mathbf{i} \in \mathcal{I}_6} f_{\mathbf{i}}(t)$, the sum of all the entries in $\mathbf{f}(t)$ is approximately conserved. This is seen by noting that

$$\begin{aligned} \frac{d}{dt} f_{\text{sum}}(t) &= \sum_{\mathbf{i}, \mathbf{j} \in \mathcal{I}_6} (A(t))_{\mathbf{i}, \mathbf{j}} f_{\mathbf{j}}(t) \\ &= \sum_{\mathbf{i} \in \mathcal{B}_u^l} (A(t))_{\mathbf{i} + \mathbf{e}_u, \mathbf{i}} f_{\mathbf{i}}(t) + \sum_{\mathbf{i} \in \mathcal{B}_u^u} (A(t))_{\mathbf{i} - \mathbf{e}_u, \mathbf{i}} f_{\mathbf{i}}(t) \\ &\quad + (u \leftrightarrow v) + (u \leftrightarrow w), \end{aligned} \quad (24)$$

where $(u \leftrightarrow v)$ [respectively, $(u \leftrightarrow w)$] denotes the same as the first and second terms except u is replaced with v (respectively, w), and $\mathcal{B}_u^l := \{\mathbf{i} \in \mathcal{I}_6 \mid i_u = 0\}$, $\mathcal{B}_u^u := \{\mathbf{i} \in \mathcal{I}_6 \mid i_u = n_{\text{gr}} - 1\}$, and similarly defined $\mathcal{B}_v^l, \mathcal{B}_v^u, \mathcal{B}_w^l, \mathcal{B}_w^u$ are the sets of the indexes of the grid points on the boundaries in the velocity coordinates. That is, the time derivative of f_{sum} is almost vanishing except for the contributions from the boundary grids. The conservation of f_{sum} has the physical meaning that the number of the neutrino particles in the 6D simulation box is conserved, which is desirable in the simulation. The violation of the conservation corresponds to the escapes of the particles from the box, which is controllable by setting the box sufficiently large.

Although these conservation properties are desirable, there might be other properties the solution $\mathbf{f}(t)$ should have from the physical perspective. For example, since $\mathbf{f}(t)$ denotes the distribution density function in the phase space, it must be nonnegative, but, it is not clear that this is satisfied in the current approach. We leave incorporating some schemes guaranteeing the nonnegativity such as upwind difference into the quantum algorithm as future work.

B. Generating the solution-encoding quantum state

As seen above, we can regard Eq. (16) as the Schrödinger equation. If we rewrite it *a la* quantum mechanics, then it becomes

$$\frac{d}{dt} |\mathbf{f}(t)\rangle = -iH(t)|\mathbf{f}(t)\rangle, \quad (25)$$

where $|\mathbf{f}(t)\rangle$ is the quantum state encoding the value of $\mathbf{f}(t)$ in the amplitudes:

$$|\mathbf{f}(t)\rangle := \frac{1}{\|\mathbf{f}(t)\|} \sum_{\mathbf{i} \in \mathcal{I}_6} f_{\mathbf{i}}(t) |\mathbf{i}\rangle. \quad (26)$$

Here $|\mathbf{i}\rangle := |i_x\rangle |i_y\rangle |i_z\rangle |i_u\rangle |i_v\rangle |i_w\rangle$, which can be also seen as $|i\rangle$ under the correspondence between \mathbf{i} and i in Eq. (14), since concatenating the binary representations of i_x, \dots, i_w yields that of i .

Now let us consider generating the quantum state $|\mathbf{f}(T)\rangle$ encoding f at the terminal time T by Hamiltonian simulation. To apply this, we make the following assumption.

Assumption 1 (Piecewise time constancy of \mathbf{F}_{CDM} and oracles to access it). Let $T \in \mathbb{R}_+$ and $n_t \in \mathbb{N}$. Let $t_i = i_t \Delta_t$ for $i_t \in [n_t]_0$ with $\Delta_t := T/n_t$. Then, for any $\mathbf{i}_x := (i_x, i_y, i_z) \in \mathcal{I}_3 := [n_{\text{gr}}]_0 \times [n_{\text{gr}}]_0 \times [n_{\text{gr}}]_0$ and $i_t \in [n_t]_0$,

$$\begin{aligned} F_{\text{CDM},x}(t, \mathbf{x}_{\mathbf{i}_x}) &= F_{\text{CDM},x}^{i_t, \mathbf{i}_x} \\ F_{\text{CDM},y}(t, \mathbf{x}_{\mathbf{i}_x}) &= F_{\text{CDM},y}^{i_t, \mathbf{i}_x} \\ F_{\text{CDM},z}(t, \mathbf{x}_{\mathbf{i}_x}) &= F_{\text{CDM},z}^{i_t, \mathbf{i}_x}, \end{aligned} \quad (27)$$

holds for any $t \in [t_i, t_{i+1})$. Here $\mathbf{x}_{\mathbf{i}_x} := (x_{i_x}, y_{i_y}, z_{i_z})$, and, for each $(i_t, \mathbf{i}_x) \in [n_t]_0 \times \mathcal{I}_3$, $F_{\text{CDM},x}^{i_t, \mathbf{i}_x}$, $F_{\text{CDM},y}^{i_t, \mathbf{i}_x}$, and $F_{\text{CDM},z}^{i_t, \mathbf{i}_x}$ are some real numbers. Furthermore, for any $i_t \in [n_t]_0$, we are given accesses to the oracles $O_{F_{\text{CDM},x}}^{i_t}$, $O_{F_{\text{CDM},y}}^{i_t}$, and $O_{F_{\text{CDM},z}}^{i_t}$ that act as

$$\begin{aligned} O_{F_{\text{CDM},x}}^{i_t} |\mathbf{i}_x\rangle |0\rangle &= |\mathbf{i}_x\rangle |F_{\text{CDM},x}^{i_t, \mathbf{i}_x}\rangle \\ O_{F_{\text{CDM},y}}^{i_t} |\mathbf{i}_x\rangle |0\rangle &= |\mathbf{i}_x\rangle |F_{\text{CDM},y}^{i_t, \mathbf{i}_x}\rangle \\ O_{F_{\text{CDM},z}}^{i_t} |\mathbf{i}_x\rangle |0\rangle &= |\mathbf{i}_x\rangle |F_{\text{CDM},z}^{i_t, \mathbf{i}_x}\rangle \end{aligned} \quad (28)$$

for any $\mathbf{i}_x \in \mathcal{I}_3$, where $|\mathbf{i}_x\rangle := |i_x\rangle |i_y\rangle |i_z\rangle$.

The assumption that \mathbf{F}_{CDM} is piecewise constant in time matches the practical setting. From the N -body simulation, we obtain \mathbf{F}_{CDM} only on the discrete time points, since in the N -body simulation we discretize the time for time integration of the equation of motion, and interpolating them as Eq. (27) is a common approximation. We also note that, for this type of \mathbf{F}_{CDM} , the Hamiltonian $H(t)$ is also piecewise constant in time: for any $i_t \in [n_t]_0$,

$$H(t) = H_{i_t} := H(t_{i_t}) \quad (29)$$

holds for any $t \in [t_{i_t}, t_{i_t+1})$.

We also assume that we can prepare the quantum state encoding the initial value of $\mathbf{f}(t)$.

Assumption 2. For a given initial value $\mathbf{f}(0)$, we have an access to the oracle $O_{\mathbf{f}(0)}$ that acts as

$$O_{\mathbf{f}(0)} |0\rangle = |\mathbf{f}(0)\rangle. \quad (30)$$

We will discuss the implementation of the above oracles in Sec. III E.

Then, we can apply Hamiltonian simulation to generate $|\mathbf{f}(T)\rangle$.

Theorem 1. Suppose that Assumptions 1 and 2 hold. Then, for any $\epsilon \in (0, 1/2)$, we have a quantum circuit $U_{\mathbf{f}(T), \epsilon}$ on the $(2 \lg N_{\text{gr}} + 5)$ -qubit system that acts as

$$U_{\mathbf{f}(T), \epsilon} |0\rangle^{\otimes (2 \lg N_{\text{gr}} + 5)} = |0\rangle^{\otimes (\lg N_{\text{gr}} + 5)} |\mathbf{f}(T)\rangle + |\psi_{\text{gar}}\rangle, \quad (31)$$

where $|\mathbf{f}(T)\rangle$ is the quantum state that encodes the solution of Eq. (16) at time T with the initial value $\mathbf{f}(0)$ as Eq. (26), and $|\psi_{\text{gar}}\rangle$ is some unnormalized quantum state satisfying $\| |\psi_{\text{gar}}\rangle \| \leq \epsilon$. In $U_{\mathbf{f}(T), \epsilon}$, (controlled) $\{O_{F_{\text{CDM},x}}^i, O_{F_{\text{CDM},y}}^i, O_{F_{\text{CDM},z}}^i\}_{i_t}$, arithmetic circuits, and their inverses are queried

$$O\left[n_{\text{gr}}T \times \max\left\{\frac{V}{L}, \frac{F_{\text{max}}}{V}\right\} + n_t \log\left(\frac{n_t}{\epsilon}\right)\right] \quad (32)$$

times, $O_{\mathbf{f}(0)}$ is queried once, and

$$O\left[\log^{5/2}\left(\frac{n_{\text{gr}}T}{\epsilon} \times \max\left\{\frac{V}{L}, \frac{F_{\text{max}}}{V}\right\}\right)\right] \quad (33)$$

qubits are used including ancillary ones.

The proof of this theorem is given in Appendix A 2. Although we postpone the details of construction of $U_{\mathbf{f}(T), \epsilon}$, we now give a rough outline. For the piecewise constant $H(t)$, the solution of Eq. (25) is

$$|\mathbf{f}(T)\rangle = \exp(-i\Delta_t H_{n_t-1}) \cdots \exp(-i\Delta_t H_0) |\mathbf{f}(0)\rangle. \quad (34)$$

We generate this state via implementing $\exp(-i\Delta_t H_{i_t})$ by block encoding, which is possible with the oracles $O_{F_{\text{CDM},x}}^i$, etc.

We can simplify the query complexity bound (32) by taking into account how to set L and V in practice. To solve the Vlasov equation precisely, we set the considered region \mathcal{V} in the phase space so that it contains the region where the distribution function f takes a nonnegligible value. In other words, noting that f reflects the neutrino motion, we set \mathcal{V} so that no particle escapes from it. For this purpose, it is sufficient to set L and V , the side lengths of \mathcal{V} , as

$$VT \sim L, F_{\text{max}}T \sim V, \quad (35)$$

since in the simulation time T a particle traverses at most VT in the position coordinate and at most $F_{\text{max}}T$ in the velocity coordinate. Conversely, much larger values of L and V than these are too much. Thus, assuming Eq. (35), we can simplify Eq. (32) as

$$\tilde{O}(n_{\text{gr}} + n_t), \quad (36)$$

as announced in Introduction. Besides, Eq. (33) becomes

$$O\left[\log^{5/2}\left(\frac{n_{\text{gr}}}{\epsilon}\right)\right], \quad (37)$$

which means that the number of qubits used is only logarithmic.

C. Extracting the power spectra

Even though we can generate the quantum state $|\mathbf{f}(T)\rangle$ that encodes $\mathbf{f}(T)$, extracting some information of interest from it is another issue. Here we consider how to estimate the neutrino power spectrum from the quantum state. In the current setting of finite volume and discrete grid points, the power spectrum is given as [99]

$$P_\nu(\mathbf{k}_{\mathbf{i}_k}) = \langle |\tilde{\delta}_{\mathbf{i}_k}^\nu|^2 \rangle \quad (38)$$

for $\mathbf{k}_{\mathbf{i}_k} = \frac{2\pi}{L}\mathbf{i}_k$ with $\mathbf{i}_k \in \mathcal{I}_3$. Here

$$\begin{aligned} \tilde{\delta}_{\mathbf{i}_k}^\nu &:= \frac{1}{n_{\text{gr}}^3} \sum_{\mathbf{i}_x \in \mathcal{I}_3} \delta_{\mathbf{i}_x}^\nu \exp(i\mathbf{k}_{\mathbf{i}_k} \cdot \mathbf{x}_{\mathbf{i}_x}) \\ &= \frac{1}{n_{\text{gr}}^3} \sum_{\mathbf{i}_x \in \mathcal{I}_3} \delta_{\mathbf{i}_x}^\nu \exp\left(\frac{2\pi i \mathbf{i}_k \cdot \mathbf{i}_x}{n_{\text{gr}}}\right) \end{aligned} \quad (39)$$

is the discrete Fourier transform of

$$\delta_{\mathbf{i}_k}^\nu := \frac{\rho_{\mathbf{i}_k}^\nu - \bar{\rho}_\nu(T)}{\bar{\rho}_\nu(T)}, \quad (40)$$

the neutrino density perturbation at $\mathbf{x}_{\mathbf{i}_k}$ the grid point in the position space.

$$\rho_{\mathbf{i}_k}^\nu := \sum_{\mathbf{i}_v \in \mathcal{I}_3} f_{(\mathbf{i}_k, \mathbf{i}_v)}(T) \Delta_v^3 \quad (41)$$

is the neutrino density at the position space grid point, calculated by integrating f with respect to the velocity coordinates in the phase space. In Eq. (41), $(\mathbf{i}_k, \mathbf{i}_v) \in \mathcal{I}_6$ is made by concatenating $\mathbf{i}_k, \mathbf{i}_v \in \mathcal{I}_3$.

Leaving how to take the ensemble average in Eq. (38) to Sec. IIID, we now consider how to estimate $|\tilde{\delta}_{\mathbf{i}_k}^\nu|^2$ from $|\mathbf{f}(T)\rangle$. Formally, we have the following theorem.

Theorem 2. Suppose that Assumptions 1 and 2 hold. Suppose that we are given the value of

$$C := \frac{\frac{1}{n_{\text{gr}}} (f_{\text{sum}}(T))^2}{\|\mathbf{f}(T)\|^2}. \quad (42)$$

Let $\epsilon \in (0, 1/2)$, $\delta \in (0, 1)$, and $\mathbf{i}_k \in \mathcal{I}_3 \setminus \{(0, 0, 0)\}$. Then there exists a quantum algorithm that, with probability at least $1 - \delta$, outputs an ϵ -approximation of $|\tilde{\delta}_{\mathbf{i}_k}^\nu|^2$ with $\mathbf{f}(T)$ the solution of Eq. (16). In this algorithm, (controlled) $\{O_{F_{\text{CDM},x}}^i, O_{F_{\text{CDM},y}}^i, O_{F_{\text{CDM},z}}^i\}_{i_t}$, arithmetic circuits, and their inverses are queried

$$O\left\{\left[n_{\text{gr}}T \times \max\left\{\frac{V}{L}, \frac{F_{\text{max}}}{V}\right\} + n_t \log\left(\frac{n_t}{C\epsilon}\right)\right] \frac{1}{C\epsilon} \log\left(\frac{1}{\delta}\right)\right\} \quad (43)$$

times, $O_{\mathbf{f}(0)}$ is queried

$$O\left[\frac{1}{C\epsilon} \log\left(\frac{1}{\delta}\right)\right] \quad (44)$$

times, and

$$O\left[\log^{5/2}\left(\frac{n_{\text{gr}}T}{C\epsilon} \times \max\left\{\frac{V}{L}, \frac{F_{\text{max}}}{V}\right\}\right)\right] \quad (45)$$

qubits are used.

Proof. Supposing that we are given $|\mathbf{f}(T)\rangle = \frac{1}{\|\mathbf{f}(T)\|} \sum_{\mathbf{i} \in \mathcal{I}_6} f_{\mathbf{i}}(T) |\mathbf{i}_x\rangle |\mathbf{i}_y\rangle |\mathbf{i}_z\rangle |\mathbf{i}_u\rangle |\mathbf{i}_v\rangle |\mathbf{i}_w\rangle$ and regarding this as the quantum state on the system of six m_{gr} -qubit registers, we consider the following unitary on this system:

$$W := Q_{m_{\text{gr}}} \otimes Q_{m_{\text{gr}}} \otimes Q_{m_{\text{gr}}} \otimes H_{m_{\text{gr}}} \otimes H_{m_{\text{gr}}} \otimes H_{m_{\text{gr}}}. \quad (46)$$

Here $Q_{m_{\text{gr}}}$ is the quantum Fourier transform on a m_{gr} -qubit system, which acts as

$$Q_{m_{\text{gr}}} |j\rangle = \frac{1}{\sqrt{n_{\text{gr}}}} \sum_{l=0}^{n_{\text{gr}}-1} \exp\left(\frac{2\pi i j l}{n_{\text{gr}}}\right) |l\rangle \quad (47)$$

Algorithm 1. Estimation of $|\tilde{\delta}_{\mathbf{k}}^v|^2$.

Input: Accuracy $\epsilon \in (0, 1/2)$, success probability $1 - \delta \in (0, 1)$, the value of C .

- 1: Construct the unitary quantum circuit $U_{\mathbf{f}(T), C\epsilon/4}$ on a $(2 \lg N_{\text{gr}} + 5)$ -qubit system following Theorem 1.
- 2: Construct $W' := I_{32N_{\text{gr}}} \otimes W$ the unitary on the same system, where $I_{32N_{\text{gr}}}$ is the identity operator on the first $\lg N_{\text{gr}} + 5$ qubits and W is the operator in Eq. (46) that acts on the other $\lg N_{\text{gr}}$ qubits.
- 3: Estimate the squared amplitude of $|\Omega'_{\mathbf{k}}\rangle := |0\rangle^{\otimes (\lg N_{\text{gr}} + 5)} |\Omega_{\mathbf{k}}\rangle$ in the state $W'U_{\mathbf{f}(T), C\epsilon/4} |0\rangle$ by QAE with accuracy $C\epsilon/4$ and success probability $1 - \delta$, and let the estimate be \tilde{p} .
- 4: Output \tilde{p}/C .

for any $j \in [n_{\text{gr}}]_0$, and $H_{m_{\text{gr}}}$ is the operation on the same system that is applying the Hadamard gate to each qubit. With $m_{\text{gr}} = \lg n_{\text{gr}}$, $Q_{m_{\text{gr}}}$ is implemented as a circuit consisting of $O(\log^2 n_{\text{gr}})$ Hadamard gates and conditional rotation gates with depth of the same order [87], and $H_{m_{\text{gr}}}$ is just a collection of $O(\log n_{\text{gr}})$ Hadamard gates. We thus neglect their costs in the following discussion. We also note that the average density $\bar{\rho}_v(T)$ is related to $\mathbf{f}(T)$ as

$$\bar{\rho}_v(T) \Delta_{\mathbf{x}}^3 = \frac{1}{N_{\text{gr}}} \sum_{\mathbf{i} \in \mathcal{I}_6} f_{\mathbf{i}}(T). \quad (48)$$

Using these along with Eqs. (39), (40), and (41), we obtain by a straightforward calculation that, for $\mathbf{i}_{\mathbf{k}} := (i_{k_x}, i_{k_y}, i_{k_z}) \in \mathcal{I}_3$,

$$|\langle \Omega_{\mathbf{i}_{\mathbf{k}}} | W | \mathbf{f}(T) \rangle|^2 = C |\tilde{\delta}_{\mathbf{i}_{\mathbf{k}}}^v|^2, \quad (49)$$

where $|\Omega_{\mathbf{i}_{\mathbf{k}}}\rangle := |i_{k_x}\rangle |i_{k_y}\rangle |i_{k_z}\rangle |0\rangle |0\rangle |0\rangle$. Since the left-hand side is the squared amplitude of the computational basis state $|\Omega_{\mathbf{i}_{\mathbf{k}}}\rangle$ in $W | \mathbf{f}(T) \rangle$, we can estimate this by QAE.⁵

In summary, we can get an ϵ -approximation of $|\tilde{\delta}_{\mathbf{i}_{\mathbf{k}}}^v|^2$ by the procedure shown in Algorithm 1.

The rest of the proof is on the accuracy and complexity of Algorithm 1. We postpone it to Appendix A 3. ■

To illustrate the outline of Algorithm 1, we present the overview diagram in Fig. 1. Using the oracles $\mathcal{O}_{\text{FCDM}, \mathbf{x}^*}^i$ etc., that give the CDM gravity, we construct block encodings of the time-evolution operators $\exp(-i\Delta_t H_i)$. Combining these block-encodings and the initial-state-encoding oracle $\mathcal{O}_{\mathbf{f}(0)}$, we construct the unitary $U_{\mathbf{f}(T), \epsilon}$ to generate the state $|\mathbf{f}(T)\rangle$ encoding the solution $\mathbf{f}(T)$ at time T with accuracy ϵ . Then this unitary followed by W generate the quantum state, in which the squared amplitude of a specific computational basis state $|\Omega'_{\mathbf{i}_{\mathbf{k}}}\rangle$ is equal to $C |\tilde{\delta}_{\mathbf{i}_{\mathbf{k}}}^v|^2$. Thus, we estimate this squared amplitude by QAE to get $|\tilde{\delta}_{\mathbf{i}_{\mathbf{k}}}^v|^2$, the squared amplitude of the specified Fourier mode of the neutrino density perturbation. Note that in this QAE, we use $U_{\mathbf{f}(T), C\epsilon/4}$, whose accuracy is $C\epsilon/4$, to guarantee the accuracy ϵ for the estimation of $|\tilde{\delta}_{\mathbf{i}_{\mathbf{k}}}^v|^2$.

⁵Note that for $|\Omega_{\mathbf{i}_{\mathbf{k}}}\rangle$ the operator V in Theorem 5, which flips the amplitude only for $|\Omega_{\mathbf{i}_{\mathbf{k}}}\rangle$ and does nothing otherwise, can be implemented by X gates and a multicontrolled Z gate.

Let us make some comments on C . In Theorem 2, we assume that we know the value of C in advance, even though it is defined with $\mathbf{f}(T)$ the solution at the terminal time. In fact, this assumption is plausible. This is because, as explained above, $\|\mathbf{f}(t)\|$ is constant over time, and $f_{\text{sum}}(t)$ is also almost constant. After all, we can calculate C using the initial value $\mathbf{f}(0)$ instead of $\mathbf{f}(T)$. The cost for this preliminary calculation will be discussed in Sec. III E.

Also, note that C is written as

$$C = \frac{(\frac{1}{N_{\text{gr}}} \sum_{\mathbf{i} \in \mathcal{I}_6} f_{\mathbf{i}}(0))^2}{\frac{1}{N_{\text{gr}}} \sum_{\mathbf{i} \in \mathcal{I}_6} (f_{\mathbf{i}}(0))^2}, \quad (50)$$

that is, the ratio of the squared average of $f_{\mathbf{i}}(T)$ over the grid points to the average of $f_{\mathbf{i}}(T)$ squared, which we expect is of order 1.

By using Eq. (35) and $C \sim 1$, we simplify Eq. (43) to

$$\tilde{O}\left(\frac{n_{\text{gr}} + n_t}{\epsilon}\right), \quad (51)$$

which is the query complexity bound announced in Introduction. Besides, Eq. (45) becomes of order (37), the space complexity announced in the Introduction.

D. Extensions

Equipped with Algorithm 1 as a basic one, we now consider its extensions so that it matches complications in practice.

1. Integrated power spectrum

Although Algorithm 1 outputs an estimate of $|\tilde{\delta}_{\mathbf{i}_{\mathbf{k}}}^v|^2$, this might not be feasible, since the magnitude of $|\tilde{\delta}_{\mathbf{i}_{\mathbf{k}}}^v|^2$ may be tiny. In fact, in the current discrete setting with n_{gr}^3 grid points in the position coordinates, $\delta_v(T, \mathbf{x})$ consists of the n_{gr}^3 Fourier components, and thus $|\tilde{\delta}_{\mathbf{i}_{\mathbf{k}}}^v|^2$ the amplitude of each Fourier components is typically suppressed by a factor $1/n_{\text{gr}}^3$. Therefore, to obtain a nonzero estimate of a value of such an order, we need to run QAE with $O(1/n_{\text{gr}}^3)$ accuracy and $O(n_{\text{gr}}^3)$ query complexity. After all, the query complexity scaling like Eq. (51) is not achieved.

However, this issue is not serious in practice. First, under the usual assumption of isotropy, $P_v(\mathbf{k})$ does not depend on the direction of \mathbf{k} but only its norm $k = |\mathbf{k}|$. Furthermore, for the purpose of comparing the numerical simulation with observations, it often suffices to get the integrated power spectrum,

$$\bar{P}_v(k_1, k_2) := \int_{k_1}^{k_2} dk 4\pi k^2 P_v(k), \quad (52)$$

with some interval $[k_1, k_2]$, which indicates the total magnitude of the neutrino density perturbations of scales between k_1 and k_2 . In the current discrete setting, the corresponding quantity is

$$\sum_{\mathbf{i}_{\mathbf{k}} \in \mathcal{I}_{3, [k_1, k_2]}} |\tilde{\delta}_{\mathbf{i}_{\mathbf{k}}}^v|^2, \quad (53)$$

where

$$\mathcal{I}_{3, [k_1, k_2]} := \{\mathbf{i}_{\mathbf{k}} \in \mathcal{I}_3 \mid k_1 \leq \|\mathbf{k}_{\mathbf{i}_{\mathbf{k}}}\| \leq k_2\}. \quad (54)$$

Thus, if we want a quantity like this, whose magnitude is larger than the Fourier-component wise amplitude, then the

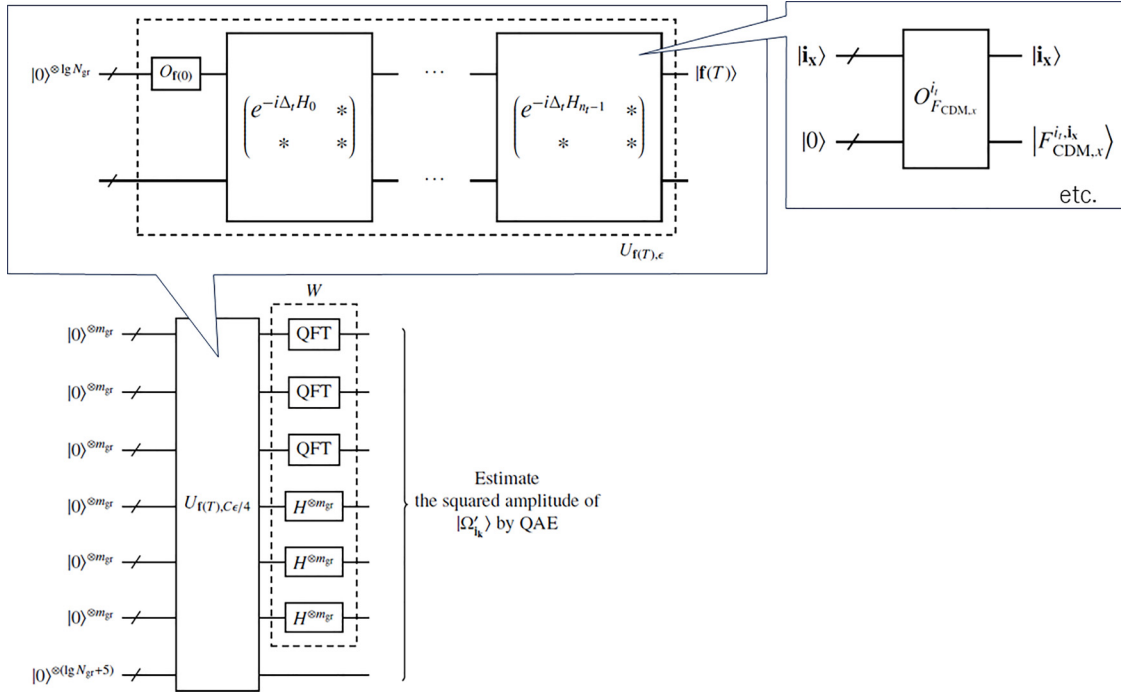


FIG. 1. Overview of the quantum circuit used in Algorithm 1.

query complexity of our quantum algorithm might not blow up.

Equation (53) can be estimated by Algorithm 1 with a slight modification. Instead of Eq. (A23), we estimate

$$\sum_{\mathbf{k} \in \mathcal{I}_{3,[k_1,k_2]}} \left| \langle \Omega'_{\mathbf{k}} | W' U_{f(T),C\epsilon/4} | 0 \rangle^{\otimes (2 \lg N_{gr}+5)} \right|^2, \quad (55)$$

that is, the probability that we obtain $|0\rangle^{\otimes (\lg N_{gr}+5)} |\mathbf{k}\rangle |0\rangle |0\rangle |0\rangle$ with $\mathbf{k} \in \mathcal{I}_{3,[k_1,k_2]}$ when we measure $W' U_{f(T),C\epsilon/4} |0\rangle^{\otimes (2 \lg N_{gr}+5)}$. This probability can be also estimated by QAE with query complexity in Eqs. (43) and (44).

2. Ensemble average

So far, we have considered how to obtain $|\tilde{\delta}_{\mathbf{k}}^v|^2$ for one realization, that is, the result of solving Eq. (16) with one initial value $\mathbf{f}(0)$. However, what we really want is its ensemble average in Eq. (38). In the classical Vlasov simulation, we estimate this ensemble average as

$$P_v(\mathbf{k}_{\mathbf{k}}) \simeq \frac{1}{n_{IV}} \sum_{i_{IV}=0}^{n_{IV}-1} |\tilde{\delta}_{\mathbf{k}}^{v,i_{IV}}|^2, \quad (56)$$

that is, the sample average of $|\tilde{\delta}_{\mathbf{k}}^{v,0}|^2, \dots, |\tilde{\delta}_{\mathbf{k}}^{v,n_{IV}-1}|^2$, the values of $|\tilde{\delta}_{\mathbf{k}}^v|^2$ in the n_{IV} different runs of the simulation, where we randomly take the different initial values according to the theoretical distribution of the primordial perturbation [24–26].

Also in quantum computing, we may do the same thing by not running the quantum algorithm many times separately but utilizing quantum superposition. That is, we extend Assumption 2 so that we can generate the superposition of the initial

values $\mathbf{f}^{(0)}(0), \dots, \mathbf{f}^{(n_{IV}-1)}(0)$:

$$O'_{f(0)} |0\rangle |0\rangle = \frac{1}{\sqrt{n_{IV}}} \sum_{i_{IV}=0}^{n_{IV}-1} |\mathbf{f}^{(i_{IV})}(0)\rangle |i_{IV}\rangle, \quad (57)$$

where $n_{IV} = 2^{m_{IV}}$ with $m_{IV} \in \mathbb{N}$ for convenience. Note that the N -body simulation for CDM should be also run many times with the initial values taken randomly, which leads to the different $\mathbf{F}_{\text{CDM}}(t, \mathbf{x})$ in the different runs. Thus, we also extend Assumption 1 so that we can use the oracle to access $\mathbf{F}_{\text{CDM}}(t, \mathbf{x})$ in the specified run:

$$\begin{aligned} \tilde{O}_{F_{\text{CDM},x}}^{i_{IV}} |\mathbf{x}\rangle |i_{IV}\rangle |0\rangle &= |\mathbf{x}\rangle |i_{IV}\rangle |F_{\text{CDM},x}^{i_{IV},\mathbf{x},i_{IV}}\rangle \\ \tilde{O}_{F_{\text{CDM},y}}^{i_{IV}} |\mathbf{x}\rangle |i_{IV}\rangle |0\rangle &= |\mathbf{x}\rangle |i_{IV}\rangle |F_{\text{CDM},y}^{i_{IV},\mathbf{x},i_{IV}}\rangle \\ \tilde{O}_{F_{\text{CDM},z}}^{i_{IV}} |\mathbf{x}\rangle |i_{IV}\rangle |0\rangle &= |\mathbf{x}\rangle |i_{IV}\rangle |F_{\text{CDM},z}^{i_{IV},\mathbf{x},i_{IV}}\rangle, \end{aligned} \quad (58)$$

where $F_{\text{CDM},x}^{i_{IV},\mathbf{x},i_{IV}}$ is $F_{\text{CDM},x}(t_{i_{IV}}, \mathbf{x}_{\mathbf{x}})$ in the i_{IV} th run of the N -body simulation for CDM, and so on. We postpone the implementation of these oracles to Sec. III E.

Equipped with these oracles, we can easily modify Algorithm 1 so that it outputs Eq. (56). That is, we add m_{IV} qubits and replace $O_{f(0)}$ with $O'_{f(0)}$. We also use $\tilde{O}_{F_{\text{CDM},x}}^{i_{IV}}$ instead of $O_{F_{\text{CDM},x}}^{i_{IV}}$, and so on. By this change, $U_{f(T),C\epsilon/4}$ in Algorithm 1 is modified to the operator $U'_{f(T),C\epsilon/4}$ on the $(2 \lg N_{gr} + m_{IV} + 5)$ -qubit system that acts as

$$\begin{aligned} U'_{f(T),C\epsilon/4} |0\rangle^{\otimes (2 \lg N_{gr} + m_{IV} + 5)} \\ = \frac{1}{\sqrt{n_{IV}}} \sum_{i_{IV}=0}^{n_{IV}-1} |0\rangle^{\otimes (\lg N_{gr} + 5)} |\mathbf{f}^{(i_{IV})}(T)\rangle |i_{IV}\rangle + |\xi_{\text{gar}}\rangle, \end{aligned} \quad (59)$$

where $\mathbf{f}^{(i_{IV})}(T)$ is the solution of Eq. (16) for the i_{IV} th initial value $\mathbf{f}^{(i_{IV})}(0)$, and $|\xi_{\text{gar}}\rangle$ is some unnormalized quantum

state satisfying $\|\xi_{\text{gar}}\| \leq \epsilon$. Then, we see that the following quantity is equal to the right-hand side of Eq. (56),

$$\sum_{i_{IV}=0}^{n_{IV}-1} \left| \langle \Omega_{\mathbf{i}_k, i_{IV}}'' | W'' U'_{\mathbf{f}(T), C\epsilon/4} | 0 \rangle^{\otimes (2 \lg N_{\text{gr}} + m_{IV} + 5)} \right|^2, \quad (60)$$

where $|\Omega_{\mathbf{i}_k, i_{IV}}''\rangle := |\Omega_{\mathbf{i}_k}'\rangle |i_{IV}\rangle$ and $W'' := W' \otimes I_{n_{IV}}$. Equation (60) is the probability that we obtain $|0\rangle^{\otimes (\lg N_{\text{gr}} + 5)} |\mathbf{i}_k\rangle |0\rangle |0\rangle |i_{IV}\rangle$ with any $i_{IV} \in [n_{IV}]_0$ when we measure $W'' U'_{\mathbf{f}(T), C\epsilon/4} |0\rangle^{\otimes (2 \lg N_{\text{gr}} + m_{IV} + 5)}$ and can be estimated by QAE. The numbers of queries to the replaced oracles are still of order (43) and (44).

Note that in the above discussion, we have implicitly assumed that the value of C is the same for different $\mathbf{f}^{(i_{IV})}(0)$. This in fact holds approximately for large N_{gr} , since C is written with the averages of the values of $f(0, \mathbf{x}, \mathbf{v})$ and its square on the N_{gr} grid points as Eq. (50), and the realizations of $\mathbf{f}^{(i_{IV})}(0)$ for various i_{IV} are generated based on the same distribution of the primordial perturbation.

E. Implementation of the oracles

Now, let us consider the implementations of the oracles used in our quantum algorithm.

1. Oracles to access \mathbf{F}_{CDM}

We first consider $\{O_{F_{\text{CDM},x}}^{i_{\mathbf{x}}}, O_{F_{\text{CDM},y}}^{i_{\mathbf{x}}}, O_{F_{\text{CDM},z}}^{i_{\mathbf{x}}}\}_{i_{\mathbf{x}}}$ the oracles to access $\mathbf{F}_{\text{CDM}}(t, \mathbf{x})$, the gravitational force by CDM as Eq. (28). Since we now assume that the N -body simulation for CDM gives the values of this on the n_{gr}^3 grid points in the position coordinates, it seems that we need to resort to QRAM that stores those n_{gr}^3 real numbers. In the recent Vlasov simulations with massive neutrino, the number of the grid points in the position coordinates is of order 10^9 [25]. Because of the issues on QRAM mentioned in Sec. II C 5, realizing a QRAM with such a large number of entries seems challenging. Nevertheless, compared to the classical Vlasov simulation, in which the $O(n_{\text{gr}}^6)$ memory space is used, the QRAM size of $O(n_{\text{gr}}^3)$ indicates a large improvement with respect to scaling on n_{gr} .

We should also note that calculating $\{F_{\text{CDM},x}^{i_{\mathbf{x}}}, F_{\text{CDM},y}^{i_{\mathbf{x}}}, F_{\text{CDM},z}^{i_{\mathbf{x}}}\}_{i_{\mathbf{x}}}$ by the N -body simulation and preparing QRAMs that store them takes $O(n_t n_{\text{gr}}^3)$ time, which has the worse scaling on n_{gr} than the query complexity of our quantum algorithm in Eq. (43). However, as mentioned in Sec. II B, in the current classical computing, the N -body simulation for only CDM is less heavy than the Vlasov and N -body simulations including massive neutrino. Therefore, it is reasonable to consider only the application of quantum computing to the Vlasov simulation for neutrino, leaving the N -body simulation for CDM classical. Also note that, once QRAMs storing $F_{\text{CDM},x}^{i_{\mathbf{x}}}$ and so on are prepared, we can reuse it for the neutrino Vlasov simulations in the various settings, e.g., various neutrino masses, which relatively diminishes the cost for the N -body simulation for CDM.

We also comment that, the approximation that $\mathbf{F}_{\text{CDM}}(t, \mathbf{x})$ is piecewise constant in time reduces the size of QRAMs to be prepared. Although we are given $\{F_{\text{CDM},x}^{i_{\mathbf{x}}}, F_{\text{CDM},y}^{i_{\mathbf{x}}}, F_{\text{CDM},z}^{i_{\mathbf{x}}}\}_{i_{\mathbf{x}}}$, whose number per vector element, say, $F_{\text{CDM},x}$, is $n_t n_{\text{gr}}^3$, our algorithm uses not one QRAM with $n_t n_{\text{gr}}^3$ entries per vector

element, but n_t QRAMs with n_{gr}^3 entries, which are $\{F_{\text{CDM},x}^{i_{\mathbf{x}}}\}_{i_{\mathbf{x}}}$ for one value of $i_t \in [n_t]_0$. This is because, as explained in Sec. III B, to get $|\mathbf{f}(T)\rangle$, we successively apply Hamiltonian simulations for the time intervals $[t_0, t_1), \dots, [t_{n_t-1}, t_{n_t})$, and in each of them we use only the constant value of $\mathbf{F}_{\text{CDM}}(t, \mathbf{x})$ in the interval. Dividing the QRAM into smaller ones reduces the difficulty of implementation. In particular, for the circuit-based QRAM, we do not need to use $n_t n_{\text{gr}}^3$ qubits, but it suffices to reuse n_{gr}^3 qubits, with different QRAMs implemented in the Hamiltonian simulation for the different time interval. We do not enjoy this reduction if $\mathbf{F}_{\text{CDM}}(t, \mathbf{x})$ continuously varies in time and we use the time-dependent Hamiltonian simulation [100], in which we use the oracle $O_{\text{ent}}^H : |t\rangle |i\rangle |j\rangle |0\rangle \mapsto |t\rangle |i\rangle |j\rangle |H_{ij}(t)\rangle$ and thus the oracles such as $O_{F_{\text{CDM},x}} : |t\rangle |\mathbf{i}_x\rangle |0\rangle \mapsto |t\rangle |\mathbf{i}_x\rangle |F_{\text{CDM},x}(t, \mathbf{x}_{\mathbf{i}_x})\rangle$ in our case.

As the last comment, we note that when we want to estimate Eq. (56), the average of $|\delta_{\mathbf{i}_k}^{v,0}|^2, \dots, |\delta_{\mathbf{i}_k}^{v,n_{IV}-1}|^2$, the number of the entries in each QRAM increases to $n_{\text{gr}}^3 n_{IV}$.

2. Oracle to generate the state encoding the initial value

Next, we consider the oracle $O_{\mathbf{f}(0)}$ in Eq. (30), which generates the quantum state $|\mathbf{f}(0)\rangle$ encoding the initial value $\mathbf{f}(0)$.

In the neutrino Vlasov simulation, the initial distribution function is set to a Fermi-Dirac distribution [26],

$$f(0, \mathbf{x}, \mathbf{v}) \propto [1 + \delta_v(0, \mathbf{x})] F_{\text{FD}}[\mathbf{v} - \mathbf{v}_b(\mathbf{x})]. \quad (61)$$

Here $\mathbf{v}_b(\mathbf{x})$ is the neutrino bulk velocity at position \mathbf{x} , and

$$F_{\text{FD}}(\mathbf{v} - \mathbf{v}_b) := \frac{1}{\exp\left(\frac{m_v |\mathbf{v} - \mathbf{v}_b|}{k_B T_v}\right) + 1}, \quad (62)$$

where k_B is the Boltzmann constant and T_v is the neutrino temperature at the initial time. The initial density fluctuation and bulk velocity are calculated based on the theory for the primordial cosmological perturbation [26].

Then, our aim is generating the following quantum state $|\mathbf{f}(0)\rangle$ encoding Eq. (61). It is written as

$$|\mathbf{f}(0)\rangle = \mathcal{C} \sum_{\mathbf{i}_x \in \mathcal{I}_3} [1 + \delta_v(0, \mathbf{x}_{\mathbf{i}_x})] |\mathbf{i}_x\rangle |F_{\text{FD}, \mathbf{v}_b(\mathbf{x}_{\mathbf{i}_x})}\rangle, \quad (63)$$

where

$$|F_{\text{FD}, \mathbf{v}_b}\rangle := \mathcal{D}_{\mathbf{v}_b} \sum_{\mathbf{i}_v \in \mathcal{I}_3} F_{\text{FD}}(\mathbf{v}_{\mathbf{i}_v} - \mathbf{v}_b) |\mathbf{v}_{\mathbf{i}_v}\rangle \quad (64)$$

and \mathcal{C} and $\mathcal{D}_{\mathbf{v}_b}$ are normalization constants.

We now assume that $\{\delta_v(0, \mathbf{x}_{\mathbf{i}_x})\}_{\mathbf{i}_x}$ are classically computed as a preparation of our quantum algorithm and stored in a QRAM. Given such a QRAM, we can generate the quantum state $\mathcal{C} \sum_{\mathbf{i}_x \in \mathcal{I}_3} (1 + \delta_v(0, \mathbf{x}_{\mathbf{i}_x})) |\mathbf{i}_x\rangle$ querying the QRAM $O(\log n_{\text{gr}}^3)$ times by the method in [101]. We also assume that $\{\mathbf{v}_b(\mathbf{x}_{\mathbf{i}_x})\}_{\mathbf{i}_x}$ are precomputed and stored in a QRAM. There are some methods to generate a quantum state encoding a function given as an explicit formula in the amplitudes [102–105]. By such a method, we can perform the operation $|\mathbf{v}_b\rangle |0\rangle \rightarrow |\mathbf{v}_b\rangle |F_{\text{FD}, \mathbf{v}_b}\rangle$, querying the quantum circuit to compute $F_{\text{FD}}(\mathbf{v} - \mathbf{v}_b)$ $O(\log n_{\text{gr}}^3)$ times. Combining these circuits and QRAMs, we can generate $|\mathbf{f}(0)\rangle$ with $O(\text{polylog}(n_{\text{gr}}))$ query complexity. Note that the sizes of the aforementioned

QRAMs are $O(n_{\text{gr}}^3)$, which is of the same order as QRAMs storing \mathbf{F}_{CDM} . Also note that preparing those QRAMs takes a $O(n_{\text{gr}}^3)$ time and this does not increase the order of the preparation time, which already includes the $O(n_{\text{gr}}^3)$ time to prepare the QRAMs for \mathbf{F}_{CDM} .

In the above discussion, we have assumed that there is only one neutrino flavor with mass m_ν . This is just for simplicity, and the extension to the more realistic multiflavor setting is straightforward as follows. The point is that we are now approximating the neutrino Vlasov equation as Eq. (4) and thus the different masses of the different flavors affect the solution only through the initial value, which depends on the mass. Since Eq. (4) is linear with respect to f , if we set the initial value to the linear combination of those for the various flavors, then the solution becomes the distribution function for the mixture of the flavors. In our quantum algorithm, the modification is only replacing $|F_{\text{FD},\mathbf{v}_b}\rangle$, which encodes the single-flavor distribution, with the quantum state encoding the initial distribution of the mixture. Since it is still given as an explicit formula, we can use the aforementioned methods for function loading to the quantum state.

Last, we make a comment on the cost to calculate C , which has been postponed. The numerator of Eq. (50) is equal to

$$\frac{1}{64V^6}(\bar{\rho}_\nu(0))^2, \quad (65)$$

and thus written with $\bar{\rho}_\nu(0)$, which is determined by the normalization of f . On the other hand, the denominator of Eq. (50) becomes

$$\frac{1}{N_{\text{gr}}} \sum_{\mathbf{i}_k \in \mathcal{I}_3} [1 + \delta_\nu(0, \mathbf{x}_{\mathbf{i}_k})]^2 \sum_{\mathbf{i}_v \in \mathcal{I}_3} \{F_{\text{FD}}[\mathbf{v}_{\mathbf{i}_v} - \mathbf{v}_b(\mathbf{x}_{\mathbf{i}_k})]\}^2 \quad (66)$$

for $f(0, \mathbf{x}, \mathbf{v})$ in the form of Eq. (61). The latter part of Eq. (66) is calculated as

$$\begin{aligned} & \sum_{\mathbf{i}_v \in \mathcal{I}_3} \{F_{\text{FD}}[\mathbf{v}_{\mathbf{i}_v} - \mathbf{v}_b(\mathbf{x}_{\mathbf{i}_k})]\}^2 \\ & \simeq \frac{1}{\Delta_V^3} \int_{-V}^V du \int_{-V}^V dv \int_{-V}^V dw \{F_{\text{FD}}[\mathbf{v} - \mathbf{v}_b(\mathbf{x}_{\mathbf{i}_k})]\}^2 \\ & \simeq \frac{1}{\Delta_V^3} \int_{-V}^V du \int_{-V}^V dv \int_{-V}^V dw (F_{\text{FD}}(\mathbf{v}))^2, \end{aligned} \quad (67)$$

where we replace the sum with the integral at the first approximation, and at the second approximation we neglect the contribution to the integral from the region outside $[-V, V] \times [-V, V] \times [-V, V]$. Equation (67) is independent of \mathbf{i}_k and evaluated numerically. The remaining part is $\sum_{\mathbf{i}_k \in \mathcal{I}_3} [1 + \delta_\nu(0, \mathbf{x}_{\mathbf{i}_k})]^2$. We can calculate this at the same time as generating $\{\delta_\nu(0, \mathbf{x}_{\mathbf{i}_k})\}_{\mathbf{i}_k}$, keeping the order of the preparation time $O(n_{\text{gr}}^3)$.

IV. DEMONSTRATION

Last, we present an illustrative numerical model. It is impossible to run our algorithm on a current real quantum computer or a quantum circuit simulator. Instead, we demonstrate a core part of our algorithm, namely, the Hamiltonian simulation-based time evolution of the neutrino distribution function $f(t, \mathbf{x}, \mathbf{v})$. By considering a toy problem, we find

the solution of Eq. (25), the discretized Vlasov equation, as Eq. (34) with the matrix exponentiation $\exp(-i\Delta_t H_i)$ calculated explicitly on a classical computer. This is what we aim at by the quantum algorithm of Hamiltonian simulation, and differs from common classical methods for time integration.

We consider the 1D setting with the position coordinate x and the velocity coordinate v . We give the CDM gravity by the following explicit function

$$F_{\text{CDM},x}(x) = A \sin(Kx) \quad (68)$$

with real constants A and K . Then, under the discretization described in Sec. III A, we find $\mathbf{f}(T) = (f(T, x_0, u_0), \dots, f(T, x_{n_{\text{gr}}-1}, u_{n_{\text{gr}}-1}))$, the vector of the values of $f(T, x, u)$ on the grid points. Since $F_{\text{CDM},x}$ does not depend on time and thus the Hamiltonian H does not either, we can get $\mathbf{f}(T)$ by applying a single operator $\exp(-iHT)$ to $\mathbf{f}(0)$:

$$|\mathbf{f}(T)\rangle = \exp(-iHT)|\mathbf{f}(0)\rangle. \quad (69)$$

The initial value is set as follows: $f(0, x, u)$ is constant in x , and the Maxwell distribution is adopted in the u direction. That is,

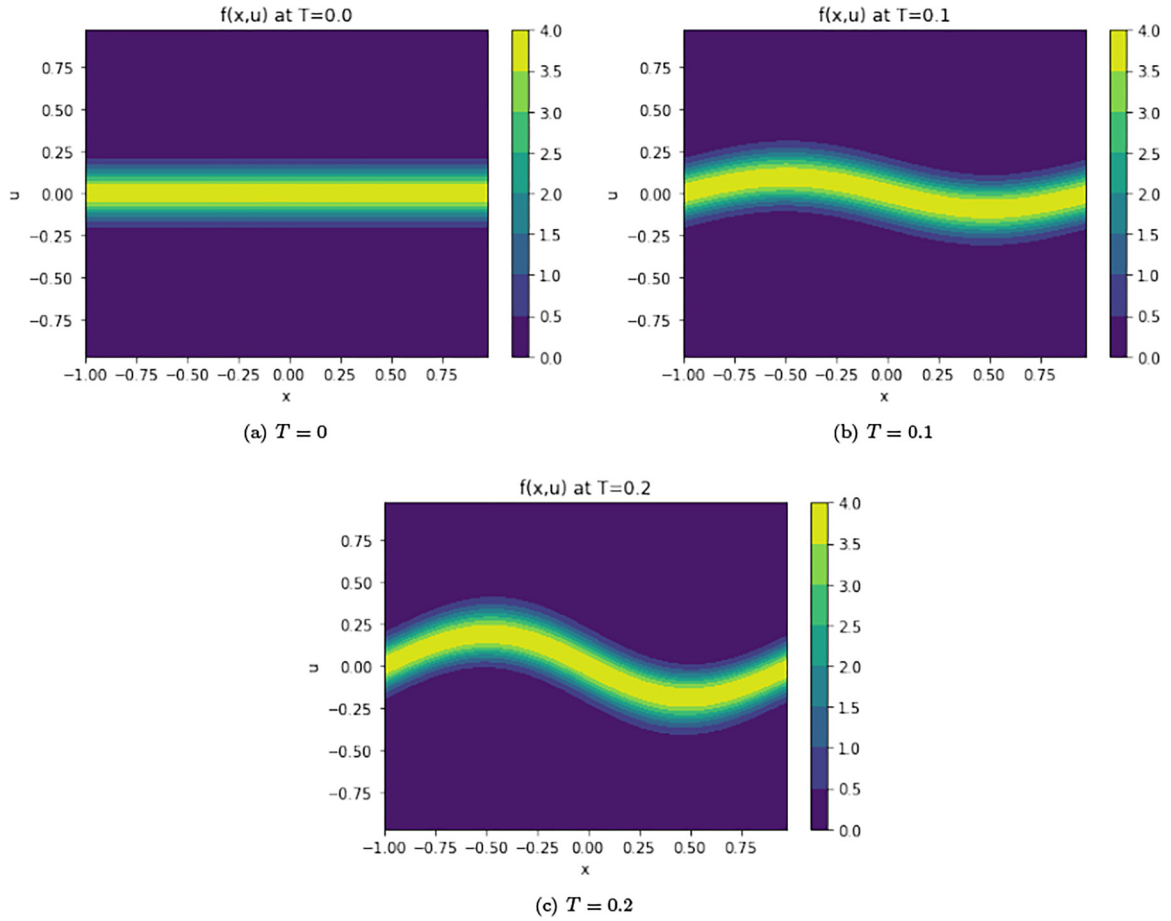
$$f(0, x, u) = \frac{1}{\sqrt{2\pi}\sigma_v^2} \exp\left(-\frac{u^2}{2\sigma_v^2}\right) \quad (70)$$

with $\sigma_v > 0$.

We show the result for $A = -1$, $K = \pi$, and $\sigma_v = 0.1$ under the discretization with $L = 2$, $V = 1$, and $n_{\text{gr}} = 64$. Figure 2 shows the heatmaps of $f(T, x, u)$ drawn with $\mathbf{f}(T)$ obtained by Eq. (69) with $T = 0, 0.1$, and 0.2 . From this $\mathbf{f}(T)$, we calculate $\delta_{i_k}^\nu$, the neutrino density perturbation at each grid point, as Eq. (40), and plot it as a function of $x = i_k L / n_{\text{gr}}$ in Fig. 3. Because of the functional form of $F_{\text{CDM},x}$, we expect that the density is enhanced at $x = 1$ and suppressed at $x = 0, 2$, and Fig. 3 matches this expectation. Using this $\delta_{i_k}^\nu$, we calculate $|\tilde{\delta}_{i_k}^\nu|^2$, the squared amplitude of the Fourier mode of the neutrino density perturbation, via Eq. (39), and plot it as a function of $k = 2\pi i_k / L$ in Fig. 4. Since the neutrino distribution evolves under the force as a single sinusoidal function with wave number $k = K$, we expect that only the Fourier mode with that wave number is enhanced. In fact, this is observed in Fig. 4. In summary, these figures imply that our calculation is working in this demonstration.

V. SUMMARY

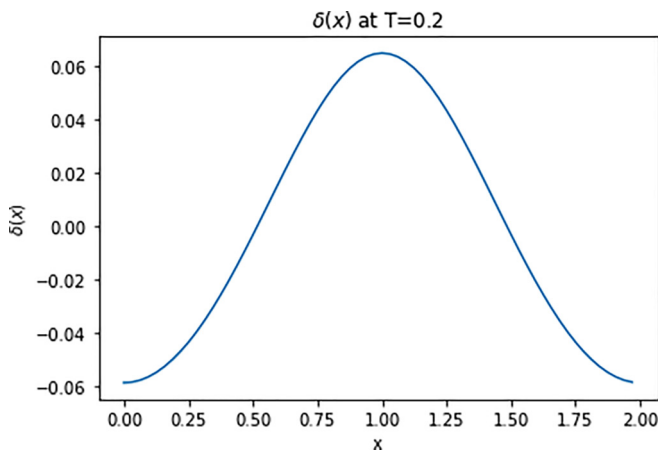
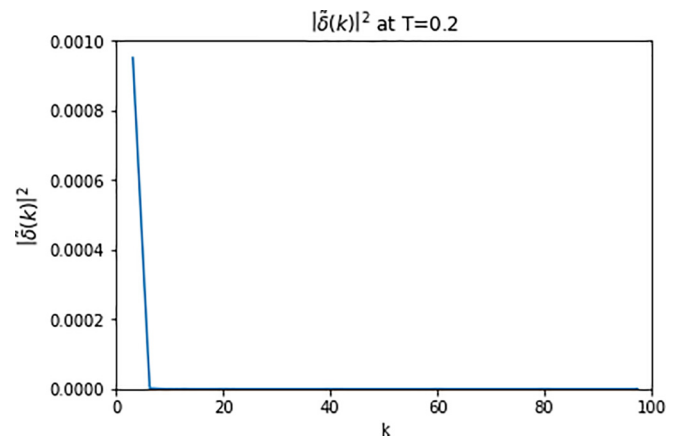
In this paper, we considered the quantum algorithm for simulations of LSS formation with massive neutrinos. The large-scale neutrino distribution is an important issue for both cosmology and particle physics. In order to follow the growth of density perturbations of neutrinos with a large velocity dispersion, it is desirable to solve directly the Vlasov equation in an efficient way rather than performing conventional N -body simulations. However, one needs to solve the PDE in a $(6 + 1)$ -dimensional space, and thus it is a challenging task: Taking n_{gr} grid points in each of six space coordinates and n_t time grid points leads to $O(n_t n_{\text{gr}}^6)$ and $O(n_{\text{gr}}^6)$ space complexity. We thus proposed a quantum algorithm for this task. First, by neglecting the gravity generated by the neutrinos that

FIG. 2. $f(T, x, u)$ at $T = 0, 0.1$ and 0.2 .

contribute only a tiny fraction in mass, and hence by taking into account only the gravity by CDM, we approximated the Vlasov equation in a linear form. We showed that the discretized Vlasov equation can be regarded as a Schrödinger equation. Then, by calculating the distribution of CDM using a fast N -body simulation in advance, we applied the Hamiltonian simulation to solve the equation. We proposed not only how to obtain the solution-encoding quantum state but also a way to extract the neutrino power spectrum, which is an

important quantity for comparisons with cosmological observations and thus is of practical interest, from the quantum state by QAE. Our method outputs a ϵ -approximation of the power spectrum with $\tilde{O}[(n_{\text{gr}} + n_t)/\epsilon]$ query complexity, using $O(\log^{5/2}(n_{\text{gr}}/\epsilon))$ qubits.

Although our algorithm has some shortcomings such as the assumption on the availability of large-sized QRAMs, it is the first proposal of a quantum algorithm for the LSS simulation that outputs the quantity of practical interest with guaranteed

FIG. 3. δ_{ix}^ν at $T = 0.2$.FIG. 4. $|\tilde{\delta}_{ik}^\nu|^2$ at $T = 0.2$.

accuracy. We thus believe that our algorithm is an important step in the application of quantum computing to the LSS simulation or, more broadly, challenging numerical tasks in cosmology and particle physics.

In future works, we will explore the possibility of extending our algorithm. Obviously, taking into account the neutrino self-gravity via, e.g., Carleman linearization, is one possible way. Another possibility is incorporating higher-order schemes for partial derivatives, for which we are now adopting the central difference as Eq. (15). Adopting higher-order schemes has two effects. On the one hand, it makes the coefficient matrix $A(t)$ less sparse: If we adopt a n th-order scheme, then the sparsity s scales as $O(n)$. In the sparse-access setting, the query complexity of Hamiltonian simulation scales as $O(s)$ on s and thus so does our algorithm. In total, the query complexity of our algorithm scales as $O(n)$ on n . On the other hand, by adopting a higher-order scheme, we may configure spatial grid points with larger interval, which means smaller n_{gr} , keeping the accuracy of the solution. Since the query complexity of our algorithm scales as $O(n_{\text{gr}})$, in total, if an n th-order scheme leads to n_{gr} scaling better than $n_{\text{gr}} = O(1/n)$, then higher-order schemes would be beneficial. As a previous study on the effect of higher-order schemes in practice, we refer to Ref. [106], which performed a set of simulations with third-, fifth-, and seventh-order schemes and in fact observed the improvement of the accuracy. However, in general, the extent to which higher-order schemes improve accuracy is highly problem dependent, and thus we leave incorporating higher-order schemes into our algorithm for future works.

ACKNOWLEDGMENTS

K.M. is supported by MEXT Quantum Leap Flagship Program (MEXT Q-LEAP) Grant No. JPMXS0120319794, JSPS KAKENHI Grant No. JP22K11924, and JST COI-NEXT Program Grant No. JPMJPF2014. S.Y. is supported by the Forefront Physics and Mathematics Program to Drive Transformation (FoPM). F.U. is supported by JSPS KAKENHI Grant No. JP23KJ0642 and the Forefront Physics and Mathematics Program to Drive Transformation (FoPM). K.F. is supported by JSPS KAKENHI Grant No. JP20K14512.

APPENDIX A: PROOFS

1. Building-block theorems

Here we present some theorems from previous papers, which are used to prove our main results, theorems on the complexity of our quantum algorithm.

a. Block encoding

Mathematically, block encoding is defined as follows:

Definition 1 (Reference [91], Definition 24). Let $s, a \in \mathbb{N}$, $\alpha, \epsilon \in \mathbb{R}_+$, and $A \in \mathbb{C}^{2^s \times 2^s}$. We say that a unitary U on a $(s + a)$ -qubit system is an (α, a, ϵ) -block encoding of A if

$$\|A - \alpha(|0\rangle^{\otimes a} \otimes I_{2^s})U(|0\rangle^{\otimes a} \otimes I_{2^s})\| \leq \epsilon. \quad (\text{A1})$$

We can efficiently construct a block encoding of a sparse matrix if we have access to its entries. The cost for this is summarized as the following theorem.

Theorem 3 (Reference [91], Lemma 48 in the full version, modified). Let $A = (A_{ij}) \in \mathbb{C}^{2^w \times 2^w}$ be a s -sparse matrix. Suppose that we have accesses to the oracles O_{row}^A and O_{col}^A that act on a two register system as

$$O_{\text{row}}^A |i\rangle |k\rangle = |i\rangle |r_{ik}\rangle, \quad O_{\text{col}}^A |k\rangle |i\rangle = |c_{ki}\rangle |i\rangle \quad (\text{A2})$$

for any $i \in [2^w]_0$ and $k \in [s]_0$, where r_{ik} (respectively, c_{ki}) is the index of the k th nonzero entry in the i th row (respectively, i th column) in A , or $i + 2^w$ if there are less than k nonzero entries. Besides, suppose that we have accesses to the oracle O_{ent}^A that acts on a three register system as

$$O_{\text{ent}}^A |i\rangle |j\rangle |0\rangle = |i\rangle |j\rangle |A_{ij}\rangle. \quad (\text{A3})$$

Then, for any $\epsilon \in \mathbb{R}_+$, there exists a $(s\|A\|_{\max}, w + 3, \epsilon)$ -block encoding of A , in which O_{row}^A and O_{col}^A are each queried once, O_{ent}^A is queried twice, additional $O[w + \log^{5/2}(\frac{s^2\|A\|_{\max}}{\epsilon})]$ one and two qubit gates are used, and $O[\log^{5/2}(\frac{s^2\|A\|_{\max}}{\epsilon})]$ ancilla qubits are used.

b. Hamiltonian simulation

The theorem on the complexity of Hamiltonian simulation, which was postponed in Sec. IIC, is as follows.

Theorem 4 (Reference [91], Corollary 62 in the full version). Let $\epsilon \in (0, \frac{1}{2})$, $t \in \mathbb{R}$, $\alpha \in \mathbb{R}_+$ and $s, a \in \mathbb{N}$. Let H be a $2^s \times 2^s$ Hermitian matrix and U be a $(\alpha, a, \epsilon/|2t|)$ -block encoding of H . Then we can implement $(1, a + 2, \epsilon)$ -block encoding of $\exp(-itH)$ using U or its inverse $O[6\alpha|t| + 9\log(12/\epsilon)]$ times, controlled U or its inverse 3 times, additional $O\{\alpha[\alpha|t| + \log(2/\epsilon)]\}$ two-qubit gates, and $O(1)$ ancilla qubits.

c. Quantum amplitude estimation

The complexity of QAE is evaluated as the following theorem.

Theorem 5 (Ref. [64], Theorem 12, modified). Suppose that we are given an access to a quantum circuit A that acts on a quantum register as $A|0\rangle = |\Phi\rangle := \sqrt{a}|\phi\rangle + \sqrt{1-a}|\phi_{\perp}\rangle$, where $|\phi\rangle$ and $|\phi_{\perp}\rangle$ are mutually orthogonal states and $a \in (0, 1)$. Also, suppose that we have access to a quantum circuit V on the same system that acts as $V|\phi\rangle = -|\phi\rangle$ and $V|\phi_{\perp}\rangle = |\phi_{\perp}\rangle$. Then, for any $\epsilon \in \mathbb{R}_+$ and $\delta \in (0, 1)$, there exists a quantum algorithm that with probability at least $1 - \delta$ outputs a ϵ -approximation of a calling A and V

$$O\left[\frac{1}{\epsilon} \log\left(\frac{1}{\delta}\right)\right] \quad (\text{A4})$$

times.

In this paper, we use QAE for the case that the target state $|\phi\rangle$ is a superposition of some set \mathcal{S} of computational basis states and $|\phi_{\perp}\rangle$ is a superposition of the other ones. In this case, a is the probability that we get a bit string corresponding to any state in \mathcal{S} when we measure $|\Phi\rangle$.

Note that, although the success probability in the original theorem in Ref. [64] is a constant, it is $1 - \delta$ in Theorem 5 with a factor $\log(1/\delta)$ added in the query complexity bound. This is due to taking the median of the results in multiple runs of the algorithm [107], which is based on the powering lemma in Ref. [108].

We also comment on the qubit number. The original version of QAE proposed in Ref. [64] is based on quantum phase estimation (QPE) [109], and uses $O(\log(1/\epsilon))$ qubits to output the estimate of a in addition to qubits used in A and V . Furthermore, it requires the controlled versions of A and V , for which we may need to use additional qubits and gates compared to the uncontrolled ones. Fortunately, after Ref. [64], many variants without QPE, which require neither additional qubits nor controlled oracles, have been proposed [110–113]. Thus, we hereafter consider that in QAE we use only qubits needed to operate A and V .

2. Proof of Theorem 1

As a preparation, let us see that we can implement the oracles used to construct the block encoding of H .

Lemma 1. Suppose that Assumption 1 holds. Then, for $H(t)$ in Eq. (23) with any $t \in [0, T)$, we have $O_{\text{row}}^{H(t)}$, $O_{\text{col}}^{H(t)}$, and $O_{\text{ent}}^{H(t)}$, in which $\{O_{\text{FCDM},x}^{i_t}, O_{\text{FCDM},y}^{i_t}, O_{\text{FCDM},z}^{i_t}\}_{i_t}$ and arithmetic circuits are queried $O(1)$ times.

Proof. If we have $O_{\text{row}}^{A(t)}$, $O_{\text{col}}^{A(t)}$, and $O_{\text{ent}}^{A(t)}$ for $A(t)$ in Eq. (18), then we immediately have those for $H(t)$ in Eq. (23), since the latter is just the former multiplied by i . Thus, we hereafter consider $O_{\text{row}}^{A(t)}$, $O_{\text{col}}^{A(t)}$, and $O_{\text{ent}}^{A(t)}$.

Let us start from $O_{\text{row}}^{A(t)}$. For $A(t)$, whose sparsity is 12, r_{ik} is given as⁶

$$r_{ik} = \sigma(\sigma^{-1}(i) + \mathbf{d}_k) \quad (\text{A5})$$

with

$$\begin{aligned} \mathbf{d}_1 &= (-1, 0, 0, 0, 0, 0) \\ \mathbf{d}_2 &= (1, 0, 0, 0, 0, 0) \\ &\vdots \\ \mathbf{d}_{11} &= (0, 0, 0, 0, 0, -1) \\ \mathbf{d}_{12} &= (0, 0, 0, 0, 0, 1). \end{aligned} \quad (\text{A6})$$

Here $\sigma : \mathcal{I}_6 \rightarrow [N_{\text{gr}}]_0$ is the map in Eq. (14) and σ^{-1} is its inverse. Note that, in the binary representation on qubits, these are in fact doing nothing, or just changing whether we consider that a computational basis state corresponds to $\mathbf{i} \in \mathcal{I}_6$ or $\sigma(\mathbf{i}) \in [N_{\text{gr}}]_0$. Thus, $O_{\text{row}}^{A(t)}$ is implemented with controlled adders, thus as a combination of arithmetic circuits. $O_{\text{col}}^{A(t)}$ is implemented similarly.

On the implementation of $O_{\text{ent}}^{A(t)}$, we consider each component of $A(t)$ separately. Let us first consider A_x . For $i, j \in [N_{\text{gr}}]_0$, we can perform the following operation:

$$\begin{aligned} &|i\rangle|j\rangle|0\rangle|0\rangle|0\rangle \\ &\rightarrow |i\rangle|j\rangle|\mathbf{1}_{\sigma^{-1}(i)-\sigma^{-1}(j)=(1,0,0,0,0,0)}\rangle \\ &\quad \times \otimes |\mathbf{1}_{\sigma^{-1}(i)-\sigma^{-1}(j)=(-1,0,0,0,0,0)}\rangle|0\rangle \end{aligned}$$

⁶Strictly speaking, this expression for r_{ik} holds for only i such that none of the entries of $\sigma^{-1}(i)$ is 0 or $n_{\text{gr}} - 1$, and otherwise the expression is slightly modified. However, such a handling is straightforward and thus we do not show the complete expression here for conciseness.

$$\begin{aligned} &= |i\rangle|j\rangle(\mathbf{1}_{\sigma^{-1}(i)-\sigma^{-1}(j) \neq (\pm 1, 0, 0, 0, 0, 0)}|0\rangle|0\rangle \\ &\quad + \mathbf{1}_{\sigma^{-1}(i)-\sigma^{-1}(j)=(1,0,0,0,0,0)}|1\rangle|0\rangle \\ &\quad + \mathbf{1}_{\sigma^{-1}(i)-\sigma^{-1}(j)=(-1,0,0,0,0,0)}|0\rangle|1\rangle)|0\rangle \\ &\rightarrow |i\rangle|j\rangle \otimes (\mathbf{1}_{\sigma^{-1}(i)-\sigma^{-1}(j) \neq (\pm 1, 0, 0, 0, 0, 0)}|0\rangle|0\rangle|0\rangle \\ &\quad + \mathbf{1}_{\sigma^{-1}(i)-\sigma^{-1}(j)=(1,0,0,0,0,0)}|1\rangle|0\rangle \left| \frac{V - (i_u + 1)\Delta_v}{2\Delta_x} \right\rangle \\ &\quad + \mathbf{1}_{\sigma^{-1}(i)-\sigma^{-1}(j)=(-1,0,0,0,0,0)}|0\rangle|1\rangle \left| \frac{-V + (i_u + 1)\Delta_v}{2\Delta_x} \right\rangle) \\ &= |i\rangle|j\rangle \\ &\quad \times \otimes |\mathbf{1}_{\mathbf{i}(i)-\mathbf{i}(j)=(1,0,0,0,0,0)}\rangle|\mathbf{1}_{\mathbf{i}(i)-\mathbf{i}(j)=(-1,0,0,0,0,0)}\rangle|(A_x)_{ij}\rangle \\ &\rightarrow |i\rangle|j\rangle|0\rangle|0\rangle|(A_x)_{ij}\rangle. \end{aligned} \quad (\text{A7})$$

Here the transformation at the first arrow is done by some arithmetic circuits. At the second arrow, regarding $|i\rangle$ as $|\sigma^{-1}(i)\rangle = |i_x\rangle|i_y\rangle|i_z\rangle|i_u\rangle|i_v\rangle|i_w\rangle$, we use the controlled versions of

$$\begin{aligned} U_{A_x,1} : |i_u\rangle|0\rangle &\mapsto |i_u\rangle \left| \frac{V - (i_u + 1)\Delta_v}{2\Delta_x} \right\rangle \\ U_{A_x,2} : |i_u\rangle|0\rangle &\mapsto |i_u\rangle \left| \frac{-V + (i_u + 1)\Delta_v}{2\Delta_x} \right\rangle, \end{aligned} \quad (\text{A8})$$

which are implemented with some arithmetic circuits. In the last arrow, the operation at the first arrow is uncomputed. The operation in Eq. (A7) is in fact $O_{\text{ent}}^{A_x}$, with some registers regarded as ancillary. $O_{\text{ent}}^{A_y}$ and $O_{\text{ent}}^{A_z}$ are implemented similarly.

To implement $O_{\text{ent}}^{A_u(t)}$ for any $t \in [0, T)$, it is sufficient to implement $O_{\text{ent}}^{A_u^{i_t}}$ for any $i_t \in [n_t]_0$, where $A_u^{i_t} := A_u(t_{i_t})$. This is done as follows:

$$\begin{aligned} &|i\rangle|j\rangle|0\rangle|0\rangle|0\rangle \\ &\rightarrow |i\rangle|j\rangle(\mathbf{1}_{\sigma^{-1}(i)-\sigma^{-1}(j) \neq (\pm 1, 0, 0, 0, 0, 0)}|0\rangle|0\rangle \\ &\quad + \mathbf{1}_{\sigma^{-1}(i)-\sigma^{-1}(j)=(1,0,0,0,0,0)}|1\rangle|0\rangle \\ &\quad + \mathbf{1}_{\sigma^{-1}(i)-\sigma^{-1}(j)=(-1,0,0,0,0,0)}|0\rangle|1\rangle)|0\rangle \\ &\rightarrow |i\rangle|j\rangle \otimes (\mathbf{1}_{\sigma^{-1}(i)-\sigma^{-1}(j) \neq (0,0,0,\pm 1,0,0)}|0\rangle|0\rangle|0\rangle \\ &\quad + \mathbf{1}_{\sigma^{-1}(i)-\sigma^{-1}(j)=(0,0,0,1,0,0)}|1\rangle|0\rangle \left| -\frac{F_{\text{CDM},x}^{i_t,\mathbf{i}_x}}{2\Delta_v} \right\rangle \\ &\quad + \mathbf{1}_{\sigma^{-1}(i)-\sigma^{-1}(j)=(0,0,0,-1,0,0)}|0\rangle|1\rangle \left| \frac{F_{\text{CDM},x}^{i_t,\mathbf{i}_x}}{2\Delta_v} \right\rangle) \\ &\rightarrow |i\rangle|j\rangle|0\rangle|0\rangle|(A_u^{i_t})_{ij}\rangle. \end{aligned} \quad (\text{A9})$$

Here the first arrow is similar to Eq. (A7). At the second arrow, we use the controlled versions of

$$\begin{aligned} U_{A_u^{i_t},1} : |\mathbf{i}_x\rangle|0\rangle &\mapsto |\mathbf{i}_x\rangle \left| -\frac{F_{\text{CDM},x}^{i_t,\mathbf{i}_x}}{2\Delta_v} \right\rangle \\ U_{A_u^{i_t},2} : |\mathbf{i}_x\rangle|0\rangle &\mapsto |\mathbf{i}_x\rangle \left| \frac{F_{\text{CDM},x}^{i_t,\mathbf{i}_x}}{2\Delta_v} \right\rangle, \end{aligned} \quad (\text{A10})$$

which are implemented with $O_{\text{FCDM},x}^i$ and some arithmetic circuits. The last arrow is uncomputation. $O_{\text{ent}}^{A_v(t)}$ and $O_{\text{ent}}^{A_w(t)}$ are implemented similarly.

Then, we can perform the following operation:

$$\begin{aligned}
 & |i\rangle|j\rangle|0\rangle|0\rangle|0\rangle|0\rangle|0\rangle|0\rangle \\
 & \rightarrow |i\rangle|j\rangle|(A_x)_{ij}\rangle|(A_y)_{ij}\rangle|(A_z)_{ij}\rangle \\
 & \quad \otimes |(A_u(t))_{ij}\rangle|(A_v(t))_{ij}\rangle|(A_w(t))_{ij}\rangle|0\rangle \\
 & \rightarrow |i\rangle|j\rangle|(A_x)_{ij}\rangle|(A_y)_{ij}\rangle|(A_z)_{ij}\rangle \\
 & \quad \otimes |(A_u(t))_{ij}\rangle|(A_v(t))_{ij}\rangle|(A_w(t))_{ij}\rangle|(A(t))_{ij}\rangle \\
 & \rightarrow |i\rangle|j\rangle|0\rangle|0\rangle|0\rangle|0\rangle|0\rangle|(A(t))_{ij}\rangle, \quad (\text{A11})
 \end{aligned}$$

where the first arrow is by $O_{\text{ent}}^{A_x}$, $O_{\text{ent}}^{A_y}$, $O_{\text{ent}}^{A_z}$, $O_{\text{ent}}^{A_u(t)}$, $O_{\text{ent}}^{A_v(t)}$, and $O_{\text{ent}}^{A_w(t)}$, we use the adder to compute $(A(t))_{ij} = (A_x)_{ij} + (A_y)_{ij} + (A_z)_{ij} + (A_u(t))_{ij} + (A_v(t))_{ij} + (A_w(t))_{ij}$ at the second arrow, and the last arrow is uncomputation. This is nothing but $O_{\text{ent}}^{A(t)}$.

To complete the proof, we note that the number of the queries to $\{O_{\text{FCDM},x}^i, O_{\text{FCDM},y}^i, O_{\text{FCDM},z}^i\}_{i_t}$ and arithmetic circuits in the operations (A7) and (A9) are $O(1)$ and so is that in Eq. (A11), which proves the statement on the query number. ■

Then, let us prove Theorem 1.

Proof of Theorem 1. First, let us construct the block encoding of $\exp(-i\Delta_t H_{i_t})$ for any $i_t \in [n_t]_0$. Noting that H_{i_t} is 12-sparse, because of Theorem 3 and Lemma 1, we have $(12\|H_{i_t}\|_{\max}, \lg N_{\text{gr}} + 3, \epsilon/2T)$ -block encoding $U_{H_{i_t}}$ of H_{i_t} , in which $O_{\text{FCDM},x}^i, O_{\text{FCDM},y}^i, O_{\text{FCDM},z}^i$ and arithmetic circuits are queried $O(1)$ times. Then, because of Theorem 4, we have a $(1, \lg N_{\text{gr}} + 5, \epsilon/n_t)$ -block encoding V_{i_t} of $\exp(-i\Delta_t H_{i_t})$, in which (controlled) $U_{H_{i_t}}$ and its inverse are queried

$$O\left[\|H_{i_t}\|_{\max} \Delta_t + \log\left(\frac{n_t}{\epsilon}\right)\right] \quad (\text{A12})$$

times, and thus so are (controlled) $O_{\text{FCDM},x}^i, O_{\text{FCDM},y}^i, O_{\text{FCDM},z}^i$, arithmetic circuits and their inverses.

Then, we generate $|0\rangle^{\otimes(\lg N_{\text{gr}}+5)}|\mathbf{f}(0)\rangle$ by using $O_{\mathbf{f}(0)}$ and apply V_0 to it. This yields

$$|\phi_1\rangle := |0\rangle^{\otimes(\lg N_{\text{gr}}+5)} \exp(-i\Delta_t H_0) |\mathbf{f}(0)\rangle + |\psi_{\text{gar},1}\rangle, \quad (\text{A13})$$

where $|\psi_{\text{gar},1}\rangle$ is an unnormalized quantum state satisfying $\| |\psi_{\text{gar},1}\rangle \| \leq \epsilon/n_t$. We further apply V_1 to $|\phi_1\rangle$ and get

$$|\phi_2\rangle := |0\rangle^{\otimes(\lg N_{\text{gr}}+5)} e^{-i\Delta_t H_1} e^{-i\Delta_t H_0} |\mathbf{f}(0)\rangle + |\psi_{\text{gar},2}\rangle. \quad (\text{A14})$$

Here

$$|\psi_{\text{gar},2}\rangle := e^{-i\Delta_t H_1} |\psi_{\text{gar},1}\rangle + |\psi'_{\text{gar},2}\rangle, \quad (\text{A15})$$

where $|\psi'_{\text{gar},2}\rangle$ is an unnormalized state with norm at most ϵ/n_t . The norm of $|\psi_{\text{gar},2}\rangle$ is bounded as

$$\| |\psi_{\text{gar},2}\rangle \| \leq \| e^{-i\Delta_t H_1} |\psi_{\text{gar},1}\rangle \| + \| |\psi'_{\text{gar},2}\rangle \| \leq \frac{2\epsilon}{n_t}. \quad (\text{A16})$$

Continuing this, we obtain

$$\begin{aligned}
 & V_{n_t-1} \cdots V_0 |0\rangle^{\otimes(\lg N_{\text{gr}}+5)} |\mathbf{f}(0)\rangle \\
 & = |0\rangle^{\otimes(\lg N_{\text{gr}}+5)} e^{-i\Delta_t H_{n_t-1}} \cdots e^{-i\Delta_t H_0} |\mathbf{f}(0)\rangle + |\psi_{\text{gar},n_t}\rangle, \quad (\text{A17})
 \end{aligned}$$

where the first term is nothing but

$$|0\rangle^{\otimes(\lg N_{\text{gr}}+5)} |\mathbf{f}(T)\rangle \quad (\text{A18})$$

because of Eq. (29), the piecewise time constancy of $H(t)$, and $|\psi_{\text{gar},n_t}\rangle$ is the unnormalized state with norm at most $\frac{\epsilon}{n_t} \times n_t = \epsilon$. This means that the above operation is $U_{\mathbf{f}(T),\epsilon}$ with the stated property.

In this operation, the number of queries to (controlled) $\{O_{\text{FCDM},x}^i, O_{\text{FCDM},y}^i, O_{\text{FCDM},z}^i\}_{i_t}$, arithmetic circuits and their inverses are that in each V_{i_t} , which is bounded as Eq. (A12), multiplied by n_t , that is,

$$O\left\{n_t \left[\|H_{i_t}\|_{\max} \Delta_t + \log\left(\frac{n_t}{\epsilon}\right)\right]\right\}. \quad (\text{A19})$$

Replacing $\|H_{i_t}\|_{\max}$ with its bound

$$\begin{aligned}
 \|H_{i_t}\|_{\max} & \leq \max \left\{ \max_{i_u \in [n_{\text{gr}}]_0} \left| \frac{u_{i_u}}{2\Delta_{\mathbf{x}}} \right|, \max_{i_v \in [n_{\text{gr}}]_0} \left| \frac{v_{i_v}}{2\Delta_{\mathbf{x}}} \right|, \max_{i_w \in [n_{\text{gr}}]_0} \left| \frac{w_{i_w}}{2\Delta_{\mathbf{x}}} \right|, \right. \\
 & \quad \left. \max_{i_x \in \mathcal{I}_3} \left| \frac{F_{\text{CDM},x}^{i_t, \mathbf{i}_x}}{2\Delta_{\mathbf{v}}} \right|, \max_{i_x \in \mathcal{I}_3} \left| \frac{F_{\text{CDM},y}^{i_t, \mathbf{i}_x}}{2\Delta_{\mathbf{v}}} \right|, \max_{i_x \in \mathcal{I}_3} \left| \frac{F_{\text{CDM},z}^{i_t, \mathbf{i}_x}}{2\Delta_{\mathbf{v}}} \right| \right\} \\
 & \leq \max \left\{ \frac{V}{L}, \frac{F_{\max}}{2V} \right\} \times \frac{n_{\text{gr}}}{2} \quad (\text{A20})
 \end{aligned}$$

yields Eq. (32).

Last, we prove the statement on the qubit number. Combining Theorems 3 and 4, we see that the number of the ancilla qubits used in V_{i_t} is

$$O\left(\log^{5/2}\left(\frac{\|H_{i_t}\|_{\max}}{\epsilon/2T}\right)\right). \quad (\text{A21})$$

Plugging Eq. (A20) into this, we get Eq. (33). Adding the qubits on which the quantum state in Eq. (A17) is generated, whose number is $2\lg N_{\text{gr}} + 5 = O(\log n_{\text{gr}})$, does not change the order. ■

3. Proof of Theorem 2

Continuation of the proof Let us see that the stated accuracy is achieved. By $U_{\mathbf{f}(T), C\epsilon/4}$, we get the following state:

$$U_{\mathbf{f}(T), C\epsilon/4} |0\rangle^{\otimes(2\lg N_{\text{gr}}+5)} = |0\rangle^{\otimes(\lg N_{\text{gr}}+5)} |\mathbf{f}(T)\rangle + |\phi_{\text{gar}}\rangle, \quad (\text{A22})$$

where $|\phi_{\text{gar}}\rangle$ is an unnormalized state with $\| |\phi_{\text{gar}}\rangle \| \leq C\epsilon/4$. Thus, defining

$$p := |\langle \Omega'_{\mathbf{i}_k} | W' U_{\mathbf{f}(T), C\epsilon/4} |0\rangle^{\otimes(2\lg N_{\text{gr}}+5)} |^2, \quad (\text{A23})$$

we obtain

$$\begin{aligned}
 & |p - C|\tilde{\delta}_{\mathbf{i}_k}^v|^2| \\
 & = |2\Re\langle \Omega'_{\mathbf{i}_k} | W |\mathbf{f}(T)\rangle \langle \Omega'_{\mathbf{i}_k} | W' |\phi_{\text{gar}}\rangle + |\langle \Omega'_{\mathbf{i}_k} | W' |\phi_{\text{gar}}\rangle|^2| \\
 & \leq 2\frac{C\epsilon}{4} + \left(\frac{C\epsilon}{4}\right)^2 \\
 & \leq \frac{3C\epsilon}{4}. \quad (\text{A24})
 \end{aligned}$$

Here, at the first inequality, we use $|\langle \Omega'_{\mathbf{i}_k} | W |\mathbf{f}(T)\rangle| \leq 1$ and $\| \langle \Omega'_{\mathbf{i}_k} | W' |\phi_{\text{gar}}\rangle \| \leq \| (W')^\dagger | \Omega'_{\mathbf{i}_k} \rangle \| \cdot \| |\phi_{\text{gar}}\rangle \| \leq \frac{C\epsilon}{4}$, which follows from the Cauchy-Schwarz inequality. \tilde{p} obtained in Step

3 in Algorithm 1 is an $C\epsilon/4$ -approximation of p , and combining this with Eq. (A24) yields

$$\left| \frac{\tilde{p}}{C} - |\tilde{\delta}_{\mathbf{k}}^v|^2 \right| \leq \left| \frac{\tilde{p}}{C} - \frac{p}{C} \right| + \left| \frac{p}{C} - |\tilde{\delta}_{\mathbf{k}}^v|^2 \right| \leq \frac{\epsilon}{4} + \frac{3\epsilon}{4} \leq \epsilon. \quad (\text{A25})$$

Thus, the error of the output of Algorithm 1 is bounded by ϵ .

Last, let us prove the statement on the query complexity. In the QAE in Step 3 in Algorithm 1, $U_{\mathbf{f}(T), C\epsilon/4}$ is queried

$$O\left(\frac{1}{C\epsilon} \log\left(\frac{1}{\delta}\right)\right) \quad (\text{A26})$$

times. Because of Theorem 1, in one call to $U_{\mathbf{f}(T), C\epsilon/4}$, the number of queries to (controlled) $\{O_{F_{\text{CDM},x}}^i, O_{F_{\text{CDM},y}}^i, O_{F_{\text{CDM},z}}^i\}_{i=1}^n$, arithmetic circuits and their inverses is

$$O\left[n_{\text{gr}} T \times \max\left\{\frac{V}{L}, \frac{F_{\text{max}}}{V}\right\} + n_t \log\left(\frac{n_t}{C\epsilon}\right)\right], \quad (\text{A27})$$

and that to $O_{\mathbf{f}(0)}$ is 1. Then, combining these, we bound the total query number in Algorithm 1 as Eqs. (43) and (44).

On the qubit number, we note that, to operate $U_{\mathbf{f}(T), C\epsilon/4}$, we use qubits whose number is of order (45), as implied by Theorem 1. Since operating W and QAE do not require additional qubits, the number of qubits used in the whole of Algorithm 1 is also of order (45).

-
- [1] S. Dodelson and F. Schmidt, *Modern Cosmology* (Academic Press, San Diego, CA, 2020).
 - [2] Planck Collaboration, Planck 2018 Results—VI. Cosmological parameters, *Astron. Astrophys.* **641**, A6 (2020).
 - [3] Super-Kamiokande Collaboration, Evidence for oscillation of atmospheric neutrinos, *Phys. Rev. Lett.* **81**, 1562 (1998).
 - [4] G. Pagliaroli, F. Rossi-Torres, and F. Vissani, Neutrino mass bound in the standard scenario for supernova electronic antineutrino emission, *Astropart. Phys.* **33**, 287 (2010).
 - [5] M. Aker *et al.* (KATRIN Collaboration), Direct neutrino-mass measurement with sub-electronvolt sensitivity, *Nat. Phys.* **18**, 160 (2022).
 - [6] E. Di Valentino, S. Gariazzo, and O. Mena, Most constraining cosmological neutrino mass bounds, *Phys. Rev. D* **104**, 083504 (2021).
 - [7] S. Abe *et al.* (KamLAND-Zen Collaboration), Search for the majorana nature of neutrinos in the inverted mass ordering region with KamLAND-Zen *Phys. Rev. Lett.* **130**, 051801 (2023).
 - [8] S. Weinberg, A new light boson? *Phys. Rev. Lett.* **40**, 223 (1978).
 - [9] F. Wilczek, Problem of strong P and T invariance in the presence of instantons, *Phys. Rev. Lett.* **40**, 279 (1978).
 - [10] L. F. Abbott and P. Sikivie, A cosmological bound on the invisible axion, *Phys. Lett. B* **120**, 133 (1983).
 - [11] J. Preskill, M. B. Wise, and F. Wilczek, Cosmology of the invisible axion, *Phys. Lett. B* **120**, 127 (1983).
 - [12] M. Dine and W. Fischler, The not so harmless axion, *Phys. Lett. B* **120**, 137 (1983).
 - [13] D. J. Marsh, Axion cosmology, *Phys. Rep.* **643**, 1 (2016), axion cosmology.
 - [14] J. Brandbyge and S. Hannestad, Grid based linear neutrino perturbations in cosmological N-body simulations, *J. Cosmol. Astropart. Phys.* **05** (2009) 002.
 - [15] J. Brandbyge and S. Hannestad, Resolving cosmic neutrino structure: A hybrid neutrino N-body scheme, *J. Cosmol. Astropart. Phys.* **01** (2010) 021.
 - [16] J. Brandbyge, S. Hannestad, T. Haugbølle, and Y. Y. Wong, Neutrinos in non-linear structure formation—The effect on halo properties, *J. Cosmol. Astropart. Phys.* **09** (2010) 014.
 - [17] M. Viel, M. G. Haehnelt, and V. Springel, The effect of neutrinos on the matter distribution as probed by the intergalactic medium, *J. Cosmol. Astropart. Phys.* **06** (2010) 015.
 - [18] S. Bird, M. Viel, and M. G. Haehnelt, Massive neutrinos and the non-linear matter power spectrum, *Mon. Not. R. Astron. Soc.* **420**, 2551 (2012).
 - [19] F. Villaescusa-Navarro, F. Marulli, M. Viel, E. Branchini, E. Castorina, E. Sefusatti, and S. Saito, Cosmology with massive neutrinos I: Towards a realistic modeling of the relation between matter, haloes and galaxies, *J. Cosmol. Astropart. Phys.* **03** (2014) 011.
 - [20] E. Castorina, E. Sefusatti, R. K. Sheth, F. Villaescusa-Navarro, and M. Viel, Cosmology with massive neutrinos II: On the universality of the halo mass function and bias, *J. Cosmol. Astropart. Phys.* **02** (2014) 049.
 - [21] D. Inman, J. D. Emberson, U.-L. Pen, A. Farchi, H.-R. Yu, and J. Harnois-Déraps, Precision reconstruction of the cold dark matter-neutrino relative velocity from n -body simulations, *Phys. Rev. D* **92**, 023502 (2015).
 - [22] A. Banerjee, D. Powell, T. Abel, and F. Villaescusa-Navarro, Reducing noise in cosmological N-body simulations with neutrinos, *J. Cosmol. Astropart. Phys.* **09** (2018) 028.
 - [23] F. Villaescusa-Navarro, C. Hahn, E. Massara *et al.*, The Quijote simulations, *Astrophys. J. Suppl. Ser.* **250**, 2 (2020).
 - [24] K. Yoshikawa, S. Tanaka, N. Yoshida, and S. Saito, Cosmological Vlasov-Poisson simulations of structure formation with relic neutrinos: Nonlinear clustering and the neutrino mass, *Astrophys. J.* **904**, 159 (2020).
 - [25] K. Yoshikawa, S. Tanaka, and N. Yoshida, A 400 trillion-grid Vlasov simulation on Fugaku supercomputer: Large-scale distribution of cosmic relic neutrinos in a six-dimensional phase space, in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '21* (ACM Press, New York, 2021).
 - [26] S. Tanaka, K. Yoshikawa, N. Yoshida, and S. Saito, Comparison of Vlasov and N-body simulations of massive neutrinos in the large-scale structure (unpublished).
 - [27] S. Lloyd, Universal quantum simulators, *Science* **273**, 1073 (1996).
 - [28] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Efficient quantum algorithms for simulating sparse Hamiltonians, *Commun. Math. Phys.* **270**, 359 (2007).
 - [29] A. M. Childs, On the relationship between continuous and discrete-time quantum walk, *Commun. Math. Phys.* **294**, 581 (2010).

- [30] A. M. Childs and R. Kothari, Simulating sparse Hamiltonians with star decompositions, in *Theory of Quantum Computation, Communication, and Cryptography*, edited by W. van Dam, V. M. Kendon, and S. Severini (Springer, Berlin, 2011), pp. 94–103.
- [31] A. M. Childs and N. Wiebe, Hamiltonian simulation using linear combinations of unitary operations, *Quantum Inf. Comput.* **12**, 901 (2012).
- [32] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, *Exponential Improvement in Precision for Simulating Sparse Hamiltonians* (ACM Press, New York, 2014), pp. 283–292.
- [33] D. W. Berry, A. M. Childs, and R. Kothari, Hamiltonian simulation with nearly optimal dependence on all parameters, in *Proceedings of the IEEE 56th Annual Symposium on Foundations of Computer Science* (IEEE, Los Alamitos, CA, 2015), pp. 792–809.
- [34] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Simulating Hamiltonian dynamics with a truncated Taylor series, *Phys. Rev. Lett.* **114**, 090502 (2015).
- [35] G. H. Low and I. L. Chuang, Optimal Hamiltonian simulation by quantum signal processing, *Phys. Rev. Lett.* **118**, 010501 (2017).
- [36] G. H. Low and I. L. Chuang, Hamiltonian simulation by qubitization, *Quantum* **3**, 163 (2019).
- [37] Y. Cao, A. Papageorgiou, I. Petras, J. Traub, and S. Kais, Quantum algorithm and circuit design solving the Poisson equation, *New J. Phys.* **15**, 013021 (2013).
- [38] D. W. Berry, High-order quantum algorithm for solving linear differential equations, *J. Phys. A: Math. Theor.* **47**, 105301 (2014).
- [39] A. Montanaro and S. Pallister, Quantum algorithms and the finite element method, *Phys. Rev. A* **93**, 032324 (2016).
- [40] D. W. Berry, A. M. Childs, A. Ostrander, and G. Wang, Quantum algorithm for linear differential equations with exponentially improved dependence on precision, *Commun. Math. Phys.* **356**, 1057 (2017).
- [41] P. C. S. Costa, S. Jordan, and A. Ostrander, Quantum algorithm for simulating the wave equation, *Phys. Rev. A* **99**, 012323 (2019).
- [42] T. Xin, S. Wei, J. Cui, J. Xiao, I. n. Arrazola, L. Lamata, X. Kong, D. Lu, E. Solano, and G. Long, Quantum algorithm for solving linear differential equations: Theory and experiment, *Phys. Rev. A* **101**, 032307 (2020).
- [43] A. M. Childs and J.-P. Liu, Quantum spectral methods for differential equations, *Commun. Math. Phys.* **375**, 1427 (2020).
- [44] S. Wang, Z. Wang, W. Li, L. Fan, Z. Wei, and Y. Gu, Quantum fast Poisson solver: The algorithm and complete and modular circuit design, *Quant. Info. Proc.* **19**, 1 (2020).
- [45] A. M. Childs, J.-P. Liu, and A. Ostrander, High-precision quantum algorithms for partial differential equations, *Quantum* **5**, 574 (2021).
- [46] N. Linden, A. Montanaro, and C. Shao, Quantum vs. classical algorithms for solving the heat equation, *Commun. Math. Phys.* **395**, 601 (2022).
- [47] D. An, J.-P. Liu, D. Wang, and Q. Zhao, A theory of quantum differential equation solvers: Limitations and fast-forwarding, [arXiv:2211.05246](#).
- [48] D. W. Berry and P. Costa, Quantum algorithm for time-dependent differential equations using Dyson series, [arXiv:2212.03544](#).
- [49] S. Jin, N. Liu, and Y. Yu, Quantum simulation of partial differential equations via Schrödingerisation: Technical details, [arXiv:2212.14703](#).
- [50] H. Krovi, Improved quantum algorithms for linear and nonlinear differential equations, *Quantum* **7**, 913 (2023).
- [51] D. Fang, L. Lin, and Y. Tong, Time-marching based quantum solvers for time-dependent linear differential equations, *Quantum* **7**, 955 (2023).
- [52] J. Hu, S. Jin, and L. Zhang, Quantum algorithms for multiscale partial differential equations, [arXiv:2304.06902](#).
- [53] S. Jin, X. Li, N. Liu, and Y. Yu, Quantum simulation for partial differential equations with physical boundary or interface conditions, [arXiv:2305.02710](#).
- [54] M. Bagherimehrab, K. Nakaji, N. Wiebe, and A. Aspuru-Guzik, Fast quantum algorithm for differential equations, [arXiv:2306.11802](#).
- [55] S.-J. Wei, C. Wei, P. Lv, C. Shao, P. Gao, Z. Zhou, K. Li, T. Xin, and G.-L. Long, A quantum algorithm for heat conduction with symmetrization, *Sci. Bull.* **68**, 494 (2023).
- [56] S. Lloyd, G. De Palma, C. Gokler, B. Kiani, Z.-W. Liu, M. Marvian, F. Tennie, and T. Palmer, Quantum algorithm for nonlinear differential equations, [arXiv:2011.06571](#).
- [57] J.-P. Liu, H. Ø. Kolden, H. K. Krovi, N. F. Loureiro, K. Trivisa, and A. M. Childs, Efficient quantum algorithm for dissipative nonlinear differential equations, *Proc. Natl. Acad. Sci. USA* **118**, e2026805118 (2021).
- [58] C. Xue, Y.-C. Wu, and G.-P. Guo, Quantum homotopy perturbation method for nonlinear dissipative ordinary differential equations, *New J. Phys.* **23**, 123035 (2021).
- [59] J.-P. Liu, D. An, D. Fang, J. Wang, G. H. Low, and S. Jordan, Efficient quantum algorithm for nonlinear reaction-diffusion equations and energy estimation, *Commun. Math. Phys.* **404**, 963 (2023).
- [60] S. Jin and N. Liu, Quantum algorithms for computing observables of nonlinear partial differential equations, [arXiv:2202.07834](#).
- [61] S. Jin, N. Liu, and Y. Yu, Time complexity analysis of quantum algorithms via linear representations for nonlinear ordinary and partial differential equations, *J. Comput. Phys.* **487**, 112149 (2023).
- [62] Y. Tanaka and K. Fujii, Quantum solvable nonlinear differential equations, [arXiv:2305.00653](#).
- [63] S. Aaronson, Read the fine print, *Nat. Phys.* **11**, 291 (2015).
- [64] G. Brassard *et al.*, Quantum amplitude amplification and estimation, *Contemp. Math.* **305**, 53 (2002).
- [65] V. Giovannetti, S. Lloyd, and L. Maccone, Quantum random access memory, *Phys. Rev. Lett.* **100**, 160501 (2008).
- [66] A. Engel, G. Smith, and S. E. Parker, Quantum algorithm for the Vlasov equation, *Phys. Rev. A* **100**, 062315 (2019).
- [67] A. Ameri, E. Ye, P. Cappellaro, H. Krovi, and N. F. Loureiro, Quantum algorithm for the linear Vlasov equation with collisions, *Phys. Rev. A* **107**, 062412 (2023).
- [68] K. Toyozumi, N. Yamamoto, and K. Hoshino, Hamiltonian simulation using quantum singular value transformation: Complexity analysis and application to the linearized Vlasov-Poisson equation, *Phys. Rev. A* **109**, 012430 (2024).

- [69] H.-R. Yu, J. D. Emberson, D. Inman, T.-J. Zhang, U.-L. Pen, J. Harnois-Déraps, S. Yuan, H.-Y. Teng, H.-M. Zhu, X. Chen, Z.-Z. Xing, Y. Du, L. Zhang, Y. Lu, and X. Liao, Differential neutrino condensation onto cosmic structure, *Nat. Astron.* **1**, 0143 (2017).
- [70] A. Engel, G. Smith, and S. E. Parker, Linear embedding of nonlinear dynamical systems and prospects for efficient quantum algorithms, *Phys. Plasmas* **28**, 062305 (2021).
- [71] T. Carleman, Application de la théorie des équations intégrales linéaires aux systèmes d'équations différentielles non linéaires, *Acta Math.* **59**, 63 (1932).
- [72] I. Y. Dodin and E. A. Startsev, On applications of quantum computing to plasma simulations, *Phys. Plasmas* **28**, 092101 (2021).
- [73] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio *et al.*, Variational quantum algorithms, *Nat. Rev. Phys.* **3**, 625 (2021).
- [74] P. Mocz and A. Szasz, Toward cosmological simulations of dark matter on quantum computers, *Astrophys. J.* **910**, 29 (2021).
- [75] L. Cappelli, F. Tacchino, G. Murante, S. Borgani, and I. Tavernelli, From Vlasov-Poisson to Schrödinger-Poisson: Dark matter simulation with a quantum variational time evolution algorithm, [arXiv:2307.06032](https://arxiv.org/abs/2307.06032).
- [76] W. Hu, R. Barkana, and A. Gruzinov, Fuzzy cold dark matter: The wave properties of ultralight particles, *Phys. Rev. Lett.* **85**, 1158 (2000).
- [77] B. N. Todorova and R. Steijl, Quantum algorithm for the collisionless Boltzmann equation, *J. Comput. Phys.* **409**, 109347 (2020).
- [78] S. Yamazaki, F. Uchida, K. Fujisawa, and N. Yoshida, Quantum algorithm for collisionless Boltzmann simulation of self-gravitating systems, [arXiv:2303.16490](https://arxiv.org/abs/2303.16490).
- [79] G. Mangano, G. Miele, S. Pastor, T. Pinto, O. Pisanti, and P. D. Serpico, Relic neutrino decoupling including flavor oscillations, *Nucl. Phys. B* **729**, 221 (2005).
- [80] C. Nascimento and M. Loverde, Neutrino winds on the sky, *J. Cosmol. Astropart. Phys.* **11** (2023) 036.
- [81] M. Costanzi, F. Villaescusa-Navarro, M. Viel, J.-Q. Xia, S. Borgani, E. Castorina, and E. Sefusatti, Cosmology with massive neutrinos III: The halo mass function and an application to galaxy clusters, *J. Cosmol. Astropart. Phys.* **12** (2013) 012.
- [82] M. LoVerde, Halo bias in mixed dark matter cosmologies, *Phys. Rev. D* **90**, 083530 (2014).
- [83] E. Castorina, C. Carbone, J. Bel, E. Sefusatti, and K. Dolag, DEMNUni: The clustering of large-scale structures in the presence of massive neutrinos, *J. Cosmol. Astropart. Phys.* **07** (2015) 043.
- [84] F. Villaescusa-Navarro, A. Banerjee, N. Dalal, E. Castorina, R. Scoccimarro, R. Angulo, and D. N. Spergel, The imprint of neutrinos on clustering in redshift space, *Astrophys. J.* **861**, 53 (2018).
- [85] S. Vagnozzi, T. Brinckmann, M. Archidiacono, K. Freese, M. Gerbino, J. Lesgourgues, and T. Sprenger, Bias due to neutrinos must not uncorrect'd go, *J. Cosmol. Astropart. Phys.* **09** (2018) 001.
- [86] A. Raccanelli, L. Verde, and F. Villaescusa-Navarro, Biases from neutrino bias: To worry or not to worry? *Mon. Not. R. Astron. Soc.* **483**, 734 (2019).
- [87] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, UK, 2002).
- [88] E. Muñoz-Coreas and H. Thapliyal, Everything you always wanted to know about quantum circuits, in *Wiley Encyclopedia of Electrical and Electronics Engineering* (John Wiley & Sons, New York, 2022), pp. 1–17.
- [89] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su, Toward the first quantum simulation with quantum speedup, *Proc. Natl. Acad. Sci. USA* **115**, 9456 (2018).
- [90] S. Chakraborty, A. Gilyén, and S. Jeffery, The power of block-encoded matrix powers: Improved regression techniques via faster Hamiltonian simulation, in *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP '19)* (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2019), Vol. 132, pp. 33:1–33:14.
- [91] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019 (ACM Press, New York, 2019), pp. 193–204.
- [92] S. Arunachalam, V. Gheorghiu, T. Jochym-O'Connor, M. Mosca, and P. V. Srinivasan, On the robustness of bucket brigade quantum RAM, *New J. Phys.* **17**, 123010 (2015).
- [93] S. Jaques and A. G. Rattew, QRAM: A survey and critique, [arXiv:2305.10310](https://arxiv.org/abs/2305.10310).
- [94] G. H. Low, V. Kliuchnikov, and L. Schaeffer, Trading T-gates for dirty qubits in state preparation and unitary synthesis, [arXiv:1812.00954](https://arxiv.org/abs/1812.00954).
- [95] D. K. Park, F. Petruccione, and J.-K. K. Rhee, Circuit-based quantum random access memory for classical data, *Sci. Rep.* **9**, 3949 (2019).
- [96] A. Paler, O. Oumarou, and R. Basmadjian, Parallelizing the queries in a bucket-brigade quantum random access memory, *Phys. Rev. A* **102**, 032608 (2020).
- [97] O. D. Matteo, V. Gheorghiu, and M. Mosca, Fault-tolerant resource estimation of quantum random-access memories, *IEEE Trans. Quantum Eng.* **1**, 1 (2020).
- [98] C. T. Hann, G. Lee, S. M. Girvin, and L. Jiang, Resilience of quantum random access memory to generic noise, *PRX Quantum* **2**, 020311 (2021).
- [99] Y. P. Jing, Correcting for the alias effect when measuring the power spectrum using a fast Fourier transform, *Astrophys. J.* **620**, 559 (2005).
- [100] M. Kieferová, A. Scherer, and D. W. Berry, Simulating the dynamics of time-dependent Hamiltonians with a truncated Dyson series, *Phys. Rev. A* **99**, 042314 (2019).
- [101] I. Kerenidis and A. Prakash, Quantum recommendation systems, in *8th Innovations in Theoretical Computer Science Conference (ITCS '17)* (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2017).
- [102] L. Grover and T. Rudolph, Creating superpositions that correspond to efficiently integrable probability distributions, [arXiv:quant-ph/0208112](https://arxiv.org/abs/quant-ph/0208112).
- [103] Y. R. Sanders, G. H. Low, A. Scherer, and D. W. Berry, Black-box quantum state preparation without arithmetic, *Phys. Rev. Lett.* **122**, 020502 (2019).

- [104] G. Marin-Sanchez, J. Gonzalez-Conde, and M. Sanz, Quantum algorithms for approximate function loading, [Phys. Rev. Res.](#) **5**, 033114 (2023).
- [105] A. G. Rattew and B. Koczor, Preparing arbitrary continuous functions in quantum registers with logarithmic complexity, [arXiv:2205.00519](#).
- [106] S. Tanaka, K. Yoshikawa, T. Minoshima, and N. Yoshida, Multidimensional Vlasov-Poisson simulations with high-order monotonicity- and positivity-preserving schemes, [Astrophys. J.](#) **849**, 76 (2017).
- [107] A. Montanaro, Quantum speedup of Monte Carlo methods, [Proc. R. Soc. A](#) **471**, 20150301 (2015).
- [108] M. R. Jerrum, L. G. Valiant, and V. V. Vazirani, Random generation of combinatorial structures from a uniform distribution, [Theor. Comput. Sci.](#) **43**, 169 (1986).
- [109] A. Y. Kitaev, Quantum measurements and the Abelian stabilizer problem, [Electr. Colloq. Comput. Complex.](#) **3** (1995).
- [110] Y. Suzuki, S. Uno, R. Raymond, T. Tanaka, T. Onodera, and N. Yamamoto, Amplitude estimation without phase estimation, [Quantum Inf. Process.](#) **19**, 75 (2020).
- [111] D. Grinko, J. Gacon, C. Zoufal, and S. Woerner, Iterative quantum amplitude estimation, [npj Quant. Inf.](#) **7**, 52 (2021).
- [112] S. Aaronson and P. Rall, Quantum approximate counting, simplified, in *Symposium on Simplicity in Algorithms (SOSA '20)* (SIAM, Philadelphia, 2020), pp. 24–32.
- [113] S. Fukuzawa, C. Ho, S. Irani, and J. Zion, Modified iterative quantum amplitude estimation is asymptotically optimal, in *Proceedings of the Symposium on Algorithm Engineering and Experiments (ALENEX)* (2023), pp. 135–147.