Million-Atom Ab Initio Electron Dynamics: Discontinuous Galerkin Real-Time Time-Dependent Density Functional Theory

Junwei Feng
University of Science and Technology of China
Hefei, Anhui, China
fengjw@mail.ustc.edu.cn

Junshi Chen
University of Science and Technology of China
Hefei, Anhui, China
Laoshan Laboratory
Qingdao, Shandong, China
cjuns@ustc.edu.cn

Xiangyu Zhang
State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
University of Chinese Academy of Sciences
Beijing, China
zhangxiangyu22z@ict.ac.cn

Junhui Liu
University of Science and Technology of China
Hefei, Anhui, China
ljh23011@mail.ustc.edu.cn

Xinming Qin
University of Science and Technology of China
Hefei, Anhui, China
xmqin03@ustc.edu.cn

Lingyun Wan
University of Science and Technology of China
Hefei, Anhui, China
wanly@mail.ustc.edu.cn

Sheng Chen
University of Science and Technology of China
Hefei, Anhui, China
chens98@mail.ustc.edu.cn

Wentiao Wu
University of Science and Technology of China
Hefei, Anhui, China
wtwu@mail.ustc.edu.cn

Bingkun Hou
University of Science and Technology of China
Hefei, Anhui, China
bingkunhou@mail.ustc.edu.cn

Yexuan Lin
University of Science and Technology of China
Hefei, Anhui, China
lyx0724@mail.ustc.edu.cn

Yihong Zhang
University of Science and Technology of China
Hefei, Anhui, China
zhangyihong@mail.ustc.edu.cn

Zechuan Zhang
University of Science and Technology of China
Hefei, Anhui, China
zhangzechuan@mail.ustc.edu.cn

Yijun Hu
University of Science and Technology of China
Hefei, Anhui, China
huyijun@mail.ustc.edu.cn

Weile Jia
State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
University of Chinese Academy of Sciences
Beijing, China
jiaweile@ict.ac.cn

Hong An
University of Science and Technology of China
Hefei, Anhui, China
Laoshan Laboratory
Qingdao, Shandong, China
han@ustc.edu.cn

Jinlong Yang
University of Science and Technology of China
Hefei, Anhui, China
jlyang@ustc.edu.cn

Wei Hu
University of Science and Technology of China
Hefei, Anhui, China
whuustc@ustc.edu.cn

†Junwei Feng, Junshi Chen, Xiangyu Zhang, and Junhui Liu contributed equally to this work.

*Corresponding Author: Xinming Qin (xmqin03@ustc.edu.cn), Weile Jia (jiaweile@ict.ac.cn), Hong An (han@ustc.edu.cn), Wei Hu (whuustc@ustc.edu.cn).

SC '25, St Louis, MO, USA

Abstract

Over the past decades, first-principles real-time time dependent density functional theory(rt-TDDFT) simulations have been limited to systems with only thousands of atoms. We propose a novel method based on the discontinuous Galerkin adaptive local basis, significantly reducing global communication in rt-TDDFT. We further introduce a tensor compression technique that leverages basis locality to avoid repeated evaluation of multi-center integrals in hybrid functionals, greatly reducing computational cost. To overcome the projection bottleneck in our basis sets, we design a fused Gemm-Reduce operation that achieves several times higher floating-point efficiency than standard BLAS combination. Our implementation reaches 34.8% of theoretical peak performance on 524,288 CGs of the New Sunway supercomputer and simulates electronic dynamics of systems with over one million atoms for both local-semi-local and hybrid functionals. This work improves computational scale by two orders of magnitude, opening new possibilities for exploring ultrafast dynamics in large-scale materials and nanophotonic devices.

CCS Concepts

• **Applied computing → Physics**.

Keywords

High performance computing, discontinuous Galerkin method, real-time time-dependent density functional theory, supercomputer

ACM Reference Format:

Junwei Feng, Junshi Chen, Xiangyu Zhang, Junhui Liu, Xinming Qin, Lingyun Wan, Sheng Chen, Wentiao Wu, Bingkun Hou, Yexuan Lin, Yihong Zhang, Zechuan Zhang, Yijun Hu, Weile Jia, Hong An, Jinlong Yang, and Wei Hu. 2025. Million-Atom Ab Initio Electron Dynamics: Discontinuous Galerkin Real-Time Time-Dependent Density Functional Theory. In *The International Conference for High Performance Computing, Networking, Storage and Analysis (SC '25), November 16–21, 2025, St Louis, MO, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3712285.3759813

1 OVERVIEW OF THE PROBLEM

Time-dependent phenomena, such as photoexcitation, electron relaxation, and non-equilibrium charge dynamics, play a crucial role in various scientific and engineering applications, including ultrafast spectroscopy[39], photocatalysis[13], solar cells[19], laser-induced phase transitions[31], organic optoelectronics[4], nonlinear optics[18], and other related phenomena. These processes, driven by external perturbations like light or electric fields, involve transient excited states and evolve on ultrafast timescales. Conventional density functional theory (DFT)[24] has been proven highly successful in describing ground-state properties, but it is inherently a static theory and thus incapable of capturing time-dependent behavior. Likewise, *ab initio* molecular dynamics (AIMD) considers only nuclear motion and therefore cannot resolve purely electronic processes occurring on femtosecond timescales.

ACM ISBN 979-8-4007-1466-5/25/11
https://doi.org/10.1145/3712285.3759813

To capture such ultrafast phenomena from first-principles simulations, a theoretical framework capable of resolving the time evolution of electronic states is required. To this end, time-dependent density functional theory (TDDFT), proposed by E. Runge and E. K. U. Gross in 1984[36], extends the original DFT formalism into the time domain. In this framework, the static Kohn–Sham equations arise as a special case of the time-dependent Kohn–Sham (TDKS) equations, as illustrated in Figure 1. TDDFT thus provides a rigorous *ab initio* approach for simulating excited-state dynamics in real time, making it an essential tool for studying excited-state dynamics in complex systems.

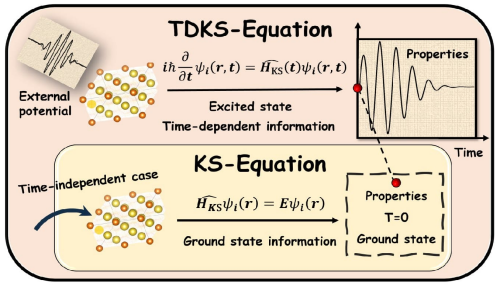

Figure 1: Differences between Kohn-Sham equation and time-dependent Kohn-Sham equation.

In recent years, large-scale *ab initio* simulations, widely applied in condensed matter physics, computational chemistry, and material simulations have primarily focused on static DFT calculations[7] or nuclear dynamics at the atomic level through ground-state AIMD calculations[37]. However, large-scale electron dynamics studies have yet to emerge.

A major challenge is that state-of-the-art large-scale electron dynamics simulations primarily rely on grid-based methods, which inherently exhibit computational characteristics that hinder scalability. These methods include a large number of required basis functions, leading to significant memory overhead, and the reliance on global Fourier transforms for Hamiltonian–vector multiplications in plane-wave methods[28], or large-scale sparse matrix–vector multiplications in real-space methods[14], both of which result in severe communication and memory bandwidth bottlenecks.

Another key challenge lies in excited-state simulations, which require a more accurate band gap description compared to ground-state calculations. To address this, hybrid functionals are typically employed to provide a more reliable treatment of exchange interactions. However, in conventional grid-based implementations, hybrid functionals require the reconstruction of complex multi-center integrals at each time step, with a computational cost that scales as $O(N^4)$. As a result, time propagation at hybrid-functional accuracy becomes computationally prohibitive for large-scale systems.

Additionally, when using non-grid-based methods to reduce the computational bottlenecks mentioned earlier, projecting wavefunctions onto real-space grids for electron density calculation often becomes the main cost. This projection involves a tall and narrow matrix multiplication followed by a reduction, both of which are limited by memory bandwidth and reduce floating-point performance.

In ground-state calculations, the diagonalization procedure typically dominates the computational cost. As a result, the scalability bottlenecks of other components are often masked by the time spent on eigensolver routines, and ensuring a scalable eigenvalue solver is usually sufficient to achieve overall scalability. In contrast, excited-state simulations require that all computational components exhibit both good parallel scalability and high floating-point efficiency in order to maintain overall performance.

1.1 Summary of Contributions

This work presents significant algorithmic innovations and system optimizations that, for the first time, enable large-scale electron dynamics simulations with both local/semi-local and hybrid functionals. These developments enable efficient *ab initio* simulations of electron dynamics in large, complex systems at practical scales.

Our algorithmic innovations can be summarized as follows:

(1) A novel discretization scheme is introduced for solving the time-dependent Kohn–Sham equations. By exploiting the locality of the discontinuous Galerkin adaptive local basis set, the method eliminates most of the global communication present in grid-based approaches. Meanwhile, irregular sparse matrix multiplications are converted into a combination of nearest-neighbor communication and small dense matrix operations, significantly reducing both communication bottlenecks and memory bandwidth limitations.

(2) A linear-scaling algorithm based on the DG-ALB basis set is proposed for hybrid-TDDFT, which reduces the computational complexity of electron repulsion integrals from $O(N^4)$ to $O(N)$. Furthermore, to reduce the high cost of repeatedly evaluating multicenter integrals in hybrid-functional rt-TDDFT, we adopt the interpolative separable density fitting method for efficient tensor compression. This enables precomputing and storing all costly integrals with modest memory overhead, eliminating redundant operations during real-time propagation and achieving up to a nearly 100× speedup over plane-wave method.

(3) A fused Gemm-Reduce operator is introduced to address performance bottlenecks in our basis set calculations. Resolves the high cache-to-memory data movement cost from decoupled tall skinny matrix zGemm and reduction operations. This optimization boosts the floating-point efficiency of hotspot computations from 10% to 82%.

These innovative methods enable our TDDFT simulations to perform real-time electron dynamics with hybrid functional accuracy for systems containing over one million atoms on the nearly full Sunway system (524,832 CGs) — a feat that is unprecedented even for lower-accuracy functional simulations.

2 CURRENT STATE OF THE ART

The mainstream methods for excited-state simulations currently include linear-response time-dependent density functional theory (lr-TDDFT)[38], Green's function–based approaches[8], real-time time-dependent density functional theory (rt-TDDFT) , and others.

lr-TDDFT can be viewed as a linear-response approximation to real-time TDDFT. It cannot capture nonlinear phenomena induced by strong external fields, and while its applicability is somewhat limited and its accuracy moderate, it remains the most widely used method for excited-state simulations due to its key advantage of avoiding explicit time propagation. However, its computational scaling is theoretically $O(N^5 - N^6)$, making large-scale simulations challenging. To date, the largest reported lr-TDDFT calculation has been limited to systems containing no more than 4,096 silicon atoms[23].

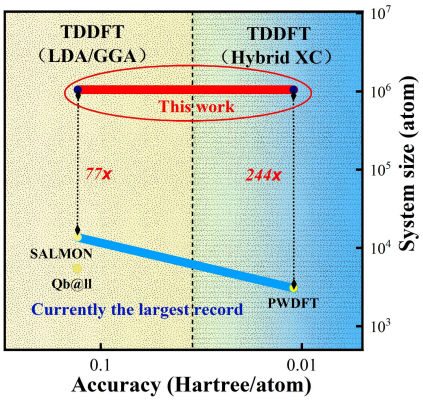

Figure 2: The relationship between system size and accuracy in different methods for rt-TDDFT simulation.

Among Green's function–based methods, the GW approximation is the most representative[9]. It provides an accurate description of quasiparticles and enables the computation of quasiparticle band structures, offering access to a subset of excited-state properties. When combined with the Bethe–Salpeter equation (BSE), the approach can be extended to describe additional phenomena such as exciton binding energies, thereby broadening its applicability. However, the explicit inversion of large matrices required in GW calculations poses a significant computational bottleneck, limiting its scalability. Currently, GW-BSE simulations are restricted to systems containing up to 3,903 silicon atoms[32].

rt-TDDFT is an *ab initio* approach that simulates the time evolution of a system by directly solving the time-dependent Kohn–Sham equations. In contrast to LR-TDDFT, rt-TDDFT does not rely on the linear approximation and is thus capable of capturing nonlinear effects induced by strong external fields. Furthermore, due to its explicit time-evolution scheme, it can simulate ultrafast nonequilibrium phenomena that are inaccessible to GW-BSE methods, offering excellent generality. The accuracy of rt-TDDFT depends on the choice of exchange–correlation functional: local and semi-local functionals are typically used for low-precision simulations, while hybrid functionals are preferred when high accuracy is required. The computational scaling of rt-TDDFT is typically between $O(N^3 - N^4)$. Due to the broad applicability of real-time TDDFT and its potential for reduced computational scaling through appropriate basis sets and linear-scaling algorithms, it offers promising prospects for extension to large-scale systems. Therefore, in this section, we focus on recent advances related to rt-TDDFT.

Numerous efforts have been made in recent years to extend rt-TDDFT to larger-scale systems. To date, the two largest local and semi-local functionals TDDFT simulations have been performed by SALMON[14] and Qb@ll[10]. SALMON simulated a SiO_2 system

Table 1: Characteristics of rt-TDDFT packages, including functionals Level, basis, test systems, number of atoms, system scale and machine. GTO denotes Gaussian-type orbitals, NAO refers to numerical atomic orbitals, PW stands for plane waves, RS represents real-space basis sets, and DG-ALB indicates Discontinuous Galerkin adaptive local basis sets. Instances of "/" indicate that the original text does not provide information for that specific item.

Level	TDDFT packages	Basis	System	Atoms	Scale	Machine	PFLOPS (% of peak)
LDA/GGA	FHI-aims [12]	NAO	oligoacene	800	128 CPU cores	Blue Gene/L	/
	RMG[20]	RS	Ag	2,160	64 nodes	Frontier	/
	QRCODE [5]	PW	Mg porphyrin, h-BN	4,536	16,388 CPU cores	Perlmutter	/
	Qb@ll [10]	PW	Al	5,400	98,304 nodes	Blue Gene/Q	8.75 (43.0%)
	OCTOPUS [1]	RS	chromophores	5,819	32,768 nodes	Blue Gene/P	0.011(10.0%)
	SALMON [14]	RS	SiO_2	13,632	27,648 nodes	Fugaku	2.69 (3.17%)
	DG-TDDFT (This work)	DG-ALB	Li/Na, Ag, $LaAlO_3$/$SrAlO_3$	1M	262,144 CGs	New Sunway, Fugaku	273.7 (45.4%)
Hybrid	Qb@ll[25]	PW	Si	512	704 CPU cores	Dogwood cluster	/
	PWDFT[22]	PW	Si	1,532	786 V100	Summit	0.11 (2%)
	PWDFT[28]	PW	Si	3,072	768 A100	Wuhan	/
	DG-TDDFT (This work)	DG-ALB	Li/Na, Ag, $LaAlO_3$/$SrAlO_3$	1M	524,288 CGs	New Sunway, Fugaku	419.9 (34.8%)

with 13,632 atoms (77,824 electrons) on 27,842 nodes of Fugaku, with a peak performance of 2.69 PFLOP/s, achieving a single-step solution time of 1.2 seconds. QB@ll simulated an aluminum system with 5,700 atoms (59,400 electrons) on 98,304 nodes of Blue Gene/Q, with a peak performance of 8.75 PFLOP/s, and reported a single-step time of 52 seconds. In terms of scalability, both SALMON and QB@ll are fundamentally limited by the large number of basis functions required by grid-based methods, which lead to high memory pressure, as well as by the significant memory bandwidth constraints and global communication overhead introduced by their basis sets. Even with extensive communication optimizations, scaling to larger parallel configurations remains challenging.

The largest real-time TDDFT calculation with hybrid functionals reported so far was performed using PWDFT[28], where the adaptively compressed exchange (ACE) method[26] was employed to reduce the cost associated with repeated applications of the Fock exchange operator. Which simulated a silicon system with 12,288 electrons using 768 A100 GPUs on the Wuhan supercomputing center. The simulation achieved a single-step time of 429.29 seconds. In this work, peak performance was not reported. However, in the 2019 PWDFT work[22], hybrid functional simulations were carried out on a system consisting of 1,532 silicon atoms using Summit, achieving a peak performance of 0.11 PFLOP/s. In terms of scalability, the PWDFT work in Wuhan supercomputer reported that simulations of systems of 6,144 Si atoms (24,576 electrons) could not be performed on A100 GPUs due to memory limitations.

In contrast, our implementation achieves orders-of-magnitude improvements in single-step runtime, peak efficiency, and scalability. By combining the linear-scaling tensor compression with an integral store-and-reuse strategy, we perform high-accuracy hybrid functional simulations of a silver system containing 1024 atoms/19,456 electrons (slightly larger than the 12,288-electron system used in PWDFT) on 512 CGs. This results in nearly 100× speedup in time-to-solution (4.41 s vs. 429.29 s), while requiring significantly lower theoretical peak performance (512CGs: 1.18 PFLOP/s vs. 786 A100: 7.45 PFLOP/s).

In terms of scalability, our distributed basis set greatly reduces global communication, achieving excellent weak scaling. On the New Sunway supercomputer, we simulate a 1,048,576-atom LiNa alloy, reaching 45.4% and 34.8% of peak performance for local/semi-local and hybrid functionals. The hybrid case delivers 419.9 PFLOP/s, 3,817× faster than the previous rt-TDDFT hybrid record.

Current studies on electronic dynamics are summarized in Table 1. The largest-scale simulations achieved by each method are shown in Figure 2 for comparison with our work. Our implementation achieved a scale-up of 77x and 244x for the two precision levels, respectively.

3 INNOVATION AND REALIZATION

3.1 Time-Dependent Kohn-Sham equations

The following is the set of time-Dependent Kohn-Sham (TDKS) equations:

$$i\partial_t \psi_i(t) = H(t, \rho(t))\psi_i(t). \tag{1}$$

Here, $\rho(t)$ represents the time-dependent electron density and $\psi(t)$ represents the wavefunction, both of which are time-dependent, unlike in the simpler ground-state KS equations.

For this type of time-dependent partial differential equations, the choice of time integration scheme must be carefully considered. Commonly used integrators include the fourth-order Runge–Kutta (RK4) method, Crank–Nicolson (CN) scheme, Exponential midpoint (EM) method, and so on. Although the DG-TDDFT framework does not impose specific constraints on the choice of integrator, we adopt the RK4 method in our implementation due to its higher numerical accuracy and better suitability for large-scale systems.

3.2 Parallel Distributed Adaptive Local Basis Function Construction

We construct an adaptive basis set using the discontinuous Galerkin (DG) method to discretize the TDKS equations, yielding a time-dependent Hamiltonian with a regular block-sparse structure. This avoids conventional grid-based approaches, which require redundant pseudopotential evaluations, global Fourier transforms, or large irregular sparse-matrix multiplications. The DG method[6] is widely used for PDEs such as the Navier-Stokes[29] and Hamilton-Jacobi equations[15]. Lin et al.[27] first applied it to ground-state KS equations in 2012, introducing an orthogonal, complete, and localized adaptive basis (ALB) set.

The DG-ALB methodology involves partitioning the global system and leveraging the local systems within each partition to construct basis functions for the overall solution. We refer to each partition as an element E_6, and the surrounding enlarged region as the extended element Q_6, as shown in Figure 3 (a).

Within each element, the corresponding number of Kohn–Sham equations is solved fully in parallel using a steady-state self-consistent

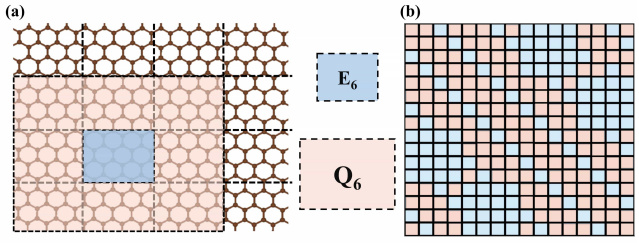

Figure 3: (a) Schematic diagram of localized basis set construction in the DG method, using a graphene system as an example. (b) The block tridiagonal sparse time-dependent Hamiltonian matrix constructed using the DG-ALB.

method to construct the basis functions of the overall Hamiltonian.

$$H^{Q_k}\phi^{Q_k}_{k,j} = \left(-\frac{1}{2}\Delta + V^{Q_k}_{\text{eff}} + V^{Q_k}_{\text{nl}}\right)\phi^{Q_k}_{k,j} = \lambda^{Q_k}_{k,j}\phi^{Q_k}_{k,j} \tag{2}$$

Q_k corresponds to the extended elements in Figure 3 (a), $\lambda^{Q_k}_{k,j}$ represents the j-th eigenvalue in the k-th extended element, and $\phi^{Q_k}_{k,j}$ represents the j-th ALB basis set in the k-th extended element. Subsequently, they will be truncated to the central element. All integrals involving the basis sets will be performed within the extended elements to account for the influence of the adjacent chemical environment on the central element. For example, in Figure 3 (a) Q_6 will be truncated to E_6.

Subsequently, the time-dependent wavefunction can be expanded in terms of the adaptive basis set as

$$\psi_i(\mathbf{r},t) = \sum_{k=1}^{M}\sum_{j=1}^{J_b}\phi_{k,j}(\mathbf{r})C_{i,k,j}(t). \tag{3}$$

3.3 Constructing Time-Dependent DG Hamiltonian

Within the discontinuous Galerkin framework, matrix elements of the time-dependent DG Hamiltonian $H^{\text{DG}}_{k',j';k,j}(t)$ expanded in the DG-ALB basis set will be defined as following form.

$$\begin{aligned}
H^{DG}_{k',j';k,j}(t) = {} & \frac{1}{2}\delta_{k,k'}\sum_{i=1}^{M}\left(\int_{E_i}\nabla\phi^*_{k',j'}(\mathbf{r})\nabla\phi_{k,j}(\mathbf{r})\,d\mathbf{r}\right)\\
& + \frac{1}{2}\sum_{i=1}^{M}\left(\int_{E_i}\phi^*_{k',j'}(\mathbf{r})V_{\text{eff}}(\mathbf{r},t)\phi_{k,j}(\mathbf{r})\,d\mathbf{r}\right)\\
& + \sum_{l,\ell}\gamma_{l,\ell}\sum_{i=1}^{M}\left(\int_{E_i}\phi^*_{k',j'}(\mathbf{r})b_{l,\ell}(\mathbf{r})\,d\mathbf{r}\int_{E_i}b^*_{l,\ell}(\mathbf{r})\phi_{k,j}(\mathbf{r})\,d\mathbf{r}\right)\\
& + \frac{1}{2}\sum_{S\in\mathcal{S}}\int_S\langle\phi_{k',j'}(\mathbf{r})\rangle^*\langle\nabla\phi_{k,j}(\mathbf{r})\rangle\,d\mathbf{r}\\
& - \frac{1}{2}\sum_{S\in\mathcal{S}}\int_S\langle\nabla\phi_{k',j'}(\mathbf{r})\rangle^*[\phi_{k,j}(\mathbf{r})]\,d\mathbf{r}\\
& + \alpha\sum_{S\in\mathcal{S}}\int_S[\phi_{k',j'}(\mathbf{r})]^*[\phi_{k,j}(\mathbf{r})]d\mathbf{r}
\end{aligned} \tag{4}$$

Here, M denotes the total number of elements in the system, $[\cdot]$ represents the jump operator across surfaces, and $\langle\cdot\rangle$ stands for the average across surfaces. $V_{\text{eff}}(\mathbf{r},t)$ is the effective potential, which includes the time-dependent external potential field.

The fundamental mathematical properties provided by the Discontinuous Galerkin method are as equation 4: the first and second terms in the Hamiltonian contribute only to the diagonal blocks. The third term contributes to both the diagonal blocks and the blocks adjacent to edges and corners. The boundary integral contributions, appearing in the fourth through sixth terms, affect only the blocks adjacent to the edges.

Since the adjacency relationships between elements are determined when the block pattern is established, we can construct a sparsity pattern that depends solely on the block structure.

As shown in Figure 3 (b), the time-dependent Hamiltonian constructed using the DG-TDDFT method has a regular block-diagonal sparse matrix structure. Each row block contains several fully dense red blocks, whose number is determined by the dimensionality of the domain partitioning: 3 for 1D, 9 for 2D, and 27 for 3D, corresponding to 3^{Dim} nearest-neighbor interactions. This number remains constant regardless of the overall system size. All dense block indices are fixed by the DG discretization.

By applying Equations 2, 3 and 4, we can expand the TDKS equation in Equation 1 within the adaptive local basis set.

$$i\partial_t C_{i,k',j'}(\mathbf{t}) = \sum_{k,j}H^{\text{DG}}_{k',j';k,j}(\mathbf{t})C_{i,k,j}(\mathbf{t}) \tag{5}$$

3.4 Constructing Exact Hartree-Fock Exchange Matrix

3.4.1 Linear scaling of constructing the HFX matrix. For hybrid-functional TDDFT, due to the large scale of electron repulsion integrals (ERIs), traditional methods need to compute them on the fly at each time step. We developed a low-memory, reusable alternative strategy based on tensor compression. In our implementation, the short-range HSE06 exchange involves a multi-center integral of the following form:

$$(\mu\nu|\lambda\sigma) = \int\int\phi_\mu(\mathbf{r})\phi_\nu(\mathbf{r})\frac{\text{erfc}(\omega|\mathbf{r}-\mathbf{r}'|)}{|\mathbf{r}-\mathbf{r}'|}\phi_\lambda(\mathbf{r}')\phi_\sigma(\mathbf{r}')\,d\mathbf{r}\,d\mathbf{r}' \tag{6}$$

Formally, such integrals contain $(MN_{\text{ALB}})^2\times(MN_{\text{ALB}})^2$ terms, where M is the total number of elements and N_{ALB} is the number of ALBs per element. The complexity is $O(M^4)$, as the number of ALBs per element is constant. However, due to the special locality of the ALBs, only the orbital product pairs within the same element are non-zero. As a result, the integrals can be simplified to the form $(\phi_{I,\mu}\phi_{I,\lambda}\mid\phi_{J,\nu}\phi_{J,\sigma})$, which contains $MN^2_{\text{ALB}}\times MN^2_{\text{ALB}}$ terms. At this point, the complexity is reduced to $O(M^2)$.

Additionally, we can further exploit the short-range nature of the HSE06 Coulomb kernel, $\frac{\text{erfc}(\omega|\mathbf{r}-\mathbf{r}'|)}{|\mathbf{r}-\mathbf{r}'|}$, which becomes nearly zero outside the boundary of a sufficiently large extended element. In our tests, we found that an element size of 8 bohr is sufficient to achieve good accuracy. As a result, this truncation imposes an additional constraint on the ERIs $(\phi_{I,\mu}\phi_{I,\lambda}\mid\phi_{J,\nu}\phi_{J,\sigma})$, namely that the index J in $\phi_{J,\nu}\phi_{J,\sigma}$ must lie within the extended element associated with I. In one, two, and three dimensions, the number of such extended elements is limited to Q = 3, 9, and 27, respectively. Consequently,

the total number of terms scales as $MN_{\text{ALB}}^2 \times QN_{\text{ALB}}^2$, resulting in a linearly scaling method with overall complexity $O(M)$. As shown in the upper part of the "Low-Scaling ERIs compression" section of Figure 4.

3.4.2 Interpolation separable density fitting algorithm. Although the evaluation of these integrals has now become a linear-scaling problem, computing this tensor at each TDDFT time propagation step still incurs significant communication overhead. In practice, however, the ERIs of the basis functions are fixed and do not need to be recomputed. The remaining challenge lies in the large size of the tensor of ALB orbital product of shape $N_{\text{ALB}}^2 \times N_r$, which poses a significant burden on limited memory resources.

Fortunately, a series of studies have shown that the orbital product pairs in this type of multicenter integrals typically exhibit low-rank structure in the resulting tensors. Among the N_{ALB}^2 vectors, many are linearly dependent and can be efficiently approximated using low-rank decomposition[30][17][34]. To reduce the computational cost, we further apply the grid-based ISDF algorithm[35] to decompose the ALB-pair tensor in each element: $\phi_{I,\mu}(\mathbf{r})\phi_{I,\lambda}(\mathbf{r}) \approx \sum_p^{N_p} \zeta_{I,p}(\mathbf{r})\phi_{I,\mu}(\hat{r}_p)\phi_{I,\lambda}(\hat{r}_p)$. As shown in the lower part of the "Low-Scaling ERIs compression" section of Figure 4. Where N_p is the total number of interpolation vectors, given by $N_p = N_u N_{\text{ALB}} \approx 6N_{\text{ALB}}-12N_{\text{ALB}}$, and N_u is the truncated rank parameter. In our tests, we found that when N_u is chosen between 6 and 12, the resulting approximation already achieves good accuracy. Given that the number of ALBs per element typically ranges from 60 to 120 ALBs, the tensor can be compressed to approximately one-tenth of its original size. $\{\hat{r}_p\}$ denotes a set of interpolation points (IPs) obtained by using the K-means algorithm, and $\{\zeta_{I,p}(\mathbf{r})\}$ is the interpolation vectors (IVs) determined by least-square fitting.

Finally, the HFX matrix can be obtained using the following matrix expression: $H_{IJ}^{\text{HFX}} \approx -\frac{1}{2}X_I^T\left[(X_I D_{IJ} X_J^T) \circ M_{IJ}\right]X_J$. It is easy to see that the integrals over the entire basis set are transformed into a small sized auxiliary Coulomb matrix M_{IJ} and reduced ALB $X_I = \{\phi_{I,\mu}(\hat{r}_p)\}$. In the entire TDDFT calculation, we only need to repeatedly compute the reduced density matrix D_{IJ}, and then apply matrix multiplication on both sides. The cost of calculating the reduced density matrix is $O(N)$ complexity and the same point-to-point communication as the wavefunction multiplication. This allows us to achieve hybrid functional TDDFT with the computational cost of GGA-level calculations.

3.5 Solving TDKS Equations and System Responses

rt-TDDFT primarily simulates the system's response to various forms of external fields. In our subsequent tests, we will use an ultrafast Gaussian laser pulse as an example. This type of laser field can be used to study various interesting phenomena such as electronic spectra and ion collisions.

The most commonly used response quantities of the system under an external field are the time-dependent dipole moment, which Fourier transform directly yields the electronic spectrum of the system. It can be shown that in the DG-ALB basis set, the dipole moment of the system has the following form:

$$\begin{aligned} D(t) &= \sum_{i=1}^{M}\int_{E_i}\mathbf{r}\rho_i(\mathbf{r},t)d\mathbf{r} \\ &= \sum_{i=1}^{M}\int_{E_i}\mathbf{r}\sum_{k=1}^{M}\sum_{l=1}^{J_b}\sum_{k'=1}^{M}\sum_{l'=1}^{J_b}\phi_{k,l}(\mathbf{r})\phi_{k',l'}(\mathbf{r})P_{k,l;k',l'}(t)d\mathbf{r} \end{aligned} \quad (7)$$

$P_{k,l;k',l'}(t)$ is time-dependent reduced density matrix over the ALBs.

Use all of the above formulas, we provide the computational workflow of Discontinuous-Galerkin-TDKS-RK4, detailed in Algorithm 1.

Algorithm 1 One time-evolution step for Discontinuous-Galerkin-TDKS-RK4 method

Input: t, Δt and $C_{i,k,j}(t)^{(0)}$.

Output: $C_{i,k',j'}(t+\Delta t)^{(0)}$, $D(t)$, $E(t)$

1: Calculate $\rho_t^{(0)}$ from $C_{i,k,j}(t)^{(0)}$, and compute the external potential at time t, constructing $H_t^{(0)}$.
2: $k_1 = -i\Delta t H_t^{(0)} C_{i,k,j}(t)^{(0)}, \quad C_{i,k,j}(t)^{(1)} = C_{i,k,j}(t)^{(0)} + \frac{1}{2}k_1$
3: Calculate $\rho_t^{(1)}$ from $C_{i,k,j}(t)^{(1)}$, and compute the external potential at time $t+\frac{1}{2}\Delta t$, constructing $H_t^{(1)}$.
4: $H_t^{(1)} = H\left(t+\frac{1}{2}\Delta t, \rho_t^{(1)}\right)$
5: $k_2 = -i\Delta t H_t^{(1)} C_{i,k,j}(t)^{(1)}, \quad C_{i,k,j}(t)^{(2)} = C_{i,k,j}(t)^{(0)} + \frac{1}{2}k_2$
6: Calculate $\rho_t^{(2)}$ from $C_{i,k,j}(t)^{(2)}$, constructing $H_t^{(2)}$.
7: $H_t^{(2)} = H\left(t+\frac{1}{2}\Delta t, \rho_t^{(2)}\right)$
8: $k_3 = -i\Delta t H_t^{(2)} C_{i,k,j}(t)^{(2)}, \quad C_{i,k,j}(t)^{(3)} = C_{i,k,j}(t)^{(0)} + k_3$
9: Calculate $\rho_t^{(3)}$ from $C_{i,k,j}(t)^{(3)}$, and compute the external potential at time $t+\Delta t$, constructing $H_t^{(3)}$.
10: $H_t^{(3)} = H\left(t+\Delta t, \rho_t^{(3)}\right)$
11: $k_4 = -i\Delta t H_t^{(3)} C_{i,k,j}(t)^{(3)}$
12: $C_{i,k,j}(t+\Delta t)^{(0)} = C_{i,k,j}(t)^{(0)} + \frac{1}{6}(k_1+2k_2+2k_3+k_4)$
13: Calculate dipole $D(t)$ moments using Equations 7.

3.6 System Optimization

The DG-TDDFT program is developed based on the DGDFT framework [27], with extensions to support hybrid functionals and excited-state dynamics. It is implemented in C++ and employs MPI for all inter-process communication, with the global communicator divided into row and column communicators. In general, intra-element column parallelization is used to handle local computations within an individual element or its extended region, while inter-element row parallelization is responsible for data exchange between neighboring elements.

3.6.1 Low Communication Distributed Matrix Multiplication. As shown in the "Fixed neighbor communication" part of Figure 4, the time-dependent Hamiltonian constructed with the DG-ALB basis exhibits a special, regular sparse structure that allows matrix multiplication to be performed with only local communication. This

optimization fully leverages the intrinsic properties of the DG-ALB basis, which are not available in traditional basis functions.

The rows and columns of the Hamiltonian are both of size $M \times N_{ALB}$. Due to the adjacency property of the integrals, the $N_{ALB} \times N_{ALB}$ block matrix formed by elements M_i and M_j is non-zero only when j is an adjacent element of i. For example, in the case of a two-dimensional block structure, each row block contains only 9 non-zero blocks. All the non-zero matrix blocks of the M_i block are stored on all the processes corresponding to the M_i element.

Furthermore, due to the localized nature of the ALB, the multiplication of ALBs between different elements is zero. Therefore, the wave function has a completely block-structured form, and the coefficients of the total wave function within each element can be independently stored on the process corresponding to the element.

For such matrix properties and storage structures, when calculating the multiplication of the time-dependent Hamiltonian matrix and the wavefunction matrix, for the multiplication of each row block, we only need to communicate with the processes corresponding to the non-zero blocks of the Hamiltonian. Clearly, this data is stored on several adjacent processes. Therefore, the global communication required for dense matrix multiplication is transformed into local communication on adjacent processes. Furthermore, this structure allows for easy computation-communication overlap. We will use non-blocking ISend/IRecv to sequentially fetch data from each adjacent process and perform local dense block Gemm calculations, further reducing communication costs.

3.6.2 Gemm-Reduce Fusion Operation. In Algorithm 1, the computation of the electron density $\rho_t^{(i)}$ constitutes a computational hotspot at each time propagation step. This density is used to construct the updated Hamiltonian $H_t^{(i)}$, as well as to evaluate the dipole moment in equation 7 at each step.

The projection of wavefunctions onto real-space grids—required for non-grid-based basis sets because the electronic density is defined on grid points—has become the primary computational hotspot in our implementation, accounting for over 40% of the total runtime due to the massive number of grid points and orbitals. Similar projection operations also arise in traditional basis sets, such as atomic orbitals. By leveraging the same underlying principles, the fused operation optimization strategy we propose holds promise for extension to these conventional approaches as well.

This projection step„ which we refer to as a Gemm-Reduce operation. The operation computes the row-wise sum of squared magnitudes of the complex matrix product A multiplied by B, where A has dimensions M by K and B has dimensions K by N. The result is a column vector of size M by 1. The dimensions M and N range from 10,000 to several hundred thousand, while the inner dimension K is only in the tens to hundreds. As this operation generates a large intermediate matrix, it is highly memory-bound; standard implementation on the New Sunway supercomputer generally yield less than 10% floating-point utilization. In Fugaku, HBM provides a notable advantage in memory bandwidth over the New Sunway (1024 GB/s vs. 307.2 GB/s). This results in slightly improved performance. However, without the application of fused operations, the floating-point efficiency of standard implementation still reaches only around 20%.

However, in our case, the full result matrix is unnecessary-we only require the column vector obtained through row-wise reduction. Exploiting this property, we implemented a fused Gemm-Reduce operation library on New Sunway (Figure 4 Gemm-Reduce fusion operation), which eliminates the costly read and write operations associated with the full result matrix and effectively overcomes the memory-bound bottleneck. As a result, the floating point utilization of the projection step increases from below 10% to 82%, significantly improving performance on memory bandwidth constrained architectures. We tested a wide range of matrix dimensions. For various narrow-strip matrices, our fused operation consistently achieved substantial speedup over the standard implementation. The performance results for matrices corresponding to the problem sizes used in the weak scaling tests are summarized in Table 2. On the New Sunway system, a hand-optimized assembly kernel is used to replace the default zGemm kernel. All performance values are measured on a single CG and reported in GFLOP/s. The theoretical peak performance of a single CG on the SW26010-Pro is 2300 GFLOP/s. On the LS-Pilot system (the system configuration is described in Section 4.1), the default zGemm kernel is used for both baseline and optimized results, and all performance values are measured on a single NUMA domain with 36 compute cores with a theoretical peak performance of 8100 GFLOPS, also reported in GFLOP/s. The higher observed speedup on the LS pilot system is primarily attributed to the lack of optimization for narrow-strip matrix multiplication in its default zGemm library, which leads to a lower baseline performance.

Implementation details: Both the ALB matrix and the coefficient matrix are partitioned into fixed-size blocks that fit within each slave core's Local Data Memory (LDM) in a Compute Processing Element (CPE), with each slave core independently performing the multiplication of its assigned matrix block. We hand-wrote custom complex matrix multiplication kernels in assembly for each slave core, replacing the standard `zGemm` kernels that operate across the entire CPE array. During each multiplication, blocks of matrix A are split into real and imaginary parts using the shuffle instruction, while matrix B remains unchanged. The resulting matrix C also stores real and imaginary parts separately. Since only the row-wise sum of squares is required, the final shuffle can be omitted, and the reductions are performed directly on the real and imaginary components. The optimized `zGemm` kernel achieves a floating point utilization of 92.3%. This assembly kernel is specifically optimized for the Sunway architecture and was adopted in place of the native kernel due to its 20%–30% performance improvement. It is worth noting, however, that the core objective of the fusion strategy—reducing memory access complexity—is architecture-agnostic. For simplicity or portability, the assembly kernel can be replaced with the native zGemm kernel without compromising the effectiveness of the fusion scheme. As demonstrated by the LS-Pilot system results in Table 2, even without a custom kernel, the fusion operation alone yields significant performance gains. Although the optimized floating point efficiency on LS Pilot (31.4%–58.2%) is lower than that on Sunway (76.6%–84.4%), it still represents a remarkable 13.6x–20.8x improvement over the baseline, which was below 5%.

The reduced results are directly accumulated within the LDM. Matrix A remains fixed while matrix B traverses the row blocks of

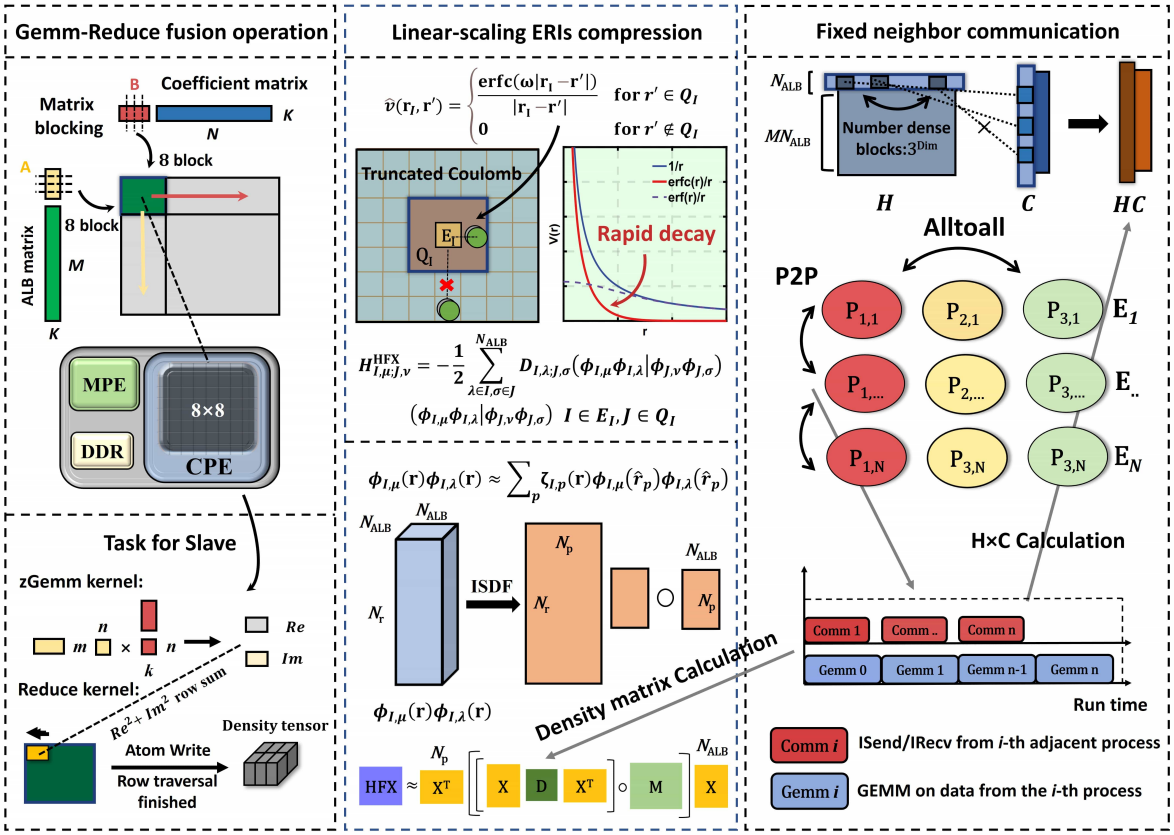

Figure 4: Schematic diagram of fusion operations, low-scaling HFX construction, ISDF data compression, and neighbor communication optimization techniques.

the coefficient matrix (as indicated by the red arrow). Intermediate reduction results are stored locally in each slave core. Once all row blocks have been processed, each slave core performs a single atomic write to the global density tensor. Matrix A is then advanced (yellow arrow) to load the next ALB matrix block. This process is repeated until the full density is computed. By profiling the DMA functions in the fusion operation, we obtained the memory bandwidth for this hotspot section, approximately 41GB/s, and the GFLOPS range is 1762.1–1942.3 GFLOPS (Table 2). The assembly microkernel can achieve a peak of 2000 GFLOPS, but the LDM (local data memory) size constraints and the lack of latency hiding for inter-thread communication should introduce some overhead. After applying the Gemm-Reduce fusion, the bottleneck shifts from memory-bound to compute-bound.

3.6.3 Customized libraries. Another major computational hotspot is the construction of the Hamiltonian matrix, particularly for HSE06, which involves extensive global Fourier transforms and typically accounts for around 50% of the total runtime. It is therefore necessary to employ specialized libraries to reduce communication overhead and mitigate the loss in floating-point efficiency. The time-dependent electron density in DG-TDDFT can be reduced to a real-space block format without communication overhead by the time-dependent wavefunction within each element. In the ground-state treatment, data is typically rearranged into a Slab format for

Table 2: Gemm refers to the floating-point performance of the standard BLAS zGemm. To handle the large intermediate matrix in the separate implementation, matrix B needs to be partitoned into column blocks. We use 1024 columns per block, as larger sizes show no clear speedup. Total includes the combined performance of separate zGemm and Reduce operations. Fuse-operator represents the performance of our fused implementation. Speedup indicates the performance gain over the standard approach. All floating-point performance values are reported in GFLOPS/s.

New Sunway						
M	*N*	*K*	zGemm	Total	Fuse-operator	Speed-up
65536	32768	64	171.4	130.2	1762.1	13.5x
196608	36864	64	163.8	118.3	1763.2	14.9x
65536	32768	128	290.4	229.2	1888.1	8.2x
196608	36864	128	280.7	211.2	1889.3	8.9x
65536	32768	256	441.4	366.7	1942.1	5.3x
196608	36864	256	430.0	343.4	1942.3	5.7x
LS Pilot System						
M	*N*	*K*	zGemm	Total	Fuse-operator	Speed-up
65536	32768	64	125.0	122.3	2542.0	20.8x
196608	36864	64	122.8	118.6	2703.3	22.8x
65536	32768	128	200.5	195.9	3713.4	19.0x
196608	36864	128	200.8	195.2	3786.8	19.4x
65536	32768	256	332.4	321.7	4625.0	14.4x
196608	36864	256	357.3	347.7	4717.6	13.6x

processing using fftw/swfftw. For excited states, due to more frequent global construction (LDA functional requires 4 global forward and inverse Fourier transforms per time step, while HSE functional requires 20 forward transforms and 28 inverse transforms.), we aim to minimize data movement costs. The MFFT library[40] is specifically adapted to the New Sunway's Pencil-blocked FFT library, optimized through a series of memory optimizations and message combinations to fit the New Sunway's communication structure. On the New Sunway, we use MFFT to avoid the overhead caused by data rearrangement, while heffte[2] are used as alternatives on the Fugaku supercomputer.

Moreover, we employ the SWAF library to replace multiple looped Gemv operations in grid-based potential interpolation with two tensor transpositions and a single Gemm, significantly improving performance.

4 HOW PERFORMANCE WAS MEASURED

All large-scale computations are conducted on two different supercomputers: New Sunway and Fugaku. Additional tests on the general applicability of the fusion operation are carried out on the LS Pilot system.

4.1 Setup of Testing Environment

New Sunway features 107,520 compute nodes and a theoretical peak performance of 1.5 EFLOPS. Each compute node includes a sw26010 pro chip along with 96 GB of memory, partitioned across 6 core groups (CGs), each with 16 GB of memory. The core groups are organized with one management processing element (MPE) and 64 computing processing elements (CPEs), arranged in an 8 × 8 grid within the Cmesh network.

Fugaku is an ARM many-core supercomputer with a theoretical peak performance of 537.21 PFLOPS. It consists of 158,976 computing nodes interconnected through a 6D-torus network. Each node is equipped with an A64FX ARM CPU, which has four core memory groups (CMGs). Each CMG contains 13 cores (1 for OS and 12 for compute) and 8 GB of HBM2 memory (32 GB per node). The A64FX supports 512-bit SVE vector instructions, achieving a peak performance of 3.38 TFLOPS at 2.2 GHz, with a memory bandwidth of 1024 GB/s. Fugaku is currently running in Boost Eco mode with one pipeline disabled, and the peak floating point performance per CPU is 1.6896 TFLOPS (base on official confirmation).

Each node of the LS pilot system is equipped with two high-performance LX2 processors, providing a total of over 256 cores. The LX2 is a system-on-chip (SoC) that integrates two computing dies within a single package. Each die is equipped with 128 GB of off-die DDR memory and divided into four NUMA domains. To enhance data transfer efficiency between DDR and the on-package high-bandwidth memory, a System Direct Memory Access (SDMA) interface is implemented within each CPU die. The two dies are interconnected via the LXLink network, with each network card offering a bidirectional bandwidth of 48 GB/s. Each core supports Scalable Vector and Matrix Extension (SVME), enabling double-precision SIMD instructions and in-pipeline 8×8 matrix operations. The cores operate at a frequency of no less than 1.3 GHz.

4.2 Measurement

The executed floating-point measurements were obtained using the performance analysis library functions penv_host1_float_muletc_count and penv_slave_fd_float_sum_count on the New Sunway system, while the computation time was measured using MPI_Wtime.

- **Time-to-solution**, defined as the per time-evolution step time, which includes all operations within one time-evolution step (including I/O operations).
- **Peak performance**, defined as "total FLOPs per time-evolution step / time-to-solution".

Setup time, such as the initialization of MPI, the system, and the ground-state wavefunction, is not included. In the New Sunway system, we use the nominal 2.3 TFLOPS per CG as the theoretical peak performance. For Fugaku, we take 1.6896 TFLOPS/CPU (Boost Eco mode) as the theoretical peak performance. The total FLOPS count is approximated by referencing the same case on New Sunway. The ground-state inputs of the program are provided by DGDFT[16]. All numerical tests employed double-precision floating-point operations.

4.3 Benchmark and Numerical Accuracy

We conducted an accuracy comparison between our code and the PWDFT software, which implements the rt-TDDFT module using the plane-wave method. The rt-TDDFT functionality in PWDFT has been previously benchmarked against other TDDFT software[21, 22], such as Quantum ESPRESSO[11] software package, in numerous studies. The results of this comparison are shown in Figure 5.

To systematically evaluate the accuracy of the DG-rtTDDFT method, we selected representative examples from molecular, semiconductor, and metallic systems for comparison with the PW-rtTDDFT method. The chosen systems include a benzene molecule contained within a $19.99 \times 19.99 \times 19.99$ Bohr simulation box, a Si_8 unit cell with lattice constants of $10.26 \times 10.26 \times 10.26$ Bohr, and a Na_4Mg_{12} unit cell with lattice constants of $20.23 \times 20.23 \times 10.12$ Bohr. The type of pseudopotential used is SG15 Optimized Norm-Conserving Vanderbilt (ONCV), the exchange correlation functional is the Local Density Approximation (LDA) and Heyd–Scuseria–Ernzerhof 2006 hybrid functional (HSE06). Since hybrid functionals involve the use of the ISDF method, we additionally tested the impact of different rank parameters Nu on the accuracy of DG-TDDFT hybrid functional calculations, and plotted the corresponding absorption spectra for a simulation time of 120 fs.

The comparison results are shown in Figure 5, where it is evident that the results obtained using our method are virtually indistinguishable from those produced by the PW-rtTDDFT method, indicating that both approaches achieve comparable simulation accuracy.

4.4 Large Systems Performance Results

4.4.1 Large-scale system selection. For TDDFT, a well-known application is the study of the spectra of clusters or nanoscale biological structures. A recent TDDFT study of Ag clusters mentioned that there are currently no suitable programs capable of using hybrid functionals for TDDFT to study the spectra of Ag_{923}

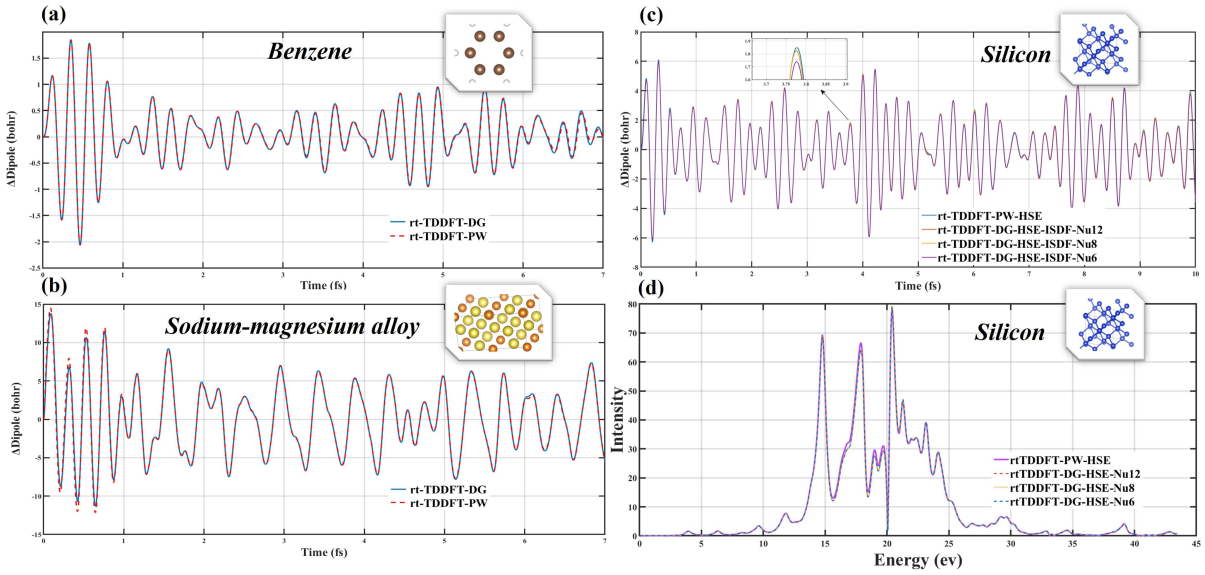

Figure 5: Figures (a), (b), and (c) show the time evolution of the dipole moment for three different materials under Gaussian laser with an electric field peak of 1 eV/Å, a frequency of 0.6615 au, and a laser width of 13.6 au, central time of the laser pulse is set to 0.11 au, calculated using DG-ALB and plane waves basis sets. (a): Benzene molecule dipole using the LDA functional. (b): Na_4Mg_{12} alloy system dipole using the LDA functional. (c): Si_8 system dipole using the HSE06 functional for different ISDF rank parameters. (d): Electronic absorption spectrum of Si_8 using the HSE06 for different ISDF rank parameters (120fs).

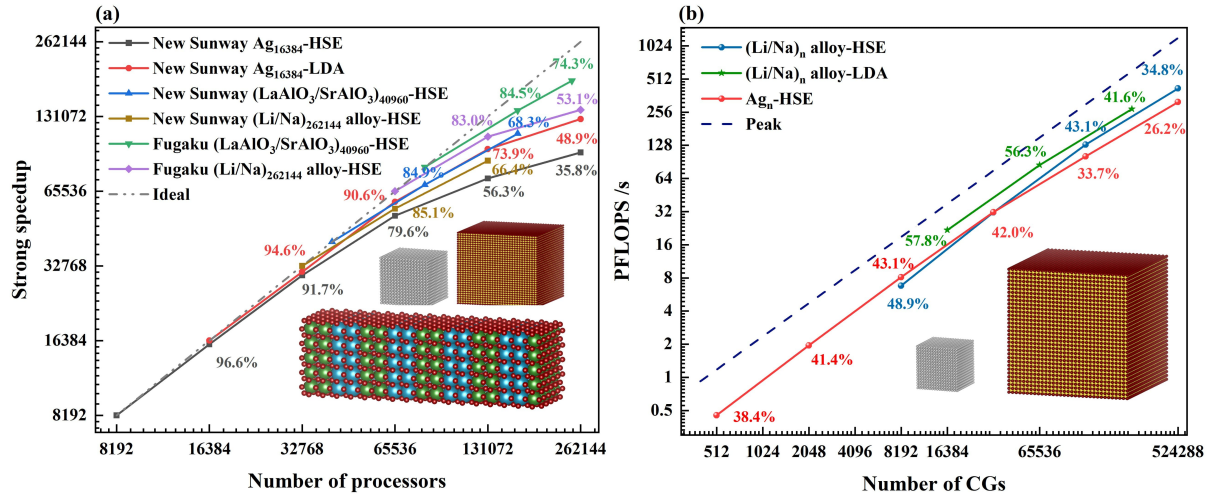

Figure 6: (a) Strong scaling of TDDFT on New Sunway and Fugaku with 8,192 to 262,144 processors, with parallel efficiency benchmarked against the smallest processor count. (b): Weak scaling of three different materials on New Sunway, with the marked points indicating the achieved peak flops rate. The LiNa tests (LDA and HSE) were scaled from 262,144 to 1,048,576 atoms (1,048,576 electrons), while the Ag system was scaled from 1,024 to 32,768 atoms (622,592 electrons).

nanostructures[3]. Considering practical applications, we specifically chose structures ranging from $(Ag)_{1024}$ to $(Ag)_{32768}$ as an example for large-scale testing. Additionally, the free electron gas in metal systems can easily give rise to non-local excitations, which are difficult to simulate using fragment-based methods. Therefore, we selected the $(Li/Na)_{262144}$-$(Li/Na)_{1048576}$ alloy systems as the second large-scale test example. The last example corresponds to a superconducting heterojunction, $LaAlO_3$-$SrTiO_3$, with a two-dimensional electron gas[33]. For such systems, laser control of electronic transport is a common technique. We selected $(LaAlO_3$-$SrTiO_3)_{40960}$ as the third test system. In all test cases involving HSE06, the ISDF rank parameter Nu was uniformly set to 12. The number of processors scales quadratically with the system size in weak scaling test, consistent with the $O(N^2)$ complexity of our algorithm.

4.4.2 Strong Scaling. Figure 6 (a) shows the strong scaling of the $(Ag)_{16384}$ system. We used a cutoff energy of 60 Ha and employed 8 ALB basis functions per orbital for DG-TDDFT calculations with

the HSE functional. Due to the global Fourier transform involved in constructing the Hamiltonian, the strong scalability of the entire system decreases as the number of processor increases. However, even with half of the machine, the simulation using the HSE functional for a $(Ag)_{16384}$ system with 622,592 electrons still maintains 35.8% scalability and achieves 29.9% floating-point efficiency, reaching a breakthrough single-step time of just 4.84 seconds. As shown in the strong scaling results, New Sunway and Fugaku exhibit consistent scaling trends, which demonstrates the generality of our method across architectures. Sunway achieves lower parallel efficiency than Fugaku under identical test cases, primarily due to the higher theoretical peak performance of each CG compared to Fugaku's CMG (2.3 TFLOPS vs. 0.42 TFLOPS), which amplifies the effects of load imbalance and communication overhead. Nevertheless, owing to the significant reduction in memory access cost enabled by the fusion strategy, Sunway consistently delivers higher peak performance—e.g., $(Li/Na)_{262144}$: 27.8% vs. 17.1%, $(LaAlO_3\text{-}SrTiO_3)_{40960}$: 41.1% vs. 32.8%—despite relying on DDR4 memory with lower bandwidth than Fugaku's HBM.

4.4.3 Weak Scaling. Figure 6 (b) presents the weak scaling performance of LiNa alloy and silver systems using LDA and HSE hybrid functionals on the new Sunway supercomputer. For the $(Ag)_n$ systems, a 60 Hartree cutoff and 9 ALB basis functions per orbital were used; for $(Li/Na)_n$, a 20 Hartree cutoff and 8 ALBs per orbital were used.

In LDA simulations, the LiNa system was scaled from 16,384 to 262,144 CGs, with atom counts from 262,144 to 1,048,576. For HSE, LiNa scaled from 32,768 to 524,288 CGs with the same system sizes, while the Ag system scaled from 512 to 524,288 CGs, corresponding to 1,024 to 32,768 atoms.

The observed rise-then-fall trend in floating-point performance is primarily due to the limited parallelism in FFT-based gradient calculations. In HSE functionals, three independent Fourier transforms are performed along each spatial direction, which can be parallelized up to three processes per element. Beyond this point, no further intra-element parallelism is gained, and global communication associated with FFTs becomes the dominant source of floating-point efficiency loss. In contrast, LDA requires fewer FFTs, leading to better scalability.

At the largest scale—524,288 CGs and 1,048,576 atoms—the HSE simulation of the LiNa system achieved 34.8% floating-point efficiency and 419.9 PFLOP/s, which is 3,817x higher than the previously reported largest hybrid-functional TDDFT simulation[22] (Table 1).

5 IMPLICATIONS

5.1 Scientific Applications

The heterojunction formed by the insulating materials $LaAlO_3$ and $SrTiO_3$ exhibits spontaneous and piezoelectric polarization phenomena on its surface. At the interface, residual electrons emerge, leading to the formation of a two-dimensional superconducting electron gas[33]. Using the DG-TDDFT method, we have conducted a first-principles investigation into the electronic dynamics of this system. Figure 7 presents their ground-state electronic density as well as the electronic transport phenomena under the control of a Gaussian laser field. These simulation scheme for the electronic dynamics of complex systems under external field control give a new physical insight into fundamental properties of the complex metallic heterojunctions for designing high-performance next-generation electronic devices.

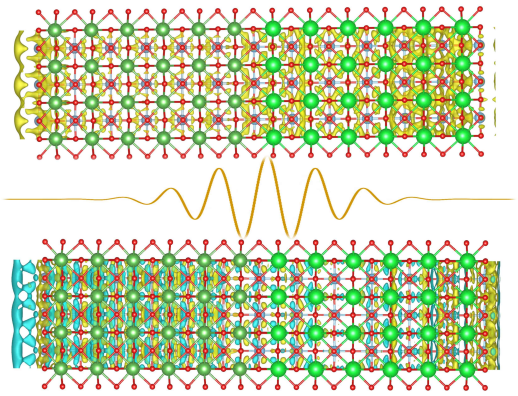

Figure 7: Top: Yellow shows the ground-state electron density isosurface. Bottom: Blue and yellow indicate regions of decreased and increased electron density in the laser-driven system, respectively.

5.2 Conclusion and Outlook

Together with density functional theory and molecular dynamics, *ab initio* electron dynamics completes the triad of essential components in first-principles simulations. For the first time, we have achieved *ab initio* electron dynamics simulations at the million-atom scale. This work delivers breakthroughs in both accuracy and scale. In systems with up to 10^5 electrons, we perform hybrid-functional-accuracy simulations with a single-step time of just a few seconds—something previously unattainable. Our work redefines the boundaries of what is computationally feasible in electron dynamics. Problems such as mesoscopic-scale photochemical reactions, external-potential-driven phenomena from low-concentration doping, and excited-state spectra of large biomolecular systems will be reconsidered in terms of their simulability.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (2016YFA0200604, 2021YFB0300600), the Strategic Priority Research Program of the Chinese Academy of Sciences (XDB1170000, XDB0450101, XDB0500102), the Innovation Program for Quantum Science and Technology (2021ZD0303306), the National Natural Science Foundation of China (22288201, 22173093, 22373096, 21688102), the CAS Project for Young Scientists in Basic Research (YSBR-005), and the Laoshan Laboratory (LSKJ202300305). We sincerely thank Professor Lijun Liu (Osaka University) for providing access to the Fugaku platform, and Professor Yutong Lu (Sun Yat-sen University) for supporting our use of the LS-Pilot platform. The authors thank the Laoshan Laboratory, AI-Scientist platform of Chinese Academy of Science, and the Supercomputing Center of USTC for the computational resources. We gratefully acknowledge ISCAS for providing MFFT support.

References

[1] J Alberdi-Rodriguez, X Andrade, A Arruabarrena, J Muguerza, and A Rubio. 2011. Improving Octopus towards the new generation of HPC systems. In *Jülich Blue Gene/P Extreme Scaling Workshop*. 3–6.

[2] Alan Ayala, Stanimire Tomov, Azzam Haidar, and Jack Dongarra. 2020. heffte: Highly efficient fft for exascale. In *International Conference on Computational Science*. Springer, 262–275.

[3] Mohit Chaudhary and Hans-Christian Weissker. 2024. Optical spectra of silver clusters and nanoparticles from 4 to 923 atoms from the TDDFT + U method. *Nat. Commun.* 15, 1 (2024), 9225.

[4] Feixiang Chen, Xueli Wang, Hong-Jun Zhuang, Mengnan Guo, Luo Wang, Junze Xie, Le Zhang, Hao Liu, Yuhan Shi, Jiajia Zhou, Xinjie Mao, Muyao Lv, Xingwu Jiang, Jinquan Chen, Yanyan Liu, Da-Yong Jin, and Wenbo Bu. 2024. Ultra-fast photoelectron transfer in bimetallic porphyrin optoelectrode for single neuron modulation. *Nat. Commun.* 15 (11 2024). https://doi.org/10.1038/s41467-024-54325-8

[5] Min Choi, Mahmut Sait Okyay, Adrian Perez Dieguez, Mauro Del Ben, Khaled Z Ibrahim, and Bryan M Wong. 2024. QRCODE: Massively Parallelized Real-Time Time-Dependent Density Functional Theory for Periodic Systems. *Comput. Phys. Commun.* (2024), 109349.

[6] Bernardo Cockburn, George E Karniadakis, and Chi-Wang Shu. 2000. The development of discontinuous Galerkin methods. In *Discontinuous Galerkin methods: theory, computation and applications*. Springer, 3–50.

[7] Sambit Das, Bikash Kanungo, Vishal Subramanian, Gourab Panigrahi, Phani Motamarri, David Rogers, Paul Zimmerman, and Vikram Gavini. 2023. Large-scale materials modeling at quantum accuracy: Ab initio simulations of quasicrystals and interacting extended defects in metallic alloys. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 1–12.

[8] Mauro Del Ben, Charlene Yang, Zhenglu Li, Felipe H da Jornada, Steven G Louie, and Jack Deslippe. 2020. Accelerating large-scale excited-state GW calculations on leadership HPC systems. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 1–11.

[9] J. Deslippe, G. Samsonidze, D. A. Strubbe, M. Jain, M. L. Cohen, and S. G. Louie. 2012. BerkeleyGW: A Massively Parallel Computer Package for the Calculation of the Quasiparticle and Optical Properties of Materials and Nanostructures. *Comput. Phys. Commun.* 183, 6 (2012), 1269–1289.

[10] Erik W Draeger, Xavier Andrade, John A Gunnels, Abhinav Bhatele, André Schleife, and Alfredo A Correa. 2017. Massively parallel first-principles simulation of electron dynamics in materials. *J. Parallel and Distrib. Comput.* 106 (2017), 205–214.

[11] Paolo Giannozzi, Stefano Baroni, Nicola Bonini, Matteo Calandra, Roberto Car, Carlo Cavazzoni, Davide Ceresoli, Guido L Chiarotti, Matteo Cococcioni, Ismaila Dabo, et al. 2009. QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials. *J. Phys.: Condens. Matter* 21, 39 (2009), 395502.

[12] Joscha Hekele, Yi Yao, Yosuke Kanai, Volker Blum, and Peter Kratzer. 2021. All-electron real-time and imaginary-time time-dependent density functional theory within a numeric atom-centered basis function framework. *J. Chem. Phys* 155, 15 (2021).

[13] Connor J. Herring and Matthew M. Montemore. 2023. Recent Advances in Real-Time Time-Dependent Density Functional Theory Simulations of Plasmonic Nanostructures and Plasmonic Photocatalysis. *ACS Nanosci. Au* 3, 4 (2023), 269–279. https://doi.org/10.1021/acsnanoscienceau.2c00061 arXiv:https://doi.org/10.1021/acsnanoscienceau.2c00061

[14] Yuta Hirokawa, Atsushi Yamada, Shunsuke Yamada, Masashi Noda, Mitsuharu Uemoto, Taisuke Boku, and Kazuhiro Yabana. 2022. Large-scale ab initio simulation of light–matter interaction at the atomic scale in Fugaku. *Int J High Perform C* 36, 2 (2022), 182–197.

[15] Changqing Hu and Chi-Wang Shu. 1999. A discontinuous Galerkin finite element method for Hamilton–Jacobi equations. *SIAM J. Sci. Comput.* 21, 2 (1999), 666–690.

[16] Wei Hu, Hong An, Zhuoqiang Guo, Qingcai Jiang, Xinming Qin, Junshi Chen, Weile Jia, Chao Yang, Zhaolong Luo, Jielan Li, et al. 2022. 2.5 Million-Atom Ab Initio Electronic-Structure Simulation of Complex Metallic Heterostructures with DGDFT. In *2022 SC22: International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*. IEEE Computer Society, 48–60.

[17] Wei Hu, Lin Lin, and Chao Yang. 2017. Interpolative Separable Density Fitting Decomposition for Accelerating Hybrid Density Functional Calculations with Applications to Defects in Silicon. *J. Chem. Theory Comput.* 13, 11 (2017), 5420–5431.

[18] Zhongwei Hu, Jochen Autschbach, and Lasse Jensen. 2016. Simulating Third-Order Nonlinear Optical Properties Using Damped Cubic Response Theory within Time-Dependent Density Functional Theory. *J. Chem. Theory Comput.* 12, 3 (2016), 1294–1304. https://doi.org/10.1021/acs.jctc.5b01060 arXiv:https://doi.org/10.1021/acs.jctc.5b01060 PMID: 26841327.

[19] Jing Huang, Zihao Li, Lei Yang, Rongfang Hu, and Guoyu Shi. 2025. A comprehensive DFT/TDDFT investigation into the influence of electron acceptors on the photophysical properties of ullazine-based D-π-A-π-A photosensitizers. *Sci Rep-uk* 15 (2025). https://api.semanticscholar.org/CorpusID:275901976

[20] Jacek Jakowski, Wenchang Lu, Emil Briggs, David Lingerfelt, Bobby G Sumpter, Panchapakesan Ganesh, and Jerzy Bernholc. 2025. Simulation of 24,000 Electron Dynamics: Real-Time Time-Dependent Density Functional Theory (TDDFT) with the Real-Space Multigrids (RMG). *J. Chem. Theory Comput.* (2025).

[21] Weile Jia, Dong An, Lin-Wang Wang, and Lin Lin. 2018. Fast real-time time-dependent density functional theory calculations with the parallel transport gauge. *J. Chem. Theory Comput.* 14, 11 (2018), 5645–5652.

[22] Weile Jia, Lin-Wang Wang, and Lin Lin. 2019. Parallel transport time-dependent density functional theory calculations with hybrid functional on summit. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 1–23.

[23] Qingcai Jiang, Zhenwei Cao, Xinhui Cui, Lingyun Wan, Xinming Qin, Huanqi Cao, Hong An, Junshi Chen, Jie Liu, Wei Hu, et al. 2024. Extending the limit of LR-TDDFT on two different approaches: Numerical algorithms and new Sunway heterogeneous supercomputer. *Parallel Comput.* 120 (2024), 103085.

[24] W. Kohn and L. Sham. 1965. Self-consistent equations including exchange and correlation effects. *Phys. Rev.* 140 (1965), A1133–A1138.

[25] Alina Kononov, Cheng-Wei Lee, Tatiane Pereira dos Santos, Brian Robinson, Yifan Yao, Yi Yao, Xavier Andrade, Andrew David Baczewski, Emil Constantinescu, Alfredo A Correa, et al. 2022. Electron dynamics in extended systems within real-time time-dependent density-functional theory. *MRS Commun.* 12, 6 (2022), 1002–1014.

[26] Lin Lin and Michael Lindsey. 2019. Convergence of Adaptive Compression Methods for Hartree-Fock-Like Equations. *Commun. Pur. Appl. Math.* 72, 3 (2019), 451–499.

[27] Lin Lin, Jianfeng Lu, Lexing Ying, and E Weinan. 2012. Adaptive local basis set for Kohn–Sham density functional theory in a discontinuous Galerkin framework I: Total energy calculation. *J. Comput. Phys* 231, 4 (2012), 2140–2154.

[28] Rongrong Liu, Zhuoqiang Guo, Qiuchen Sha, Tong Zhao, Haibo Li, Wei Hu, Lijun Liu, Guangming Tan, and Weile Jia. 2025. Large Scale Finite-Temperature Real-time Time Dependent Density Functional Theory Calculation with Hybrid Functional on ARM and GPU Systems. *arXiv preprint arXiv:2501.03061* (2025).

[29] Igor Lomtev and George Em Karniadakis. 1999. A discontinuous Galerkin method for the Navier–Stokes equations. *International journal for numerical methods in fluids* 29, 5 (1999), 587–603.

[30] J. Lu and L. Ying. 2015. Compression of the Electron Repulsion Integral Tensor in Tensor Hypercontraction Format with Cubic Scaling Cost. *J. Comput. Phys.* 302 (2015), 329–335.

[31] Yirui Lu, Yang Li, and Bing Huang. 2024. Recent progress on photoinduced phase transitions in materials from first-principles calculations. *Computational Materials Today* 2-3 (2024), 100012. https://doi.org/10.1016/j.commt.2024.100012

[32] Mit H Naik, Emma C Regan, Zuocheng Zhang, Yang-Hao Chan, Zhenglu Li, Danqing Wang, Yoseob Yoon, Chin Shen Ong, Wenyu Zhao, Sihan Zhao, et al. 2022. Intralayer charge-transfer moiré excitons in van der Waals superlattices. *Nature* 609, 7925 (2022), 52–57.

[33] A Ohtomo and HY Hwang. 2004. A high-mobility electron gas at the LaAlO3/SrTiO3 heterointerface. *Nature* 427, 6973 (2004), 423–426.

[34] Xinming Qin, Wei Hu, and Jinlong Yang. 2023. Interpolative separable density fitting for accelerating two-electron integrals: A theoretical perspective. *J. Chem. Theory Comput.* 19, 3 (2023), 679–693.

[35] Xinming Qin, Wei Hu, and Jinlong Yang. 2023. Interpolative separable density fitting for accelerating two-electron integrals: A theoretical perspective. *J. Chem. Theory Comput.* 19, 3 (2023), 679–693.

[36] Erich Runge and Eberhard KU Gross. 1984. Density-functional theory for time-dependent systems. *Phys. Rev. Lett* 52, 12 (1984), 997.

[37] Ryan Stocks, Jorge L Galvez Vallejo, CY Fiona, Calum Snowdon, Elise Palethorpe, Jakub Kurzak, Dmytro Bykov, and Giuseppe MJ Barca. 2024. Breaking the Million-Electron and 1 EFLOP/s Barriers: Biomolecular-Scale Ab Initio Molecular Dynamics Using MP2 Potentials. In *SC24: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 1–12.

[38] Jack Strand, Sergey K Chulkov, Matthew B Watkins, and Alexander L Shluger. 2019. First principles calculations of optical properties for oxygen vacancies in binary metal oxides. *J. Chem. Phys.* 150, 4 (2019).

[39] Jiyu Xu, Daqiang Chen, and Sheng Meng. 2022. Decoupled ultrafast electronic and structural phase transitions in photoexcited monoclinic VO_2. *Sci. Adv.* 8, 44 (2022), eadd2392. https://doi.org/10.1126/sciadv.add2392 arXiv:https://www.science.org/doi/pdf/10.1126/sciadv.add2392

[40] Yuwen Zhao, Fangfang Liu, Wenjing Ma, Huiyuan Li, Yuanchi Peng, and Cui Wang. 2023. MFFT: A GPU Accelerated Highly Efficient Mixed-Precision Large-Scale FFT Framework. *ACM T. Archit. Code Op.* 20, 3 (2023), 1–23.

Appendix: Artifact Description

A Overview of Contributions and Artifacts

A.1 Paper's Main Contributions

A.2 Computational Artifacts

A_1 DGDFT/DGTDDFT-Sunway DOI:N/A (proprietary)
A_2 DGDFT/DGTDDFT-Fugaku DOI:N/A (proprietary)

Artifact ID	Contributions Supported	Related Paper Elements
A_1	C_1, C_2, C_3, C_4	Tables 2 Figure 5-7
A_2	C_1, C_2, C_3	Figures 6

A.2.1 ARTIFACT AVAILABILITY. Software Artifact Availability: Some author-developed software artifacts are NOT maintained in a public repository or are NOT released under an OSI-approved open-source license.

Hardware Artifact Availability: No author-developed hardware artifacts are associated with this work.

Data Artifact Availability: No author-developed data artifacts are associated with this work.

Proprietary Artifacts: No author-developed artifacts are proprietary.

This work was developed based on DGDFT. An early open-source version of DGDFT from Berkeley Lab is available to help readers understand its principles, and we have included it in the Zenodo repository.

The latest version of DGDFT has been jointly developed by multiple authors from LBNL and USTC, and includes a substantial amount of unpublished work. As such, we do not have the right to release the contributions of others without consent. We kindly ask for your understanding.

If you are interested in accessing the code, please contact me at fengjw@mail.ustc.edu.cn. The code will be made available upon obtaining consent from all co-developers.

B ARTIFACT IDENTIFICATION

B.1 Computational Artifact A_1

Relation To Contributions

This artifact is the Sunway version of DGDFT/DGTDDFT. Figures 5–7 and Table 2 present results obtained using this code, including accuracy, scaling, and application tests, as well as fused computation performance.

Expected Results

C1: This code uses a novel DG-ALB basis, and the results in Figure 5 show that it achieves nearly the same accuracy as the plane-wave basis. C2: It adopts linear-scaling algorithms, achieving up to 100× speedup over plane-wave methods on systems of similar size. C3: As shown in Figure 6, our code scales efficiently to 524,288 processors on the New Sunway supercomputer. C4: In Table 2, the fused operation library in this code improves floating-point efficiency by several times to over tenfold compared to the unfused version.

Expected Reproduction Time (in Minutes)

The time for ground-state calculations varies with the system and resources, typically ranging from minutes to hours. Excited-state simulations similarly depend on these factors, but are also influenced by the time step: although each step takes only seconds, high-accuracy propagation requires many steps, leading to simulation times of several days for large systems.

Artifact Setup (incl. Inputs)

Hardware. **New Sunway:** Relevant hardware details: 6CGs (Core Groups) per node. A CG consists of 64 Compute Processing Elements (CPEs) on a 4×4 mesh, with four CPEs sharing each mesh stop.

Software. Applications and versions: DGDFT/DGTDDFT-Sunway. Libraries and versions: swgcc710-tools-SEA-1449-mpi, libxc/6.2.2, mswfft, yaml-cpp/0.5.1, swBlas/xMath, swLapack/xMath, scalapack-2.0.2, swfftw, elpa/2023.05.001, PEXSI/Sunway

Datasets / Inputs. Ag (1,024-32,768 atom), Li/Na (262,144-1,048,57 atom)

Installation and Deployment. Compilers and versions: swgcc710-tools-SEA-1449

Artifact Execution

```
source ./env.1449.sh
echo $DGDFT_DIR
rm -rf $DGDFT_DIR/external/yaml-cpp
cd $DGDFT_DIR/external/yaml-cpp-0.7.0
rm -rf build/*
cd build
cmake -DCMAKE_INSTALL_PREFIX=$DGDFT_DIR/external/yaml-cpp ..
make
make install
cd $DGDFT_DIR/external/lbfgs
make cleanall && make
cd $DGDFT_DIR/external/rqrcp
make cleanall && make
cd $DGDFT_DIR/external/blopex/blopex_abstract
make clean && make
cd $DGDFT_DIR/src
make cleanall && make -j
cd $DGDFT_DIR/examples
make cleanall && make pwdft && make dgtddft
bsub -b -J name -o runlog -exclu -share_size 15000 -host_stack 2048 -n 1024 -cgsp 64 -q queue dgtddft
```

Artifact Analysis (incl. Outputs)

The rt-tddft-pw code used for benchmarking, the benchmark data with DGTDDFT, the early open-source version of DGDFT, and

the fused operation library used in our work (available on the Sunway platform) have all been organized in the following Zenodo repository.

(1) https://doi.org/10.5281/zenodo.16933545

Starting from "RTTDDFT-Hybrid/RTTDDFT", the simulation is entirely handled by our own implementation. The ground-state input is not read from I/O files; instead, the initial wavefunctions are directly copied from memory using DGDFT. For each RK4 step, the outputs of interest are as follows:
Dx: -6.08925869337 Dy: -6.08925395339 Dz -6.08918443486
This is dipole moment data of Si_8. By performing a Fourier transform of the time-dependent dipole moment, the absorption spectrum can be obtained. Due to differences in the initial coordinate origins used in the simulations, each dipole value must be shifted by subtracting the dipole moment at 0 fs.

The timing breakdown of each component, as well as the per-step runtime, for the LiNa example output is shown below:
DensityTime: 5.60260750095 s
Times Ham vector: 0.474219603333 s
Times Update: 0.122302035241 s
Times Dipole: 0.00671734857315 s
Times Vext: 0.00856638095138 s
Times Construct Ham: 7.94920213905 s
Time for RK4 a cycle = 14.4039612314 s.

Finally, we provide the most critical fused operator library for floating-point optimization. If you have access to the Sunway environment, you can directly link the provided .a file. The interface of the routine is as follows
sw_ColMajor_zgemm_reduce_master (char trans_1, char trans_2, int M, int N, int K, std::complex<double>* A, int lda, std::complex<double>* B, int ldb, double* R);

B.2 Computational Artifact A_2

Relation To Contributions

This artifact is the Fugaku version of DGDFT/DGTDDFT. The scalability results on Fugaku in Figure 6 were obtained using this code.

Expected Results

C1 & C2: Same as A1. C3: As shown in Figure 6, the code was successfully scaled to 262,144 processors on Fugaku, reaching half-machine scale.

Expected Reproduction Time (in Minutes)

Same as A1.

Artifact Setup (incl. Inputs)

Hardware. **Fugaku:** Relevant hardware details: Each node is equipped with an A64FX ARM CPU, which has four core memory groups (CMGs). Each CMG contains 13 cores (1 for OS and 12 for compute) and 8 GB of HBM2 memory (32 GB per node).

Software. Applications and versions: DGTDDFT-Fugaku.Libraries and versions: tcsds-1.2.41-mpi, libxc/9.1.5, heffte/2.0.0, yaml-cpp/0.7.0, fftw/3.6.10, Scalapack (fugaku official LP64, SVE version: -lfjscalapacksve -SSL2 -SCALAPACK), Lapack (fugaku official LP64, SVE version version: -lfjlapackexsve), elpa/2023.05.001, PEXSI/2.0.0

Datasets / Inputs. Li/Na (262,144 atom), $LaAlO_3/SrAlO_3$ (10,240, 40,960 atom)

Installation and Deployment. Compilers and versions: compiler/tcsds-1.2.41

Artifact Execution

```
source ./env.sh
rm -rf $DGDFT_DIR/external/yaml-cpp
cd $DGDFT_DIR/external/yaml-cpp-0.7.0
rm -rf build/*
cd build
cmake -DCMAKE_INSTALL_PREFIX=$DGDFT_DIR/external/yaml-cpp ..
make
make install
echo $DGDFT_DIR
cd $DGDFT_DIR/external/lbfgs
make cleanall && make
cd $DGDFT_DIR/external/rqrcp
make cleanall && make
cd $DGDFT_DIR/external/blopex/blopex_abstract
make clean && make
cd $DGDFT_DIR/src
make cleanall && make -j
cd $DGDFT_DIR/examples
make cleanall && make pwdft && make dgtddft
pjsub submit_large.sh
```

Artifact Analysis (incl. Outputs)

Same as A1.