

# Advancing Quantum Many-Body GW Calculations on Exascale Supercomputing Platforms

Benran Zhang<sup>1</sup>, Daniel Weinberg<sup>2</sup>, Chih-En Hsu<sup>1,3</sup>, Aaron R. Altman<sup>4</sup>, Yuming Shi<sup>4</sup>, James B. White III<sup>5</sup>, Derek Vigil-Fowler<sup>6</sup>, Steven G. Louie<sup>2,7</sup>, Jack R. Deslippe<sup>2</sup>, Felipe H. da Jornada<sup>4,8</sup>, Zhenglu Li<sup>1,\*</sup>, Mauro Del Ben<sup>2,\*</sup>

<sup>1</sup>University of Southern California, USA. <sup>2</sup>Lawrence Berkeley National Laboratory, USA. <sup>3</sup>Tamkang University, Taiwan.

<sup>4</sup>Stanford University, USA. <sup>5</sup>Oak Ridge National Laboratory, USA. <sup>6</sup>National Renewable Energy Laboratory, USA.

<sup>7</sup>University of California at Berkeley, USA. <sup>8</sup>SLAC National Accelerator Laboratory, USA.

\*Correspondence to be addressed to: zhenglul@usc.edu (Z.L.), mdelben@lbl.gov (M.D.B.)

## Abstract

Advanced *ab initio* materials simulations face growing challenges as increasing systems and phenomena complexity requires higher accuracy, driving up computational demands. Quantum many-body GW methods are state-of-the-art for treating electronic excited states and couplings but often hindered due to the costly numerical complexity. Here, we present innovative implementations of advanced GW methods within the BerkeleyGW package, enabling large-scale simulations on Frontier and Aurora exascale platforms. Our approach demonstrates exceptional versatility for complex heterogeneous systems with up to 17,574 atoms, along with achieving true performance portability across GPU architectures. We demonstrate excellent strong and weak scaling to thousands of nodes, reaching double-precision core-kernel performance of 1.069 ExaFLOP/s on Frontier (9,408 nodes) and 707.52 PetaFLOP/s on Aurora (9,600 nodes), corresponding to 59.45% and 48.79% of peak, respectively. Our work demonstrates a breakthrough in utilizing exascale computing for quantum materials simulations, delivering unprecedented predictive capabilities for rational designs of future quantum technologies.

## 1 Justification for ACM Gordon Bell Prize

BerkeleyGW delivers a breakthrough in exascale computational quantum many-body materials simulations, achieving true performance portability across all leadership-class supercomputing architectures, excellent strong scaling, and high fraction of peak for core computing kernels. BerkeleyGW enables predictive quantum materials modeling with outstanding time-to-solution and double-precision throughput above 1.0 ExaFLOP/s performance.

## 2 Performance Attributes

Category of Achievement	Scalability, Time-to-solution, Peak Performance
Type of Method Used	Both Explicit and Implicit
Results Reported	Kernel Only
Precision Reported	Double Precision
System Scale	Results Measured on Full System
Measurement Mechanism	FLOP Count

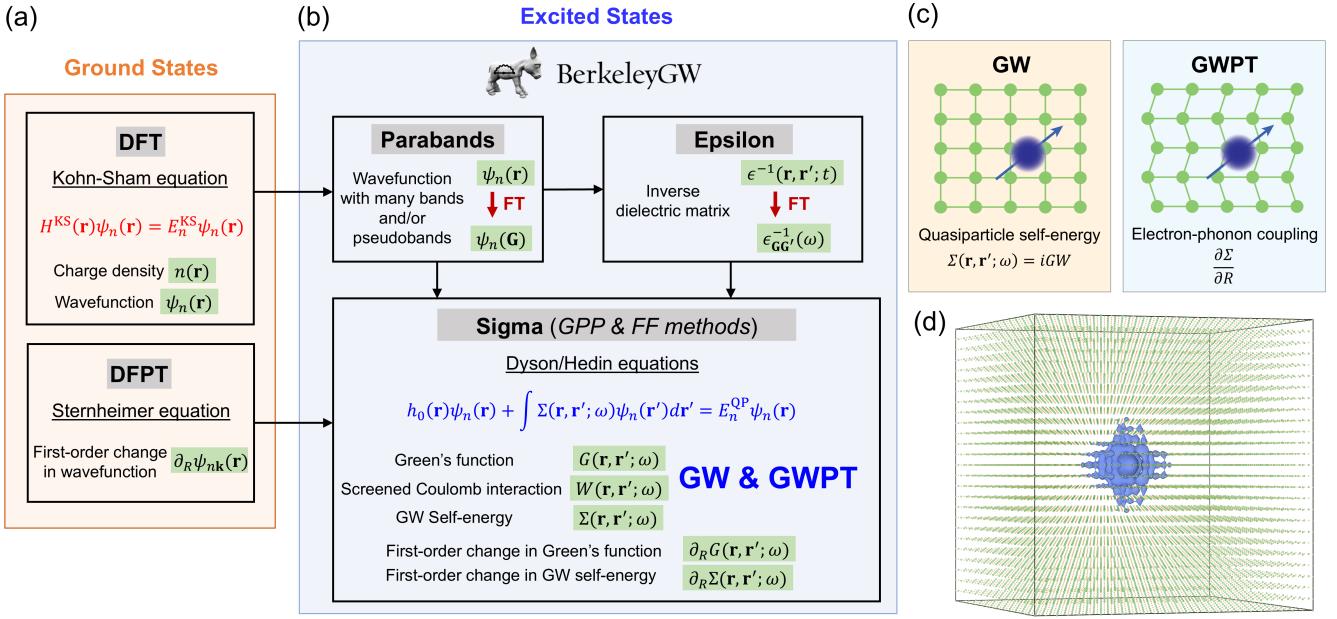
## 3 Overview of the Problem

Quantum materials research has entered a new era where a broad range of emerging materials systems and various many-body phenomena are becoming central topics of studies. This manifests in two aspects: first, the materials systems are becoming increasingly heterogeneous, such as defects in semiconductors as solid-state qubits (e.g., nitrogen-vacancy center) and nanometer-scale moiré superlattices (e.g., twisted bilayer graphene); second, the dominant interactions of interest often have quantum many-body nature, such as electron-electron, electron-phonon, and electron-hole interactions. Predictive first-principles, or *ab initio* methods, are heavily demanded to understand these complex phenomena and systems and to design next-generation electronic, optical, and quantum devices.

Standard density functional theory (DFT) approaches nowadays are able to process heterogeneous systems of  $O(10^4)$  of atoms, as achieved recently in the Gordon Bell Prize in 2023 [1]. However, DFT methods bear significant limitations in describing excited-state properties of materials, which requires explicit calculations of the electron-electron interactions that are not captured within DFT. DFT-based approaches have lagged, both quantitatively and even qualitatively, behind the desired predictive power from first principles for issues such as band gaps and electron-phonon coupling strengths – quantities critical for the design of novel materials for energy and quantum science applications, for instance.

The GW approximation [2–4] (where  $G$  stands for Green's function and  $W$  for the screened Coulomb interaction) is one approach for capturing the explicit electron-electron interactions in materials, and has been successful in predicting band gaps, band widths, and molecular excitation energy levels accurately. Moreover, extensions based on other first-principles formalisms building on top of the GW approximation have demonstrated excellent accuracy in many excited-state phenomena. For example, the first-principles GW plus Bethe-Salpeter equation approach [5] can comprehensively describe optical spectra and excitonic properties of materials ranging from bulk solids to two-dimensional (2D) materials to molecules. More recently, the development of GW perturbation theory [6] (GWPT) has enabled systematic electron-phonon coupling calculations at the many-electron level and shown excellent improvement against the results based on density-functional perturbation theory (DFPT) in several quantum materials [6, 7].

The BerkeleyGW software package offers a range of widely-adopted first-principles methodologies for electronic and optical



**Figure 1: Workflow of GW and GWPT calculations.** (a) Ground-state DFT and DFPT calculations (starting-point for GW). (b) Excited-state calculations using BerkeleyGW. Various key quantities are highlighted in the green shade (FT = Fourier transform). (c) The GW method computes quasiparticle excitation. The GWPT method computes electron-phonon coupling ( $R$  represents atom positions). (d) Charge density of a defect state in the LiH17,574 system.

excitations based on the GW approximation, as well as unique developments such as GWPT for correlated electron-phonon coupling. To apply these accurate quantum many-body methods to study complex quantum materials systems such as solid-state defects and moiré superlattices, the computational bottleneck of such approaches must be addressed. In 2020, BerkeleyGW was successfully scaled to the full machine of Summit with the NVIDIA GPU architecture [8]. In the last few years, the first-ever exascale supercomputers Frontier and Aurora became available with, however, different hardware architectures using AMD and Intel GPUs, respectively. It poses a compelling demand to port large-scale computational software packages to adapt to various GPU architectures (namely, AMD, Intel, and NVIDIA) while keeping the high performance.

The GW method computes electronic quasiparticle (QP) excitations in materials by solving the Dyson's equation,

$$h_0(\mathbf{r})\psi_i(\mathbf{r}) + \int \Sigma(\mathbf{r}, \mathbf{r}'; E_i^{QP})\psi_i(\mathbf{r}')d\mathbf{r}' = E_i^{QP}\psi_i(\mathbf{r}) , \quad (1)$$

where  $\mathbf{r}$  is the electron position,  $\psi_i$  is the quasiparticle wavefunction of state  $i$ ,  $E_i^{QP}$  is the quasiparticle energy,  $h_0$  is the mean-field Hamiltonian, and  $\Sigma$  is the self-energy in the GW approximation derived from Hedin's equations [4]. The self-energy operator describes the non-local and frequency-dependent quantum mechanical interactions among the electrons. The complexity of the GW method is much higher than the widely used DFT method, as shown in Fig. 1 and can be easily seen in the number of arguments in corresponding operators, i.e., DFT:  $(\mathbf{r})$ ; GW:  $(\mathbf{r}, \mathbf{r}'; \omega)$ .

In solving Eq. 1, the self-energy matrix elements  $\Sigma_{lm}(E)$  can be constructed by a set of (many)  $N_b$  wavefunctions  $\{\psi_n\}_{n=1..N_b}$  (also

referred to as bands or states) [2, 3]:

$$\Sigma_{lm}(E) = \frac{i}{2\pi} \sum_{nGG'} M_{ln}^{-G^*} M_{mn}^{-G'} \int_0^\infty d\omega \frac{\epsilon_{GG'}^{-1}(\omega) v_{G'}}{E - E_n - \omega} , \quad (2)$$

where  $l, m$  are quasiparticle band or state indices of interest,  $n$  runs over the whole  $N_b$  bands range,  $G$  labels planewave (PW) basis elements,  $\epsilon_{GG'}^{-1}$  is the inverse dielectric matrix of the system in planewave basis,  $E_n$  is the orbital energy,  $v_{G'}$  is the Coulomb interaction in the reciprocal space, and  $M$  represents the plane-wave matrix elements of wavefunctions,  $M_{mn}^G = \int d\mathbf{r} \psi_m^*(\mathbf{r}) e^{iG \cdot \mathbf{r}} \psi_n(\mathbf{r})$ . The frequency ( $\omega$ ) integration can be very well treated via the generalized plasmon-pole (GPP) model [2], as well as the direct full-frequency (FF) sampling.

The inverse epsilon matrix  $\epsilon^{-1}$  is constructed with the polarizability matrix  $\chi_{GG'}$ ,

$$\epsilon^{-1}(\omega) = [\mathbf{I} - \mathbf{v}\chi(\omega)]^{-1} , \quad (3)$$

where  $\mathbf{I}$  and  $\mathbf{v}$  are diagonal identity and Coulomb matrices, and,

$$\chi_{GG'}(\omega) = 2 \sum_{vc} M_{vc}^{G^*} \Delta_{vc}(\omega) M_{vc}^{G'} . \quad (4)$$

Here  $v, c$  are wavefunction indices spanning the  $N_v$  valence and  $N_c$  conduction states. Note that  $N_v + N_c = N_b$ .  $\Delta_{vc}(\omega)$  is an energy factor containing the orbital energies  $E_v, E_c$ , and dependence on frequency  $\omega$ .

A standard BerkeleyGW workflow (see Fig. 1) starts with the ground-state DFT (and DFPT) calculations to generate input for excited-state GW (and GWPT) calculations. Typically, many bands (up to thousands or tens of thousands) are needed for convergence: a challenge for iterative solvers in most DFT codes. BerkeleyGW

provides a Parabands module that can generate a large set of wavefunctions  $\{\psi_n\}$  of  $N_b$  bands based on DFT output. The Epsilon module computes the inverse dielectric matrix. The Sigma module constructs the self-energy operator and evaluates a set of self-energy matrix elements ( $N_\Sigma$  diagonal elements, or  $N_\Sigma^2$  full matrix elements including off-diagonal ones), for the GW quasiparticle excitation energies  $E^{QP}$  and GWPT electron-phonon matrix elements. For practical implementations of Eqs. 2, 3 and 4, the canonical GW method has an overall  $O(N^4)$  scaling with standard calculation parameters summarized in Table 1.

In this work, we present several significant methodological and algorithmic innovations implemented in BerkeleyGW:

- True portability across AMD, Intel, and NVIDIA GPU architectures using *directive-based* OpenACC and OpenMP models demonstrated on Frontier, Aurora, and Perlmutter.
- Kernel optimizations with *hardware-optimized* programming languages, i.e., HIP for AMD, SYCL for Intel, and CUDA for NVIDIA GPUs, reaching high fraction of peak performance.
- Excellent strong and weak scaling up to (nearly) the full machine of Aurora and Frontier.
- New optimized kernels achieving high FP64 throughput of 1.069 EFLOP/s on Frontier (9,408 nodes or 75,264 GPUs) and 707.52 PFLOP/s on Aurora (9,600 nodes or 115,200 GPUs), corresponding to 59.45% and 48.79% of the theoretical and attainable peak, respectively.
- Massive materials applications including silicon (Si) divacancy (up to 2,742 atoms) and lithium hydride (LiH) defect (up to 17,574 atoms).
- Advanced GW methods, including GWPT for electron-phonon coupling, full-frequency (FF) GW, and reduced scaling GW with mixed stochastic-deterministic approach.

**Table 1: Computational parameters in the GW workflow.**

Symbol	Synopsis
$N_G^\psi$	No. of PWs ( $G$ vectors) for wavefunctions $\{\psi_n\}$
$N_G$	No. of PWs ( $G$ vectors) for $\epsilon, \chi$ (Eq. 3,4)
$N_v$	No. of valence bands (Eq. 4)
$N_c$	No. of conduction bands (Eq. 4)
$N_b$	No. of total bands $N_v + N_c$ (Eq. 2)
$N_\Sigma$	Dimension of $\Sigma(E)$ self-energy matrix (Eq. 2)
$N_E$	No. of $E$ grid points for $\Sigma(E)$ (Eq. 2)
$N_\omega$	No. of $\omega$ integration points (Eq. 2)
$N_{\text{Eig}}$	No. of eigenvectors for low rank $\chi^0(\omega)$
$N_p$	No. of phonon perturbations $R_p$ (Eq. 5)

All parameters grow linearly with system size except  $N_E$  and  $N_\omega$ .

## 4 Current State of the Art

The GW approximation, originally derived by Hedin [4], has been a successful *ab initio* approach to obtaining quasiparticle properties since the seminal work by Hybertsen and Louie [2, 3]. These properties allow for accurate understanding of energy levels and their alignments in a variety of environments, making GW the theory of choice for studying heterogeneous and extended systems.

As described above, the traditional sum-over-states formulation of GW calculations has a formal scaling of  $O(N^4)$ , and different approaches have been taken to enhance the computational efficiency

and scaling of GW calculations. In one approach, the summation over empty states is eliminated by using DFPT. This has been implemented successfully in multiple code bases by Umari *et al.* [9], Giustino *et al.* [10], and Govoni *et al.* [11–13]. While the scaling of this approach remains  $O(N^4)$ , it has a distinct advantage in avoiding the generation of wavefunctions of many empty states. Another approach that stays within the traditional sum-over-states paradigm is to use so-called pseudobands that are stochastic averages over the Kohn-Sham (KS) states within defined energy windows. Altman *et al.* [14] have shown that the use of pseudobands greatly reduces the number of bands needed to obtain accurate quasiparticle energies, and reduces the scaling of GW calculation to  $O(N^{2.4})$ . The pseudobands concept is inspired by the fully stochastic GW approach [15], which shows linear scaling for the computation of certain materials properties. However, stochastic GW introduces uncorrelated stochastic errors, and cannot compute all electronic properties available from deterministic GW calculations. Real-space, imaginary-time GW was originally proposed and implemented by Rieger *et al.* [16], reducing the system size scaling to  $O(N^3)$ . Several implementations have used this scheme or related approaches [17–19]. However, this is achieved through various transformations that lead to a significant prefactor, so that the system size at which the  $O(N^3)$  space-time GW method becomes favorable is highly system dependent. Yeh and Morales used interpolative density fitting to also achieve  $O(N^3)$  scaling and argued that the prefactor should generally be smaller than that in the space-time approach [20]. However, their calculations were done on down-folded model Hamiltonians, so the full promise of this approach for first-principles GW calculations remains to be determined.

The GW method is often implemented within the planewave basis set, such as the popular software packages BerkeleyGW [21] (this work), WEST [11, 22], Quantum ESPRESSO [23], Abinit [24], Yambo [25], SternheimerGW [26], and VASP [27, 28]. Other implementations of the GW method use localized basis functions, such as numerical atomic orbitals (FHI-aims [29]), Gaussians (Fiesta [30], MolGW [31]), linearized augmented-planewave with local orbitals (Exciting [32], ELK [33]), and mixed Gaussian and planewaves (CP2K [34]). Localized basis sets typically have reduced computational cost due to their smaller basis size, but convergence has to be checked carefully in systems with diffused states. Generally, planewave implementations are more suitable for extended systems, while localized basis sets are used for localized systems such as molecules and nanoclusters.

As most GW studies still focus on systems with tens to hundreds of atoms because of the high computational complexity, the community is pushing towards much larger systems to access new phenomena. Among the largest GW calculations to date, we mention here (i) the twisted bilayer phosphorene structure containing ~2,700 atoms (~13.5k electrons) using linear-scaling stochastic GW by Brooks *et al.* [35], (ii) our previously reported result of silicon divacancy defect with 2742 atoms (~11k electrons) using the planewave basis set and the deterministic approach with BerkeleyGW [8], (iii) the ~10k electrons calculation with WEST by Yu *et al.* [22], and (iv) the recent result of ~14k atoms/electrons calculation of LiH using a low-rank approximation approach by Wu *et al.* [36]. In this work, we focus on optimizing the standard GW approach with planewave basis sets as implemented in BerkeleyGW [21]. Our results achieve

unprecedented scalability and applicability to describe complex materials on exascale platforms.

## 5 Innovations Realized

We present recent theoretical, algorithmic, and HPC optimization advances in BerkeleyGW. Sec. 5.1 introduces the first-of-its-kind GW perturbation theory (GWPT) to study correlated electron-phonon coupling. Sec. 5.2 details methods to accelerate full-frequency GW calculation by reducing its  $O(N_G^2)$  cost dependence and  $O(N^3)$  memory bottleneck. Sec. 5.3 addresses the need for large  $N_b$  via a mixed stochastic-deterministic scheme that compresses the wavefunction space and effectively lowers the  $O(N^4)$  scaling. Sec. 5.4 outlines performance portability across pre-exascale and exascale platforms. Sec. 5.5 and Sec. 5.6 report GPU kernel optimizations for diagonal and off-diagonal self-energy matrix elements, pushing performance to high peak and enabling large-scale GW and GWPT calculations, including full solutions to Dyson's equation.

### 5.1 GW Perturbation Theory

Electron-phonon coupling is one of the central interactions in materials physics, and is critical to a wide range of materials properties, including carrier mobility, optical absorption, quantum decoherence, and phonon-mediated superconductivity, among others. Accurate first-principles computation of microscopic electron-phonon interactions of materials systems is essential to design and optimize next-generation electronic and optoelectronic devices. The prevailing approach for systematic calculations of electron-phonon matrix elements is DFPT, which is a linear-response theory of DFT. The linear-response formulation elegantly decomposes the phonon perturbations to an electronic system into independent modes, where each perturbation can be solved at a similar cost as a standard DFT calculation. However, DFPT inherits similar limitations as DFT, and becomes insufficient for materials with stronger electron correlation effects.

GWPT is a newly developed method that enables systematic computation of electron-phonon coupling at the many-body level within the linear-response formulation for the first time [6]. GWPT has demonstrated excellent accuracy in capturing the correlation effects in the electron-phonon coupling beyond DFPT in several quantum materials. In GWPT, the equation to compute the atom-displacement-perturbed self-energy operator matrix elements is,

$$\left[ \frac{\partial}{\partial R_p} \Sigma(E) \right]_{lm} = \frac{i}{2\pi} \sum_{nGG'} \left[ \frac{\partial M_{ln}^{-G^*}}{\partial R_p} M_{mn}^{-G'} + M_{ln}^{-G^*} \frac{\partial M_{mn}^{-G'}}{\partial R_p} \right] \times \int_0^\infty d\omega \frac{\epsilon_{GG'}^{-1}(\omega) v_{G'}}{E - E_n - \omega}, \quad (5)$$

where the operator  $\frac{\partial}{\partial R_p}$  represents an atom-displacement induced perturbation, where  $p$  labels the degrees of freedom (e.g., a particular atom moving along one direction, or a phonon eigenmode). The construction of the first-order change in self-energy  $\partial_R \Sigma$  needs the first-order changes in the matrices  $\partial_R M$ , which are constructed by first-order changes in the wavefunctions  $\partial_R \psi_n$  of all  $N_b$  bands.

BerkeleyGW is the only package offering the implementation of the unique GWPT method. Currently, the frequency dependence is treated within the GPP model, which provides a straightforward

strategy for implementation. Moreover, the GPP kernel has been extensively optimized for excellent peak performance, alleviating the high computational burden of GWPT that introduces an additional prefactor  $N_p$  (number of perturbations) to the complexity of the standard GW. On the other hand, the  $N_p$  perturbations are independent and massively parallelized to full scale with minimal communications on exascale machines.

### 5.2 Fast Full-Frequency GW Method

Although the treatment of the frequency dependence of the polarizability  $\chi(\omega)$  has often been achieved via the GPP model, recent advances have enabled direct calculations of the full-frequency dependence, at a competitive computational cost. The key advance is the static subspace approximation [37], where the zero-frequency polarizability  $\chi(\omega = 0)$  is firstly calculated (using Eq. 4), then a subspace is defined with  $\chi(\omega = 0)$  to calculate non-zero frequencies [11–13, 38, 39]. The zero-frequency polarizability is diagonalized and the  $N_{\text{Eig}}$  most significant eigenvectors are kept as the subspace basis. The non-zero-frequencies polarizability can then be constructed as a modified Eq. 4,

$$\chi_{BB'}(\omega \neq 0) = 2 \sum_{vc} M_{vc}^{B^*} \Delta_{vc}(\omega) M_{vc}^{B'}, \quad (6)$$

where  $B$  and  $B'$  index over  $N_{\text{Eig}}$  subspace basis vectors. The subspace representations of  $M_{vc}$  and  $\chi$  are connected to their planewave representations via the  $N_G \times N_{\text{Eig}}$  projection matrix  $C_s$ ,  $M_{vc}^B = \sum_G M_{vc}^G C_s^{GB}$  and  $\chi_{BB'} = \sum_{GG'} (C_s^{GB})^* \chi_{GG'} C_s^{G'B'}$ . This subspace compression reduces the computation of  $\chi(\omega)$  from  $O(N_\omega N_v N_c N_G^2)$  to  $O(N_v N_c N_G^2 + N_\omega N_v N_c N_{\text{Eig}}^2)$  since the full planewave basis is only used for the zero frequency. In general, a subspace fraction of 10–20% is sufficient to converge GW quasiparticle energies, hence this approximation results in a speedup of  $\sim 25 – 100$  times over the full planewave implementation [37, 40].

Full-frequency polarizability calculations are further enabled by careful GPU offloading of the key computational kernels. Significant prior efforts have been spent in optimizing the Fourier transformations to obtain the planewave matrix elements  $M_{nm}^G$  (*MTXEL* kernel [8]). Calculation of the polarizability via Eq. 6 in *CHI\_SUM* kernel is most computationally intensive, which suffers from an  $O(N^3)$  memory footprint on both host and device. To address this issue, our redesigned implementation effectively divides the computation into workable blocks over the  $N_v$  valence bands, which we call the NV-Block algorithm. Combination of NV-Block and static subspace approximation enables efficient and accurate calculation of the full-frequency polarizability [41]. Full-frequency self-energy calculations also benefit from the subspace approximation by performing the  $G$  and  $G'$  sums in the reduced basis set (Eq. 2), where the key steps can be casted as dense matrix multiplication.

The full-frequency GW offloading was solely performed using the open programming models OpenMP-target and OpenACC, which enables portability across the various leadership HPC systems and reduces the development overhead. Since most of the computationally limiting kernels could be offloaded to vendor matrix multiplication libraries, the use of the open models was less of a hindrance to performance.

### 5.3 Reduced Cost and Scaling with a Mixed Stochastic-Deterministic Algorithm

A major bottleneck of GW calculations is the sum-over-bands in inverse dielectric matrix  $\epsilon^{-1}$  and self-energy  $\Sigma$  (Eqs. 2 and 4). We have developed a novel algorithm based on a stochastic compression of the Lehmann representation of the Green's function [14], which significantly compresses the high-energy bands, and reduces the actual computational scaling with system size. The method amounts to modifying the KS energies  $E_n$  and states  $|\psi_n\rangle$  that are fed into the GW calculations. First, the energy spectrum of the KS states is partitioned into slices  $\{S\}$  and a special protection region  $P$  around the Fermi energy. The KS states and energies in  $P$  remain untouched. The states in the remaining slices are replaced with stochastic linear combinations of the KS states within each slice, yielding stochastic pseudobands  $|\xi_j^S\rangle = \frac{1}{\sqrt{N_\xi}} \sum_{n \in S} e^{2\pi i \theta_n^j} |\psi_n\rangle$ . Here  $\theta \in [0, 1]$  is a random scalar, and we construct  $N_\xi$  stochastic pseudobands for each slice, with  $N_\xi$  typically between 2-5. We assign to these states  $|\xi_j^S\rangle$  the average energy of the KS states in  $S$ .

The advantage of this approach is multiple-fold. First, because the slices are chosen according to their energy, they do not scale with system size. Second, through a careful error analysis, one can gradually increase the energy spanned by each slice, leading to an *exponential* compression of the KS states necessary in the sums-over-bands in Eqs. 2 and 4. Finally, pseudobands capture the full KS Hamiltonian and eliminate band-truncation parameters in the calculation of  $\chi$  and  $\Sigma$ .

To construct pseudobands, one needs to fully diagonalize the KS Hamiltonian, a bottleneck that scales as  $O(N^3)$ . We avoid this by rewriting pseudobands as random vectors  $|x\rangle$  projected to the slice subspaces  $|\xi_j^S\rangle := f^S(H)|x\rangle$ . The projection operator  $f^S(H) = \sum_{n \in S} |\psi_n\rangle\langle\psi_n|$  can be efficiently approximated using a Chebyshev-Jackson expansion  $\tilde{f}_l^S(H)$  of order  $l$  [42, 43]. In practice, the entire construction scales as a matrix-vector operation  $\sim O(N) - O(N^2)$ . The pseudobands and Chebyshev-Jackson methods alleviate a traditional bottleneck for sum-of-bands in GW calculations.

### 5.4 Portability with Open Standard

Over the years, the optimization of BerkeleyGW [21] has enabled high-performance execution on leadership class HPC systems, from many-core CPU [44] to heterogeneous GPU architectures [8], obtaining outstanding performance and achieving excellent time to solution. However, as HPC systems grow increasingly complex, with most computational power residing in specialized accelerators, it has been realized that a greater challenge lies in ensuring performance portability across HPC platforms.

The BerkeleyGW's performance portability strategy is to leverage open, directive-based programming models, specifically OpenMP-target (OMP) [45] and OpenACC (OACC). Open standard programming models enable broad support across diverse architectures – including NVIDIA, AMD, and Intel GPUs – while preserving a unified codebase. This approach simplifies maintainability and helps obtain decent offloaded performance for novel developments. Despite the several advantages, we faced many challenges in porting the pipeline as we navigated around compiler pitfalls, especially to: (i) support the programming model and stay up-to-date with the

**Table 2: Application systems and computation sizes.**

System Name	$N_G^\psi$	$N_G$	$N_b$	$N_v$	$N_c$
Si214	31,463	11,075	$\gtrsim 5,500$	428	$\gtrsim 5,000$
Si510	74,653	26,529	$\gtrsim 15,000$	1,020	$\gtrsim 13,900$
Si998	145,837	51,627	$\gtrsim 28,000$	1,996	$\gtrsim 26,000$
Si2742	363,477	141,505	80,695	5,484	75,211
Si2742'	363,477	141,505	15,840	5,484	10,356
LiH998	81,313	52,923	$\gtrsim 3,100$	499	$\gtrsim 2,600$
LiH17574	506,991	362,733	49,920	8,787	41,133
BN867	439,769	84,585	49,920	1,734	48,186

model standards, (ii) generate offloaded kernels ensuring correctness of results, (iii) interface with the vendor-specific libraries and corresponding APIs, and (iv) perform kernel optimizations capable of exploiting the characteristics of the specific hardware.

The BerkeleyGW GPU implementation demonstrates that open standards are not only practical for portability, but also capable of delivering high performance, with OpenACC recovering almost the entirety of the performance delivered by the best CUDA implementation on NVIDIA GPU [46]. On the other hand, performance for the open models is lower on AMD and Intel GPUs, especially for large custom kernels not relying on vendors' offloaded libraries. The adopted portability strategy has been widely successful, with the public release of *BerkeleyGW-4.0* (BGW-4.0) [47] in production on NVIDIA, AMD and Intel GPU platforms alike, showcasing the effectiveness and readiness of open models in heterogeneous HPC environments moving forward.

### 5.5 Optimized GPP Kernel for Diagonal Self-Energy Matrix Elements

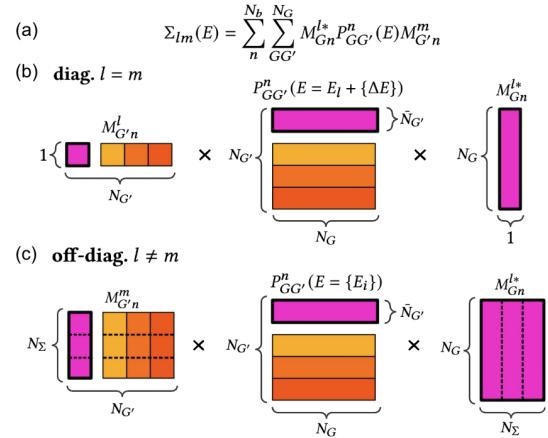
The diagonal matrix elements of the self-energy operator directly provides information of quasiparticle energy levels, which are among the most commonly desired quantities. The most computationally intensive GPP kernel (Fig. 2a) for the diagonal elements (denoted as *diag*) in the Sigma module is ported to the HIP and SYCL programming languages in order to gain the most optimal performance on Frontier and Aurora. CUDA version of the GPP *diag* kernel has been developed for NVIDIA GPUs [8]. In GPP *diag* kernel, we divide the computation over self-energy pools. Within each pool, the GPP *diag* kernel computes its assigned self-energy matrix elements, e.g.  $\Sigma_{ll}$ , i.e., the diagonal ones. The GPP *diag* kernel is executed entirely on the device, with matrix elements  $P_{GG'}^n$  being computed on the fly. The summation over all  $N_{G'} (= N_G)$  is distributed over MPI ranks within a self-energy pool ( $N_{\text{rank}}$  per pool) in the calculation, with each rank holding  $\bar{N}_{G'} = N_{G'}/N_{\text{rank}}$  elements. Thus, in each kernel invocation,  $N_G$  and  $\bar{N}_{G'}$  are different bounds for the summation of  $G$  and  $G'$ , respectively. The relation between loop indices  $N_b < \bar{N}_{G'} \ll N_G$  dictates the design of the kernel (see Fig. 2b). In both HIP and SYCL kernels, we employ two levels of two-dimensional parallelism to decompose the problem. The first level decomposes the summation over  $\bar{N}_{G'}$  and  $N_b$  to distribute the computation over work-groups. The second level further decomposes the summation over  $N_G$  within each work-group. This decomposition scheme effectively utilizes the accelerators' massive parallelism by maintaining high arithmetic intensity within each

work-group. In the following, we present detailed kernel adaptations to accelerators on Frontier and Aurora, where terms such as thread blocks (used in HIP and AMD architecture) and work-groups (used in SYCL and Intel architecture) are used interchangeably.

**5.5.1 Adapting to Frontier and Aurora Accelerators.** The Frontier's AMD MI250X and Aurora's Intel PVC GPUs have similar characteristics, therefore the GPP kernel optimizations share similar techniques on both architectures summarized below:

- (1) Explicit memory management on device to coalesce memory access within a thread block. Prior to launching the kernel, one thread from each thread block loads sections of the  $M_{G'n}^{l*}$  and  $M_{G'n}^m$  arrays, corresponding to the current  $\bar{N}_{G'}$  and  $N_b$  blocks, into the shared memory. This significantly reduces the number of memory moves incurred during execution and drastically increases the arithmetic intensity.
- (2) Block size tuning of the second level of kernel parallelism to maximize shared memory usage while avoiding memory overflow. We choose block sizes in the local parallelization to fully utilize Local Data Share (LDS) for each block on GPU.
- (3) Loops are manually unrolled to gain maximum Vector General-Purpose Registers (VGPRs) and Scalar General Purpose Registers (SGPRs) occupancy without overflow. In particular, VGPRs overflowing on AMD accelerators incur huge latency in instruction execution. By monitoring the VGPRs occupancy during compile time and carefully tuning loop unrolling and thread block sizes, we obtain over 97% Vector Arithmetic Logic Unit (VALU) utilization rate while ensuring 0 VGPRs spillage.
- (4) Replace expensive operations such as divisions and absolute values with reciprocals and multiplications as discussed in [8]. This not only avoids the execution of such operations but also adds to the Fused Multiply Add (FMA) instruction count, which utilizes hardware more efficiently. In this way, the GPP kernel contains over 57% FMA instructions with less than 4% being inefficient transcendental operations.
- (5) Two-stage reduction over thread blocks. First, each thread block designates a thread that accumulates the values using a masked intrinsic warp shuffle function. Then, we use the atomic add operation to accumulate the final result over all thread blocks. The choice of the number of thread blocks becomes a balance between register occupancy, memory access, and number of atomic reductions.

**5.5.2 Hardware-Specific Adaptations.** The HIP and SYCL kernels have been adapted to better match the characteristics of each architecture to maximize hardware parallelization performance. In the HIP kernel, we utilize a larger thread block size with more threads per block to maximize occupancy during execution. In the SYCL kernel, we instead tune the kernel to have more work-groups and fewer work-items per work-group to match the optimal Single Instruction Multiple Data (SIMD) width [48]. For AMD MI250X, more shared memory is loaded locally for each thread block with larger block size of computation to accommodate the increased memory. For Intel PVC, the layout of the work-groups require smaller chunks of shared memory for the large number of work-groups. These optimized memory layouts further decrease instruction stall rate and improve the arithmetic intensity.



**Figure 2: Parallelization and data layout in Sigma-GPP.** (a) GPP self-energy working equation. (b) Distribution of data in optimized GPP *diag.* kernel. (c) Distribution of data in optimized GPP *off-diag.* kernel. (b) and (c) represent one set of operations within the loop of summation over  $n$ .

## 5.6 Optimized GPP Kernel for Off-Diagonal Self-Energy Matrix Elements

Our optimized GPP *diag.* kernel for diagonal matrix elements is at the ceiling of achievable arithmetic intensity considering its matrix-vector-like operation nature. Furthermore this implementation minimizes memory requirements by generating the band and frequency dependence of the inner matrix on the fly, which is highly efficient for diagonal-element calculations. On the other hand, advanced GW methods (including GWPT) require the calculation of full self-energy matrix including off-diagonal elements, thus the number of elements to be computed for  $N_\Sigma$  bands becomes  $N_\Sigma^2$  (i.e., the full matrix), instead of  $N_\Sigma$  for diagonal-only elements. In this case we can gain arithmetic intensity by recasting the original GPP kernel into a matrix-matrix multiplication-like kernel. This is achieved by reformulating the formalism via generalizing the internal frequency argument  $E$  in  $\Sigma_{lm}(E)$  to a predefined uniform frequency grid  $\{E_i\}$  independent of  $(l, m)$  indices (in contrast to the GPP *diag.* kernel) over the energy range of interest (e.g., the bandwidth across the  $N_\Sigma$  bands). This generalization (see Fig. 2c) computes the full matrix of  $\Sigma_{lm}(\{E\})$  with  $l$  and  $m$  span  $N_\Sigma$ , offering much more accurate self-consistent quasiparticle energies from the full solutions of the Dyson's equation and dynamical behavior of the electron-phonon matrix elements from GWPT.

We have implemented a new full-matrix GPP kernel which efficiently computes off-diagonal matrix elements (denoted as *off-diag.*). To increase arithmetic intensity, in the GPP *off-diag.* kernel, we precompute the band ( $n = 1, \dots, N_b$ ) and frequency ( $\{E_i\}_{i=1, \dots, N_E}$ ) dependent matrices  $P$  (Fig. 2c) over all  $(n, E)$  pairs, and perform ZGEMM for each  $(n, E)$  configuration. The pre-computation (*prep.* step) utilizes the same optimizations as in the GPP *diag.* kernel (Sec. 5.5.1 and 5.5.2). For diagonal-only calculations, this new strategy provides no benefit, because it significantly increases the memory demands, and the overhead of the *prep.* step cancels the performance gain from ZGEMM. However, when the full  $\Sigma(E)$  matrix is required for large  $N_\Sigma$  (for full solutions of GW Dyson's equation

and large-scale GWPT calculations), the reuse of the precomputed  $P$  matrices for many target states makes this new formulation very competitive and efficient. The resulting computation of the GPP *off-diag.* kernel reduces to two consecutive dense matrix multiplications (ZGEMM) of dimensions  $N_\Sigma \times N_G \times N_G$  and  $N_\Sigma \times N_G \times N_\Sigma$  per  $(n, E)$  pair (Fig. 2c), achieving a two-fold increase in performance throughput compared to the GPP *diag.* kernel.

## 6 How Performance Was Measured

The performance measurements are obtained on a set of realistic applications (see Table 2) to highlight the capabilities of BerkeleyGW across methodologies and system sizes. Our general baseline performance features semiconductor defects (proxy for solid-state qubits) with systems of variable sizes from small (214 silicon atoms) to large (2742 silicon atoms) [49]. The largest system Si2742 contains a total of 80,695 bands. The mixed stochastic-deterministic pseudobands approach allows for improved convergence at a lower number of band, i.e., at  $N_b = 15,840$ . We label the same system with 15,840 bands as Si2742'. We also present massive applications of GW calculations on defects in solid LiH, with supercells containing up to 17,574 atoms, surpassing the previously reported largest GW calculation of 13,824-atom LiH [36]. Additionally, we report results of 3.88°-twisted BN moiré bilayer consisting of 867 atoms (with 1.5-nm vacuum layer), with a carbon substitution at a boron site adjacent to a nitrogen vacancy. Defects in layered BN are useful as single-photon emitters, with moiré twisting offering tunability. To demonstrate the electron-phonon coupling capabilities, we perform GWPT calculations on a LiH defect system with 998 atoms, involving six atomic displacements ( $N_p = 6$ ). These calculations help describe quantum decoherence and excitation lifetimes.

The performance results are collected from three HPC systems:

- *Frontier* (OLCF): 9,408 nodes, each with 1 AMD Milan CPU and 4 AMD Instinct MI250X GPUs, each comprised of 2 Graphics Compute Dies (GCD) for a total of 8 devices. FP64 peak performance per GPU 23.9 TeraFLOP/s and aggregated 1.80 ExaFLOP/s.
- *Aurora* (ALCF): 10,624 nodes, each with 2 Intel Xeon CPU Max Series and 6 Intel Data Center GPU Max Series “Ponte Vecchio” (PVC), each comprised of 2 tiles for a total of 12 devices. FP64 peak per GPU 17 TeraFLOP/s and aggregated 2.17 ExaFLOP/s.
- **Note:** At the time of this work, Aurora's GPUs are not running at full capacity. Therefore, we compare against the measured FP64 Vector MAD Peak of 11.4 TeraFLOP/s using Intel Advisor Profiler. Hence, the attainable peak of Aurora is 1.45 ExaFLOP/s.
- *Perlmutter* (NERSC): 1,792 nodes, each with 1 AMD Milan CPU and 4 NVIDIA A100 GPUs. FP64 peak performance per GPU 9.7 TeraFLOP/s and aggregated 69.5 PetaFLOP/s.

Unless otherwise stated, in this work, a “GPU” means a single NVIDIA A100 for Perlmutter, a single MI250X’s GCD for Frontier, and a single PVC’s tile for Aurora.

To determine the number of floating point operations (FLOPs) performed in the Sigma module, we use the canonical FLOP count from the most computationally intensive kernel. In the Sigma module, the GPP kernel takes up over 95% of the FLOPs for production calculations. The computational complexity for the GPP *diag.* kernel is  $O(N_\Sigma N_b N_G^2 N_E)$ . Through a series of tests listed in Table 3, we determine a linear relationship between the FLOP count and

**Table 3: FLOP count from measured (Meas.) and estimated (Est.) performance for Si-214 on Frontier (F) and Aurora (A).**

	$N_\Sigma$	$N_b$	$N_G$	$N_E$	Est. (TFLOP)	Meas. (TFLOP)	Accuracy
F	2	5,000	3,911	3	38.32	38.55	99.39%
	4	15,045	26,529	3	10,609.67	10,564.75	99.57%
	8	6,340	11,075	4	2,077.88	2,064.84	99.37%
A	2	3,000	11,075	6	416.27	415.17	99.74%
	1	5,000	11,075	6	346.89	345.89	99.71%
	1	2,000	11,075	6	138.76	139.42	99.52%

the computational complexity as,

$$\text{FLOP count (GPP diag.)} = \alpha \times N_\Sigma N_b N_G^2 N_E , \quad (7)$$

where  $\alpha$  is an architecture- and compiler-dependent constant prefactor. To determine the quasiparticle energy  $E_i^{\text{QP}}$  from the self-consistent relation in Eq. 1, we need a few  $N_E \sim O(1) - O(10)$  sampling points for  $E$  in evaluating specific diagonal element  $\Sigma_{ll}(E)$ , where the value of  $E$  depends on the  $l$  index. We use the ROCm profiler for AMD GPUs on Frontier and the Intel Advisor profiler for Intel GPUs on Aurora to determine the prefactor value  $\alpha$ . The prefactor for GPP *diag.* kernel on Frontier and Aurora are measured to be  $\alpha_{\text{Frontier}} = 83.50$  and  $\alpha_{\text{Aurora}} = 94.27$ , respectively. In Table 3, we also verify the accuracy of the prefactor with less than 1% discrepancy between the estimated and measured FLOP count.

For the GPP *off-diag.* kernel, we account only for the ZGEMM operations, with the *prep.* step contributing diminishing fraction of FLOPs at large  $N_\Sigma$  and  $N_E$ . We sample uniformly  $N_E \sim O(10^2)$  points for  $E$  in  $\Sigma_{lm}(E)$ , reformulated to be independent of  $(l, m)$  to cast the algorithm to ZGEMM, spanning the energy window of  $N_\Sigma$  bands for full solutions of Dyson’s equation. Our implementation performs  $2N_b N_E$  times of ZGEMM operations, with the number of FLOPs counted as (based on standard ZGEMM FLOP count):

$$\begin{aligned} \text{FLOP count (GPP off-diag.; ZGEMM only)} \\ = 2N_b N_E \times 8(N_\Sigma N_G^2 + N_G N_\Sigma^2) . \end{aligned} \quad (8)$$

Note that in the GPP *off-diag.* kernel, we only count FLOPs from ZGEMM, but still measure the full kernel runtime (including *prep.* step), hence our reported performance stands as a lower bound.

## 7 Performance Results

### 7.1 Performance Portability

We evaluate the performance portability of several GW implementations across different programming models and hardware architectures, as summarized in Table 4. We focus on five programming models: two directive-based open standards (OpenMP and OpenACC) and three hardware-optimized models (CUDA, HIP, and SYCL). These implementations are benchmarked on different GPU architectures, namely NVIDIA, AMD, and Intel.

Open standards such as OpenACC and OpenMP remain valuable tools for achieving performance portability, particularly as compiler technology continues to mature. On NVIDIA hardware, OpenACC demonstrates exceptional performance, recovering over 90% of the best CUDA implementation, highlighting good compiler support. However, the situation is less favorable on Frontier (AMD GPUs),

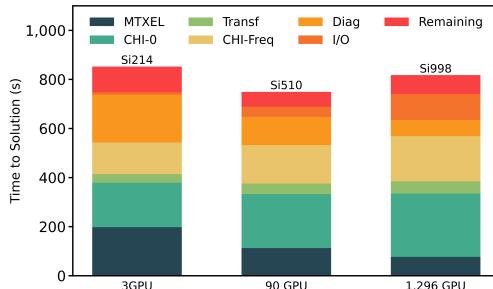
**Table 4: Sigma time to solution (seconds) for Si-510 with  $N_\Sigma = 128$  across architectures and programming models.**

# of Nodes	GW-GPP diag.								GW-FF				
	Perlmutter				Frontier			Aurora			Perlmutter	Frontier	Aurora
	OMP <sup>†</sup>	OMP	OACC	CUDA	OMP <sup>†</sup>	OACC	HIP	OMP <sup>†</sup>	OMP	SYCL	OACC	OACC	OMP
4	4,186.3	3,268.7	3,197.3	2,928.3	2,562.1	2,111.9	1,382.5	3,621.1	2,877.2	1,416.0	528.2	354.4	364.7
8	1,978.9	1,640.2	1,601.1	1,467.1	1,294.9	1,062.7	684.6	1,835.2	1,437.9	736.0	281.8	188.3	208.3
16	990.1	826.0	804.6	744.2	654.9	548.6	369.3	918.5	727.1	390.0	159.3	112.7	128.2
32	501.9	419.7	407.8	383.8	336.8	282.0	191.4	467.6	372.6	205.3	99.22	70.6	93.9
64	260.1	218.3	214.7	203.5	182.7	147.3	110.5	245.6	199.1	121.6	71.5	53.7	69.9

where OpenACC gives only 60–70% of our best HIP performance, which is likely attributed to overall less maturity and less aggressive compiler optimization capabilities. Furthermore, OpenACC is not currently supported by Intel compilers on Intel GPUs, limiting its portability across all major vendors. The OpenACC implementation discussed here is publicly available as part of the released BGW-4.0.

For OpenMP, two implementations have been evaluated here. The first one, labeled as OMP<sup>†</sup> in Table 4, was also released in BGW-4.0, and at the time of release, it did not incorporate all the optimizations present in the OpenACC version. As a result, OMP<sup>†</sup> is approximately 15–20% slower than OpenACC on both Perlmutter and Frontier. The second OpenMP version, labeled as OMP, includes additional optimizations similar to our OpenACC implementation, including reduced kernel branching and improved data reuse. However, it still lacks support for asynchronous GPU execution. Despite this, it nearly matches the performance of OpenACC on Perlmutter. On Frontier, however, this optimized OMP implementation performs poorly, taking long execution time even for small applications. The issue appears to arise from the compiler’s attempt to parallelize over the innermost strided loops of the kernel, which are correctly serialized in the OpenACC version using *loop seq*. On Intel GPUs, the optimized OpenMP version provides around 20% better performance than OMP<sup>†</sup>, but is still 50% slower than the SYCL implementation. These results underscore current limitations of the OpenMP kernel optimization capabilities on Intel software and hardware.

While these results may suggest an insurmountable performance advantage for hardware-optimized programming models, especially on non-NVIDIA hardware, we emphasize that our goal is to evaluate how current implementations perform *out of the box*. We anticipate that, with continued improvements in compilers and tool chains, the open standards could catch up to their hardware-optimized counterparts across all major GPU architectures. Despite current

**Figure 3: Weak scaling of the GW-FF Epsilon on Aurora.**

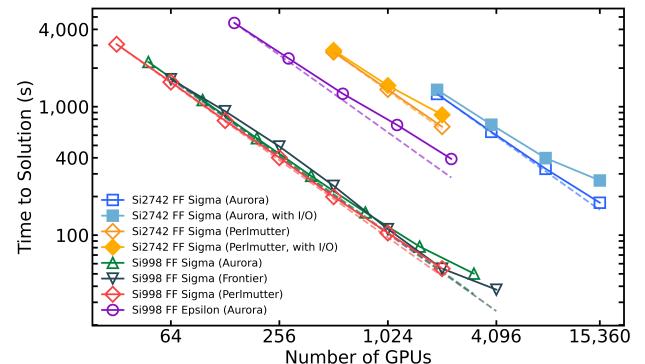
limitations, the results presented here are encouraging, showing that open standards already deliver competitive performance across different hardware platforms, and hold promise as viable, maintainable, and portable solutions for future architectures.

## 7.2 Performance of GW-FF

The full-frequency GW implementation in BerkeleyGW is only slightly more costly than the GPP method, due to the use of the static subspace approximation. This implementation is highly scalable due to the multi-layer parallelizations (including the additional level over frequencies), showing strong scaling up to thousands of GPUs, with portability across all three major vendors.

In the Epsilon module, the computational cost for full-frequency polarizability is only about twice as high as for the GPP model. The weak scaling of the FF implementation is shown in Fig. 3. The main computational kernels (*CHI-0*, *CHI-Freq*, and *Transf*) show nearly ideal weak scaling, while the lower scaling kernels (*MTXEL* and *Diag*) decrease significantly. In this case, the additional calculation of 19 frequencies with  $\sim 20\%$  subspace fraction only takes about the same time as the initial zero-frequency calculation with the full planewave basis.

The calculation of self-energy in Sigma using the full-frequency polarizability becomes very efficient with the static subspace approximation. Furthermore, the extreme parallelism offered by the number of self-energy elements allows for strong scaling up to tens of thousands of GPUs and portable scaling on all three HPC systems, as shown in Fig. 4. Weak scaling with respect to  $N_\Sigma$  shows the same favorable performance up to tens of thousands of GPUs as the GPP model, since the parallelization scheme is identical. Weak scaling

**Figure 4: Strong scaling of the GW-FF.** Results are reported excluding I/O unless noted specifically.

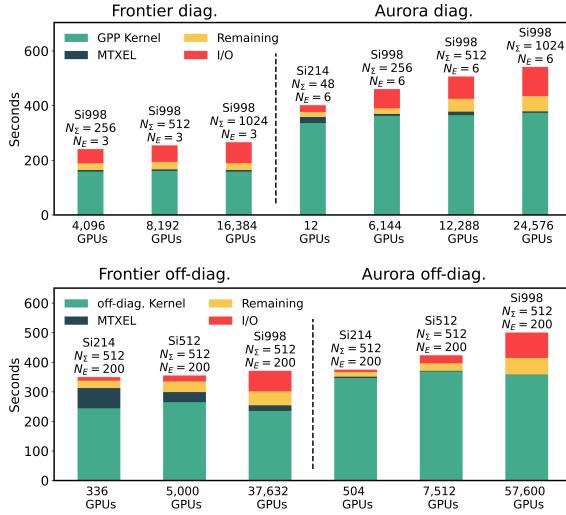


Figure 5: Weak scaling of the GW-GPP Sigma.

with the compute pool size is less favorable due to communication, but the abundant parallelism available over  $N_\Sigma$  alleviates this issue in large-scale calculations.

### 7.3 Optimized GPP Kernel Performance

Using the hardware-optimized programming models, we performed systematic scaling calculations to demonstrate the performance of the GPP implementations. We report results on Frontier and Aurora, where the kernel implementations are optimized with HIP and SYCL, respectively.

Fig. 5 shows weak scaling on both Frontier and Aurora with varying system sizes. The problem size is scaled based on Eqs. 7 and 8. The dominant computational step, the GPP *diag.* and *off-diag.* kernels, construct the GW self-energy operator and its matrix elements using the GPP model. We observe excellent weak scaling and time to solution up to tens of thousands GPUs on both Frontier and Aurora systems.

Fig. 6 shows strong scaling of GPP *diag.* and *off-diag.* calculations on Frontier and Aurora using Si998 and Si2742 systems. Our results show excellent strong scaling excluding I/O, up to the full machine of Frontier with 9,408 nodes, and up to 90.4% of the full machine of Aurora with 9,600 nodes. The GPP *diag.* kernel shows excellent scalability across a small to large number of GPUs due to its memory-efficient formalism and implementation. The GPP *off-diag.* kernel shows the best kernel performance with large-scale calculations, by leveraging the ZGEMM library, and particularly the matrix cores of AMD GPUs on Frontier. Moreover, we have explored the Tensile library on Frontier, which optimizes ZGEMM performance for specific matrix sizes in the GPP *off-diag.* kernel. Our observations show that for the large application case (Si998 with  $N_\Sigma = 512$ ), the default ZGEMM library call already reaches the best-achievable performance compared to the one with Tensile optimization, whereas for moderate problem size (Si998 with  $N_\Sigma = 384$ ), the Tensile optimization can boost the overall kernel performance by ~ 10%, to the similar best achievable level.

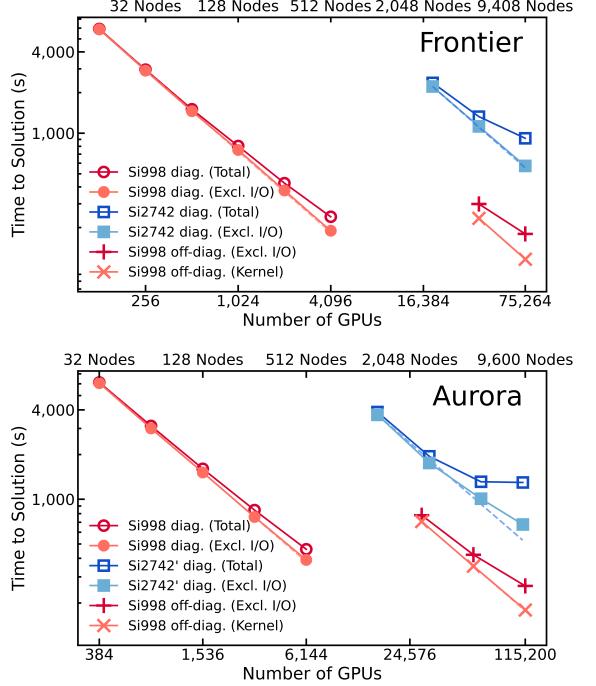


Figure 6: Strong scaling of the GW-GPP Sigma.

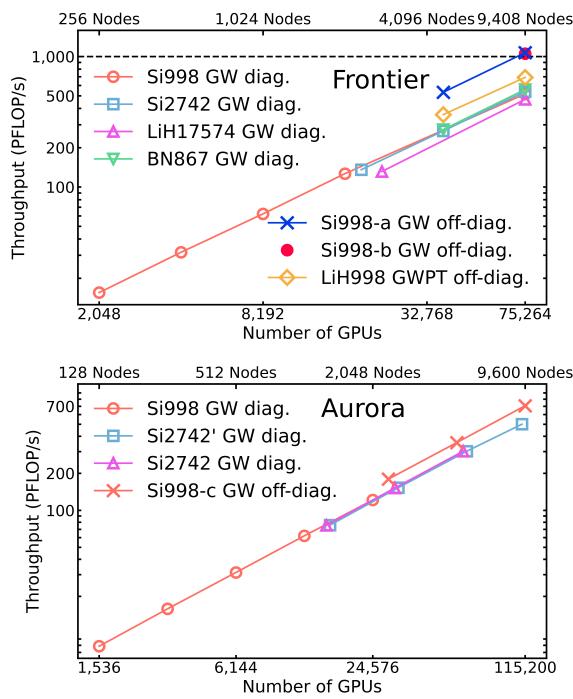
### 7.4 Full System Runs and Peak Performance

Fig. 7 shows the throughput performance of the Sigma-GPP *diag.* and *off-diag.* kernels (the most computationally intensive kernels) on Frontier and Aurora. Here, we demonstrate applications over a wide range of systems, including solid-state defects of Si and LiH, and defects in BN moiré superlattices.

On both Frontier and Aurora, the GPP *diag.* kernel consistently reaches ~ 500 PetaFLOP/s at (nearly) the full machine scale with the hardware-optimized implementations. In particular, at full machine of Frontier (9408 nodes, or 75,264 AMD GPUs), we achieved 558.3 PetaFLOP/s in double precision, corresponding to 31.04% of the theoretical peak; and with 87.5% of the full Aurora (9,296 nodes, or 111,552 Intel GPUs), we achieved 500.97 PetaFLOPs in double precision, corresponding to 39.39% of the attainable peak.

The GPP *off-diag.* kernel at (nearly) the full machine scale achieves significantly higher double-precision throughput performance: **1.069 ExaFLOP/s on 9,408 Frontier nodes** (75,264 AMD GPUs, full machine), corresponding to **59.45% of the theoretical peak**; and **707.52 PetaFLOP/s on 9,600 Aurora nodes** (115,200 Intel GPUs, 90.4% of full machine), corresponding to **48.79% of attainable peak**. These performance gains are directly related to the reformulation of the most computationally heavy contractions into ZGEMM operations, benefiting the off-diagonal calculations. This recasting substantially increases arithmetic intensity at the cost of additional memory consumption. The resulting trade-off between memory footprint and compute efficiency proves highly favorable when a large number of  $N_\Sigma$  is calculated along with a fine grid of  $N_E$ .

In Table 5, we list some of the best achieved throughput results. The excellent scalability up to over 1.0 ExaFLOP/s and high peak



**Figure 7: Throughput of GPP kernel achieved on Frontier and Aurora.** The dashed line (upper panel) marks 1.0 ExaFLOP/s performance. Si998 demonstrates multiple configurations: Si998-a ( $N_E = 200, N_b = 28, 224$ ), Si998-b ( $N_E = 512, N_b = 28, 224$ ), and Si998-c ( $N_E = 200, N_b = 28, 800$ ).

percentage ( $\sim 50 - 60\%$ ) of the theoretical or attainable peak in *double precision* have clearly established the effectiveness and generalizability of our kernel optimizations across major GPU architectures by different vendors. Note that the performance of the whole application improves with the desired accuracy. For instance, in the Si998-b case (Table 5), computing  $N_E = 512$  frequencies yields a kernel performance of 1.051 ExaFLOP/s, along with excellent whole application performance of over 800 PetaFLOP/s excluding I/O, and over 500 PetaFLOP/s including I/O. Our work not only demonstrates flexibility of directive-based open programming models, but also highlights the transferable knowledge in hardware-optimized implementations for achieving high peak performance.

## 8 Implications

This work presents several key innovations of BerkeleyGW benchmarked on the exascale Aurora and Frontier supercomputers. On the HPC side, we have successfully enabled true portability using both directive-based open standards and hardware-optimized models, achieving high performance on AMD, Intel, and NVIDIA GPU architectures. Specifically, we have scaled the GW calculations to (nearly) the full machine of Frontier and Aurora, obtained  $\sim 700$  to over 1,000 FP64 PetaFLOP/s kernel performance ( $\sim 50 - 60\%$  of the theoretical/attainable peak). On the methodology aspect, we have successfully enabled large-scale and highly efficient GWPT calculations for correlated electron-phonon coupling, along with scalable and portable GW calculations using both GPP and full-frequency

**Table 5: Best throughput performance on Frontier (F) and Aurora (A).**

Optimized diagonal GPP Kernel					
System	Calculation	# of Nodes	Time (s)	Perf. (PFLOP/s)	% of Peak
BN867 GW	Kernel (F)	9,408	188.45	558.32	31.04
Si2742 GW	Kernel (F)	9,408	445.02	534.80	29.73
Si2742' GW	Kernel (A)	9,296	475.58	500.97	39.39
LiH998 GWPT	Kernel (F)	9,408	92.91	479.27	26.64
Optimized off-diagonal GPP Kernel					
System	Calculation	# of Nodes	Time (s)	Perf. (PFLOP/s)	% of Peak
Si998-a GW	Kernel (F)	9,408	116.4	<b>1,069.36</b>	<b>59.45</b>
Si998-b GW	Kernel (F)	9,408	303.13	1,051.21	58.44
Si998-b GW	Tot. excl. I/O (F)	9,408	390.75	815.49	45.33
Si998-b GW	Tot. incl. I/O (F)	9,408	604.96	526.73	29.28
Si998-c GW	Kernel (A)	9,600	179.52	<b>707.52</b>	<b>48.79</b>
LiH998 GWPT	Kernel (F)	9,408	30.13	691.10	38.42

schemes. The mixed stochastic-deterministic pseudobands method can further help to reduce the scaling of the GW methods. These versatile functionalities place BerkeleyGW at the forefront of first-principles many-body perturbation theory research.

With these advancements, BerkeleyGW is now able to systematically compute at scale the quasiparticle excited-state properties and the electron-phonon coupling phenomena, which are critical to transport, optical absorption, and decoherence and lifetimes of quantum states (e.g. in qubits and quantum emitters), for complex materials structures with  $O(10^3) - O(10^4)$  atoms. The portable and efficient utilization of resources, along with the capability to describe heterogeneous systems and quantum many-body interactions, sets up a new frontier for studying increasingly complex quantum materials, phenomena, and devices in the exascale age.

## Acknowledgments

This work was primarily supported by C2SEPEM at LBNL, which is funded by the US Department of Energy (DOE) under contract No. DEAC02-05CH11231, which supported the core developments of advanced theories, methods, algorithms and exascale optimizations in BerkeleyGW (B.Z., A.R.A., Y.S., S.G.L., J.R.D., F.H.J., Z.L., and M.D.B.). B.Z., C.E.H. and Z.L. acknowledge US NSF (DMR-2440763) for partial support for numerical calculations. A.R.A. acknowledges the US DOE Center NPNEQ (DE-AC52-07NA27344) for Parabands development. Y.S. acknowledges NSF (DMR-2238328) for Chebyshev-Jackson algorithm development. D.V.F. and D.W. acknowledge US DOE BEAST project (DE-SC0022247) for the GW-FF offloaded implementation. An award of computer time was provided by the INCITE program (*MAT280, ElectronPhonon*) and by ALCC (*CPH169, PredictPhotocatal*) program. This research used resources of both the ALCF and OLCF which are DOE SC User Facilities supported under contracts DE-AC02-06CH11357 and DE-AC05-00OR22725. Computational resources were also provided by DOE User Facility NERSC (DE-AC0205CH11231), and by TACC (DMR22042), which is supported by NSF (OAC-1818253). We thank Dr. Ye Luo at ALCF for helpful discussions.

## References

- [1] Sambit Das, Bikash Kanungo, Vishal Subramanian, Gourab Panigrahi, Phani Mottamarri, 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* (Denver, CO, USA) (SC '23). Association for Computing Machinery, New York, NY, USA, Article 1, 12 pages. doi:10.1145/3581784.3627037
- [2] Mark S Hybertsen and Steven G Louie. 1986. Electron correlation in semiconductors and insulators: Band gaps and quasiparticle energies. *Phys. Rev. B* 34, 8 (1986), 5390.
- [3] Mark Hybertsen and Steven G Louie. 1985. First-principles theory of quasiparticles: calculation of band gaps in semiconductors and insulators. *Phys. Rev. Lett.* 55, 13 (1985), 1418.
- [4] Lars Hedin. 1965. New method for calculating the one-particle green's function with application to the electron-gas problem. *Phys. Rev.* 139, 3A (Aug. 1965), A796–A823.
- [5] Giovanni Onida, Lucia Reining, and Angel Rubio. 2002. Electronic excitations: density-functional versus many-body Green's-function approaches. *Rev. Mod. Phys.* 74, 2 (2002), 601.
- [6] Zhenglu Li, Gabriel Antonius, Meng Wu, Felipe H da Jornada, and Steven G Louie. 2019. Electron-phonon coupling from ab initio linear-response theory within the GW method: Correlation-enhanced interactions and superconductivity in  $\text{Ba}_{1-x}\text{K}_x\text{BiO}_3$ . *Phys. Rev. Lett.* 122, 18 (May 2019), 186402.
- [7] Zhenglu Li, Meng Wu, Yang-Hao Chan, and Steven G Louie. 2021. Unmasking the origin of kinks in the photoemission spectra of cuprate superconductors. *Phys. Rev. Lett.* 126, 14 (2021), 146401.
- [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 *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (SC '20). IEEE Press, Article 4, 11 pages.
- [9] Paolo Umari, Geoffrey Stenuit, and Stefano Baroni. 2010. GW quasiparticle spectra from occupied states only. *Phys. Rev. B* 81, 11 (2010), 115104.
- [10] Feliciano Giustino, Marvin L Cohen, and Steven G Louie. 2010. GW method with the self-consistent Sternheimer equation. *Phys. Rev. B* 81, 11 (2010), 115105.
- [11] Marco Govoni and Giulia Galli. 2015. Large scale GW calculations. *J. Chem. Theory Comput.* 11, 6 (2015), 2680–2696.
- [12] Huy-Viet Nguyen, T Anh Pham, Dario Rocca, and Giulia Galli. 2012. Improving accuracy and efficiency of calculations of photoemission spectra within the many-body perturbation theory. *Phys. Rev. B* 85, 8 (2012), 081101.
- [13] T Anh Pham, Huy-Viet Nguyen, Dario Rocca, and Giulia Galli. 2013. GW calculations using the spectral decomposition of the dielectric matrix: Verification, validation, and comparison of methods. *Phys. Rev. B* 87, 15 (2013), 155148.
- [14] Aaron R Altman, Sudipta Kundu, and Felipe H da Jornada. 2024. Mixed stochastic-deterministic approach for many-body perturbation theory calculations. *Phys. Rev. Lett.* 132, 8 (Feb. 2024), 086401.
- [15] Daniel Neuhauser, Yi Gao, Christopher Arntsen, Cyrus Karshenas, Eran Rabani, and Roi Baer. 2014. Breaking the Theoretical Scaling Limit for Predicting Quasiparticle Energies: The Stochastic GW Approach. *Phys. Rev. Lett.* 113 (Aug 2014), 076402. Issue 7. doi:10.1103/PhysRevLett.113.076402
- [16] Martin M Rieger, L Steinbeck, I D White, H N Rojas, and R W Godby. 1999. The GW space-time method for the self-energy of large systems. *Comput. Phys. Commun.* 117, 3 (March 1999), 211–228.
- [17] Peitao Liu, Merzuk Kaltak, Ji ří Klimeš, and Georg Kresse. 2016. Cubic scaling GW: Towards fast quasiparticle calculations. *Phys. Rev. B* 94 (Oct 2016), 165109. Issue 16. doi:10.1103/PhysRevB.94.165109
- [18] Jan Wilhelm, Dorothea Golze, Leopold Talirz, Jürg Hutter, and Carlo A. Pignedoli. 2018. Toward GW Calculations on Thousands of Atoms. *J. Phys. Chem. Lett.* 9, 2 (2018), 306–312. doi:10.1021/acs.jpclett.7b02740 PMID: 29280376.
- [19] Minjung Kim, Glenn J. Martyna, and Sohrab Ismail-Beigi. 2020. Complex-time shredded propagator method for large-scale GW calculations. *Phys. Rev. B* 101 (Jan 2020), 035139. Issue 3. doi:10.1103/PhysRevB.101.035139
- [20] Chia-Nan Yeh and Miguel A Morales. 2024. Low-scaling algorithms for GW and constrained random phase approximation using symmetry-adapted interpolative separable density fitting. *J. Chem. Theory Comput.* 20, 8 (April 2024), 3184–3198.
- [21] Jack Deslippe et al. 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.
- [22] Victor Wen-zhe Yu and Marco Govoni. 2022. GPU Acceleration of Large-Scale Full-Frequency GW Calculations. *J. Chem. Theory and Comput.* 18, 8 (2022), 4690–4707. doi:10.1021/acs.jctc.2c00241 PMID: 35913080.
- [23] Paolo Giannozzi et al. 2009. QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials. *J. Phys. Cond. Matter* 21, 39 (2009), 395502.
- [24] Xavier Gonze et al. 2009. ABINIT: First-principles approach to material and nanosystem properties. *Comput. Phys. Commun.* 180, 12 (2009), 2582–2615.
- [25] Andrea Marini et al. 2009. Yambo: an ab initio tool for excited state calculations. *Comput. Phys. Commun.* 180, 8 (2009), 1392–1403.
- [26] Martin Schlipf et al. 2020. SternheimerGW: a program for calculating GW quasiparticle band structures and spectral functions without unoccupied states. *Comput. Phys. Commun.* 247 (2020), 106856.
- [27] Georg Kresse and Jurgen Hafner. 1993. Ab initio molecular dynamics for liquid metals. *Phys. Rev. B* 47, 1 (1993), 558.
- [28] Georg Kresse and Jurgen Hafner. 1994. Ab initio molecular-dynamics simulation of the liquid-metal-amorphous-semiconductor transition in germanium. *Phys. Rev. B* 49, 20 (1994), 14251.
- [29] Volker Blum et al. 2009. Ab initio molecular simulations with numeric atom-centered orbitals. *Comput. Phys. Commun.* 180, 11 (2009), 2175 – 2196. doi:10.1016/j.cpc.2009.06.022
- [30] Denis Jacquemin, Ivan Duchemin, and Xavier Blase. 2015. Benchmarking the Bethe-Salpeter Formalism on a Standard Organic Molecular Set. *J. Chem. Theory Comput.* 11, 7 (2015), 3290–3304. doi:10.1021/acs.jctc.5b00304 PMID: 26207104.
- [31] Fabien Brunoval et al. 2016. molgw 1: Many-body perturbation theory software for atoms, molecules, and clusters. *Comput. Phys. Commun.* 208 (2016), 149 – 161. doi:10.1016/j.cpc.2016.06.019
- [32] Andris Gulans et al. 2014. Exciting: a full-potential all-electron package implementing density-functional theory and many-body perturbation theory. *J. Phys. Condens. Matter* 26, 36 (2014), 363202.
- [33] ELK-software -. ELK: an all-electron full-potential linearised augmented-plane wave (LAPW) code. <http://elk.sourceforge.net/>.
- [34] Thomas D. Kühne, Marcella Iannuzzi, Mauro Del Ben, Vladimir V. Rybkin, et al. 2020. CP2K: An electronic structure and molecular dynamics software package - Quickstep: Efficient and accurate electronic structure calculations. *J. Chem. Phys.* 152, 19 (2020), 194103. doi:10.1063/5.0007045
- [35] Jacob Brooks, Guorong Weng, Stephanie Taylor, and Vojtech Vlcek. 2020. Stochastic many-body perturbation theory for Moiré states in twisted bilayer phosphorene. *J. Phys. Condens. Matter* 32, 23 (2020), 234001.
- [36] Wentiao Wu et al. 2024. Enabling 13K-Atom Excited-State GW Calculations via Low-Rank Approximations and HPC on the New Sunway Supercomputer. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis* (Atlanta, GA, USA) (SC '24). IEEE Press, Article 61, 14 pages. doi:10.1109/SC41406.2024.000067
- [37] Mauro Del Ben, Felipe H. da Jornada, Gabriel Antonius, Tonatiuh Rangel, Steven G. Louie, Jack Deslippe, and Andrew Canning. 2019. Static subspace approximation for the evaluation of  $G_0W_0$  quasiparticle energies within a sum-over-bands approach. *Phys. Rev. B* 99 (Mar 2019), 125128. Issue 12. doi:10.1103/PhysRevB.99.125128
- [38] Hugh F. Wilson, Françoise Gygi, and Giulia Galli. 2008. Efficient iterative method for calculations of dielectric matrices. *Phys. Rev. B* 78 (2008), 113303.
- [39] Hugh F. Wilson, Deyu Lu, Françoise Gygi, and Giulia Galli. 2009. Iterative calculations of dielectric eigenvalue spectra. *Phys. Rev. B* 79 (2009), 245106.
- [40] Jacob M. Clary et al. 0. Static Subspace Approximation for Random Phase Approximation Correlation Energies: Applications to Materials for Catalysis and Electrochemistry. *Journal of Chemical Theory and Computation* 0, 0 (0), null. doi:10.1021/acs.jctc.4c01276
- [41] Daniel Weinberg et al. 2024. Static Subspace Approximation for Random Phase Approximation Correlation Energies: Implementation and Performance. *Journal of Chemical Theory and Computation* 20, 18 (2024), 8237–8246. doi:10.1021/acs.jctc.4c00807
- [42] Alexander Weiß, Gerhard Wellein, Andreas Alvermann, and Holger Fehske. 2006. The kernel polynomial method. *Rev. Mod. Phys.* 78, 1 (2006), 275–306.
- [43] Grady Schofield, James R Chelikowsky, and Yousef Saad. 2012. A spectrum slicing method for the Kohn-Sham problem. *Comput. Phys. Commun.* 183, 3 (2012), 497–505.
- [44] Mauro Del Ben, Felipe H. da Jornada, Andrew Canning, Nathan Wichmann, Karthik Raman, Ruchira Sasanka, Chao Yang, Steven G. Louie, and Jack Deslippe. 2019. Large-scale GW calculations on pre-exascale HPC systems. *Comput. Phys. Commun.* 235 (2019), 187 – 195.
- [45] Barbara Chapman et al. 2021. Outcomes of OpenMP Hackathon: OpenMP Application Experiences with the Offloading Model (Part II). In *OpenMP: Enabling Massive Node-Level Parallelism*, Simon McIntosh-Smith, Bronis R. de Supinski, and Jannis Klinkenberg (Eds.). Springer International Publishing, Cham, 81–95.
- [46] Charlene Yang. 2020. 8 Steps to 3.7 TFLOP/s on NVIDIA V100 GPU: Roofline Analysis and Other Tricks. arXiv:2008.11326 [cs.DC] <https://arxiv.org/abs/2008.11326>
- [47] 2024. BerkeleyGW. [Computer Software] <https://berkeleygw.org/download/>. doi:10.11578/dc.20240320.1
- [48] Oscar Antepara et al. 2023. Performance Portability Evaluation of Blocked Stencil Computations on GPUs. In *Proceedings of the SC '23 Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis* (Denver, CO, USA) (SC-W '23). Association for Computing Machinery, New York, NY, USA, 1007–1018. doi:10.1145/3624062.3624177
- [49] BerkeleyGW-N10 2024. *Optical Properties of Materials Workflow*. <https://gitlab.com/NERSC/N10-benchmarks/berkeleygw-workflow>