

<https://doi.org/10.1038/s41524-024-01422-3>

# High-speed and low-power molecular dynamics processing unit (MDPU) with ab initio accuracy



Pinghui Mo<sup>1,2,16</sup>, Yujia Zhang<sup>2,16</sup>, Zhuoying Zhao<sup>2,16</sup>, Hanhan Sun<sup>3,16</sup>, Junhua Li<sup>2</sup>, Dawei Guan<sup>2</sup>, Xi Ding<sup>2</sup>, Xin Zhang<sup>2</sup>, Bo Chen<sup>4</sup>, Mengchao Shi<sup>2,5</sup>, Duo Zhang<sup>5,6,7</sup>, Denghui Lu<sup>8</sup>, Yinan Wang<sup>6,9</sup>, Jianxing Huang<sup>10</sup>, Fei Liu<sup>11</sup>, Xinyu Li<sup>6</sup>, Mohan Chen<sup>12</sup>, Jun Cheng<sup>10</sup>, Bin Liang<sup>3</sup>, Weinan E<sup>6,9,12</sup>, Jiayu Dai<sup>4</sup>✉, Linfeng Zhang<sup>5,6</sup>✉, Han Wang<sup>13</sup>✉ & Jie Liu<sup>1,6,14,15</sup>✉

Molecular dynamics (MD) is an indispensable atomistic-scale computational tool widely-used in various disciplines. In the past decades, nearly all ab initio MD and machine-learning MD have been based on the general-purpose central/graphics processing units (CPU/GPU), which are well-known to suffer from their intrinsic “memory wall” and “power wall” bottlenecks. Consequently, nowadays MD calculations with ab initio accuracy are extremely time-consuming and power-consuming, imposing serious restrictions on the MD simulation size and duration. To solve this problem, here we propose a special-purpose MD processing unit (MDPU), which could reduce MD time and power consumption by about  $10^3$  times ( $10^9$  times) compared to state-of-the-art machine-learning MD (ab initio MD) based on CPU/GPU, while keeping ab initio accuracy. With significantly-enhanced performance, the proposed MDPU may pave a way for the accurate atomistic-scale analysis of large-size and/or long-duration problems which were impossible/impractical to compute before.

As a cornerstone of atomistic-scale analysis<sup>1–3</sup>, molecular dynamics (MD) is playing increasingly important roles in various disciplines, like physics<sup>4</sup>, chemistry<sup>5</sup>, biology<sup>6,7</sup>, materials<sup>8</sup>, nanotechnology<sup>9</sup>, medicine<sup>10</sup>, geology<sup>11</sup>, semiconductor<sup>12</sup>, cosmology<sup>13</sup> and so on. As a result, the number of MD-related publications grows as an exponential function of time (e.g., Fig. 1 of ref.<sup>14</sup> and Supplementary Fig. 1). To reliably capture various atomistic-scale phenomena, energy prediction error ( $\epsilon_e$  in meV/atom) of MD should be around or below 1.0 kcal/mol (i.e., 43.4 meV/atom), which is well-known as “quantum mechanical” or “ab initio” accuracy as defined by John A. Pople<sup>15</sup>.

From the software/algorithm point of view, the existing MD methods consist of three major categories: (1) classical MD (CMD), which uses artificially-crafted analytic expressions to describe the interatomic potential (IAP)<sup>16–20</sup>; (2) ab initio MD (AIMD), which uses ab initio calculations like

density functional theory (DFT) to obtain the IAP<sup>21–26</sup>, and (3) machine-learning MD (MLMD) which uses machine learning models to compute the IAP<sup>27–33</sup>. Amongst these three categories, only AIMD and MLMD could achieve ab initio accuracy. In contrast, the IAP prediction error of CMD could be as high as 10.0 kcal/mol<sup>34–37</sup>.

From the hardware/computer point of view, although there are special-purpose hardware/computer (notably, the Anton series<sup>38–40</sup> and the MDGRAPE series<sup>41–43</sup>, for example) for CMD, the existing MD simulations with ab initio accuracy (i.e., AIMD and MLMD) in the literature are predominantly executed by using the general-purpose central/graphics processing units (CPU/GPU) which are based on the von Neumann (vN) architecture<sup>44</sup>. It is well known that the vN CPU/GPU suffer from “memory wall” and “power wall” bottlenecks — most (e.g., >90%) of the total

<sup>1</sup>College of Integrated Circuits, Hunan University, Changsha, 410082, P.R. China. <sup>2</sup>College of Electrical and Information Engineering, Hunan University, Changsha, 410082, P.R. China. <sup>3</sup>College of Computer, National University of Defense Technology, Changsha, 410073, P.R. China. <sup>4</sup>Department of Physics, National University of Defense Technology, Changsha, 410073, P.R. China. <sup>5</sup>DP Technology, 100080 Beijing, P.R. China. <sup>6</sup>AI for Science Institute, 100080 Beijing, P.R. China.

<sup>7</sup>Academy for Advanced Interdisciplinary Studies, Peking University, 100871 Beijing, P. R. China. <sup>8</sup>HEDPS, CAPT, College of Engineering, Peking University, 100871 Beijing, P.R. China. <sup>9</sup>School of Mathematical Science, Peking University, 100871 Beijing, P.R. China. <sup>10</sup>State Key Laboratory of Physical Chemistry of Solid Surfaces, iChEM, College of Chemistry and Chemical Engineering, Xiamen University, Xiamen, 361005, P.R. China. <sup>11</sup>School of Integrated Circuits, Peking University, 100871 Beijing, P.R. China. <sup>12</sup>Center for Machine Learning Research, Peking University, 100871 Beijing, P.R. China. <sup>13</sup>Institute of Applied Physics and Computational Mathematics, 100088 Beijing, P.R. China. <sup>14</sup>Greater Bay Area Institute for Innovation, Hunan University, Guangzhou, 511300, P.R. China.

<sup>15</sup>Department of Electrical and Computer Engineering, University of Washington, Seattle, WA, 98195, USA. <sup>16</sup>These authors contributed equally: Pinghui Mo, Yujia Zhang, Zhuoying Zhao, Hanhan Sun. ✉e-mail: [jydai@nudt.edu.cn](mailto:jydai@nudt.edu.cn); [zhanglf@dp.tech](mailto:zhanglf@dp.tech); [wang\\_han@iapcm.ac.cn](mailto:wang_han@iapcm.ac.cn); [jie\\_liu@hnu.edu.cn](mailto:jie_liu@hnu.edu.cn)

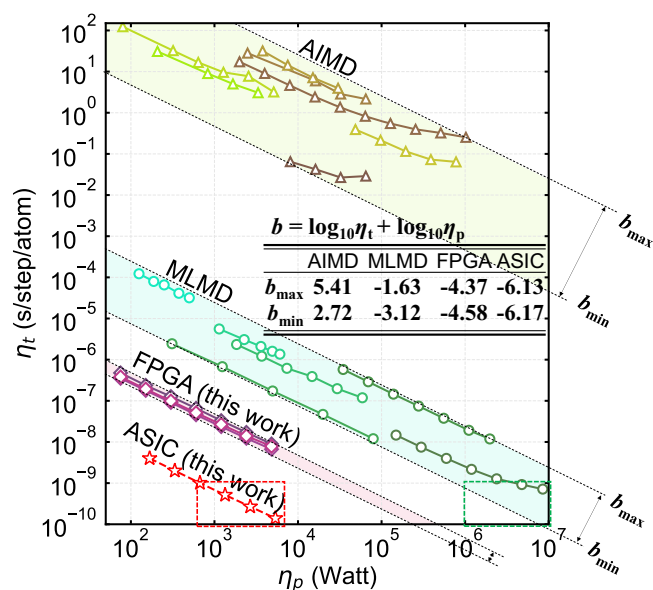


**Fig. 1 | Efficiency comparison of AIMD, MLMD, and proposed methods.** Time ( $\eta_t$ ) and power-consumption ( $\eta_p$ ) of state-of-the-art AIMD, MLMD, and the proposed MD method based on FPGA and ASIC. The equation “ $b = \log_{10}\eta_t + \log_{10}\eta_p$ ” is used to fit the data of AIMD, MLMD, FPGA, and ASIC, to obtain values of parameter  $b$ . Here,  $b_{\min}/b_{\max}$  denotes the approximate lower/upper boundary of each method. To achieve  $\eta_t = 10^{-10}$ – $10^{-9}$  s/step/atom,  $\eta_p$  is reduced from  $10^6$ – $10^7$  W (green box<sup>49,71</sup>) to about  $10^3$  W (red box).

calculation time and power is actually used to frequently shuttle data, and only a very small proportion of time and power is spent to perform the functional arithmetic/logic operations<sup>45,46</sup>. This leads to high power-consumption ( $\eta_p$  in Watt) and high time-consumption ( $\eta_t$  in second/step/atom), i.e., low MD speed<sup>47,48</sup>. To remedy low MD speed, parallelization using many CPUs/GPUs (e.g.,  $10^6$ – $10^7$  CPU cores and  $10^4$ – $10^5$  GPUs) becomes necessary, further increasing  $\eta_p$  (e.g.,  $\eta_p$  can reach about  $10^3$  MW as shown in Fig. 1)<sup>49–53</sup>. The high  $\eta_t$  and  $\eta_p$  values imposes serious restrictions on state-of-the-art MD simulation size/duration and affordability.

Aiming at mitigating these restrictions, here we propose a special-purpose MD processing unit (MDPU), which could reduce  $\eta_t$  and  $\eta_p$  by about  $10^3$  times ( $10^9$  times) compared to state-of-the-art MLMD (AIMD), while keeping ab initio accuracy (Sections “Results” and “Discussion”). Such dramatic MD performance improvement is accomplished via co-design and in-depth co-optimization of algorithm, hardware, and software (Section “Methods”)<sup>54</sup>. First, heavy-duty calculations in MD are replaced by using light-weight equivalent arithmetic/logic operations, by leveraging cutting-edge machine-learning algorithms, after extensive and carefully-crafted revisions (Section “Algorithm”)<sup>30,55</sup>. Second, a powerful computing-in-memory (CIM) hardware calculation engine is implemented, in order to mitigate the “memory wall” and “power wall” bottlenecks, based on the commodity-level or commercially-available hardware technologies like field programmable gate array (FPGA) and application specific integrated circuit (ASIC) (Section “Hardware”)<sup>56,57</sup>. Third, the software is purposefully rewritten into highly-efficient pipelines according to FPGA/ASIC characteristics (Section “Software”)<sup>54</sup>. More implementation details are available in Section “Methods”.

## Results

The proposed MD method is applied to six systems, including elementary systems (Au<sup>58</sup>, Cu<sup>59</sup>, Mg<sup>60</sup>), binary systems (H<sub>2</sub>O<sup>61,62</sup>, GeTe<sup>63,64</sup>), and quaternary system (Li<sub>10</sub>GeP<sub>2</sub>S<sub>12</sub><sup>65</sup>), to quantitatively analyze MD performance metrics like accuracy (Section “Accuracy”), speed and power (Section “Speed and Power”), cost (Section “Cost”), and system size (Section “Size”). We purposefully choose these six

systems, which are known to be challenging to achieve high accuracy, as they are either energetically subtle or under extreme external conditions<sup>58–65</sup>.

## Accuracy

As shown in Fig. 2, the proposed MD method has decent accuracy. Compared to the ab initio DFT results, the root-mean-square error (RMSE) of its potential energy surface (PES) prediction results is  $\epsilon_e = 85.35, 1.84, 2.47, 7.62, 2.95, 1.66$  meV/atom; and the RMSE of its atomic force prediction results is  $\epsilon_f = 173.20, 16.44, 13.91, 148.24, 110.69, 59.67$  meV/Å, in Au, Cu, Mg, GeTe, H<sub>2</sub>O, and Li<sub>10</sub>GeP<sub>2</sub>S<sub>12</sub>, respectively. Notably,  $\epsilon_e$  here ( $10^{-3}$ – $10^{-1}$  eV/atom) is 2–3 orders of magnitude smaller than typical interatomic bond energy ( $10^0$ – $10^1$  eV/atom<sup>66</sup>); and  $\epsilon_f$  here (about  $10^1$ – $10^2$  meV/Å) is very close to the typical default convergence threshold force value of lattice relaxation in various DFT tools (e.g., 40.0 meV/Å in SIESTA<sup>25</sup>; 25.7 meV/Å in Quantum Espresso<sup>22</sup>; and 23.1 meV/Å in CP2K<sup>67</sup>).

It worth noting that such small  $\epsilon_e$  and  $\epsilon_f$  indicates very high accuracy of the proposed MD method, since these six systems involve repeated chemical bond rupture and formation with extremely complicated and dynamically-evolving high-dimensional PES, imposing serious challenges to keep high calculation accuracy<sup>58–65</sup>. Fortunately, ab initio accuracy of energy and atomic force of the proposed MD method (Fig. 2) lays a solid foundation for reliable prediction of various physical properties. In Fig. 3, these physical properties are computed using the proposed MD method and compared directly against the established MLMD, because MLMD is known to have ab initio accuracy<sup>58–65</sup> and DFT-based AIMD is too time-consuming and power-consuming to obtain these properties (Fig. 1).

Under intensive shock wave compression (e.g.,  $10^0$ – $10^1$  km/s), which leads to extremely high pressure (e.g.,  $10^3$ – $10^4$  GPa) and temperature (e.g.,  $10^4$ – $10^5$  K), the Au physical properties calculated by using the proposed MD method agree well with MLMD results (Fig. 3a and Supplementary Fig. 2)<sup>58</sup>. Under strong (up to 10%) tensile strain, the proposed MD method can accurately predict Cu strain-stress curve and the related properties like the Young’s modulus (Fig. 3(b))<sup>59</sup>. The stacking fault energies of Mg, which is difficult to accurately compute<sup>60</sup>, are calculated using the proposed method on basal {0001}, prismatic {1010}, pyramidal I {1011}, and pyramidal II {1122} planes, and the results agree well with the established MLMD and DFT methods (Fig. 3c and Supplementary Fig. 3)<sup>60</sup>.

The proposed MD method could accurately reproduce the radial distribution function of liquid H<sub>2</sub>O, which has complicated PES and hydrogen bonds (Fig. 3d)<sup>61,62</sup>. It is known that the energy difference between amorphous and crystalline chalcogenide phase change material (PCM) is  $\epsilon_{a-c} \approx 10^2$  meV/atom, and the energy difference between PCM with and without Peierls distortion is  $\epsilon_{\text{Peierls}} \approx 10^1$  meV/atom<sup>68–70</sup>. Using the proposed MD method, we have reliably captured not only the melt-quench-recrystallization processes of PCM GeTe, but also its Peierls distortion as revealed by the angular-limited three-body correlation (ALTBC) results (Fig. 3e)<sup>63,64</sup>. ALTBC indicates the bond length distribution of two Ge–Te bonds in the three-body structure (i.e. Ge–Te–Ge and Te–Ge–Te) with an angle greater than 155°. The PES error here ( $\epsilon_e = 7.62$  meV/atom as shown in Fig. 2e) is much smaller than  $\epsilon_{a-c}$  and  $\epsilon_{\text{Peierls}}$ , enabling MDPU to correctly reproduce the microstructure of the amorphous GeTe. The proposed MD method also accurately predicts the Li-ion diffusion process in the complicated quaternary system Li<sub>10</sub>GeP<sub>2</sub>S<sub>12</sub> (Fig. 3f)<sup>65</sup>.

Our study shows that the accuracy of DPMD simulations on CPU/GPU aligns closely with those on MDPU. Minor variations observed in examples like Cu and Li<sub>10</sub>GeP<sub>2</sub>S<sub>12</sub> fall within the statistical error range and are attributed to natural fluctuations in the simulations, rather than differences in the hardware. The calculation data and codes are open-sourced (Sections “Data Availability” and “Code Availability”). More details are available in Supplementary Section 2.

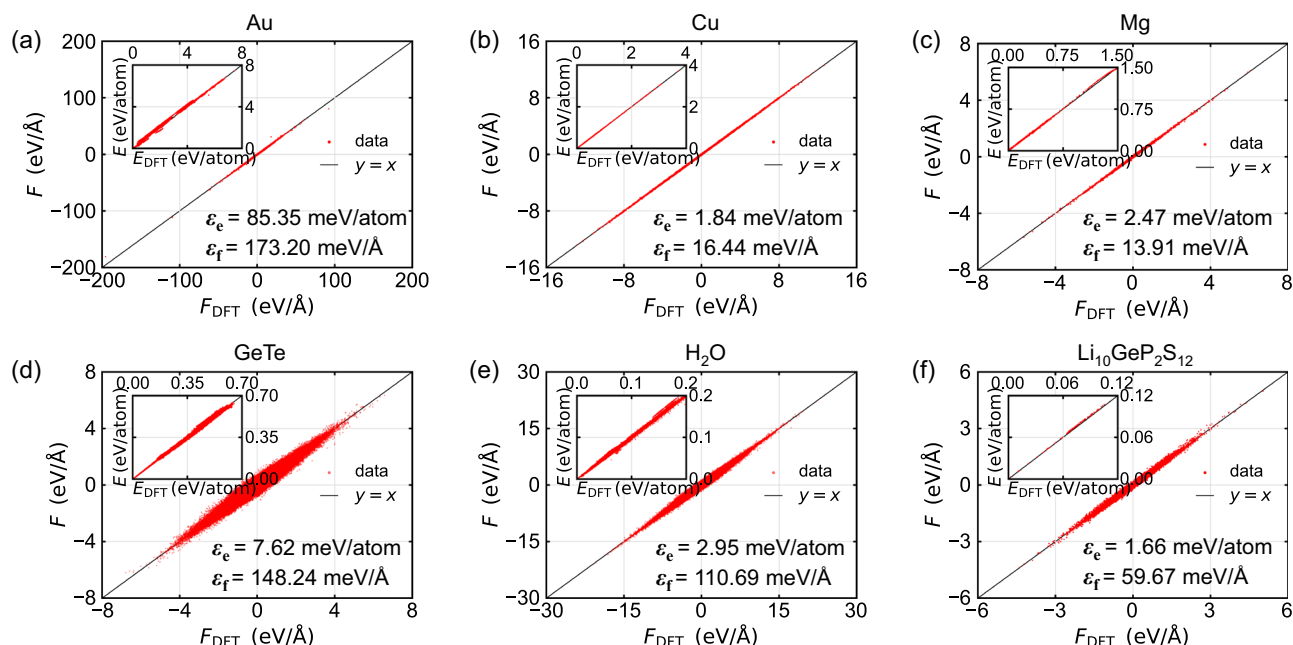


**Fig. 2 | The comparison of predicted energy and force.** Comparison between energy/force predicted by the proposed MD method ( $E/F$ ) and those computed using DFT ( $E_{\text{DFT}}/F_{\text{DFT}}$ ). Here,  $\varepsilon_e/\varepsilon_f$  is the root-mean-square error (RMSE) of  $E/F$  compared against  $E_{\text{DFT}}/F_{\text{DFT}}$ .

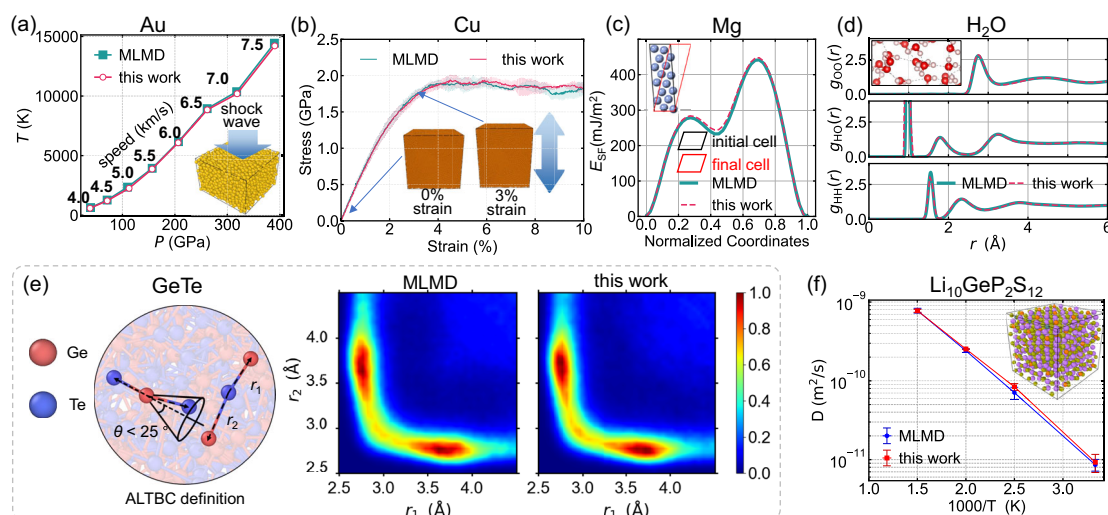


**Fig. 3 | Comparison of results between the MD method proposed in this work and the established MLMD.** **a** Au under intensive shock wave compression<sup>58</sup>; **b** Cu under strong tensile strain<sup>59</sup>; **c** stacking fault energies of Mg on the pyramid I

{1011} plane<sup>60</sup>; **d** radial distribution function (RDF) of liquid H<sub>2</sub>O<sup>63,64</sup>; **e** angular-limited three-body correlation (ALTBC) of amorphous GeTe<sup>61,62</sup>; **f** diffusion coefficient ( $D$ ) of Li in Li<sub>10</sub>GeP<sub>2</sub>S<sub>12</sub><sup>65</sup>.

### Speed and power

As shown in Fig. 1, due to the linear strong and weak scaling of MLMD<sup>49</sup>, we use

$$b = \log_{10} \eta_t + \log_{10} \eta_p \quad (1)$$

to investigate time-efficiency and power-efficiency. Though it does not hold for AIMD, Eq. (1) helps approximately define the upper ( $b_{\text{max}}$ ) and lower ( $b_{\text{min}}$ ) boundaries of each method, facilitating quantitative analysis and comparison. Figure 1 shows that the proposed MD method based on FPGA is over  $10^7$  times more efficient than CPU/GPU-based AIMD ( $b_{\text{min}}^{\text{AIMD}} - b_{\text{max}}^{\text{FPGA}} \approx 7.1$ ,  $b_{\text{max}}^{\text{AIMD}} - b_{\text{max}}^{\text{FPGA}} \approx 9.8$ ), and over  $10^1$  times more efficient than CPU/GPU-based MLMD ( $b_{\text{min}}^{\text{MLMD}} - b_{\text{max}}^{\text{FPGA}} \approx 1.3$ ,  $b_{\text{max}}^{\text{MLMD}} - b_{\text{max}}^{\text{FPGA}} \approx 2.7$ ). It deserves stressing that the FPGA hardware used here has merely  $10^2$  MHz-

level clock frequency and  $10^6$ -level configurable logic blocks (Section “Size”), which are far inferior than the CPU/GPU (GHz-level clock frequency and far more logic devices) used by AIMD/MLMD.

When deployed on ASIC (Section “Size”), whose clock frequency and logic device density are comparable to CPU/GPU, the proposed MD method can be about  $10^9$  times more efficient than CPU/GPU-based AIMD ( $b_{\text{min}}^{\text{AIMD}} - b_{\text{max}}^{\text{ASIC}} \approx 8.9$ ,  $b_{\text{max}}^{\text{AIMD}} - b_{\text{max}}^{\text{ASIC}} \approx 11.5$ ), and about  $10^3$  times more efficient than CPU/GPU-based MLMD ( $b_{\text{min}}^{\text{MLMD}} - b_{\text{max}}^{\text{ASIC}} \approx 3.0$ ,  $b_{\text{max}}^{\text{MLMD}} - b_{\text{max}}^{\text{ASIC}} \approx 6.5$ ). Such significant efficiency increase may have profound impact on future-generation high-performance computation of MD with ab initio accuracy. For example, as the recipient of 2020 Association for Computing Machinery (ACM) Gordon Bell Prize, a research team achieved record-breaking  $\eta_t = 10^{-10}$ – $10^{-9}$  s/step/atom (green dashed box at bottom right of Fig. 1), by using the entire Summit supercomputer, which was

the most powerful supercomputer in the world at that time ( $\eta_p \approx 10^6$ – $10^7$  W which needs more than 4000 gallons of water pump through cooling system every minute to carry away heat; 522 m<sup>2</sup> area;  $10^4$ – $10^5$  NVIDIA V100 GPUs;  $10^5$ – $10^6$  CPU cores)<sup>49,71</sup>. In sharp contrast, to achieve the same  $\eta_t$  ( $10^{-10}$ – $10^{-9}$  s/step/atom), the proposed MD method needs merely about  $\eta_p \approx 10^3$  W power (red dashed box at bottom of Fig. 1) by using 10 to 20 ASICs, which fits into a single desktop-size computer (Section “Methods”).

In Fig. 1, the  $\eta_t$  values of AIMD and MLMD are fetched from the literature (Supplementary Table 1); the  $\eta_t$  results of FPGA and ASIC are obtained in this work (Section “Methods”); the  $\eta_p$  values are approximately estimated using a unified equation  $\eta_p = P_{\text{CPU}} \times N_{\text{CPU}} + P_{\text{GPU}} \times N_{\text{GPU}} + P_{\text{FPGA}} \times N_{\text{FPGA}} + P_{\text{ASIC}} \times N_{\text{ASIC}}$ , where  $N_{\text{CPU}}$ ,  $N_{\text{GPU}}$ ,  $N_{\text{FPGA}}$ , and  $N_{\text{ASIC}}$  denote the number of CPU core(s), GPU(s), FPGA(s), and ASIC(s); and  $P_{\text{CPU}}$ ,  $P_{\text{GPU}}$ ,  $P_{\text{FPGA}}$ , and  $P_{\text{ASIC}}$  denote the thermal design power (TDP) of a CPU core, a GPU, an FPGA, or an ASIC, respectively (Supplementary Tables 2, 3, 4). Here, we use TDP (defined as “the maximum amount of heat generated by a computer chip or component that the cooling system is designed to dissipate under any workload”<sup>72</sup>) instead of measured power, mainly due to two reasons. First, in existing AIMD and MLMD papers, the measured power values of MD calculations are typically not available, but the types of CPU/GPU are disclosed and TDP values are available in the datasheets of CPU/GPU. Second, to economically utilize the hardware resources, the computing devices typically consume most (e.g.,  $\zeta = 65.2$ – $91.4\%$  as shown in Supplementary Table 5) of TDP. This is particularly true for the computationally-intensive tasks like MD with ab initio accuracy.

## Cost

Compared to the widely-used CPU/GPU-based MD, the proposed MD method is more cost-efficient. To accomplish an MD task with  $N_a$  atoms and  $n_s$  timesteps, the total cost is  $C_{\text{total}} = \eta_t \times N_a \times n_s \times C_{\text{rent}}$  US dollars (USD), where  $C_{\text{rent}}$  (in USD/s) is the price to rent computing time. Since  $C_{\text{rent}}$  of the proposed MD method (like FPGA servers in Supplementary Table 6) is comparable or lower than the CPU/GPU counterparts and  $\eta_t$  of the proposed MD method is significantly lower (Fig. 1), the  $C_{\text{total}}$  of the proposed MD method can be much lower than CPU/GPU-based MD. For instance, by using the commodity-level FPGA (Xilinx Virtex UltraScale+ VU9P), we have built a computing server cluster to openly provide MD computing service (available at <https://bohrium.dp.tech>), which can reduce  $C_{\text{total}}$  by over  $10^1$  times compared to the widely-used CPU/GPU service providers.

## Size

Using one CPU/GPU computer node, AIMD (MLMD) can typically calculate about  $10^3$  ( $10^6$ ) atoms, partially due to capacity limitations of the high-performance memory, e.g., the high-bandwidth memory (HBM) on the GPU card (typically  $10^1$ – $10^2$  GB)<sup>73</sup>. In order to mitigate this limitation, the proposed MD method adopted purposefully-designed pipeline (Section “Methods”), which enable us to leverage the larger-capacity (typically  $10^2$ – $10^4$  GB) dynamics random access memory (DRAM) on the mainboard instead. Consequently, the proposed MD method based on one desktop-size computer can calculate over  $10^8$  atoms with ab initio accuracy. Notably, this is comparable to the cutting-edge MLMD, which used the entire Summit supercomputer to accommodate about  $10^8$  atoms<sup>49,71</sup>.

## Discussion

The abovementioned merits (i.e., ab initio accuracy, high speed, low power, low cost, and large size) of the proposed MD method are generally-applicable to the MD calculations of any chemical elements, any composition, and any statistical ensemble. The proposed MD method has linear time complexity, which is desirable for parallelization and scaling. These merits are inherited from the deep potential smooth edition (DeepPot-SE) MLMD algorithm (Section “Algorithm”)<sup>55</sup>. All hardware (e.g., ASIC and FPGA) used in this paper are commodity-level or

commercially-available technologies. The communication networks are realized by using widely-used technologies, like internet wide-area remote-direct-memory-access protocol (iWARP), InfiniBand (IB), and peripheral component interconnect express (PCIe), etc. The parallelization is realized based on the mainstream Beowulf cluster and message passing interface (MPI). So, there is no hardware component in this paper that cannot be purchased commercially on the existing market. The software is programmed using mainstream programming languages like C/C++, and the users can control the MD calculations via popular MD tools like LAMMPS<sup>74</sup>. The proposed MD system is designed in this way, aiming at general-purpose user-friendly MD applications with ab initio accuracy.

It is worth mentioning that MDPU has high accuracy and efficiency at the same time, which makes it have obvious advantages over AIMD when simulating complex systems. Taking the Au system as an example, although AIMD can accurately predict the thermodynamic properties of shocked-compressed Au, it fails to capture the evolution of microstructures under shock compression due to the limit of simulation sizes<sup>58</sup>. In addition, as shown in Fig. 3, six examples in various fields such as special materials, semiconductors, and electrochemistry have extremely complex and dynamically evolving simulation processes, which pose a serious challenge to the accuracy of PES. The results demonstrate that MDPU is not only accurate and efficient but also general enough to be extended to more application fields, effectively addressing practical materials science problems.

It deserves noting that the DeepPot-SE algorithm (Section “Algorithm”)<sup>55</sup> has been applied successfully in many MD simulations from different disciplines<sup>75</sup>, including long-range electrostatic interactions<sup>76</sup> and polarization<sup>77</sup>. By inheriting its merits of general-applicability, the MDPU proposed here is promising to offer next-generation general-purpose MD solutions with high-speed, low-power, and high-accuracy.

Though the time- and power-consumptions ( $\eta_t$  and  $\eta_p$ ) have been significantly reduced by mitigating “memory/power wall” in CIM hardware, the remaining communication between master and slave processing units (MPU and SPU in Section “Methods”) hinders further enhancement of efficiency. It deserves future attention to solve this problem, for example, by integrating MPU and SPU in a larger-area ASIC.

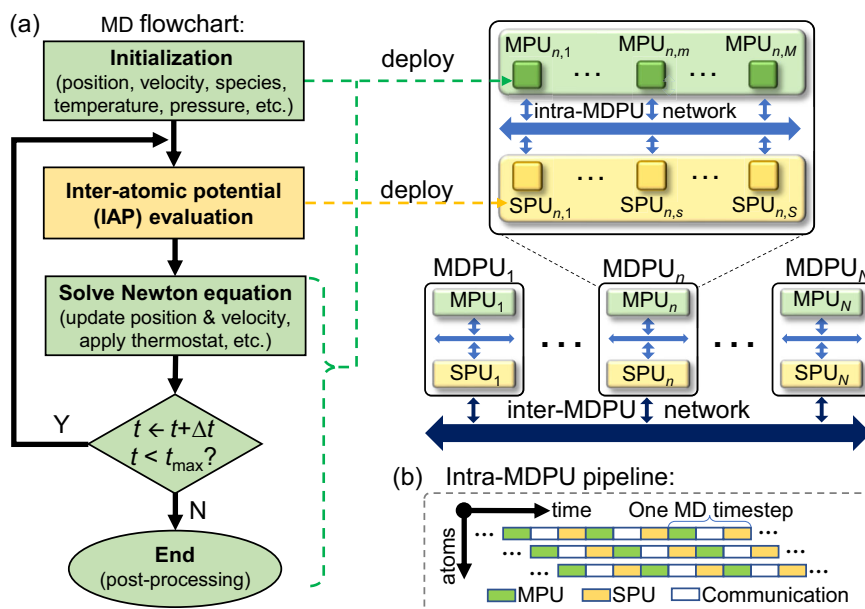
Compared to the seminar and established ASIC-based MD machines (e.g., the Anton series<sup>38–40</sup> and the MDGRAPE series<sup>41–43</sup>) and the FPGA-based MD machines<sup>78–82</sup>, the most significant merit of the proposed MDPU is its ab initio accuracy. Though the existing ASIC-based and FPGA-based MD machines are very efficient, they are all based on CMD, which intrinsically cannot achieve ab initio accuracy. Therefore, in the FPGA/ASIC-based MD machine family, the MDPU proposed here can offer a useful efficient MD modeling solution when high accuracy is required.

It is well known that general-purpose CPUs and GPUs have the advantages of low development costs and faster development cycles, but their computational efficiency is limited by the “memory/power wall” bottleneck. In contrast, special-purpose FPGAs and ASICs are more computationally efficient but come with higher development costs and longer development cycles. FPGAs offer programmability, but their limitations include lower clock frequencies and limited resources, making them more suitable for debugging and testing purposes. ASICs surpass FPGAs in terms of clock frequency and hardware resources, but they require redesign and remanufacturing for each iteration, which adds to the development complexity and time.

## Methods

The results presented above are obtained by following the same analysis procedure (Section “Analysis Procedure”), and the MD simulations are performed on the proposed MD system (Section “System Overview”), which consists of co-designed and deeply co-optimized algorithm (Section 4.0), hardware (Section “Hardware”), and software (Section “Software”).

**Fig. 4 | System overview of the proposed MD method and MD processing unit (MDPU).** MDPU consists of three parts—master processing units (MPUs), slave processing units (SPUs), intra- and inter-MDPU communication networks which link MPUs and SPUs. **a** In the MD calculation flow, the evaluation of interatomic potential (IAP) is deployed on the SPUs, which are marked in yellow; all other calculations are deployed on MPUs, which are marked in green. **b** The pipeline designed to coordinate the operations of MPUs, SPUs, and the MPU-SPU communication.



## Analysis procedure

To simulate each system, the analysis procedure consists of three consecutive steps. In the first step, the emerging active learning tools (e.g., DP-GEN<sup>59</sup>) are used to iteratively call the DFT calculations, to generate training datasets. In this work, we use the training datasets, which were generated in this way and published previously in the literature (Supplementary Table 1). In the second step, the interatomic potential (IAP) model is trained using the emerging deep learning tools, like DeepPot-SE in DeepMD<sup>30,55,73</sup>, which is revised extensively here according to characteristics of FPGA and ASIC hardware (Section “Algorithm”). In the third step, MD simulation is performed based on the proposed MD system (Sections “System Overview”, “Algorithm”, “Hardware”, “Software”), by using the IAP model.

## System overview

As shown in Fig. 4a, the proposed system consists of  $N$  MD processing units (i.e., MDPU <sub>$n$</sub> , where  $n = 1, 2, \dots, N$ ). Inside each MDPU, there are  $M$  master processing units (i.e., MPU <sub>$n,m$</sub> , where  $m = 1, 2, \dots, M$ ) and  $S$  slave processing units (i.e., SPU <sub>$n,s$</sub> , where  $s = 1, 2, \dots, S$ ). MPU can be implemented using CPU/GPU, which is based on the traditional von Neumann architecture (lower time-efficiency and power-efficiency); and SPU can be implemented using FPGA/ASIC, which is based on the emerging CIM non von Neumann architecture (higher time-efficiency and power-efficiency). The MPU and SPU jointly form a heterogeneous architecture parallelization system via intra-MDPU and inter-MDPU communication networks.

The system is designed in this way, in order to realize three goals. First, to boost MD efficiency. As most of MD calculation workload (sometimes over 99%) is to evaluate the interatomic potential (IAP), IAP is completely offloaded to SPU, which is a coprocessor with an internal architecture specially-designed to significantly enhance time and power efficiency of IAP evaluation.

Second, to enable user-friendly and flexible MD calculations. Though SPU is efficient, its FPGA/ASIC implementation is inconvenient to control via programs that users are familiar with. So MPU is implemented based on CPU/GPU, which can be controlled using programs like C/C++/Fortran, facilitating arbitrarily-defined post-processing, meta-dynamics, and MD with all kinds of statistical ensembles, e.g., constant-energy constant-volume (NVE), constant-temperature constant-volume (NVT), constant-temperature constant-pressure (NPT), constant-temperature constant-stress (NST), constant-pressure constant-enthalpy (NPH), etc.

Third, to seamlessly integrate both high-efficiency and high-flexibility. Though MPU helps realizing flexibility, it is slower and may introduce extra

latency besides the much faster SPU. To annihilate the potential MPU-induced  $\eta_t$  increase, a pipeline is used to coordinate the execution of MPU, SPU, and the MPU-SPU communication. As shown in Fig. 4(b), system design and parameters (e.g.,  $M$  and  $S$ ) are optimized such that time consumption of MPU ( $t_{\text{MPU}}$ ) and intra-MDPU MPU-SPU communication ( $t_{\text{comm}}$ ) can be completely covered by that of SPU ( $t_{\text{SPU}}$ ), i.e.,  $t_{\text{MPU}} \leq t_{\text{SPU}}$  and  $t_{\text{comm}} \leq t_{\text{SPU}}$ . Consequently, the total MD system time consumption is just the time used by the highly-efficient SPU.

## Algorithm

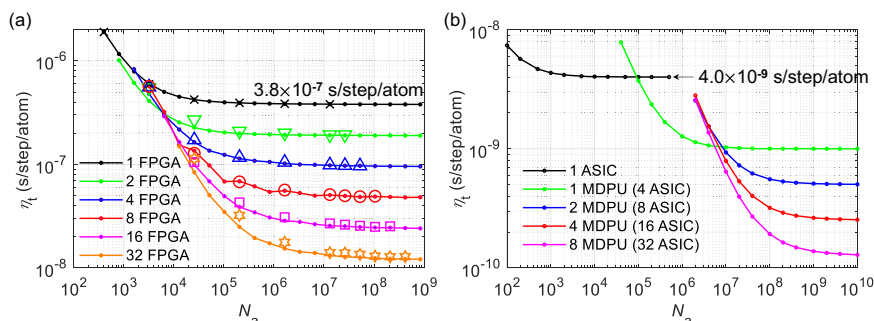
In this work, we use deep potential smooth edition (DeepPot-SE) MLMD algorithm<sup>55</sup>, after extensive modifications, to model IAP. In a  $N_a$ -atom system, the set of atomic coordinates is  $\mathbf{R} = \{\mathbf{R}_i | i = 1, 2, \dots, N_a\}$ , where  $\mathbf{R}_i = (x_i, y_i, z_i)$ . The set of chemical species is  $Z = \{Z_i | i = 1, 2, \dots, N_a\}$ . The neighbor list of atom  $i$  is defined as  $I_i = \{j | R_{ji} < R_c\}$ , where  $R_{ji} = |\mathbf{R}_j - \mathbf{R}_i|$ ,  $\mathbf{R}_j = (x_j, y_j, z_j)$ ,  $R_c$  is the cutoff radius.

To ensure the translational symmetry, the global coordinate system  $\{\mathbf{R}_j\} = \{\mathbf{R}_j | j \in I_i\}$  is translated into a generalized local environment matrix  $\mathbf{R}_{ji} = (s_{ji}, s_{ji}x_{ji}/R_{ji}, s_{ji}y_{ji}/R_{ji}, s_{ji}z_{ji}/R_{ji})$ , where  $s_{ji} = 1/R_{ji}$  if  $0 \leq R_{ji} < R_{cs}$ ;  $s_{ji} = 1/R_{ji} \times \{0.5 + 0.5 \times \cos[\pi(R_{ji} - R_{cs})/(R_c - R_{cs})]\}$ , if  $R_{cs} \leq R_{ji} < R_c$ ; and  $s_{ji} = 0$ , if  $R_{cs} \leq R_{ji}$ . Here,  $R_{cs}$  is a smooth cutoff parameter. Then,  $s_{ji}$  is mapped into a  $M_G$ -dimension embedding matrix  $\mathbf{G}_{ji}$  by a multilayer perceptron (MLP) neural network, which is referred to as “FeaNet” hereafter. The  $l^{\text{th}}$  layer of the MLP is  $d^{l+1} = \phi^l(d^l \times w^l + b^l)$ , where  $d^l$ ,  $w^l$ ,  $b^l$  and  $\phi^l$  are the input vector, weight matrix, bias vector, and the nonlinear activation function, respectively.

To ensure the permutation symmetry and the rotational symmetry, an  $M_G \times 4$  matrix  $\mathbf{P}_i = \sum_j \mathbf{G}_{ji}^T \times \mathbf{R}_{ji}$  and an  $M_G \times M_G$  matrix  $\mathbf{D}_i^T = \mathbf{P}_i \times \mathbf{P}_i^T$  are computed. Then, an  $M_G \times M_H$  ( $1 \leq M_H \leq M_G$ ) matrix  $\mathbf{D}_i[r, c] = \mathbf{D}_i^T[r, (c + r) \% M_G]$  is extracted from  $\mathbf{D}_i^T$ , where  $r$  and  $c$  denote row and column indices, respectively; and  $\%$  denotes the modulo operation. By using another MLP, which is referred to as “FitNet” hereafter,  $\mathbf{D}_i$  is mapped into energy  $E_i$ , and the total system energy is computed using  $E = \sum_i E_i$ . Finally, the atomic force  $\mathbf{F}_i = -\nabla_i E(\mathbf{R}_i)$  and the virial  $\mathbf{V} = \sum_i \mathbf{R}_i^T \mathbf{F}_i$  can be computed.

Though this MLMD algorithm is much more efficient than AIMD (Fig. 1), it has been designed to run on CPU/GPU, whose architecture (von Neumann) is fundamentally different from that of FPGA/ASIC (CIM non von Neumann) used here. So if this MLMD algorithm was directly deployed on FPGA/ASIC hardware without revision, it would be too transistor-consuming to be efficient enough (i.e., large  $\eta_t$  and  $\eta_p$ ). To minimize  $\eta_t$  and

**Fig. 5 | Performance and parallelization of FPGA/ASIC-based MD systems.**  $N_a$ -dependent time consumption ( $\eta_t$ ) and its parallelization performance of the proposed MD system based on FPGA (a) and ASIC (b), where  $N_a$  is the number of atoms. In (a), one MDPU consists of a multicore CPU and an FPGA. In (b), one MDPU consists of one GPU and four ASICs; The lines with small dots are analytical prediction values; and the symbols (i.e., crosses, upward/downward triangles, circles, squares, and stars) are the measured values.



$\eta_p$ , this MLMD algorithm is extensively revised, in the following four aspects, according to the characteristics of FPGA/ASIC hardware (Section “Hardware”).

First, the emerging highly-efficient reconfigurable digital CIM technology<sup>56</sup> is utilized to replace the conventional floating-point multiply-accumulation (MAC) operation, which is known as one of the most computationally-intensive parts. Specifically,  $MAC(A, B) = \sum_i a_i \times b_i$  between the two floating-point vectors  $A = (a_1, a_2, \dots, a_K)$  and  $B = (b_1, b_2, \dots, b_K)$  is replaced by: (1) pre-alignment of elements of  $A$  (i.e.,  $a'_i = h_i^a \times 2^{e_i^a}$ ) and  $B$ ; and (2) subsequent fixed-point MAC operation (i.e.,  $MAC(A', B') = \sum_i a'_i \times b'_i = 2^{e_a+e_b} \times \sum_i h_i^a \times h_i^b$ ). Here,  $h_i^a = a_i \times 2^{-e_i^a}$ ;  $e_i^a = e_{\max}^a - e_h$ ;  $e_{\max}^a$  is the largest element of  $A$ ;  $e_h$  is the fractional digits of fixed-point number  $h_i^a$  for  $A$ . Similar definitions are valid for  $B$  by replacing superscript “a” with “b”. This could significantly enhance the time and power efficiency while maintaining high accuracy<sup>56</sup>, since inputs are pre-aligned to their local maximum exponents, avoiding complex alignment logic circuit in hardware implementation (Section “Hardware”).

Second, though the above pre-alignment has replaced the resource-consuming floating-point operations (e.g.,  $a_i \times b_i$ ) by using much more resource-saving fixed-point operations (e.g.,  $h_i^a \times h_i^b$ ), the time and power efficiency could be further enhanced via replacing the resource-consuming multiplications by using the resource-saving bitwise shift and/or add operations<sup>83,84</sup>. The consequent quantization error can be effectively controlled by using multiple bitwise shift/add operations to replace one multiplication, by leveraging MLP’s merit of high tolerance of weight variation<sup>85,86</sup> and the training algorithms purposefully designed to control the error due to weight variation<sup>87</sup>.

Third, the conventional trigonometric function-based nonlinear activation function,  $\varphi(\cdot)$ ,<sup>55</sup> which is very transistor-consuming, is replaced here by using much more transistor-saving light-weight activation function  $\psi(x) = (1/16) \cdot x_c^2 \cdot (x_c \cdot |x_c| - 4x_c) + x_c$ , where  $x_c = \max(\min(x, 2), -2)$ . The  $\psi(x)$  needs only two multiplications, excluding complicated trigonometric, exponential or division operations; the remaining coefficient multiplications are efficiently implemented as inexpensive shift operations  $2^n$  (i.e.,  $1/16 = 2^{-4}$  and  $4 = 2^2$ ). It worth noting that both the first-order and second-order derivatives of  $\psi(x)$  are continuous, paving a way to the accurate evaluation of energy, atomic force, and virial.

Fourth, though the above three revisions have significantly reduced  $\eta_t$  and  $\eta_p$  of MLP, the efficiency of “FeaNet” MLP can be further enhanced by leveraging its special algorithm characteristics<sup>73</sup>. Since the dimensionality of “FeaNet” input (output) is 1 ( $M_G$ ), here this low-dimensional 1-to- $M_G$  MLP mapping is replaced by using  $M_G$  interpolation functions. For one of these  $M_G$  continuous functions,  $y = f(x)$ , as defined on a finite range  $[x_{\min}, x_{\max}]$ , its input range is uniformly divided into segments ( $x \in [x_k, x_{k+1}]$ ), where  $k = \lfloor (x - x_{\min}) / dx \rfloor$ . Here,  $dx$  is set small enough such that a polynomial can be used to approximate the function of the  $k^{\text{th}}$  segment. The cubic spline interpolation function  $g_k(x) = a \cdot \Delta x^3 + b \cdot \Delta x^2 + c \cdot \Delta x + d$  is used to approximate  $y = f(x)$ , where  $\Delta x = x - x_k$ ; and four coefficients  $a, b, c$  and  $d$  are first determined by solving the four linear equations  $g_k(x_k) = f(x_k)$ ,  $g'_k(x_k) = f'(x_k)$ ,  $g_k(x_{k+1}) = f(x_{k+1})$ , and  $g'_k(x_{k+1}) = f'(x_{k+1})$  and then stored in lookup tables. In both feed-forward (to evaluate energy) and back-

propagation (to evaluate atomic force) “FeaNet” calculations, the above techniques are applied, to guarantee continuity of the function value and its first-order derivative.

## Hardware

In the widely-used von Neumann architecture hardware (e.g., CPU/GPU), a huge proportion (sometimes over 90%) of total MD calculation time and power is spent to frequently shuttle data back-and-forth between the information processing device and the information storage device (i.e., “memory wall” and “power wall” bottlenecks)<sup>45–47</sup>. To solve this problem, we have designed and implemented a non von Neumann architecture CIM hardware, to deploy the revised IAP MLMD algorithm (Section “Algorithm”), based on commodity-level or commercially-available FPGA and ASIC technologies.

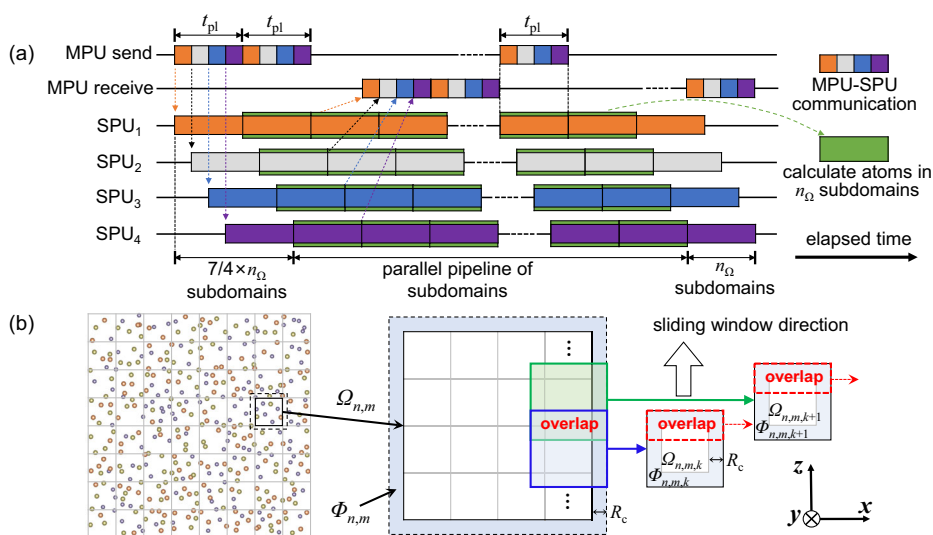
While FPGA has the merits of fast design iteration and low development cost, it suffers from drawbacks of inferior performance, power, and area (PPA) metrics. In contrast, the PPA metrics of ASIC is typically about  $10^1$ – $10^2$  times superior than FPGA, at the expense of slow design iteration and high development cost<sup>57</sup>. So, following the common practice of the semiconductor industry, the CIM hardware here is developed and implemented in two consecutive phases (first FPGA and then ASIC), both of which are based on the emerging in-memory processing technologies to mitigate the “memory wall” and “power wall” bottlenecks<sup>56</sup>.

In the first phase, FPGA (e.g., Xilinx Virtex UltraScale+ VU9P with  $f_{\text{FPGA}} = 350$  MHz clock frequency) is used to realize SPU; and CPU (e.g., Intel Xeon Platinum 8163) can be used to realize MPU. Inside one commodity-level desktop, we can install several FPGA cards (e.g.,  $S = 1, 2, 3, 4$  as shown in Fig. 4). To enable intra-MDPU pipeline (i.e.,  $t_{\text{MDPU}} \leq t_{\text{SPU}}$  in Fig. 4a),  $M = \kappa \times S$  CPU cores are needed inside each MDPU. We found that typically  $\kappa < 10$  suffices (e.g.,  $\kappa = 4$  to 8), up to the CPU performance, if software on MPU is well-programmed, and further increasing  $\kappa$  does not help reducing  $\eta_t$ . To realize inter-MDPU communication, we use internet wide-area remote-direct-memory-access protocol (iWARP), whose bandwidth and latency are shown in Supplementary Fig. 4. The black curve of Fig. 5a shows that an MDPU with one FPGA achieves  $\eta_t = 3.8 \times 10^{-7}$  s/step/atom, consuming merely  $\eta_p = 7.5 \times 10^1$  W.

As a direct comparison, the established MLMD based on one state-of-the-art GPU, after the IAP model compression and in-depth software optimization, can only achieve  $\eta_t = 10^{-6}$ – $10^{-5}$  s/step/atom and  $\eta_p = 10^2$ – $10^3$  W<sup>49,71,73</sup>. Notably, such low  $\eta_t$  and  $\eta_p$  based on FPGA are achieved by using merely  $f_{\text{FPGA}} = 10^{-1}$ – $10^0$  GHz clock frequency and about  $d_{\text{FPGA}} \approx 10^6$  devices (i.e., logic units). It is well known that ASIC has much higher clock frequency (e.g.,  $f_{\text{ASIC}} = 10^0$ – $10^1$  GHz) and significantly more devices (e.g.,  $d_{\text{ASIC}} > 10^9$  transistors). So, here we use the first phase (FPGA) as a springboard to the second phase (ASIC)<sup>57</sup>, in order to achieve even lower  $\eta_t$  and  $\eta_p$ .

In the second phase, ASIC is used to realize SPU; and GPU can be used to implement MPU. Since the Xilinx Virtex UltraScale+ FPGA is fabricated using 14 nm/16 nm technology nodes with about 30–40 million transistors/mm<sup>288</sup>, we use a process design kit (PDK) with similar transistor density to

**Fig. 6 | Parallel pipelining and sliding window techniques in MD calculation.** Schematic figures to illustrate the MPU-SPU-communication parallel pipeline used to enhance intra-MDPU calculation efficiency where  $t_{pl}$  is the pipeline period (a), and sliding window used to enhance inter-MDPU calculation efficiency, in the second phase of implementation (ASIC).



design ASIC. The digital circuit design, which has been well-tested on FPGA in the first phase, is simulated and synthesized using mainstream electronic design automation tools like Synopsys Verilog Compiler and Simulator (VCS<sup>®</sup>), Synopsys Design Compiler (DC<sup>®</sup>), and so on. To realize the inter-MDPU communication, we use the InfiniBand (IB) technology<sup>91</sup>, whose bandwidth and latency are shown in Supplementary Fig. 4.

As shown in Supplementary Fig. 5, the ASIC is designed to calculate atoms by using the time division multiplexing (TDM) and pipeline technologies. In our ASIC, it takes  $n_0 = 676$  clock cycles to complete calculation of the first atom. After that, the pipeline uses 8 extra clock cycles to finish calculating each additional atom. Since the ASIC clock frequency here is 2.0 GHz, one clock cycle takes 0.5 ns. So, the time consumption of one ASIC is

$$\eta_{t(\text{ASIC})} \approx (n_0 + 8 \times N_{a(\text{ASIC})}) / N_{a(\text{ASIC})} \times (0.5 \times 10^{-9}) = 3.38 \times 10^{-7} / N_{a(\text{ASIC})} + 4.00 \times 10^{-9} \text{ s/step/atom} \quad (2)$$

When the number of atoms sent to an ASIC ( $N_{a(\text{ASIC})}$ ) is large (e.g.,  $10^3$ – $10^4$  or larger), we have  $n_0 \ll 8 \times N_{a(\text{ASIC})}$  and  $\eta_{t(\text{ASIC})} \approx 4.00 \times 10^{-9}$  s/step/atom as indicated by black arrow in Fig. 5b. One ASIC consumes about  $\eta_{p(\text{ASIC})} = 1.3 \times 10^2$  W power.

Notably, though  $\eta_{p(\text{ASIC})}$  of one ASIC here is comparable to that of one state-of-the-art GPU (i.e.,  $10^2$ – $10^3$  W),  $\eta_{t(\text{ASIC})}$  of one ASIC is about 3 orders of magnitude lower than that of one state-of-the-art GPU (i.e., about  $10^{-6}$  s/step/atom)<sup>49,71,73</sup>. To strike a balance between ASIC fabrication cost and calculation performance, the ASIC area is designed to be a moderate value (about 110 mm<sup>2</sup>) and four ASICs are used together with one GPU (i.e.,  $S/M = 4$ ) to form an MD processing unit (MDPU). Since each GPU (i.e., MPU) typically has 16 peripheral component interconnect express (PCIe) lanes, so each ASIC (i.e., SPU<sub>*i*</sub>,  $i = 1, 2, 3, 4$ ) is designed to have 4 PCIe lanes, such that  $S/M = 4$ .

As shown in Fig. 6a, in the second phase (ASIC), each MPU connects with four SPUs via an PCIe switch, in order to further reduce  $\eta_t$ . So, the number of atoms sent to one MDPU ( $N_{a(\text{MDPU})}$ ) is four times of that sent to one SPU ( $N_{a(\text{SPU})}$ ), i.e.,  $N_{a(\text{MDPU})} = N_{a(\text{SPU})} \times S/M = 4 \times N_{a(\text{SPU})}$ . Time efficiency of an MDPU is

$$\eta_{t(\text{MDPU})} \approx (c_0 \times N_{\Omega} \times n_{\Omega} + N_{a(\text{MDPU})}/4) / N_{a(\text{MDPU})} \times \eta_{t(\text{SPU})} = 9.14 \times 10^{-2} / N_{a(\text{MDPU})}^2 + 2.71 \times 10^{-4} / N_{a(\text{MDPU})} + 1.00 \times 10^{-9} \text{ s/step/atom} \quad (3)$$

where  $N_{\Omega} = 4096$  is the number of atoms in a subdomain (defined in Fig. 6b and Section “Software”);  $n_{\Omega} = 6$  subdomains of atoms are sent together into

an ASIC to ensure  $t_{\text{comm}} \leq t_{\text{SPU}}$ ;  $c_0 = 7/4 + 1 = 2.75$  means that it takes MPU  $7/4 \times t_{pl}$  to send data to SPUs and  $1 \times t_{pl}$  to receive data from SPU in one pipeline period ( $t_{pl}$  as defined in Fig. 6a). Equation (3) is visualized as the green curve in Fig. 5b. It worth noting that one common blade server node<sup>92</sup> can accommodate at least 4 MDPU, which can achieve  $\eta_t \approx 2.5 \times 10^{-10}$  s/step/atom (red curve of Fig. 5b) consuming merely  $\eta_p \approx 2.7 \times 10^3$  W power. This  $\eta_t$  value is comparable to the speed using the whole top-tier supercomputer in the world with  $\eta_p \approx 10^7$  W<sup>49,71</sup>.

## Software

The software is developed for deployment on von Neumann architecture MPU, to perform three tasks: (1) to coordinate the intra-MDPU MPU-SPU-communication pipeline; (2) to realize the inter-MDPU parallelization via message passing interface (MPI); and (3) to run all calculations other than IAP evaluation. To efficiently accomplish these three tasks, as shown in Fig. 6b, the supercell with  $N_a$  atoms is partitioned into  $N_d = N \times M$  domains (denoted as  $\Omega_{n,m}$  with  $n = 1, 2, \dots, N$  and  $m = 1, 2, \dots, M$ ), where  $N$  is the number of MDPU and  $M$  is the number of MPU per MDPU. Each domain ( $\Omega_{n,m}$ ), which is surrounded by a domain shell ( $\Phi_{n,m}$ ) of thickness  $R_c$ , is further partitioned into subdomains ( $\Omega_{n,m,k}$ ), which is surrounded by their subdomain shells ( $\Phi_{n,m,k}$ ).

In the first phase (FPGA), to run an MD step, MPU<sub>*n,m*</sub> first encodes atomic information (e.g., the atomic coordinates  $\{R_i | i \in \Omega_{n,m,k} \cup \Phi_{n,m,k}\}$ , the atomic specie  $\{z_i | i \in \Omega_{n,m,k} \cup \Phi_{n,m,k}\}$ , and the neighbor list  $\{I_i | i \in \Omega_{n,m,k}\}$ ) and save it into buffer memory. Then, MPU<sub>*n,m*</sub> sends the encoded data to SPU<sub>*ng*</sub> ( $g = [m/k]$ ). After SPU<sub>*ng*</sub> calculation of PES, CPU<sub>*n,m*</sub> reads the encoded PES data from SPU<sub>*ng*</sub>. Next, MPU<sub>*n,m*</sub> sends the enable signal to the following CPU core in the  $n^{\text{th}}$  MDPU. Finally, the data is decoded to obtain the atomic energies  $\{E_i | i \in \Omega_{n,m,k}\}$ , forces  $\{F_i | i \in \Omega_{n,m,k} \cup \Phi_{n,m,k}\}$ , and virials  $\{V_i | i \in \Omega_{n,m,k} \cup \Phi_{n,m,k}\}$ . Following the processing of the subdomains, the MPU<sub>*n,m*</sub> must communicate with each other via MPI, and collect all components to obtain the final values of system energy, atomic forces, and atomic virials.

In the second phase (ASIC), since an ASIC is much faster than an FPGA (i.e., significantly reduced  $t_{\text{SPU}}$ ), it becomes much more challenging to achieve  $t_{\text{MPU}} \leq t_{\text{SPU}}$  and  $t_{\text{comm}} \leq t_{\text{SPU}}$ . So, besides the above domain and subdomain partitioning scheme, several other highly-efficient software technologies are used to further reduce  $t_{\text{MPU}}$  and  $t_{\text{comm}}$ . To reduce  $t_{\text{MPU}}$ , MPU is based on GPU instead of CPU, as a GPU typically has about  $10^2$  times more cores than a CPU. And GPU kernel fusion programming technology is used to dramatically reduce  $t_{\text{MPU}}$  on GPU<sup>93</sup>. To reduce  $t_{\text{comm}}$ , we use sliding window (Fig. 6b), such that the communication data size can be significantly reduced – after computing  $\{\Omega_{n,m,k} \cup \Phi_{n,m,k}\}$ , the atomic

information in the overlap region does not need to be resent to the SPU to compute  $\{\Omega_{n,m,k+1} \cup \Phi_{n,m,k+1}\}$ . Since there is limited hardware resource, to enable highly-efficient CIM processing, we restrict the total number of atoms in  $\Omega_{n,m,k}$  to be 4096 and that of  $\{\Omega_{n,m,k} \cup \Phi_{n,m,k}\}$  to be 16384.

To enhance efficiency to run MD with large number of atoms and/or long duration, we use the  $N$ -MDPU parallelization, based on inter-MDPU MPI. According to the Amdahl's law, the time consumption is  $\eta_t = \eta_{t(\text{MDPU})}/N + \zeta$  (plotted in Fig. 5), where  $\eta_{t(\text{MDPU})}$  ( $\zeta$ ) is computation (communication) time consumption (unit: s/step/atom); and  $N$  is the number of MDPU. Here,  $\zeta = (\zeta_1 + \zeta_2)/N_a$ , where  $\zeta_1 = \log_2 N \times T_c(b_c)$  is synchronous collective communication time (e.g., MPI function "MPI\_Allreduce")<sup>94</sup>;  $\zeta_2 = T_n(6b_n)/v + (2v-1)/v \times T_n(3b_n)$  is the asynchronous communication time (e.g., MPI functions "MPI\_Isend", "MPI\_Irecv");  $T_c(b_c) = b_c / G_{\text{bandwidth}}(b_c) + G_{\text{latency}}(b_c)$  is a function depends on the bandwidth ( $G_{\text{bandwidth}}$ ) and latency ( $G_{\text{latency}}$ ), which is shown in the Supplementary Fig. 4;  $T_n(b_n) = \sum_a T_c(N_{a(a)} \times b_n)$ ;  $N_{a(a)}$  is number of atoms transmitted in the  $a^{\text{th}}$  MPI communication;  $b_c$  is the number of bits transmitted via inter-MDPU synchronous collective MPI communication ( $b_c = 64$  bits here since only temperature is transmitted);  $b_n$  is the number of bits per atom transmitted via inter-MDPU asynchronous MPI communication (here  $b_n = 32$  bits is used to represent each number); and the (sub) domain shell is constructed every  $v$  steps (e.g.,  $v = 10$ ). In  $\zeta_2$ , the first term means that 6 numbers (3 coordinates, chemical species, atomic index, and region label) per atom should be transmitted every  $v$  steps; the second term means updating the atomic coordinates (3 numbers per atom)  $v-1$  times every  $v$  step and updating atomic forces (3 numbers per atom) every step.

## Data availability

All data needed to reproduce the analysis results in this paper (e.g., Fig. 2 and Fig. 3) can be freely downloaded from: <https://github.com/LiuGroupHNU/md-data/tree/master/data>.

## Code availability

The codes are open-sourced at: <https://github.com/LiuGroupHNU/md-data>, with a step-by-step tutorial: <https://github.com/LiuGroupHNU/md-data/blob/master/code/doc/mdpu/mdpu.md>, based on GNU lesser General Public License 3.0: <https://github.com/LiuGroupHNU/md-data/blob/master/code/LICENSE>.

Received: 24 October 2023; Accepted: 28 September 2024;

Published online: 07 November 2024

## References

- Alder, B. J. & Wainwright, T. E. Phase transition for a hard sphere system. *J. Chem. Phys.* **27**, 1208–1209 (1957).
- Iftimie, R., Minari, P. & Tuckerman, M. E. Ab initio molecular dynamics: Concepts, recent developments, and future trends. *Proc. Natl. Acad. Sci.* **102**, 6654–6659 (2005).
- Tuckerman, M. E. & Martyna, G. J. Understanding modern molecular dynamics: 0.167em techniques and applications. *J. Phys. Chem. B* **104**, 159–178 (1999).
- Hu, Y.-C. & Tanaka, H. Origin of the boson peak in amorphous solids. *Nat. Phys.* **18**, 669–677 (2022).
- Galib, M. & Limmer, D. T. Reactive uptake of  $\text{N}_2\text{O}_5$  by atmospheric aerosol is dominated by interfacial processes. *Science* **371**, 921–925 (2021).
- Wang, D. et al. Efficient sampling of high-dimensional free energy landscapes using adaptive reinforced dynamics. *Nat. Comput. Sci.* **2**, 20–29 (2021).
- Karplus, M. & Petsko, G. A. Molecular dynamics simulations in biology. *Nature* **347**, 631–639 (1990).
- Rao, F. et al. Reducing the stochasticity of crystal nucleation to enable subnanosecond memory writing. *Science* **358**, 1423–1427 (2017).
- Ding, K. et al. Phase-change heterostructure enables ultralow noise and drift for memory operation. *Science* **366**, 210–215 (2019).
- Gorgulla, C. et al. An open-source drug discovery platform enables ultra-large virtual screens. *Nature* **580**, 663–668 (2020).
- Laio, A., Bernard, S., Chiarotti, G. L., Scandolo, S. & Tosatti, E. Physics of iron at earth's core conditions. *Science* **287**, 1027–1030 (2000).
- Ihara, S. & Itoh, S. Molecular dynamics in semiconductor physics. *Comput. Mater. Sci.* **10**, 80–87 (1998).
- Cheng, B., Mazzola, G., Pickard, C. J. & Ceriotti, M. Evidence for supercritical behaviour of high-pressure liquid hydrogen. *Nature* **585**, 217–220 (2020).
- Kadau, K., Germann, T. C. & Lomdahl, P. S. Molecular dynamics comes of age: 320-billion-atom simulation on BlueGene/L. *Int. J. Modern Phys. C* **17**, 1755–1761 (2006).
- Pople, J. A. Nobel lecture: Quantum chemical models. *Rev. Mod. Phys.* **71**, 1267–1274 (1999).
- Thompson, A. P. et al. LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comput. Phys. Commun.* **271**, 108171 (2022).
- Berendsen, H., van der Spoel, D. & van Drunen, R. GROMACS: A message-passing parallel molecular dynamics implementation. *Comput. Phys. Commun.* **91**, 43–56 (1995).
- Phillips, J. C. et al. Scalable molecular dynamics on CPU and GPU architectures with NAMD. *J. Chem. Phys.* **153**, 044130 (2020).
- Case, D. A. et al. AmberTools. *J. Chem. Inf. Model.* **63**, 6183–6191 (2023).
- Brooks, B. R. et al. CHARMM: The biomolecular simulation program. *J. Comput. Chem.* **30**, 1545–1614 (2009).
- Kresse, G. & Furthmüller, J. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Phys. Rev. B* **54**, 11169–11186 (1996).
- Giannozzi, P. et al. QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials. *J. Phys.: Condens. Matter* **21**, 395502 (2009).
- Kühne, T. D. et al. CP2K: An electronic structure and molecular dynamics software package - quickstep: Efficient and accurate electronic structure calculations. *J. Chem. Phys.* **152** (2020).
- Boero, M. & Oshiyama, A. Car-Parrinello Molecular Dynamics, 489–497 (Springer Netherlands, Dordrecht, 2016).
- Soler, J. M. et al. The siesta method for ab initio order- $n$  materials simulation. *J. Phys.: Condens. Matter* **14**, 2745–2779 (2002).
- Chen, M., Guo, G.-C. & He, L. Systematically improvable optimized atomic basis sets for ab initio calculations. *J. Phys.: Condens. Matter* **22**, 445501 (2010).
- Thompson, A., Swiler, L., Trott, C., Foiles, S. & Tucker, G. Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials. *J. Computational Phys.* **285**, 316–330 (2015).
- Schütt, K. T., Arbabzadah, F., Chmiela, S., Müller, K. R. & Tkatchenko, A. Quantum-chemical insights from deep tensor neural networks. *Nat. Commun.* **8** (2017).
- Smith, J. S., Isayev, O. & Roitberg, A. E. ANI-1: an extensible neural network potential with dft accuracy at force field computational cost. *Chem. Sci.* **8**, 3192–3203 (2017).
- Zhang, L., Han, J., Wang, H., Car, R. & E, W. Deep potential molecular dynamics: A scalable model with the accuracy of quantum mechanics. *Phys. Rev. Lett.* **120**, 143001 (2018).
- Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A. & Müller, K.-R. SchNet - a deep learning architecture for molecules and materials. *J. Chem. Phys.* **148**, 241722 (2018).
- Lee, K., Yoo, D., Jeong, W. & Han, S. SIMPLE-NN: An efficient package for training and executing neural-network interatomic potentials. *Comput. Phys. Commun.* **242**, 95–103 (2019).
- Chen, X. et al. TensorAlloy: An automatic atomistic neural network program for alloys. *Comput. Phys. Commun.* **250**, 107057 (2020).
- Kanal, I. Y., Keith, J. A. & Hutchison, G. R. A sobering assessment of small-molecule force field methods for low energy conformer predictions. *Int. J. Quant. Chem.* **118**, e25512 (2018).

35. Zgarbová, M., Otyepka, M., Šponer, J., Hobza, P. & Jurecka, P. Large-scale compensation of errors in pairwise-additive empirical force fields: comparison of amber intermolecular terms with rigorous dft-sapt calculations. *Phys. Chem. Chem. Phys.* **12**, 10476 (2010).
36. Demir, H. et al. DFT-based force field development for noble gas adsorption in metal organic frameworks. *J. Mater. Chem. A* **3**, 23539–23548 (2015).
37. Lu, C. et al. OPLS4: Improving force field accuracy on challenging regimes of chemical space. *J. Chem. Theory Comput.* **17**, 4291–4300 (2021).
38. Shaw, D. E. et al. Anton, a special-purpose machine for molecular dynamics simulation. *Commun. ACM* **51**, 91–97 (2008).
39. Shaw, D. E. et al. Anton 2: Raising the bar for performance and programmability in a special-purpose molecular dynamics supercomputer. In *SC14: International Conference for High Performance Computing, Networking, Storage and Analysis* (IEEE, 2014).
40. Shaw, D. E. et al. Anton 3: twenty microseconds of molecular dynamics simulation before lunch. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (ACM, 2021).
41. Susukita, R. et al. Hardware accelerator for molecular dynamics: MDGRAPE-2. *Comput. Phys. Commun.* **155**, 115–131 (2003).
42. Taiji, M., Narumi, T., Ohno, Y. & Konagaya, A. Mdgape-3: A petaflops special-purpose computer system for molecular dynamics simulations. In Joubert, G., Nagel, W., Peters, F. & Walter, W. (eds.) *Parallel Computing*, vol. 13 of *Advances in Parallel Computing*, 669–676 (North-Holland, 2004).
43. Ohmura, I., Morimoto, G., Ohno, Y., Hasegawa, A. & Taiji, M. MDGRAPE-4: A special-purpose computer system for molecular dynamics simulations. *Philos. Trans. R. Soc. A: Math. Phys. Eng. Sci.* **372**, 20130387 (2014).
44. [https://en.wikipedia.org/wiki/von\\_neumann\\_architecture](https://en.wikipedia.org/wiki/von_neumann_architecture).
45. Horowitz, M. 1.1 computing's energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)* (IEEE, (2014).
46. Stegailov, V., Smirnov, G. & Vechev, V. VASP hits the memory wall: Processors efficiency comparison. *Concurrency and Comput. Pract. Exp.* **31**, e5136 (2019).
47. Mo, P. et al. Accurate and efficient molecular dynamics based on machine learning and non von Neumann architecture. *npj Comput. Mater.* **8** (2022).
48. Zhao, Z. et al. A heterogeneous parallel non-von Neumann architecture system for accurate and efficient machine learning molecular dynamics. *IEEE Transactions on Circuits and Systems I: Regular Papers* 1–11 (2023).
49. Jia, W. et al. Pushing the limit of molecular dynamics with ab initio accuracy to 100 million atoms with machine learning. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, Article 5 (IEEE Press, Atlanta, Georgia, 2020).
50. Das, S. et al. Fast, scalable and accurate finite-element based ab initio calculations using mixed precision computing. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (ACM, 2019).
51. Schade, R. et al. Towards electronic structure-based ab-initio molecular dynamics simulations with hundreds of millions of atoms. *Parallel Comput* **111**, 102920 (2022).
52. Fattebert, J.-L., Osei-Kuffuor, D., Draeger, E. W., Ogitsu, T. & Krauss, W. D. Modeling dilute solutions using first-principles molecular dynamics: Computing more than a million atoms with over a million cores. In *SC16: International Conference for High Performance Computing, Networking, Storage and Analysis* (IEEE, 2016).
53. Guo, Z. et al. Extending the limit of molecular dynamics with ab initio accuracy to 10 billion atoms. In *Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (ACM, 2022).
54. Leiserson, C. E. et al. There's plenty of room at the top: What will drive computer performance after moore's law? *Science* **368**, eaam9744 (2020).
55. Zhang, L. et al. End-to-end symmetry preserving inter-atomic potential energy model for finite and extended systems. In Bengio, S. et al. (eds.) *Advances in Neural Information Processing Systems 31*, 4436–4446 (Curran Associates, Inc., 2018).
56. Tu, F. et al. A 28 nm 29.2tflops/w BF16 and 36.5stops/w INT8 reconfigurable digital CIM processor with unified FP/INT pipeline and bitwise in-memory booth multiplication for cloud deep learning acceleration. In *2022 IEEE International Solid-State Circuits Conference (ISSCC)* (IEEE, 2022).
57. Kuon, I. & Rose, J. Measuring the gap between FPGAs and ASICs. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **26**, 203–215 (2007).
58. Chen, B. et al. Atomistic mechanism of phase transition in shock compressed gold revealed by deep potential (2020). <https://arxiv.org/abs/2006.13136>. 2006.13136.
59. Zhang, Y. et al. DP-GEN: A concurrent learning platform for the generation of reliable deep learning based potential energy models. *Comput. Phys. Commun.* **253**, 107206 (2020).
60. Zhang, L., Lin, D.-Y., Wang, H., Car, R. & E, W. Active learning of uniformly accurate interatomic potentials for materials simulation. *Phys. Rev. Mater.* **3**, 023804 (2019).
61. DiStasio, R. A., Santra, B., Li, Z., Wu, X. & Car, R. The individual and collective effects of exact exchange and dispersion interactions on the ab initio structure of liquid water. *J. Chem. Phys.* **141**, 084502 (2014).
62. Zhang, L., Wang, H., Car, R. & E, W. Phase diagram of a deep potential water model. *Phys. Rev. Lett.* **126**, 236001 (2021).
63. Shi, M., Mo, P. & Liu, J. Deep neural network for accurate and efficient atomistic modeling of phase change memory. *IEEE Electron Device Lett* **41**, 365–368 (2020).
64. Zhang, W. & Ma, E. Unveiling the structural origin to control resistance drift in phase-change memory materials. *Mater. Today* **41**, 156–176 (2020).
65. Huang, J. et al. Deep potential generation scheme and simulation protocol for the Li<sub>10</sub>GeP<sub>2</sub>S<sub>12</sub>-type superionic conductors. *J. Chem. Phys.* **154**, 094703 (2021).
66. Sanderson, R. T. Interrelation of bond dissociation energies and contributing bond energies. *J. Am. Chem. Soc.* **97**, 1367–1372 (1975).
67. VandeVondele, J. et al. Quickstep: Fast and accurate density functional calculations using a mixed gaussian and plane waves approach. *Comput. Phys. Commun.* **167**, 103–128 (2005).
68. Liu, J., Xu, X., Brush, L. & Anantram, M. P. A multi-scale analysis of the crystallization of amorphous germanium telluride using ab initio simulations and classical crystallization theory. *J. Appl. Phys.* **115**, 023513 (2014).
69. Akola, J. & Jones, R. O. Structure of amorphous Ge<sub>8</sub>Sb<sub>2</sub>Te<sub>11</sub>: GeTe-Sb<sub>2</sub>Te<sub>3</sub> alloys and optical storage. *Phys. Rev. B* **79**, 134118 (2009).
70. Gaspard, J.-P. Structure of covalently bonded materials: From the peierls distortion to phase-change materials. *Comptes Rendus Phys* **17**, 389–405 (2016).
71. <https://awards.acm.org/bell>.
72. [https://en.wikipedia.org/wiki/thermal\\_design\\_power](https://en.wikipedia.org/wiki/thermal_design_power).
73. Lu, D. et al. Dp compress: A model compression scheme for generating efficient deep potential models. *J. Chem. Theory Comput.* **18**, 5559–5567 (2022).
74. <https://www.lammps.org/>.
75. <https://deepmodeling.com/blog/papers/deepmd-kit/>.
76. Zhang, L. et al. A deep potential model with long-range electrostatic interactions. *J. Chem. Phys.* **156**, 124107 (2022).
77. Zhang, C., Yue, S., Panagiotopoulos, A. Z., Klein, M. L. & Wu, X. Why dissolving salt in water decreases its dielectric permittivity. *Phys. Rev. Lett.* **131**, 076801 (2023).

78. Deng, Q. & Liu, Q. Field-programmable gate array acceleration of the tersoff potential in lammmps. *Eng. Rep.* e12694 (2023).
79. Wu, C. et al. FASDA: An FPGA-aided, scalable and distributed accelerator for range-limited molecular dynamics. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '23 (Association for Computing Machinery, New York, NY, USA, 2023).
80. Sheng, N. et al. Microsecond simulation in a special-purpose molecular dynamics computer cluster. In *2023 11th International Conference on Bioinformatics and Computational Biology (ICBCB)* (IEEE, 2023).
81. Yuan, M. et al. FPGA-accelerated tersoff multi-body potential for molecular dynamics simulations. In *Applied Reconfigurable Computing. Architectures, Tools, and Applications*, 17–31 (Springer Nature Switzerland, 2022).
82. Yang, C. et al. Fully integrated fpga molecular dynamics simulations. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '19 (ACM, 2019).
83. Chen, H. et al. Addernet: Do we really need multiplications in deep learning? (2019). <https://arxiv.org/abs/1912.13200v3>.
84. Elhoushi, M., Chen, Z., Shafiq, F., Tian, Y. H. & Li, J. Y. DeepShift: Towards multiplication-less neural networks (2019). <https://arxiv.org/abs/1905.13298v3>.
85. Sackinger, E. Measurement of finite-precision effects in handwriting- and speech-recognition algorithms. In *Lecture Notes in Computer Science*, 1223–1228 (Springer Berlin Heidelberg, 1997).
86. Merrikh-Bayat, F. et al. High-performance mixed-signal neurocomputing with nanoscale floating-gate memory cell arrays. *IEEE Trans. Neural Netw. Learn. Syst.* **29**, 4782–4790 (2018).
87. Conti, M., Orcioni, S. & Turchetti, C. Training neural networks to be insensitive to weight random variations. *Neural Netw.* **13**, 125–132 (2000).
88. <https://www.xilinx.com/products/silicon-devices/fpga/virtex-ultrascale-plus.html>.
89. <https://www.synopsys.com/verification/simulation/vcs.html>.
90. <https://www.synopsys.com/implementation-and-signoff/rtl-synthesis-test/dc-ultra.html>.
91. Li, Y. et al. Understanding hot interconnects with an extensive benchmark survey. *BenchCouncil Trans. Benchmarks Stand. Evaluat.* **2**, 100074 (2022).
92. <https://www.gigabyte.cn/enterprise/gpu-server/g292-z20-rev-100>.
93. Wang, G., Lin, Y. & Yi, W. Kernel fusion: An effective method for better power efficiency on multithreaded GPU. In *2010 IEEE/ACM Intl Conference on Green Computing and Communications & Intl Conference on Cyber, Physical and Social Computing* (IEEE, 2010).
94. Rabenseifner, R. & Traff, J. L. More efficient reduction algorithms for non-power-of-two number of processors in message-passing parallel systems. In *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, 36–46 (Springer Berlin Heidelberg, 2004).

## Acknowledgements

This work is supported by the National Natural Science Foundation of China (62474058 and 61804049); the Yuelushan Center for Industrial Innovation (2023YCII0104); the Huxiang High Level Talent Gathering Project (2019RS1023); the Technology Innovation and Entrepreneurship Funds of Hunan Province, P.R. China (2019GK5029); the Fund for Distinguished Young Scholars of Changsha (kq1905012).

## Author contributions

P.H.M., Y.J.Z., Z.Y.Z. and J.L. proposed the methodology and performed the experiments. P.H.M., Y.J.Z., Z.Y.Z., J.H.L., H.H.S. and J.L. prepared the visualized tables and figures. P.H.M., Y.J.Z., Z.Y.Z., J.H.L., D.W.G., X.D., X.Z., B.C., J.X.H., J.C., J.Y.D., and J.L. collected the data. P.H.M., Z.Y.Z., and J.L. wrote the initial manuscript. P.H.M., Y.J.Z., Z.Y.Z., H.H.S., J.H.L., D.W.G., X.D., X.Z., B.C., M.C.S., D.Z., D.H.L., Y.N.W., J.X.H., F.L., X.Y.L., M.H.C., J.C., B.L., W.N.E., J.Y.D., L.F.Z., H.W., and J.L. provided the review and revision for manuscript. H.W. provided suggestions to improve the implementation. J.L. provided the oversight and guidance for the research execution.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary information** The online version contains supplementary material available at <https://doi.org/10.1038/s41524-024-01422-3>.

**Correspondence** and requests for materials should be addressed to Jiayu Dai, Linfeng Zhang, Han Wang or Jie Liu.

**Reprints and permissions information** is available at <http://www.nature.com/reprints>

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024