科学通报 2025 年 第 70 卷 第 24 期 : 4109 ~ 4115
人工智能驱动的材料科学研究专题 论 文
《中国科学》杂志社
SCIENCE CHINA PRESS

第一性原理精度的分子动力学模拟强可扩展性研究

李剑雄[1,2], 谭光明[1,2], 贾伟乐[1,2*]

1. 中国科学院计算技术研究所, 处理器芯片全国重点实验室, 北京 100190
2. 中国科学院大学, 北京 100190
* 联系人, E-mail: jiaweile@ict.ac.cn

2025-01-06收稿, 2025-03-05修回, 2025-03-05接受, 2025-03-06网络版发表
中国科学院战略性先导科技专项(XDB0500102)、国家自然科学基金(92270206, T2125013, 62372435, 62032023, 61972377, 61972380, T2293702)和中国科学院稳定支持基础研究领域青年团队计划(YSBR-005)资助

摘要 基于神经网络的第一性原理精度的分子动力学模拟方法, 目前已成为研究物质相变、材料缺陷等量子物理现象的重要工具. 然而, 尽管该方法在模拟空间尺度、计算精度等方面已取得了显著进展, 但是由于其计算过程需要进行神经网络模型推理, 包含大量复杂算子, 模拟速度仍然受限, 每天仅能实现纳秒级时间尺度的模拟. 一些复杂物理化学现象的研究, 需要微秒甚至毫秒级的模拟. 即使是当前模拟速度最快的工具也需要数周时间的计算, 这严重制约了相关领域的研究进程. 提升模拟工具的强可扩展性能, 加快模拟速度, 减少到解时间, 具有重要的研究意义. 本文对DeePMD-kit这一广泛使用的开源工具展开了一系列细粒度的优化. 我们针对其在强可扩展场景下的计算效率低下的问题, 提出了无框架代码实现、面向瘦高矩阵的GEMM算子优化, 以及大内存下Tabulate算法优化等优化方法. 我们在ARM和X86两个平台上进行了代码实现, 在每个核心只计算一个原子时, 将单个时间步所需的计算时间降低到了最低331 μs, 在两个平台上的计算速度分别提升了34和303倍.

关键词 高性能计算, DeePMD, 分子动力学, 神经网络力场

分子动力学模拟一直是研究物质微观物理化学性质的重要工具, 被广泛应用于药物研制、新材料研究等领域[1]. 其通过模拟微观粒子的动力学过程, 可以在原子层面揭示反应细节. 相比于实际实验, 利用分子动力学模拟工具可以经济、高效、安全地完成物理或化学实验探索, 减少高昂精密仪器的使用和实验材料的消耗[2].

分子动力学模拟的核心是原子势能建模. 传统的分子动力学模拟采用经验势函数法(empirical force fields, EFFs), 利用含经验参数的解析式来定义原子势能和作用力, 如Lennard-Jones(L-J)势[3]、Tersoff势[4]等. 虽然其计算复杂度低, 但是由于其表达能力受限于解析式本身的建模程度, 精度相对较低, 只能实现牛顿力学精度的模拟. 对于弱非共价键分子间相互作用、核量子效应等[5,6], 无法进行有效的建模, 需要利用第一性原理精度的分子动力学模拟工具进行模拟.

当前第一性原理精度的分子动力学模拟可以分为两种方法. 一种是第一性原理分子动力学方法(*ab initio* molecular dynamic, AIMD)[7], 其从电子结构出发, 通过求解Kohn-Sham方程来进行微观粒子的受力计算, 当前如VASP[8]、PWmat[9,10]等工具均基于AIMD进行求解, 被广泛应用于超导材料[11]、纳米材料[12]研究等领域. 然而, AIMD计算与物理体系的原子规模呈三次方增长, 很难进行大尺度的模拟[13]. 为了提高模拟速

引用格式: 李剑雄, 谭光明, 贾伟乐. 第一性原理精度的分子动力学模拟强可扩展性研究. 科学通报, 2025, 70: 4109–4115
Li J, Tan G, Jia W. Strong scaling of molecular dynamics simulations with *ab initio* accuracy (in Chinese). Chin Sci Bull, 2025, 70: 4109–4115, doi: 10.1360/TB-2025-0027

度, 减少计算复杂度, Behler和Parrinello [14]提出了基于神经网络的分子动力学(neural-network-based molecular dynamics, NNMD)方法. NNMD利用AIMD结果来训练神经网络模型, 并通过模型推理进行原子势能和作用力计算, 将计算复杂度从AIMD的$O(N^3)$降低为$O(N)$, 在模拟空间尺度和模拟速度方面均有了巨大的提升, 同时能保持AIMD精度, 成为分子动力学模拟的新方法.

随着研究的深入和计算机性能的提升, 当前各种NNMD方法被广泛提出, 以进一步提高精度并减少训练复杂度. 例如, 基于图神经网络方法的NequIP [15], 利用等变性原理进行原子特征提取, 在训练数据集减少了三个数量级的同时能保持较高的精度; Allegro [16]通过局部等变性深度神经网络, 实现了良好的可扩展性, 在128个GPU节点上实现了百万银原子体系每秒114个时间步的模拟速度; SNAP [17]方法则利用频谱领域分析方法提取原子表征, 并通过对称性减少了一个数量级的计算成本, 在4650个summit超算节点上实现了200亿碳原子的模拟. 除此之外, DeePMD-kit [18]、SPONGE [19]、UNEP-v1 [20]、SpookyNet [21]和DimeNet++ [22,23]等方法均在模型结构和计算上进行了积极的探索. 在这些方法中, DeePMD-kit以其较高的精度、完善的社区生态获得了广泛关注, 并应用于纳米材料研究 [24]、物质相变 [25]等领域. 2020年, 贾伟乐团队利用DeePMD-kit实现了1亿原子规模的仿真, 获得了当年的戈登贝尔奖 [13]; 随后在2022年, 通过对内部计算和访存的细粒度优化, 进一步将原子规模扩展到了100亿, 在富岳超级计算机4560个计算节点的强可扩展实验中, 200万水分子体系实现了4.7 ns/天的模拟速度 [26].

DeePMD-kit采用深度势能模型(deep potential, 以下简称DeePMD势)进行原子势能建模. 在DeePMD势中, 每个原子i的能量E_i由其邻居原子类型和分布决定. 在计算过程中, 其首先根据原子i的邻居列表, 生成原子的环境矩阵$\tilde{\mathcal{R}}_i$, $\tilde{\mathcal{R}}_i = s(\boldsymbol{r}_{ij}) \times (1, x_{ij}/|\boldsymbol{r}_{ij}|, y_{ij}/|\boldsymbol{r}_{ij}|, z_{ij}/|\boldsymbol{r}_{ij}|)$. 其中, $\boldsymbol{r}_{ij} = \boldsymbol{r}_j - \boldsymbol{r}_i$, 表示中心原子$i$到其邻居原子$j$的向量距离; $s(r_{ij})$为平滑函数. 之后, 再将$s(r_{ij})$输入到一个三层全连接的嵌入网络中(embedding net), 得到$\mathcal{G}_i$矩阵, 即可计算得到描述符矩阵$\mathcal{D}_i$, $\mathcal{D}_i = (\mathcal{G}_i^<)^{\mathrm{T}}\tilde{\mathcal{R}}_i(\tilde{\mathcal{R}}_i)^{\mathrm{T}}\mathcal{G}_i$, 其中$\mathcal{G}_i^<$为$\mathcal{G}_i$矩阵的前16列. $\mathcal{D}_i$保持了原子系统的平移、旋转和对称不变性的物理信息. 然后, 再将$\mathcal{D}_i$输入到一个三层全连接神经网络中(fitting net), 即可输出原子的能量E_i. 原子受力F_i可以通过总能E对原子位置的梯度求导得出. 由于嵌入网络的计算复杂度较高, DeePMD-kit进一步提出了DP Compress方法 [27], 利用五阶多项式拟合方法替换嵌入网络中的三层全连接神经网络的计算, 利用查表方法降低了计算复杂度, 并通过算子融合减少了内存占用, 同时保持了原有的精度, 成为了推理时的首选方法.

尽管DeePMD-kit在模拟空间尺度、模拟精度等方面已取得了显著提升, 当前代码在推理时每个时间步仍需数微秒的计算时间, 每天仅能实现纳秒级的模拟. 一些复杂的物理化学现象, 如蛋白质折叠等, 需要微秒甚至毫秒尺度的模拟才会出现 [28,29]. 这使得即使在超级计算机数千节点上, 也需要数周时间的计算才会得到有效的结果, 严重制约了研究进程. 对DeePMD势开展强可扩展性优化, 也就是利用更多的计算资源, 加快模拟速度, 减少到解时间, 具有重要的现实意义. 同时, 探索DeePMD势的模拟速度上限, 尤其是每个核心只计算一个原子时的模拟速度极限, 对于探索硬件性能瓶颈、指导软硬协同设计也有重要的科研价值.

在强可扩展场景下, 尤其是每个核心只计算一个原子时, DeePMD势与传统经验势以及AIMD相比, 面临着不同挑战. 首先, DeePMD势计算需要神经网络框架的支持, 框架开销在每个线程计算任务较少时变得不可忽略, 且框架生成的算子冗余且低效; 其次, 计算过程涉及众多的矩阵计算, 而当前BLAS库在处理瘦高矩阵时, 计算效率较低, 无法最大化利用硬件计算资源; 最后, 由于每个节点分配的原子数较少, 大量的内存闲置, 造成严重的内存资源浪费.

针对以上问题, 我们对DeePMD势的计算过程展开了细粒度的优化, 特别是在每个核心只计算一个原子的场景下, 加快模拟速度, 实现最短的到解时间.

1 优化方法

(1) 无框架代码实现. DeePMD势采用TensorFlow框架进行模型推理, 在每个计算核心只计算一个原子时, 框架的算子调度、内存管理等额外开销会占据超过60%的计算时间. 为了彻底消除框架的额外开销, 我们梳理了DeePMD势的执行流程, 重新编写

了DeePMD势代码, 并进行了深度的算子融合. 我们将计算流程合并为6个步骤, 分别为环境矩阵生成、嵌入网络计算、拟合网络计算、拟合网络梯度计算、嵌入网络梯度计算和受力计算. 我们进一步通过算子融合和计算数据复用, 减少了计算量, 提升了访存效率. 在内存管理方面, 原始代码采用c++标准库中的std::vector进行数据管理, 而我们发现在模拟过程中, 由于原子数量变动, 会进行内存重新分配, 引入大量额外开销. 因此, 在初始化时, 根据本地原子数量的理论最大值, 为每个参与计算的数组均开辟了相应的最大空间, 避免了模拟过程中的内存动态分配的开销. 同时, 为了提高可移植性, 减小用户使用难度, 我们提供了两种运行方式：一种是保留了TensorFlow框架的版本, 但是TensorFlow框架仅用于读取模型参数, 不涉及计算; 另一种是利用我们的脚本提取TensorFlow的模型参数, 然后即可使用无框架的DeePMD势代码, 减少用户配置模拟环境时的时间开销.

(2) 面向瘦高矩阵的GEMM算子优化. DeePMD势采用MPI+OPENMP的并行编程模型, 计算任务会以原子为单位平均分配到每个OPENMP线程上, 而不是算子内部进行多线程并行. 由于$\mathcal{D}$矩阵的M维度大小等于原子数量, 这种并行模式在强可扩展场景下, 会导致每个线程在拟合网络中的GEMM算子计算时, M很小,计算效率很低. 为了提高fitting net中GEMM算子的计算效率, 我们基于CPU提供的向量指令, 实现了面相瘦高矩阵的高性能GEMM算子库. 通过切分矩阵N维度, 尽可能复用cache的中间结果. 另外, 在拟合网络梯度计算时, 需要计算梯度值与参数矩阵转置的乘积, 当M维度较小时, GEMM-NT算子的计算效率较低. 因此, 我们在初始化阶段即生成拟合网络中的参数矩阵的转置矩阵, 将拟合网络梯度计算的GEMM-NT算子转换为GEMM-NN算子, 进一步提升计算效率.

(3) 大内存下的Tabulate算子优化. 在强可扩展场景下, 节点内原子数量较少, 50%以上内存闲置. 我们发现可以通过较大的Tabulate算子表项来利用这部分空余内存, 提高算子计算效率. 在基准工作中, Tabulate算子以0.01为间隔进行建表, 对于水体系而言表项仅1360项. 在进行查表后, 还要完成五阶多项式计算才能得到最终结果. 若减小表项间隔, 增加表项数量, 在保持精度的同时可以减少多项式阶数, 减少计算量. 我们在此设多项式阶数为m, 嵌入网络最后一层神经元数量为d, 每个表项参数个数为$(m+1)\times d$. 假设取经验值$d=128$, 若间隔减小两个数量级, 设置为0.0001, 表项数量为$0.136M$. 采用三阶多项式和一阶多项式, 使用单精度的情况下, 所占内存分别为1.1 GB和557 MB(水分子中包含两种原子, 需要保存4份参数), 空余内存足够保存表项数据, 且表项密度足以保持计算精度. 我们同时使用了混合精度方法, 将嵌入网络计算、tanh算子、fitting net第二层和第三层GEMM算子转换为fp32精度, fitting net第一层GEMM算子使用fp16精度, 进一步提升计算效率.

2 结果与讨论

我们在A64FX(ARM)和Intel Xeon(R)Gold 6248 (X86)两种架构平台上分别进行了代码实现和测试, 平台具体参数如表 1所示. 由于使用的Intel平台为skylake架构, 不支持fp16计算, 因此在混合精度实现时Intel平台采用mixed-fp32配置. 我们选取了水分子和铜原子两个典型的物理体系作为测试用例. 水和铜的截断半径分别设置为6和8 Å, 模拟时间步长分别为0.5和1.0 fs, 氢、氧和铜原子的邻居原子数分别设置为46、92和512. 铜体系采用NVE系综, 水体系采用NVT系综(初始温度和结束温度均为330 K, 阻尼系数为0.5). Intel平台基准代码为DeePMD-kit-2.0.3, 而A64FX平台为DeePMD-kit-2.0.3-fugaku, 由于后者针对A64FX平台进行了优化, 因此基准性能较高. 分子动力学引擎采用LAMMPS-22Dec22.

首先是准确性测试, 表 2为不同多项式阶数和表项间隔配置下, 单步计算的能量和受力结果与DFT数据的误差. 可以看出, 利用混合精度、减少多项式阶数、减小表项间隔可以保持第一性原理计算精度. 我们选取mixed-fp16、一阶多项式、表项间隔0.0001作为最终版本. 图 1为水体系的径向分布函数(radial distribution function, RDF)图像, 用于描述水体系的结构. 在5 K原子体系下, 进行了10 K时间步的模拟. baseline为基准代码的结果, 其余为优化后的代码的结果, 图中4条曲线重合, 可以证明优化后的代码的正确性, 同时可以证明混合精度方法以及低阶多项式优化可以保持AIMD精度.

表 1 测试平台参数
Table 1 Test platform specifications

CPU	制造商	主频(GHz)	核心数	制造工艺	内存	指令集	特性	BLAS库
A64FX	富士	2.2	52	TSMC 7nm	4x8GB HBM2	Armv8.2-A	512-bit SVE	fjlapack
Xeon(R) Gold 6248R	英特尔	2.9	32	Intel 14nm	512GB DDR4	x86_64	AVX-512	MKL-2025

表 2 原子能量和受力与DFT数据误差
Table 2 Error between atomic energy and force with DFT data

精度	表项间隔	能量			受力		
		v5	v3	v1	v5	v3	v1
double	0.01	1.7×10^{-3}	——	——	4.4×10^{-2}	——	——
mixed-fp32	0.01	1.7×10^{-3}	1.7×10^{-3}	2.2×10^{-3}	4.4×10^{-2}	4.4×10^{-2}	5.0×10^{-2}
	0.001	1.7×10^{-3}	1.6×10^{-3}	1.7×10^{-3}	4.4×10^{-2}	5.1×10^{-2}	4.4×10^{-2}
	0.0001	——	——	1.2×10^{-3}	——	——	4.3×10^{-2}
mixed-fp16	0.01	4.0×10^{-3}	4.0×10^{-3}	3.5×10^{-3}	4.4×10^{-2}	4.4×10^{-2}	5.0×10^{-2}
	0.001	4.0×10^{-3}	4.0×10^{-3}	4.0×10^{-3}	4.4×10^{-2}	5.1×10^{-2}	4.4×10^{-2}
	0.0001	——	——	4.5×10^{-3}	——	——	4.3×10^{-2}

图 2所示为各优化方法带来的性能提升. 我们要测试在强可扩展场景下, 能实现的最快模拟速度. 在测试时, 采用仅使用一个进程, 原子体系大小为12, 每个核心绑定一个原子的方式, 避免了原子移动时的负载不均和进程间通信对计算性能的影响. 优化后的代码在两个平台上分别实现了34和303倍的性能提升. 由于Intel平台基准代码性能较低, 在图中未能显示. 我们的无框架代码实现和混合精度方法大幅提升了计算性能. 针对高性能GEMM算子的优化, 由于Intel的MKL库针对瘦高矩阵有着专门的优化, 因此替换为我们的GEMM库后, 并没有显著的性能提升; 而A64FX平台由于其fjlapack库在瘦高矩阵计算时效率较低, 因此性能提升较高. 对于Tabulate算子优化, 由于在强可扩展场景下, 主要是访存瓶颈, A64FX 平台采用了更大带宽的HBM2, 因此实现了更高的计算性能; 而在Intel平台上, 由于访存速度较慢, Tabulate 算子优化并没有带来理想的优化效果.

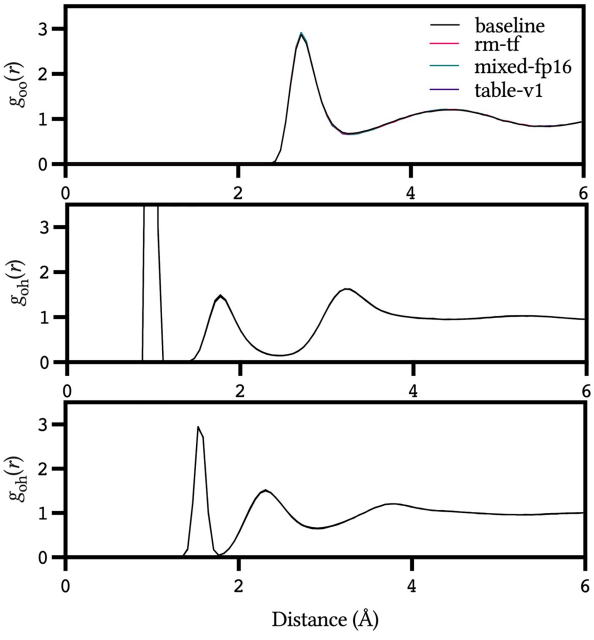

图 1 (网络版彩色)水体系RDF图像
Figure 1 (Color online) RDF of the water system

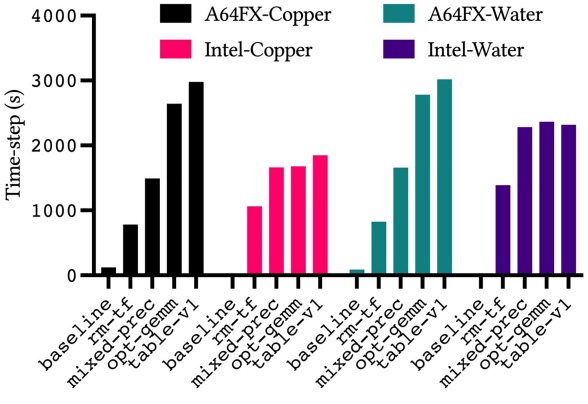

图 2 (网络版彩色)各种优化方法下的模拟速度
Figure 2 (Color online) Simulation speed under various optimization methods

3 结论

本文针对DeePMD-kit这一基于神经网络的第一性原理精度的分子动力学模拟工具展开了一系列优化. 我们移除了推理时的TensorFlow框架, 进行了细粒度的算子融合, 并通过设计面向瘦高矩阵的高性能GEMM算子库和大内存下的Tabulate算子优化方法进一步提升了计算效率. 我们的优化在单核性能上, 在A64FX和Intel两个平台上, 相比基准工作分别提升了34和303倍. 我们的代码当前已开源, 仓库地址为 https://github.com/HPC-AI-Team/lammps-deepmd-sc24.git. 其中table_v1_a64fx分支对应A64FX平台代码, table_v1_x86分支对应Intel平台实现.

虽然我们的优化带来了极大的性能提升, 但是测试结果也反映出, 在通用超算平台上, 即使进行极致的性能优化, 也很难实现1 μs/d(模拟时间间隔为1 fs/时间步时, 2.6万时间步/秒)的模拟速度. 这主要受限于通用平台的算力、访存性能,并且通用平台的数据通路和DeePMD计算数据流也无法完全匹配, 难以达到硬件的峰值性能. 另外, 通信问题和负载不均也是大规模模拟时的瓶颈, 由于进程间通信开销和原子移动带来的进程间负载不均衡, 大规模模拟时性能会下降约40%. 未来, 我们期望能够根据DeePMD-kit的计算和通信模式, 设计领域专用硬件, 进一步提升DeePMD-kit的模拟速度.

参考文献

1 Shaw D E, Dror R O, Salmon J K, et al. Millisecond-scale molecular dynamics simulations on Anton. In: Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis. 2009. 1–11
2 Duan X H. Optimization of melecular dynamics algorithms based on the Sunway TaihuLight supercomputer (in Chinese). Doctor Dissertation. Jinnan: Shandong University, 2020 [段晓辉. 基于“神威 • 太湖之光”的分子动力学算法优化. 博士学位论文. 济南: 山东大学, 2020]
3 Lennard-Jones J E. Cohesion. Proc Phys Soc, 1931, 43: 461
4 Tersoff J. New empirical approach for the structure and energy of covalent systems. Phys Rev B, 1988, 37: 6991
5 DiStasio R A, Santra B, Li Z, et al. The individual and collective effects of exact exchange and dispersion interactions on the *ab initio* structure of liquid water. J Chem Phys, 2014, 141: 084502
6 Ko H Y, Zhang L, Santra B, et al. Isotope effects in liquid water via deep potential molecular dynamics. Mol Phys, 2019, 117: 3269–3281
7 Car R, Parrinello M. Unified approach for molecular dynamics and density-functional theory. Phys Rev Lett, 1985, 55: 2471
8 Hafner J. *Ab-initio* simulations of materials using VASP: density-functional theory and beyond. J Comput Chem, 2008, 29: 2044–2078
9 Jia W, Cao Z, Wang L, et al. The analysis of a plane wave pseudopotential density functional theory code on a GPU machine. Comput Phys Commun, 2013, 184: 9–18
10 Jia W, Fu J, Cao Z, et al. Fast plane wave density functional theory molecular dynamics calculations on multi-GPU machines. J Comput Phys, 2013, 251: 102–115
11 Zhang Y Z, Kandpal H C, Opahle I, et al. Microscopic origin of pressure-induced phase transitions in the iron pnictide superconductors $A\mathrm{Fe_2As_2}$: an *ab initio* molecular dynamics study. Phys Rev B Condens Matter, 2009, 80: 094530
12 Raty J Y, Gygi F, Galli G. Growth of carbon nanotubes on metal nanoparticles: a microscopic mechanism from *ab initio* molecular dynamics simulations. Phys Rev Lett, 2005, 95: 096103
13 Jia W, Wang H, Chen M, et al. Pushing the limit of molecular dynamics with *ab initio* accuracy to 100 million atoms with machine learning. In: International Conference for High Performance Computing, Networking, Storage and Analysis. New York: IEEE, 2020. 1–14
14 Behler J, Parrinello M. Generalized neural-network representation of high-dimensional potential-energy surfaces. Phys Rev Lett, 2007, 98: 146401
15 Batzner S, Musaelian A, Sun L, et al. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. Nat Commun, 2022, 13: 2453
16 Musaelian A, Batzner S, Johansson A, et al. Learning local equivariant representations for large-scale atomistic dynamics. Nat Commun, 2023, 14: 579
17 Thompson A P, Swiler L P, Trott C R, et al. Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials. J Comput Phys, 2015, 285: 316–330
18 Wang H, Zhang L, Han J, et al. DeePMD-kit: a deep learning package for many-body potential energy representation and molecular dynamics. Comput Phys Commun, 2018, 228: 178–184

19 Huang Y P, Xia Y, Yang L, et al. SPONGE: a GPU-accelerated molecular dynamics package with enhanced sampling and AI-driven algorithms. Chin J Chem, 2022, 40: 160–168
20 Song K, Zhao R, Liu J, et al. General-purpose machine-learned potential for 16 elemental metals and their alloys. Nat Commun, 2024, 15: 10208
21 Unke O T, Chmiela S, Gastegger M, et al. SpookyNet: learning force fields with electronic degrees of freedom and nonlocal effects. Nat Commun, 2021, 12: 7273
22 Gasteiger J, Groß J, Günnemann S. Directional message passing for molecular graphs. 2020, arXiv: 2003.03123
23 Gasteiger J, Giri S, Margraf J T, et al. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. 2021, arXiv: 2011.14115
24 Hedman D, McLean B, Bichara C, et al. Dynamics of growing carbon nanotube interfaces probed by machine learning-enabled molecular simulations. Nat Commun, 2024, 15: 4076
25 Feng T, Zhao J, Lu G. Machine learning model to efficiently predict the structure and properties of $MgCl_2$-NaCl-KCl melts. Sol Energy Mater Sol, 2024, 272: 112903
26 Guo Z, Lu D, Yan Y, 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, 2022. 205–218
27 Lu D, Jiang W, Chen Y, et al. DP compress: a model compression scheme for generating efficient deep potential models. J Chem Theory Comput, 2022, 18: 5559–5567
28 Shaw D E, Adams P J, Azaria A, 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, 2021. 1–11
29 Santos K, Moore S, Oppelstrup T, et al. Breaking the molecular dynamics timescale barrier using a wafer-scale system. In: International Conference for High Performance Computing, Networking, Storage and Analysis. New York: IEEE, 2024. 1–13

Summary for “第一性原理精度的分子动力学模拟强可扩展性研究”

Strong scaling of molecular dynamics simulations with *ab initio* accuracy

Jianxiong Li[1,2], Guangming Tan[1,2] & Weile Jia[1,2*]

[1] *State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China*
[2] *University of Chinese Academy of Sciences, Beijing 100190, China*
* Corresponding author, E-mail: jiaweile@ict.ac.cn

Neural-network-based molecular dynamics (NNMD) simulations with *ab initio* accuracy have currently become the preferred method for modeling quantum physical phenomena in large-scale systems, such as phase transition and nanotechnology. Compared to empirical force fields (EFF), NNMD achieves superior accuracy and enables the simulations of more complex physical phenomena. Furthermore, in contrast to *ab initio* molecular dynamics (AIMD), NNMD offers significantly improved computational efficiency. Despite notable advancements have been made in scaling system sizes and improving accuracy by various NNMD methods, the simulation speed remains a bottleneck due to the intensive computations involved in neural network inference and high memory access. As a result, the computational time required for each time-step is at least on the order of several microseconds, limiting simulation speed to nanoseconds per day. However, numerous complex physical and chemical phenomena, such as chemical reactions in combustion processes and protein folding, require simulations at the microsecond or even millisecond scale to get meaningful results. Even state-of-the-art tools, such as DeePMD-kit, one of the fastest NNMD packages, require weeks of computation to reach these timescales. This significantly hinders research progress in fields such as materials science and pharmaceuticals. Therefore, enhancing the strong scaling of simulation tools, accelerating simulation speed, and reducing the time-to-solution are of substantial research significance. Meanwhile, Exploring the limits of strong scalability, particularly in scenarios where each core computes only a single atom, is crucial for identifying hardware performance bottlenecks and providing insights for future hardware design.

In this study, we present a series of fine-grained computational optimizations for DeePMD-kit, an open-source neural-network-based MD simulation tool with *ab initio* accuracy. We restructure the computational workflow of DeePMD-kit's inference process and implement a framework-free version, which effectively eliminates the massive overhead introduced by the AI frameworks such as TensorFlow. Additionally, we perform advanced kernel fusion and optimize data reuse to enhance memory access efficiency. We further optimize the GEMM kernel for tall-and-skinny matrices, which can maximize the vector unit utilization and make the data adapt to the cache size. And then, we refine the tabulate kernel by decreasing the interval and degree of polynomial to leverage the idle memory and reduce the computation overhead. Moreover, we bring out the mixed precision to further improve the simulation speed while maintain the *ab initio* accuracy.

We implement two optimized versions, which are oriented to ARM and X86 platforms, respectively. Numerical results show that our optimizations can preserve the *ab initio* accuracy while substantially improving computation efficiency, achieving speedups of 34x and 303x on the ARM and X86 platforms, respectively. As a result, the computational time for one time-step is reduced to as low as 331 microseconds. Despite our optimizations have brought significant performance improvements, the test indicates that achieving a simulation speed of 1 microsecond per day (26K time-steps per second, assuming the time-step unit is 1 femtosecond) remains challenging on the general-purpose platforms. This is mainly limited by the hardware computing power and the misalignment between the DeePMD-kit's dataflow and the hardware architecture. In the future, we aim to address these challenges by exploring domain-specific hardware acceleration through software-hardware co-design, further enhancing the simulation efficiency of DeePMD-kit.

high performance computing, DeePMD, molecular dynamics, neural network force field

doi: 10.1360/TB-2025-0027