EXTENDING QUANTUM TRANSPORT SIMULATIONS TO REALISTIC DEVICE DIMENSIONS THROUGH MACHINE-LEARNED ELECTRONIC STRUCTURES

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine-learned (ML) operator models can be trained to predict *ab initio* Hamiltonian matrices at significantly reduced computational cost compared to density-functional theory (DFT), extending electronic structure calculations to previously unfeasible scales and allowing for their integration with advanced technology computer aided design (TCAD) tools. Here, we introduce *MALOQ*, a model to train on and predict electronic structure matrices for complex materials consisting of thousands of atoms, described by large orbital basis sets, and covering a wide range of atomic elements. MALOQ extends a state-of-the-art, SO(2)-equivariant architecture with custom data-processing kernels to handle high-rank sparse Hamiltonian matrix data and a scalable edge-wise distribution to treat large atomic graphs. We demonstrate training (inference) on structures made of 3,000 (48,000) atoms, up to 384 (128) GPUs. For systems with 10k+ atoms, MALOQ produces Hamiltonian matrices 10,000$\times$ faster than standard DFT codes, at high spectral accuracy. These matrices can be input to a quantum transport (QT) solver, enabling the simulation of entirely-atomistic devices consisting of 21,600 atoms ($>$180k orbitals). Overall, the full MALOQ+QT pipeline outperforms the reference DFT+QT one by a factor $>$150$\times$, while maintaining comparable accuracy.

1 INTRODUCTION

The electronic Hamiltonian in matrix form ($\mathbf{H}$) plays a key role in modern technology computer aided design (TCAD) tools used in the semiconductor industry to simulate devices such as transistors *in silico* (Stettler et al., 2021; Kwon et al., 2021). Specifically, $\mathbf{H}$ serves as an input to quantum transport (QT) solvers which compute the 'current vs. voltage' characteristics of atomically-resolved 'devices'. The most advanced QT solvers operate on *ab initio* electronic structures obtained from density functional theory (DFT) calculations (Brandbyge et al., 2002). They are capable of treating systems with 10k+ atoms, in the presence of complex many-body effects (Vetsch et al., 2025), matching the physical dimensions of electronic devices fabricated by semiconductor companies (Agrawal et al., 2024). However, generating the input $\mathbf{H}$ with DFT scales with $\mathcal{O}(N_{orb}^3)$, N_{orb} being the total number of atomic orbitals in the structure of interest, ultimately limiting the problem sizes (number of atoms) that QT codes can handle. Existing works thus typically compute $\mathbf{H}$ for small, periodic systems and tile them to construct device-scale models (Ducry et al., 2026).

Meanwhile, machine-learned (ML) operator models are rapidly establishing themselves as practical means to predict the electronic structure of atomic systems and to generate their $\mathbf{H}$ with $\mathcal{O}(N_{orb})$ scalability. Early Hamiltonian models took advantage of the architectures first developed for ML interatomic potentials (MLIPs), extending them towards the higher-rank tensors that electronic interactions must be decomposed into. More recent models can predict Hamiltonians beyond small molecules and crystalline materials, tackling 'universal' datasets of structures with 100+ atoms (Kaniselvan et al., 2026; Li et al., 2022, 2025). **Similarly to how MLIPs have revolutionized MD simulations, machine-learned Hamiltonian (MLH) models will unlock new capabilities in TCAD investigations of nanoscale electronic devices.**

Training and inference on large-scale Hamiltonian matrix data, however, introduces a different set of computational challenges than previously encountered by MLIPs. First, the spherical features used

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

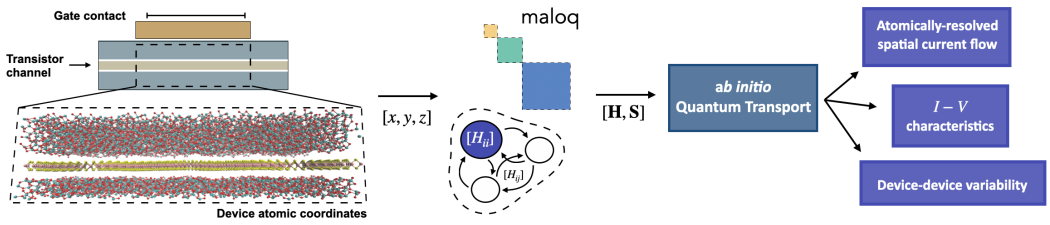


Figure 1: **Overview of the 'atomic structure (xyz) → learned H → quantum transport' workflow.** The electronic transport region is first constructed atom by atom, here a transistor made of a monolayer MoS_2 channel surrounded by HfO_2 oxide layers. The resulting set of atomic coordinates is passed to the *MALOQ* package, which uses it to predict the $\mathbf{H}$ matrix corresponding to the initial device geometry. With this matrix as input, a quantum transport solver can return the $I-V$ characteristics and the bond-resolved electronic current flowing through the structure investigated.

within embeddings must match the dimensions of high-rank orbital interaction data, typically up to f (and sometimes g) orbitals, requiring maximum angular degree (l_{max}) of 6 to 8. Second, while MLIPs typically enforce cutoffs of ∼6-8 Å between interacting atoms, treating orbital interaction data requires handling dedicated edge embeddings that extend beyond interatomic distances of 10 Å (Kaniselvan et al., 2026). Larger cutoffs create densely connected graph representations with node degrees over 100, significantly increasing the total volume of embeddings. Finally, processing orbital interaction data necessitates frequent basis transformations during training and inference to convert between 'label' and 'matrix' representations of $\mathbf{H}$. These differences with MLIPs substantially increase the computational cost and memory footprint of both MLH training and inference.

To enable operator prediction *at scale*, we introduce *MALOQ* (*M*assively *A*ccelerated *L*earning of *O*perators for *Q*uantum Transport), an accelerated ML framework to train electronic structure matrices and predict them for arbitrary atomic elements and system sizes, from large molecules to devices made of 20k+ atoms (**Fig. 1**). MALOQ builds on the ML backbone of HELM (Kaniselvan et al., 2026), which shows state-of-the-art performance in $\mathbf{H}$ prediction for universal molecule datasets. Architecturally, it uses a message-passing (MP) graph neural network (GNN) leveraging equivariant spherical channel network (eSCN) convolutions (Passaro & Zitnick, 2023). On top of it, we introduce computational innovations to enable massively parallel training and inference of electronic structure matrices. Specifically, we include custom kernels to perform (1) 'matrix ↔ label' conversions to pre-process $\mathbf{H}$ matrices into training labels and reconstruct them from predicted spherical coefficients, (2) rapid computation of the generalized rotation (Wigner-D) matrices required by eSCN convolutions, and (3) distributed graph-level training and inference on large material systems. Using an efficient point-to-point communication scheme and edge-wise partitioning of input atomic graphs, we show efficient strong scaling up to 384 (128) NVIDIA GH200 GPUs for training (inference), and weak scaling above 90% efficiency up to 48,000 atoms (128 GPUs).

Finally, we integrate trained MALOQ checkpoints with an *ab initio* QT solver (Vetsch et al., 2025) to examine a realistically extended transistor structure at atomistic resolution. While previous simulations relied on exploiting structural repetition to reach nanoscale dimensions, here we consider an MoS_2/HfO_2 transistor consisting of 20k+ atoms, establishing a new record for QT simulations of a structurally irregular atomistic system. We then visualize the electronic current flowing through it at a bond-resolved level, highlighting the influence of defects at semiconductor/oxide interfaces. MALOQ thus introduces materials-level, ML-accelerated capabilities for atomistic semiconductor device design, enabling relevant transistor architectures and memory cells to be modeled at a DFT level of accuracy.

2 Background & Related Work

From electronic structure to quantum transport. The training data for MLIP and MLH models is typically generated with a DFT code that solves the Kohn-Sham equations self-consistently until the electron density $\rho(\mathbf{r})$ and electrostatic potential $V(\mathbf{r})$ at position $\mathbf{r}$ of the atomic system of interest have converged, after N_{iter} iterations (Kohn & Sham, 1965). Many DFT implementa-

Application	Ref.	Symm.	l_{max}	Distr.	Max # Atoms (N_A)	
					Training	Inference
QHNet	Yu et al. (2023b)	SO(3)	4	✗	12	12
QHFlow	Kim et al. (2025)	SO(3)	4	✗	29	29
WANet	Li et al. (2025)	SO(2)	6	✗	150	150
Aug. part.	Xia et al. (2025)	SO(2)	4	✗	600	3,000
HamGNN	Huang et al. (2026)	SO(3)	6	✗	60	444
HELM	Kaniselvan et al. (2026)	SO(2)	6	✗	150	300
DeepH-pack	Li et al. (2026)	SO(2)	4	✗	216	4,096
MALOQ	**This work**	SO(2)	8	✓	**3,000+**	**48,000**

Table 1: Selection of GNN applications for MLH, including the operation ('Symm.') to update node/edge features (SO(3) tensor products, or SO(2) eSCN convolutions), the possibility of distributed compute environment (Distr.), the largest number of atoms reported for training and inference (N_A), and the maximum angular momentum that can be treated (l_{max}). Note that all models use multiple message-passing layers to incorporate higher-order interactions, except Xia et al. (2025).

tions operate in a basis of localized atomic orbitals $|\varphi_i\rangle$, often constructed from contracted Gaussian functions (Kühne et al., 2020; Neese, 2011) that transform like spherical harmonics ($Y_m^l(\hat{r}')$) under rotation. Each spherical harmonic is specified by an angular momentum degree l and order $m \in \{-l, \ldots, l\}$, e.g., s ($l = 0$), p ($l = 1$), d ($l = 2$), $\cdots$, giving rise to a generalized eigenvalue problem (GEVP): $\mathbf{H}\psi = \varepsilon\mathbf{S}\psi$. The Hamiltonian matrix $\mathbf{H}$ has entries $H_{ij} = \langle\varphi_i|\hat{H}(\mathbf{r})|\varphi_j\rangle$ where $\hat{H}(\mathbf{r})$ is the Hamiltonian operator. The overlap matrix $\mathbf{S}$ is made of terms $S_{ij} = \langle\varphi_i|\varphi_j\rangle$ that describe finite overlaps between the localized orbitals, and can be precomputed analytically from the orbital basis. The complexity of the full method scales as $\mathcal{O}(N_{orb}^3)$ per iteration. At the end of these calculations, atomic- and molecular-level properties, such as forces and energies, can be derived from the final, converged $\mathbf{H}$. Both $\mathbf{H}$ and $\mathbf{S}$ are matrices of size $N_{orb} = \sum_{k=1}^{N_A} N_{orb,k}$, where N_A is the total number of atoms, $N_{orb,k}$ the number of orbitals (basis elements) of atom k, and the index k runs over all atoms accounted for in the simulation domain.

To model the quantum transport properties of a device, the original GEVP must be modified by adding 'open-boundary conditions' (Datta, 1995), which turn it into a linear system of equations with the energy ε as input. From the energy-dependent $\psi(\varepsilon)$, a transmission function $T(\varepsilon)$ can be computed, either from one side of the considered device to the other or between two neighboring atoms. Through the Landauer-Büttiker equation (Datta, 1995), the electronic current I_d is obtained by weighting $T(\varepsilon)$ with the electron occupation probability of the device electrodes. This approach is known as the 'quantum transmitting boundary method' (QTBM), which we detail in Appendix A.

Hamiltonian prediction models. MLH prediction models were initially built as rotationally equivariant GNNs using Clebsch Gordon tensor products (CGTPs) to mix features (Gong et al., 2023; Yu et al., 2023b). However, CGTPs scale poorly with l_{max}, $\mathcal{O}(l_{max}^6)$ with dense-algebra kernels, $\mathcal{O}(l_{max}^5)$ when the sparsity of the coefficients is accounted for (Xie et al., 2025). MLH architectures capable of treating systems beyond small molecules and crystalline materials have thus taken a different route, using eSCN convolutions in place of CGTPs. By implementing local embedding rotations through Wigner-D matrices, the dimensionality of the spherical interactions is reduced from SO(3) to SO(2). Concretely, eSCN-based models maintain rotational equivariance through linear-layer operations, and thus scale with $\mathcal{O}(l_{max}^3)$ (Passaro & Zitnick, 2023). Currently, nearly all MLH prediction models treating structures with 100+ atoms rely on this approach (Li et al., 2025; Kaniselvan et al., 2026; Li et al., 2026) (see **Table 1**). Note that existing MLH models have so far not integrated distributed training approaches, unlike MLIPs (see Appendix B).

3 Implementation and Optimization

3.1 Architecture

MALOQ consists of two components: (1) a learnable ML model to map atomic structures to spherical tensors (2) and 'non-learnable' data-processing pipelines. The different stages of its internal

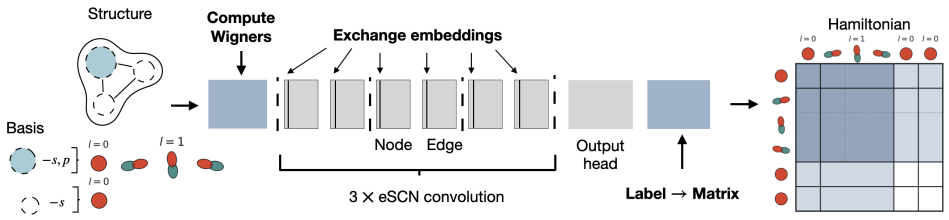


Figure 2: **Overview of the components within MALOQ.** Its non-learnable (blue) and learnable (grey) components are schematized as blocks that, from left to right, map an input atomic graph defined by constituent atomic coordinates and elements to a predicted Hamiltonian matrix.

workflow during inference are illustrated in **Fig. 2**. The GNN is initialized from the graph representation of an input atomic structure. The first 'non-learnable' step, labeled 'Compute Wigners' in **Fig. 2**, creates generalized rotation (Wigner-D) matrices rotating all edges of the atomic graph to a common $(+y)$ axis. These precomputed rotation matrices are then used within the second, learnable component of the model: the eSCN convolutions in each of three message-passing (MP) layers (center of **Fig. 2**). Learnable interactions occur through constrained linear layers that preserve rotational equivariance, following Passaro & Zitnick (2023). Finally, the directed 'messages' created by interacting nodes along each graph edge are either aggregated onto their target nodes (in the Node blocks) or used directly to update the embeddings of edges (in the Edge blocks). At the end of the MP operations, each node or edge of the atomic graph is described by an embedding of multi-channel spherical harmonic coefficients, which has been processed through three layers of learnable eSCN convolutions. At this point, the final 'non-learnable' component of MALOQ ('Label→Matrix' in the right part of **Fig. 2**) uses these predicted 'Labels' to reconstruct the Hamiltonian in matrix representation.

3.2 Data pipelines

Wigner generation. The embeddings of every node (n_i) and edge (e_{ij}) of the input atomic graph take the form of spherical harmonic coefficients of rank $l = 0...l_{max}$. Each of them is expanded into E channels - the dimensions of every n_i and e_{ij} embedding processed is thus $(l_{max}+1)^2 \times E$. The idea behind eSCN consists of performing a rotation R of the original atomic system $\hat{r}$ into $\hat{r}' = \mathbf{R}\cdot\hat{r}$ (where $\hat{r}/\hat{r}'$ are normalized direction vectors) to align all $\hat{r}_{ij}$ edge vectors to a common $(+y)$ axis. This requires multiplying the embeddings n_i/e_{ij} by a direct sum of 'Wigner-D' matrices $\mathbf{D}^l(R)$ of degree l, where the necessary rotation matrices $\mathbf{R}$ are precomputed, i.e., $n'_i/e'_{ij} = \bigoplus_{l=0}^{l_{max}} \mathbf{D}^l(\mathbf{R}) \times n^l_i/e^l_{ij}$. The upper part of **Fig. 3(a)** shows the block-diagonal structure of these Wigner-D matrices acting on the embeddings n_i/e_{ij}. Practically, creating the Wigner-D matrices involves (1) computing a set of Euler angles α, β, γ corresponding to each edge in a given atomic graph, and then (2) using them to determine the $(2l{+}1) \times (2l{+}1)$ entries of $\mathbf{D}^l(\mathbf{R})$.

A reference PyTorch implementation is provided by the e3nn library (Geiger & Smidt, 2022) to compute each $\mathbf{D}^l(\mathbf{R})$ sub-matrix separately, but for all graph edges in a batch. Each sub-computation involves a chain of four small matrix multiplications, executed via separate kernel launches, materializing the intermediate results in GPU global memory. Hence, a significant speedup can be gained by fusing all kernels participating in the full $\hat{r}_{ij} \rightarrow \mathbf{D}^l(\mathbf{R})$ operation chain for every $\hat{r}_{ij}$ and orbital degree l, vastly increasing arithmetic intensity. We thus introduce a Triton kernel which (1) fuses the calculation of the Euler angles and of all matrix products that are required to obtain the $\mathbf{D}^l(\mathbf{R})$ sub-matrices of degree $l = 0...l_{max}$, and (2) implements a block-packing strategy to group different ls (lower part of **Fig. 3(a)**). To do so, we take advantage of the regular angular dimensions of each l, which are fixed by the underlying physics. As the matrix product in Triton (tl.dot) uses fixed shapes (e.g., 16×16, 32×32...), packing the sub-matrices for multiple l-components together minimizes padding overhead and fully utilizes the fixed-size tiles.

Matrix ↔ Label Conversions. To process a dataset of Hamiltonian matrices during training, we need to (1) extract the individual orbital interaction blocks into equally-shaped buffers and (2) perform a basis transformation to convert them into labels. The reverse operation (labels to matrices)

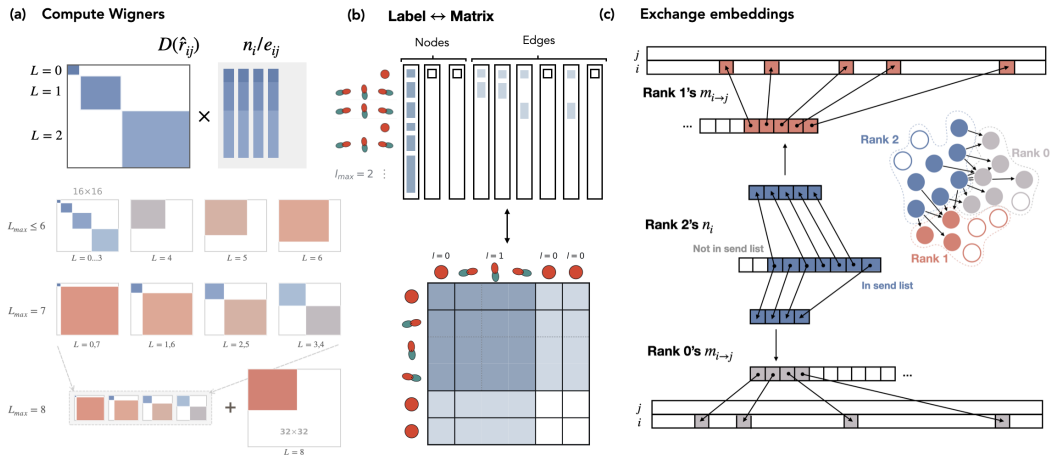


Figure 3: **Implementation and optimization of the data pipelines and custom communication scheme.** (a, top) Schematic of Wigner-D matrices $\mathbf{D}(\mathbf{R})$ multiplying node/edge embeddings n_i/e_{ij} to align every edge $\hat{r}_{ij}$ from the graph with the $+y$-axis. (a, bottom) Block-packing strategy implemented in Triton for $\mathbf{D}(\mathbf{R})$ matrix products (Triton tl.dot): ℓ-blocks are packed into 16×16 tiles (or 32×32 for ℓ=8) to maximize the tensor core utilization of GPUs. (b) Mapping between the label (top) and matrix (bottom) representations of data for a toy H_2O molecule with a minimal basis (Oxygen: s, p, Hydrogen: s), so that l_{max}=2. (c) Illustration of the send/recv operation to communicate node embeddings n_i from Rank 2 to Ranks 0 and 1, for the partitioned graph pictured on the right side. Each rank corresponds to a different GPU. The arrows indicate the edges starting from nodes belonging to Rank 2's partition and ending in one of the other partitions: 4 (5) Rank 2's nodes are connected to Rank 0 (1)'s nodes through inter-partition edges. The data is first copied to a buffer before being sent to the targeted Rank. There, it is unpacked into a local message buffer $m_{i\to j}$.

occurs during inference. The required basis transformations are described in Appendix C, whereas the 'matrix' and 'label' representations are illustrated schematically in **Fig. 3(b)**. First, the entries of the Hamiltonian, in its matrix or label form, are extracted before the matrix operations in **Eqn. (5)** can be executed. While the latter can be done efficiently using PyTorch tensors, mapping the Hamiltonian entries from their label to their matrix representations (and vice versa) rapidly becomes a bottleneck for large training datasets or inference on large atomic systems. We introduce a dedicated CUDA kernel to handle the two-way transformation between batches of (or single) matrices and labels. Within this kernel, each thread processes a single matrix element, i.e., orbital-orbital interaction block, and each thread block processes one node (diagonal blocks of $\mathbf{H}$) or edge (off-diagonal blocks of $\mathbf{H}$). The kernel uses a precomputed 'orbital template' $\mathcal{T}(Z_i, Z_j)$ containing the row and column slices of $\mathbf{H}_{ij}$ corresponding to the set of orbitals interaction between atom i of atomic element Z_i and atom j of atomic element Z_j.

3.3 Graph-level distribution of embeddings

At the start of each MP layer, a set of messages $m_{i\to j}$, consisting of the concatenated node embeddings $n_i \oplus n_j$, is assembled. This set of messages undergoes eSCN convolutions, which mix the features of n_i and n_j, and is then either aggregated onto node j (node update) or used to update the edge embedding e_{ij} (edge update). For large atomic structures, the graph's node and edge embeddings greatly exceed the memory of a single GPU. As a consequence, the graph must be partitioned and distributed across GPUs, with inter-process communication occurring at the start of each node/edge block within a MP layer (**Fig. 2**).

We implement embedding exchanges across GPUs using the PyTorch distributed NCCL backend. Each process (1) packs its own n_i into a separate send buffer for all its neighbors (j), (2) sends/receives those buffers through point-to-point communication, and (3) unpacks the received n_i before inserting them into its message tensors. This communication scheme is visualized in **Fig. 3(c)**, where a process, Rank 2, packs and sends its local n_i to its neighbors, Ranks 0 and 1. After receiving this data, they insert it into their local message buffer $m_{i\to j}$. During the backward pass,

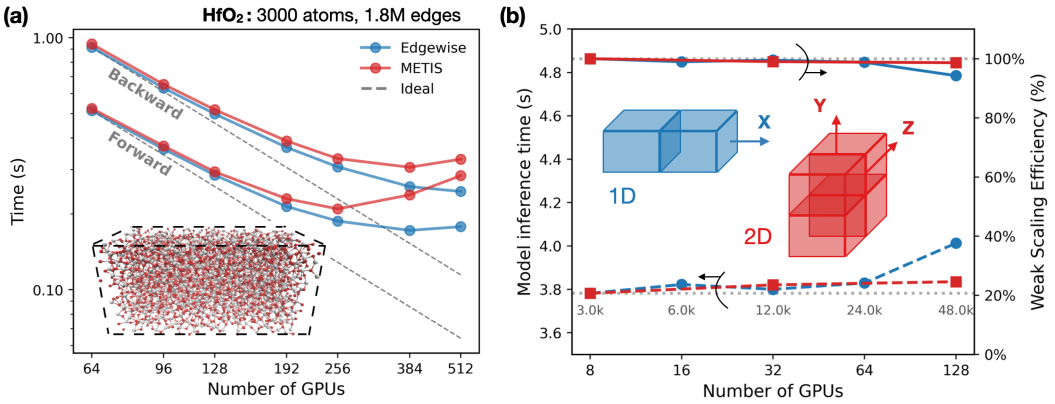


Figure 4: **Scalability of MALOQ's training and inference.** (a) Strong scaling for training, including the runtime of the forward/backwards passes distributed over different numbers of GPUs for the single 3,000-atom, 1.8M-edge HfO_2 structure in inset. Runtimes are the median over 200 epochs, with the first 20 omitted ('warmup' runs). (b) Weak scaling inference time with edge-wise partitioning. The problem size is defined by tiling the same HfO_2 structure as in (a) in 1D or 2D. The dashed lines refer to time, the solid ones to scaling efficiency.

the reverse communication pattern must be executed. In addition, we partially hide communication latency by overlapping the indexing of local nodes into message tensors with the asynchronous transfer of remote node embeddings. The list of local node embeddings is partitioned such that those requiring transmission to remote ranks are contiguous in memory (see Rank 2 in **Fig. 3(c)**). Note that this communication scheme preserves equivalence to single-GPU execution across both forward and backward passes, without impacting model accuracy beyond floating-point accumulation errors.

It remains to partition the graph in order to (1) evenly distribute the workload (better load balance) and (2) reduce communication time (latency and volume of embedding exchanges). MALOQ is designed to handle partitioning of both batches of irregularly-sized molecules or individual large structures. To cover this range of possible inputs, we include two graph partitioning strategies leveraging (1) the Metis algorithm as a standard 'minimum cut' reference to minimize communication overhead (Karypis & Kumar, 1998), or (2) an edge-wise distribution of the material graph.

4 Scaling to Large Structures

DFT calculations of materials systems typically apply 3D periodic boundary conditions. In small unit cells, two atoms can, therefore, interact with each other twice: once within their reference unit cell and once through the periodic image. To avoid learning this double interaction, the data used to train MALOQ must fully saturate the radial decay of H_{ij} orbital interaction terms, by satisfying $L_{\text{min}} = \min(L_x, L_y, L_z) \geq 2r_{\text{cut}}$, where L_{min} is the minimum cell dimension and r_{cut} is the interatomic distance beyond which H_{ij} becomes negligible. The r_{cut} identified in this way is also used as an interaction cutoff radius for the graph. Depending on the orbital basis set and atomic elements involved, orbital interactions typically occur beyond $r_{\text{cut}} = 10$ Å (Kaniselvan et al., 2026), necessitating values of L_{min} above 2 nm.

This restriction effectively implies that, for a ML Hamiltonian model to generalize to material structures significantly larger than encountered during training, *the training data must include sufficiently 'large' atomic graphs*. When processing individual structures at these scales, distributing the memory across multiple GPUs becomes necessary.

Scalable training over large structures. The graph partitioning strategies of MALOQ are first evaluated in **Fig. 4(a)** where we report its strong scaling performance for the HfO_2 material dataset from Xia et al. (2025). Here, a single structure of size $2.6\times2.6\times5.3$ nm^3 , corresponding to one batch, contains 3,000 atoms, and roughly 1.8 million edges. We run all experiments on the Alps supercomputer, which has 4 GH200 GPUs per node (further detail on the training environment in

Appendix D). Measurements start at a baseline of 64 GPUs, since this is the minimum number required to fit the full graph in memory for training. We measure strong scaling efficiency up to 512 GPUs - at this distribution, each rank has, on average $\sim 3 \times 10^3$ graph edges. Beyond 192 GPUs, the scaling performance under the two partitioning approaches implemented (Metis and edge-wise) begins to deteriorate. This occurs because (1) the runtime is mostly dictated by the number of edges, and (2) communication is partially hidden. It follows that the reduced load imbalance provided by an edge-wise partitioning outperforms the communication-avoiding Metis approach. Under an edge-wise partition, MALOQ sustains 50% strong scaling efficiency on up to 384 GH200 GPUs. After this point, decrease in parallel efficiency occurs due to loss of processing throughput through the core eSCN convolution block (**Fig. D.1(b)**).

Weak scaling for inference. Non-idealities affecting materials' properties such as phase boundaries (Zhou et al., 2023), interfaces between different compounds (Dossena et al., 2025), or amorphous layers, can only be captured in graphs representing structures made of several thousand atoms and with nanoscale dimensions, well beyond the computational capabilities of DFT. To demonstrate the ability of MALOQ to produce electronic structures at these scales, we present in **Fig. 4(b)** its weak scaling efficiency during inference, using the same amorphous-HfO_2 system as in **Fig. 4(a)**. We start on 8 GPUs, which is required to fit the baseline structure made of 3,000 atoms ($\sim$1.8 million edges) in inference mode, and scale up to 128 GPUs and 48,000 atoms ($\sim$28 million edges). To explore both ideal and realistic increases in problem size, we tile the initial 3,000-atom HfO_2 cell according to two strategies: (1) 1-D tiling, where the cell is repeated only along the x-direction, and (2) 2-D tiling, where the cell is repeated along the y- and z-dimensions. In both cases, we use an edge-wise partition of the tiled graphs. Note that we time only the model components, as matrix$\rightarrow$label conversion and Wigner-D generation remain under 10 ms, collectively several orders of magnitude faster than reference numpy/torch implementations (**Fig. F.1(c)** and **Table 3**). MALOQ achieves a negligible loss of weak-scaling efficiency for inference on these structures, 94% with 1-D scaling and 98% with 2-D scaling over 48,000-atom structures distributed on 128 GPUs.

We note that MALOQ's graph-level partitioning and communication schemes can also be used to more efficiently subdivide batches of molecules between GPUs and train them (Appendix E). Furthermore, numerical evaluations of the data pipeline optimizations are provided in Appendix F.

5 Integration with device-level transport modeling

MoS_2 and other transition metal dichalcogenides (TMDCs) are attracting a lot of attention from the semiconductor industry to replace Silicon field-effect transistors (FETs) in next-generation nanoribbon architectures (Mortelmans et al., 2024). Wafer-scale MoS_2 transistors were reported with gate contacts as short as 4.8 nm, where Si devices no longer operate properly (Kwon et al., 2026). However, the 2D technology suffers from non-ideal interfaces between the semiconductor channel and the oxide layers around it. *Ab initio* quantum transport studies found that inhomogeneity at the TMDC/oxide interface creates electrostatic potential variations against which electrons scatter (Van Troeye et al., 2025; Dossena et al., 2025), making an explicit inclusion of the oxide layers necessary for accurate device modeling. This requires constructing large atomic systems made of 10k+ atoms, obtaining their electronic structure ($\mathbf{H}$) with DFT (e.g., CP2K (Kühne et al., 2020)), and performing transport calculations with, for example QTBM (Lent & Kirkner, 1990). Practically, the computational bottleneck in this process is the electronic structure calculation. While previous studies circumvented this issue by tiling relatively small unit cells (Ducry et al., 2026), MALOQ can go beyond such approximations and provide non-periodic $\mathbf{H}$ for realistic device dimensions.

Dataset generation and training. We consider a monolayer MoS_2 FET surrounded by HfO_2 layers. To create the atomic structures of our training dataset, we extract periodic snapshots along high-temperature molecular dynamics trajectories of MoS_2/HfO_2 samples performed with LAMMPS (Thompson et al., 2022) parameterized with MACE-MH1 (Batatia et al., 2022). Using the CP2K (Kühne et al., 2020) DFT code with a double-ζ valence polarized (DZVP) basis for MoS_2, and a single-ζ (SZV) basis for HfO_2, we find that $r_{\text{cut}} \sim$10 Å (see **Fig. G.1(a)**). To satisfy the constraint of $L_{\text{min}} = \min(L_x, L_y, L_z) \geq 2r_{\text{cut}}$, our training structures are 3.16×3.16×3.29 nm^3, and consist of 2,088 atoms (16,608 orbitals). A representative example with the corresponding Hamiltonian matrix is depicted in **Fig. 5(a)**. The full training dataset contains 50 structures, 104,400 atoms,

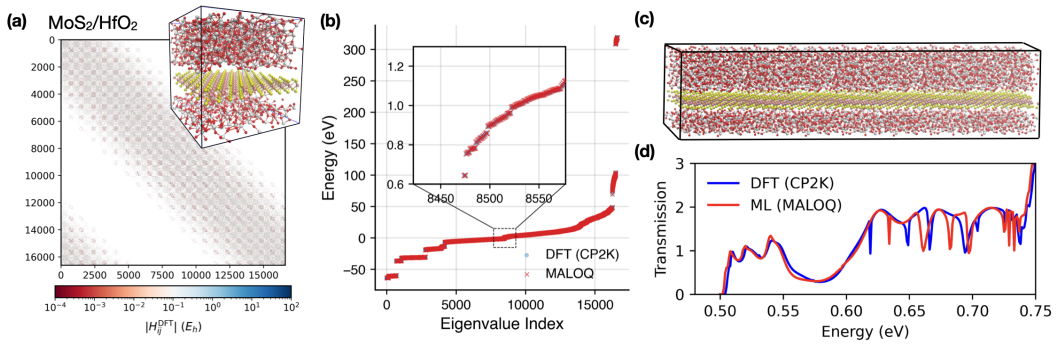


Figure 5: **Predicting Hamiltonian matrices at the device scale.** (a) Example of a 3.16×3.16×3.29 nm^3 MoS_2/HfO_2 training sample made of 2,088 atoms (Mo: pink, S: yellow, Hf: red, O: gray) and 16,608 orbitals superimposed on the corresponding DFT Hamiltonian matrix. (b) Comparison between all eigenvalues (16,608) obtained by diagonalizing a Hamiltonian matrix produced by DFT (blue circles) and by MALOQ (red crosses) for a MoS_2/HfO_2 system with the same dimensions as in (a). The inset zooms into the edge of the MoS_2 conduction band. (c) Schematic view of a 12.64×3.16×3.29 nm^3 MoS_2/HfO_2 structure consisting of 8,352 atoms. (d) Transmission function through the structure in (c), near its conduction band edge, as computed with MALOQ's and CP2K's $\mathbf{H}$, after adding pure MoS_2 extensions on each side of the device to inject electrons.

∼6.9B orbital-interaction coefficients, and required a total of **30 node hours (NHs)** to be generated. Note that, due to the nearsightedness of the H_{ij} elements, each $\mathbf{H}$ matrix is sparse, making a graph representation especially efficient. Before training, we post-process all $\mathbf{H}$ matrices by shifting their equilibrium Fermi level μ to zero: $\mathbf{H} = \mathbf{H} - \mu\mathbf{S}$. After performing this global shift, the electronic energies of elements across different atomic samples are similarly aligned, and can be referenced out according to the procedure detailed in Kaniselvan et al. (2026). Training was performed with one single structure per batch, each distributed across 40 GPUs, and consumed **120 NHs** on Alps.

Spectral accuracy for large-scale structures. On a validation structure of the same dimensions, MALOQ shows a **0.0378 meV mean absolute error (MAE)** in the matrix elements of the predicted Hamiltonian ($\mathbf{H_{ML}}$) with respect to its DFT label data ($\mathbf{H_{DFT}}$). A parity plot of the matrix elements is provided in **Fig. G.1(b)**. Solving the electronic structure GEVP for MoS_2/HfO_2 with $\mathbf{H_{ML}}$ results in excellent agreement in the eigenvalues (EVs) around the MoS_2 conduction band edge, when compared to DFT (**Fig. 5(b)**), with an average error of 38 meV across all 16,608 eigenvalues. This remaining error is mostly distributed in the highest unoccupied states (**Fig. G.1(c)**) that do not contribute to the electronic current. While it is difficult to compare MAE across distinct studies because of different structure sizes, atomic species, and basis sets, we can nevertheless put our results into perspective by mentioning (1) Li et al. (2026), which considered an amorphous SiO_2 sample made of 4,096, and reported a validation MAE of 0.99 meV for the Hamiltonian matrix elements, and (2) Xia et al. (2025) with 2.58 meV on 3,000-atom HfO_2 structures.

MALOQ's ability to handle larger test structures beyond the training data is illustrated in **Fig. 5(c-d)**, using an MoS_2/HfO_2 stack of 12.64×3.16×3.29 nm^3 with 8,352 atoms. The transmission function $T(\varepsilon)$ through this sample, as calculated with QTBM (Appendix A) and $\mathbf{H_{ML}}$, accurately reproduces the DFT one over a large energy range (**Fig. 5(d)**). This physical quantity determines the fidelity of the electronic current I_d via **Eqn. (2)**. Taking the ML (DFT) $T(\varepsilon)$ as input, setting the source Fermi level E_{FS} 100 meV above the conduction band edge of MoS_2, and applying a voltage of 0.6 V results in a current I_d equal to 6.449 (6.45) μA, **an error of <0.1%** in this case. This value can be compared to the ∼30% error obtained by Xia et al. (2025) on a similar transport problem.

Electronic transport through devices. Generating the Hamiltonian matrix of the 12.64×3.16×3.29 nm^3 MoS_2/HfO_2 sample in **Fig. 5(c)** with MALOQ costs **<0.01 NH** on Alps, while it takes >20 NHs with CP2K, assuming convergence within N_{iter} =100 self-consistent iterations in DFT. This number of iterations represents a best-case estimation based on small-scale calculations. To further quantify MALOQ's acceleration, we report in **Fig. 6(a)** the computational cost stemming from the generation of $\mathbf{H}$ with it and with CP2K, from 2,088 up to 41,760 atoms.

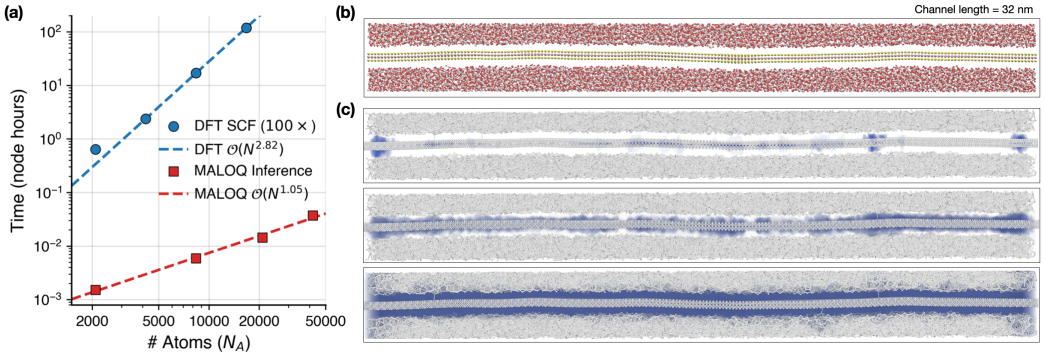


Figure 6: **Modeling electrical current flow through 2D MoS_2/HfO_2 devices.** (a) Computational cost in node hours of DFT (blue circles) and MALOQ (red squares) to produce the **H** of systems made of 2,088 up to 41,760 atoms on GH200 GPUs. The DFT calculations converge on average within N_{iter}=100 self-consistent field iterations, while the ML approach is single-shot. The dashed lines serve as visual guides and show a time complexity of $\mathcal{O}(N^{2.82})$ for DFT and $\mathcal{O}(N^{1.01})$ for MALOQ. (b) Atomic structure of a 32nm-long MoS_2/HfO_2 sample made of 21,600 atoms and used as the channel of a transistor architecture similar to the one in **Fig. 1**. (c) Electronic current flow through the device in (b) represented with three isosurfaces of decreasing magnitude (from top to bottom). Current crowding due to defects at the semiconductor/oxide interface and leakage currents penetrating into the surrounding oxide can be observed.

The DFT simulation of the largest structure is projected to exceed ∼800 NHs based on a measured time complexity of $\mathcal{O}(N^{2.82})$, close to its algorithmic complexity of $\mathcal{O}(N^3)$. For the same size, MALOQ's linear-scaling inference produces **H** in $<$**0.1 NH**, a $>$10,000$\times$ speedup over CP2K's standard algorithm. This massive acceleration allows us to go beyond DFT capabilities and model a realistic MoS_2/HfO_2 FET made of 21,600 atoms with a length of 32 nm, no structural tiling nor internal repetition, and the geometry of **Fig. 1**. For this system, **a single inference takes 0.02 NH, instead of 200 NHs with DFT, thus fully amortizing the computational cost of dataset generation (30 NHs) and training (120 NHs).**

This large-scale atomic structure, schematized in **Fig. 6(b)**, was created through a separate melt-and-quench procedure. After generating its Hamiltonian matrix with MALOQ, we pass it to the QT code QuaTrEx (Vetsch et al., 2025), self-consistently coupling *ab initio* quantum transport via QTBM and Poisson's equation. This returns the bond-resolved electronic current of the 21,600-atom device within **1.2 NHs**. The MALOQ+QT pipeline thus outperforms its DFT+QT counterpart by **a factor $>$150$\times$** ($>$200 vs. 1.22 NHs). Contrary to standard simulations of MoS_2 transistors (Ghafoor et al., 2026), we explicitly include the surrounding amorphous dielectric in the same atomically-resolved domain as the 2D monolayer. This let us track not only how the electronic current flows through the semiconductor channel, but also how it leaks into the oxide layers, and is occasionally diverted by the presence of interface defects. To visualize these phenomena, we plot in **Fig. 6(c)** three isosurfaces of the device ON-state current corresponding to different magnitudes, highlighting current crowding due to the roughness of the neighboring oxides. These effects can only be captured in full DFT-level simulations, without tiled cells to avoid non-physical repetitions of the same defects.

6 Conclusion & Outlook

We introduce MALOQ, a scalable, distributed MLH model capable of handling atomic structures at scales of 48,000 atoms. It shares its backbone architecture with the HELM model, but introduces additional data pipeline optimizations and graph partitioning for distribution across multiple GPU nodes. Combined with an advanced quantum transport solver, MALOQ enables the modeling of unprecedentedly large device structures at an atomistic level of resolution, with DFT-like accuracy. We demonstrate its capabilities in providing insights into the physics of a monolayer MoS_2 FET with HfO_2 oxide layers. Our model is ready to be deployed for *in silico* designs of next-generation nanoelectronic devices as diverse as phase-change memories (Zhou et al., 2023) and silicon nanoribbon field-effect transistors (Yeap et al., 2024). Code will be released and linked upon publication.

AI USE STATEMENT

GitHub Copilot was used during code implementation. No LLM tools were used in the writing of this paper.

REFERENCES

A. Agrawal, W. Chakraborty, W. Li, H. Ryu, B. Markman, S. H. Hoon, R. K Paul, C. Y. Huang, S. M. Choi, K. Rho, A. Shu, R. Iglesias, P. Wallace, S. Ghosh, K. L. Cheong, J. L. Hockel, R. Thorman, L. Baumgartel, L. Shoer, V. Mishra, S. Berrada, A. Ashita, C. Weber, B. Obradovic, A. A. Oni, Z. Brooks, N. Franco, J. Kavalieros, and G. Dewey. Silicon ribbonfet CMOS at 6nm gate length. In *2024 IEEE International Electron Devices Meeting (IEDM)*, 2024. URL https://doi.org/10.1109/IEDM50854.2024.10873367. (cited on page 1)

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, C. K. Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Shunting Zhang, Michael Suo, Phil Tillet, Xu Zhao, Eikan Wang, Keren Zhou, Richard Zou, Xiaodong Wang, Ajit Mathews, William Wen, Gregory Chanan, Peng Wu, and Soumith Chintala. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ASPLOS '24, pp. 929–947, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703850. doi: 10.1145/3620665.3640366. URL https://doi.org/10.1145/3620665.3640366. (cited on page 17)

Ilyes Batatia, Dávid Péter Kovács, Gregor N. C. Simm, Christoph Ortner, and Gábor Csányi. MACE: Higher order equivariant message passing neural networks for fast and accurate force fields, 2022. URL https://arxiv.org/abs/2206.07697. (cited on page 7)

Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E. Smidt, and Boris Kozinsky. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13(1), May 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-29939-5. URL http://dx.doi.org/10.1038/s41467-022-29939-5. (cited on page 16)

Mads Brandbyge, José-Luis Mozos, Pablo Ordejón, Jeremy Taylor, and Kurt Stokbro. Density-functional method for nonequilibrium electron transport. *Physical Review B*, 65(16), March 2002. ISSN 1095-3795. doi: 10.1103/physrevb.65.165401. URL http://dx.doi.org/10.1103/PhysRevB.65.165401. (cited on page 1)

Sascha Brück, Mauro Calderara, Mohammad Hossein Bani-Hashemian, Joost VandeVondele, and Mathieu Luisier. Efficient algorithms for large-scale quantum transport calculations. *The Journal of Chemical Physics*, 147(7), August 2017. ISSN 1089-7690. doi: 10.1063/1.4998421. URL http://dx.doi.org/10.1063/1.4998421. (cited on pages 15 and 16)

Supriyo Datta. *Non-equilibrium Green's function formalism*, pp. 293–342. Cambridge Studies in Semiconductor Physics and Microelectronic Engineering. Cambridge University Press, 1995. doi: 10.1017/CBO9780511805776.009. (cited on pages 3 and 15)

Mauro Dossena, Benoit Van Troeye, Fabian Ducry, Jiang Cao, Aryan Afzalian, Geoffrey Pourtois, and Mathieu Luisier. Mobility calculation in disordered ws2-al2o3 stacks from first principles. *npj 2D Materials and Applications*, 9(1), August 2025. ISSN 2397-7132. doi: 10.1038/s41699-025-00587-9. URL http://dx.doi.org/10.1038/s41699-025-00587-9. (cited on page 7)

Fabian Ducry et al. Oxide induced degradation in MoS_2 field-effect transistors. *npj 2D Materials and Applications*, 10(1), March 2026. ISSN 2397-7132. doi: 10.1038/s41699-026-00677-2. URL http://dx.doi.org/10.1038/s41699-026-00677-2. (cited on pages 1 and 7)

Luigi Fusco, Mikhail Khalilov, Marcin Chrapek, Giridhar Chukkapalli, Thomas Schulthess, and Torsten Hoefler. Understanding data movement in tightly coupled heterogeneous systems: A case study with the grace hopper superchip, 2024. URL https://doi.org/10.48550/ARXIV.2408.11556. (cited on page 17)

Mario Geiger and Tess Smidt. e3nn: Euclidean neural networks, 2022. URL https://doi.org/10.48550/ARXIV.2207.09453. (cited on page 4)

Mughira Ghafoor, Salah Ud Din, Sufaid Shah, Matin Ashurov, Muhammad Sajid, Rajwali Khan, and Jing Lu. Quantum transport insights into layer-dependent behavior of sub-1 nm mos2 transistors for advanced scaling. *RSC Advances*, 16(34):32397–32408, 07 2026. ISSN 2046-2069. doi: 10.1039/d6ra00943c. URL https://doi.org/10.1039/d6ra00943c. (cited on page 9)

Xiaoxun Gong, He Li, Nianlong Zou, Runzhang Xu, Wenhui Duan, and Yong Xu. General framework for E(3)-equivariant neural network representation of density functional theory hamiltonian. *Nature Communications*, 14(1), May 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-38468-8. URL http://dx.doi.org/10.1038/s41467-023-38468-8. (cited on page 3)

Kevin Han, Bowen Deng, Amir Barati Farimani, and Gerbrand Ceder. DistMLIP: A distributed inference platform for machine learning interatomic potentials. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=4tasfBIPxp. (cited on page 16)

Yuewen Huang, Pin Chen, and Yutong Lu. Uniham: A large-scale soc-complete dataset and benchmark for hamiltonian learning in materials. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 9486–9497. ACM, August 2026. doi: 10.1145/3770855.3817531. URL http://dx.doi.org/10.1145/3770855.3817531. (cited on page 3)

Weile Jia, Han Wang, Mohan Chen, Denghui Lu, Lin Lin, Roberto Car, Weinan E, and Linfeng Zhang. 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*, SC '20. IEEE Press, 2020. ISBN 9781728199986. URL https://dl.acm.org/doi/abs/10.5555/3433701.3433707. (cited on page 16)

Manasa Kaniselvan, Benjamin Kurt Miller, Meng Gao, Juno Nam, and Daniel S. Levine. Learning from the electronic structure of molecules across the periodic table. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=PS1YS8Wv4t. (cited on pages 1, 2, 3, 6, 8, and 18)

George Karypis and Vipin Kumar. Multilevelk-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed Computing*, 48(1):96–129, January 1998. ISSN 0743-7315. doi: 10.1006/jpdc.1997.1404. URL http://dx.doi.org/10.1006/jpdc.1997.1404. (cited on pages 6 and 18)

Kuzma Khrabrov, Anton Ber, Artem Tsypin, Konstantin Ushenin, Egor Rumiantsev, Alexander Telepov, Dmitry Protasov, Ilya Shenbin, Anton M. Alekseev, Mikhail Shirokikh, Sergey Nikolenko, Elena Tutubalina, and Artur Kadurin. ∇^2DFT: A universal quantum chemistry dataset of drug-like molecules and a benchmark for neural network potentials. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=ElUrNM9U8c. (cited on page 18)

Seongsu Kim, Nayoung Kim, Dongwoo Kim, and Sungsoo Ahn. High-order equivariant flow matching for density functional theory hamiltonian prediction, 2025. URL https://arxiv.org/abs/2505.18817. (cited on page 3)

W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Physical Review*, 140(4A):A1133–A1138, November 1965. ISSN 0031-899X. doi: 10.1103/physrev.140.a1133. URL http://dx.doi.org/10.1103/PhysRev.140.A1133. (cited on page 2)

Boris Kozinsky, Albert Musaelian, Anders Johansson, and Simon Batzner. Scaling the leading accuracy of deep equivariant models to biomolecular simulations of realistic size. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage*

and Analysis, SC '23, pp. 1–12. ACM, November 2023. doi: 10.1145/3581784.3627041. URL http://dx.doi.org/10.1145/3581784.3627041. (cited on page 16)

Thomas D. Kühne, Marcella Iannuzzi, Mauro Del Ben, Vladimir V. Rybkin, Patrick Seewald, Frederick Stein, Teodoro Laino, Rustam Z. Khaliullin, Ole Schütt, Florian Schiffmann, Dorothea Golze, Jan Wilhelm, Sergey Chulkov, Mohammad Hossein Bani-Hashemian, Valéry Weber, Urban Borštnik, Mathieu Taillefumier, Alice Shoshana Jakobovits, Alfio Lazzaro, Hans Pabst, and *et al.* CP2K: An electronic structure and molecular dynamics software package - quickstep: Efficient and accurate electronic structure calculations. *J. Chem. Phys.*, 152(19):194103, May 2020. URL https://doi.org/10.1063/5.0007045. (cited on pages 3 and 7)

Junyoung Kwon et al. Gate structuring on n-type bilayer MoS_2 field-effect transistors for ultra-high current density. *Nature Materials*, 25(5):824–831, January 2026. ISSN 1476-4660. doi: 10.1038/s41563-025-02452-y. URL http://dx.doi.org/10.1038/s41563-025-02452-y. (cited on page 7)

Uihui Kwon, Yonghee Park, Yoon-Suk Kim, Jaehyun Yoo, and Dae Sin Kim. Tcad challenges and opportunities to find a feasible device architecture for sub-3nm scaling. In *2021 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD)*, pp. 5, 2021. doi: 10.1109/SISPAD54002.2021.9592554. (cited on page 1)

Craig S. Lent and David J. Kirkner. The quantum transmitting boundary method. *Journal of Applied Physics*, 67(10):6353–6359, 05 1990. ISSN 0021-8979. doi: 10.1063/1.345156. URL https://doi.org/10.1063/1.345156. (cited on pages 7 and 15)

Daniel S. Levine, Muhammed Shuaibi, Evan Walter Clark Spotte-Smith, Michael G. Taylor, Muhammad R. Hasyim, Kyle Michel, Ilyes Batatia, Gábor Csányi, Misko Dzamba, Peter Eastman, Nathan C. Frey, Xiang Fu, Vahe Gharakhanyan, Aditi S. Krishnapriyan, Joshua A. Rackers, Sanjeev Raja, Ammar Rizvi, Andrew S. Rosen, Zachary Ulissi, Santiago Vargas, C. Lawrence Zitnick, Samuel M. Blau, and Brandon M. Wood. The open molecules 2025 (omol25) dataset, evaluations, and models, 2025. URL https://doi.org/10.48550/ARXIV.2505.08762. (cited on pages 18 and 20)

He Li, Zun Wang, Nianlong Zou, Meng Ye, Runzhang Xu, Xiaoxun Gong, Wenhui Duan, and Yong Xu. Deep-learning density functional theory hamiltonian for efficient ab initio electronic-structure calculation. *Nature Computational Science*, 2(6):367–377, June 2022. ISSN 2662-8457. doi: 10.1038/s43588-022-00265-6. URL http://dx.doi.org/10.1038/s43588-022-00265-6. (cited on page 1)

Yang Li, Yanzhen Wang, Boheng Zhao, Xiaoxun Gong, Yuxiang Wang, Zechen Tang, Zixu Wang, Zilong Yuan, Jialin Li, Minghui Sun, Zezhou Chen, Honggeng Tao, Baochun Wu, Yuhang Yu, He Li, Felipe H. da Jornada, Wenhui Duan, and Yong Xu. Deeph-pack: A general-purpose neural network package for deep-learning electronic structure calculations, 2026. URL https://doi.org/10.48550/ARXIV.2601.02938. (cited on pages 3 and 8)

Yunyang Li, Zaishuo Xia, Lin Huang, Xinran Wei, Samuel Harshe, Han Yang, Erpai Luo, Zun Wang, Jia Zhang, Chang Liu, Bin Shao, and Mark Gerstein. Enhancing the scalability and applicability of kohn-sham hamiltonians for molecular systems. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=twEvvkQqPS. (cited on pages 1 and 3)

Mathieu Luisier and Andreas Schenk. Atomistic simulation of nanowire transistors. *Journal of Computational and Theoretical Nanoscience*, 5(6):1031–1045, 2008. doi: 10.1166/jctn.2008.2415. (cited on pages 15 and 16)

W. Mortelmans, P. Buragohain, A. Kitamura, C.J. Dorow, C. Rogan, L. Siddiqui, R. Ramamurthy, J. Lux, T. Zhong, S. Harlson, E. Gillispie, T. Wilson, R. Toku, A. Oni, A. Penumatcha, M. Kavrik, M. Jaikissoon, K. Maxey, A. Kozhakhmetov, C-Y. Cheng, C-C. Lin, S. Lee, A. Vyatskikh, N. Arefin, D. Kencke, J. Kevek, T. Tronic, M. Metz, S.B. Clendenning, K.P. O'Brien, and U. Avci. Gate oxide module development for scaled GAA 2D FETs enabling SS<75mV/d and record $I_{d,max}$>900A/m at L_g<50nm. In *2024 IEEE International Electron Devices Meeting (IEDM)*, pp. 1–4. IEEE, December 2024. doi: 10.1109/iedm50854.2024.10873417. URL http://dx.doi.org/10.1109/IEDM50854.2024.10873417. (cited on page 7)

Frank Neese. The orca program system. *WIREs Computational Molecular Science*, 2(1):73–78, June 2011. ISSN 1759-0884. doi: 10.1002/wcms.81. URL http://dx.doi.org/10.1002/wcms.81. (cited on page 3)

Yutack Park, Jaesun Kim, Seungwoo Hwang, and Seungwu Han. Scalable parallel algorithm for graph neural network interatomic potentials in molecular dynamics simulations. *Journal of Chemical Theory and Computation*, 20(11):4857–4868, May 2024. ISSN 1549-9626. doi: 10.1021/acs.jctc.4c00190. URL http://dx.doi.org/10.1021/acs.jctc.4c00190. (cited on page 16)

Saro Passaro and C. Lawrence Zitnick. Reducing SO(3) convolutions to SO(2) for efficient equivariant GNNs. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 27420–27438. PMLR, Jul 2023. URL https://proceedings.mlr.press/v202/passaro23a.html. (cited on pages 2, 3, and 4)

K. T. Schütt, M. Gastegger, A. Tkatchenko, K.-R. Müller, and R. J. Maurer. Unifying machine learning and quantum chemistry with a deep neural network for molecular wavefunctions. *Nature Communications*, 10(1), November 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-12875-2. URL http://dx.doi.org/10.1038/s41467-019-12875-2. (cited on page 18)

Mark A. Stettler, Stephen M. Cea, Sayed Hasan, Lei Jiang, Patrick H. Keys, Colin D. Landon, Prabakar Marepalli, Daniel Pantuso, and Cory E. Weber. Industrial tcad: Modeling atoms to chips. *IEEE Transactions on Electron Devices*, 68:5350, 2021. doi: 10.1109/TED.2021.3076976. (cited on page 1)

A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton. LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comput. Phys. Commun.*, 271:108171, 2022. URL https://doi.org/10.1016/j.cpc.2021.108171. (cited on pages 7 and 16)

Benoît Van Troeye, Fabian Ducry, Mauro Dossena, Mathieu Luisier, Aryan Afzalian, and Geoffrey Pourtois. Impact of interface and surface oxide defects on WS_2 electronic properties from first principles. *ACS Nano*, 19:11664, 2025. doi: 10.1021/acsnano.4c08959. (cited on page 7)

Nicolas Vetsch, Alexander Maeder, Vincent Maillou, Anders Winka, Jiang Cao, Grzegorz Kwasniewski, Leonard Deuschle, Torsten Hoefler, Alexandros Nikolaos Ziogas, and Mathieu Luisier. Ab-initio quantum transport with the gw approximation, 42, 240 atoms, and sustained exascale performance. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '25, pp. 1–13. ACM, November 2025. doi: 10.1145/3712285.3771784. URL http://dx.doi.org/10.1145/3712285.3771784. (cited on pages 1, 2, and 9)

Chen Hao Xia, Manasa Kaniselvan, Alexandros Nikolaos Ziogas, Marko Mladenović, Rayen Mahjoub, Alexander Maeder, and Mathieu Luisier. Learning the electronic hamiltonian of large atomic structures, 2025. URL https://arxiv.org/abs/2501.19110. (cited on pages 3, 6, and 8)

YuQing Xie, Ameya Daigavane, Mit Kotak, and Tess Smidt. The price of freedom: Exploring expressivity and runtime tradeoffs in equivariant tensor products. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=EvIwwGYTLc. (cited on page 3)

Geoffrey Yeap, S.S. Lin, H.L. Shang, H.C. Lin, Y.C. Peng, M. Wang, PW Wang, CP Lin, KF Yu, WY Lee, HK Chen, DW Lin, BR Yang, CC Yeh, CT Chan, JM Kuo, C-M Liu, TH Chiu, MC Wen, T.L. Lee, CY Chang, R. Chen, P-H Huang, C.S. Hou, YK Lin, FK Yang, J. Wang, S. Fung, Ryan Chen, C.H. Lee, TL Lee, W. Chang, DY Lee, CY Ting, T. Chang, HC Huang, HJ Lin, C. Tseng, CW Chang, KB Huang, YC Lu, C-H Chen, C.O. Chui, KW Chen, MH Tsai, CC Chen, N. Wu, HT Chiang, XM Chen, SH Sun, JT Tzeng, K. Wang, YC Peng, HJ Liao, T. Chen, YK Cheng, J. Chang, K. Hsieh, A. Cheng, G. Liu, A. Chen, HT Lin, KC Chiang, CW Tsai, H. Wang, W. Sheu, J. Yeh, YM Chen, CK Lin, J. Wu, M. Cao, LS Juang, F. Lai, Y. Ku, S.M. Jang, and L.C. Lu.

2nm platform technology featuring energy-efficient nanosheet transistors and interconnects co-optimized with 3DIC for AI, HPC and mobile SoC applications. In *2024 IEEE International Electron Devices Meeting (IEDM)*, pp. 1–4. IEEE, December 2024. doi: 10.1109/iedm50854.2024.10873475. URL http://dx.doi.org/10.1109/IEDM50854.2024.10873475. (cited on page 9)

Haiyang Yu, Meng Liu, Youzhi Luo, Alex Strasser, Xiaofeng Qian, Xiaoning Qian, and Shuiwang Ji. QH9: A quantum hamiltonian prediction benchmark for qm9 molecules. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 40487–40503. Curran Associates, Inc., 2023a. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/7f755e271717450020fda40f020922dd-Paper-Datasets_and_Benchmarks.pdf. (cited on page 18)

Haiyang Yu, Zhao Xu, Xiaofeng Qian, Xiaoning Qian, and Shuiwang Ji. Efficient and equivariant graph networks for predicting quantum Hamiltonian. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 40412–40424. PMLR, Jul 2023b. URL https://proceedings.mlr.press/v202/yu23i.html. (cited on page 3)

Yuxing Zhou, Wei Zhang, En Ma, and Volker L. Deringer. Device-scale atomistic modelling of phase-change memory materials. *Nature Electronics*, 6(10):746–754, September 2023. ISSN 2520-1131. doi: 10.1038/s41928-023-01030-x. URL http://dx.doi.org/10.1038/s41928-023-01030-x. (cited on pages 7 and 9)

APPENDIX

A Quantum transmitting boundary method (QTBM) **15**

B Domain decomposition and graph partitioning **16**

C Label and matrix representation of the Hamiltonian **16**

D Training environment **17**

E MALOQ on molecules **17**
E.1 Molecular Hamiltonian datasets 17
E.2 Partitioning molecular batches 18
E.3 Scalable training for molecular batches 18

F Data processing acceleration **19**

G MoS_2/HfO_2 structures - additional data **21**

A QUANTUM TRANSMITTING BOUNDARY METHOD (QTBM)

To compute the current flowing through nanoscale devices described by a Hamiltonian matrix $\mathbf{H}$, the quantum transmitting boundary method (QTBM) can be used (Lent & Kirkner, 1990). It models the transmission of charge carriers through a domain defined by a given 'electronic structure', without gain or loss in energy through interactions with other quanta. We refer to this as a 'coherent' transport regime. The unknowns are the energy-dependent $\psi(\varepsilon)$ wavefunctions, which obey:

$$(\varepsilon \cdot \mathbf{S} - \mathbf{H} - \Sigma^L(\varepsilon) - \Sigma^R(\varepsilon)) \cdot \psi(\varepsilon) = Inj(\varepsilon) \tag{1}$$

Here, $Inj(\varepsilon)$ describes carrier injection from the contacts (at input energy ε), which are coupled to the device through left/right 'open boundary self-energies' $\Sigma^L(\varepsilon)$/$\Sigma^R(\varepsilon)$ in case of a two-terminal structure. The process of computing $Inj(\varepsilon)$ and $\Sigma^{L/R}(\varepsilon)$ involves solving eigenvalue problems to determine the injected states from the left/right contacts at a given energy. We refer to the methodology in Brück et al. (2017).

Once $Inj(\varepsilon)$ and $\Sigma^{L/R}(\varepsilon)$ are known, **Eqn. (1)** becomes a system of linear equations that can be solved for $\psi(\varepsilon)$ by a sparse direct solver. Concretely, $\psi(\varepsilon)$ is a multi-column vector that can be decomposed into states coming from the left ($\psi^L(\varepsilon)$) and right ($\psi^R(\varepsilon)$) electrodes. Furthermore, different modes with index n can be injected from a given contact. Hence, the wavefunctions depend on the mode they originate from and have the following form $\psi_n^{L/R}(\varepsilon)$.

To compute the net current, the device geometry is first decomposed into unit cells J, each of them being connected to its previous and next neighbor through a Hamiltonian block $\mathbf{H_{J,J-1}}$ and $\mathbf{H_{J,J+1}}$. We can then apply Landauer-Büttiker's formula to obtain the electronic current I_d (Datta, 1995):

$$I_d = \frac{2e}{h} \int d\varepsilon \, T(\varepsilon) \, \left(f_L(\varepsilon) - f_R(\varepsilon)\right), \tag{2}$$

where $f_L(\varepsilon)$ and $f_R(\varepsilon)$ are the Fermi-Dirac distributions of the left and right contacts, respectively. These distributions are shifted relative to one another by the externally applied voltage bias. To compute I_d in the coherent limit of transport, a new quantity, the energy-resolved transmission function $T(\varepsilon)$, is needed. It can be evaluated between any neighboring cells with indices J and $J+1$ according to Luisier & Schenk (2008):

$$T(\varepsilon) = 2 \sum_n \mathrm{Im}\left(\psi_{n,J}^{c,T}(\varepsilon) \cdot \mathbf{H_{J,J+1}} \cdot \psi_{n,J+1}^{c}(\varepsilon)\right) \left|\frac{dE(k_{c,n})}{dk_{c,n}}\right|^{-1}_{E(k_{c,n})=\varepsilon}. \tag{3}$$

In **Eqn. (3)**, c refers to either the left (L) or right (R) contact, $\psi^c_{n,J}(\varepsilon)$ is the wavefunction injected from contact c using mode n and evaluated at the device unit cell J, T indicates that the vector is transposed, while "$E(k_{c,n})$ vs. $k_{c,n}$" is the "energy vs. momentum" dispersion of mode n inside contact c. It can be calculated, for example, with the contour integral technique from Brück et al. (2017). Note that $T(\varepsilon)$ does not depend on the choice of c (the transmissions from left to right and from right to left are identical) or J (because of current continuity, the transmission between two adjacent cells J and $J+1$ is the same everywhere).

We can also project the total current I_d into local areas of the simulation domain to analyze the influence of atomic-level features. To do this, we compute an inter-atomic current $I_{d,ij}$ with **Eqn. (2)** where $T(\varepsilon)$ is replaced by $T_{ij}(\varepsilon)$, the 'transmission function' between two atoms located at i and j. This quantity is defined as (Luisier & Schenk, 2008):

$$T_{ij}(\varepsilon) = 2\sum_n \mathrm{Im}\left(\psi^{c,T}_{n,i}(\varepsilon)\cdot \mathbf{H_{ij}}\cdot \psi^c_{n,j}(\varepsilon)\right)\left|\frac{dE(k_{c,n})}{dk_{c,n}}\right|^{-1}_{E(k_{c,n})=\varepsilon}. \tag{4}$$

The difference, as compared to **Eqn. (3)**, is that $\mathbf{H_{ij}}$ is now the block that connects all orbitals of atom i and j and no longer the coupling between two adjacent cells. As a consequence, $\psi^c_{n,i}(\varepsilon)$ and $\psi^c_{n,j}(\varepsilon)$ correspond to the wavefunctions associated with atoms i and j, respectively. Finally, the inter-atomic currents can be interpolated into a regular spatial grid to visualize specific current isovalues as surfaces over the atomic coordinates.

B Domain decomposition and graph partitioning

Distributed GNN applications in materials science are concerned with MLIPs and, to the best of our knowledge, have not yet been applied to MLH. Many of them, e.g., DeepMD Jia et al. (2020) and Allegro Kozinsky et al. (2023), take advantage of the domain decomposition functionality of the LAMMPS molecular dynamics code Thompson et al. (2022). LAMMPS relies on spatial decomposition of the atomic domain into boxes, and has a specific functionality to modify the box-boundaries and balance the number of atoms within each box. Directly partitioning the graph instead of the spatial domain appears as a promising avenue to improve load balancing and more evenly distribute communication. This is the approach pursued by SevenNet Park et al. (2024); Batzner et al. (2022) and DistMLIP Han et al. (2026), which serve as wrappers for existing MLIP architectures. SevenNet is integrated into LAMMPS, and its strong scaling abilities are primarily limited by reduced GPU utilization when distributed over many computing units. DistMLIP Han et al. (2026) supports multi-process, but single-node inference.

C Label and matrix representation of the Hamiltonian

The entries of $\mathbf{H}$ correspond to interactions between electronic orbitals located on the same or on different atoms. The full matrix can be decomposed into sub-matrices $\mathbf{H}^{\alpha\beta}_{ij} = \langle l_\alpha, m_\alpha|\,\hat{H}(r)\,|l_\beta, m_\beta\rangle$. They contain all interactions $\mathbf{H}^{\alpha\beta}_{ij}(m_\alpha, m_\beta)$ between spherical harmonic basis functions l_α (of degree α) on atom i, and l_β (of degree β) on j. Mathematically, each $\mathbf{H}^{\alpha\beta}_{ij}$ corresponds to the tensor product $l_\alpha \otimes l_\beta$, which has length $(2l_\alpha+1)\times(2l_\beta+1)$. Through a transformation T, this tensor product can be written as the direct sum ($\oplus$) of angular momentum eigenstates $|L, M\rangle$, where L ranges from degree $|\alpha-\beta|$ to $|\alpha+\beta|$, i.e., $T(l_\alpha \otimes l_\beta) = \bigoplus_{i=|\alpha-\beta|}^{(\alpha+\beta)} L_i = L_{|\alpha-\beta|}\oplus\ldots\oplus L_{(\alpha+\beta)}$. The transformation T involves a matrix of Clebsch-Gordon (CG) coefficients. The entries for a specific $|L, M\rangle$ component are given by:

$$|L, M\rangle = \sum_{m_\alpha=-l_\alpha}^{+l_\alpha}\sum_{m_\beta=-l_\beta}^{+l_\beta} C^{(L,M)}_{(l_\alpha,m_\alpha)(l_\beta,m_\beta)}\,|l_\alpha, m_\alpha\rangle\,|l_\beta, m_\beta\rangle\,. \tag{5}$$

Here, $C^{(L,M)}_{(l_\alpha,m_\alpha)(l_\beta,m_\beta)}$ is the CG coefficient describing the contribution of $|l_\alpha, m_\alpha\rangle\,|l_\beta, m_\beta\rangle$ to the state $|L, M\rangle$, and $|l_\alpha, m_\alpha\rangle\,|l_\beta, m_\beta\rangle$ corresponds to $\mathbf{H}^{\alpha\beta}_{ij}(m_\alpha, m_\beta)$. The transformation T thus decomposes the Hamiltonian sub-matrix $\mathbf{H}^{\alpha\beta}_{ij}$ into a 'vector' form, representing a direct sum of tensors

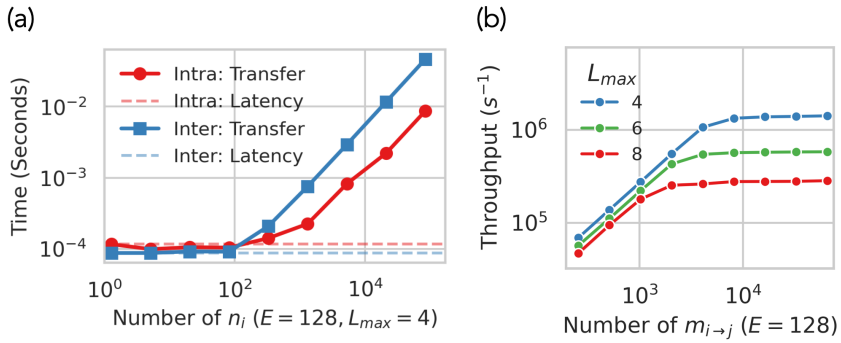


Figure D.1: **Embedding communication and processing throughput**. (a) Measured intra-/inter-node communication throughput of node embeddings n_i with embedding size E=128. (b) Processing throughput for eSCN convolutions as a function of the number of messages $m_{i\to j}$ for different l_{max}. Each data point corresponds to the median of 20 measurements.

with different angular degree l. All $|L, M\rangle$ corresponding to a single $\mathbf{H}_{ij}$ block are then concatenated into a vector $\mathbf{h}_{ij}$. The same operation is repeated for all blocks of $\mathbf{H}$. Since not all atomic elements require the same number of orbitals (basis set) to be accurately described, the resulting $\mathbf{h}_{ij}$ vectors have different lengths. They are all padded to a common dimension, that of the largest element in the dataset. The collection of all padded $\mathbf{h}_{ij}$ forms the 'labels' used to train the node and edge embeddings of the GNN (top part of **Fig. 3(b)**), and the loss is computed element-wise over the model's prediction of them. The maximum degree of the $\mathbf{h}_{ij}$ becomes the l_{max} required for embeddings.

D Training environment

Hardware Our models are trained on the Alps supercomputer at CSCS, where each node comprises 4 NVIDIA GH200 Grace Hopper superchips (Fusco et al., 2024). Every superchip combines a Hopper GPU with 132 Streaming Multiprocessors and 96 GB HBM3 memory with a Grace CPU with 72 cores and 128 GB LPDDR5. Each Hopper GPU is connected to every other GPU within the same node via NVLink and 150 GB/s bidirectional point-to-point bandwidth, resulting in a total of 900 GB/s. The Alps nodes are connected via an HPE Slingshot 11 network with a Dragonfly topology. MALOQ is written in PyTorch (Ansel et al., 2024), using torch tensors to represent the embeddings of nodes and edges. To communicate embeddings between GPUs, we use PyTorch's distributed functionality and its NVIDIA Collective Communications Library (NCCL) backend.

Throughput To quantify both inter-process communication time, we show in **Fig. D.1(a)** the time to send/receive different counts of n_i between two processes, each running on a separate GH200 superchip on Alps. Up to about 100 n_i (considering $E = 128$ channels and $l_{max} = 4$), communication is bound by latency. At the same time, the achievable runtime improvements due to graph distribution are limited by processing throughput, which decreases once each partition owns too few local edges. The number of messages required to saturate processing throughput is shown in **Fig. D.1(b)**. It is proportional to the number of local edges.

E MALOQ on molecules

E.1 Molecular Hamiltonian datasets

All DFT-level datasets available today have necessitated at least one electronic structure calculation per atomic system considered, but almost none of them stored the corresponding Hamiltonian matrices. In **Table 2** we summarize the parameters of several datasets that indeed include Hamiltonian matrices. Note that the nature of the atomic elements as well as the DFT features (functional, localized basis set) do not change the computational problem at hand, but heavier elements typically require a larger basis, thus determining l_{max}. To benchmark the computational performance

	Dataset	Elements	Basis ($\mathbf{L_{max}}$)	Func.	r_{cut} (Å)
Small molecule	MD17	H, C, N, O	def2_SVP (**4**)	PBE	(FC)
	QH9	H, C, N, O, F	def2_SVP (**4**)	B3LYP	(FC)
	∇^2DFT	H, C, N, O, F, S, Cl, Br	def2_SVP (**4**)	ωB97X-D	(FC)
Large Molecule	CSH_58k	$\times$58 elements	def2-TZVPD (**6**)	ωB97M-V	12.0
	Electrolytes	$\times$19 elements	def2-TZVPD (**6**)	ωB97M-V	12.0
	Metal-Organics	$\times$31 elements	def2-TZVPD (**8**)	ωB97M-V	12.0

Table 2: Summary of existing public Hamiltonian matrix datasets, highlighting the elements included, basis set and $\mathbf{l_{max}}$ required to describe it, functional type (Func.), and interaction cutoff radius. 'FC' means 'fully connected graph', in case molecules are small enough to avoid applying a cutoff. Data sources: MD17 (Schütt et al., 2019), QH9 (Yu et al., 2023a), ∇^2DFT (Khrabrov et al., 2024), electronic structure subsets from OMol25 (Levine et al., 2025; Kaniselvan et al., 2026).

of MALOQ when training on large-scale electronic structure matrix data, we use data from the three representative datasets in bold in **Table 2**: ∇^2DFT and its drug-like molecules with up to 50 atoms each, and custom 'unsolvated electrolytes' and 'metal organics' datasets derived from the Open Molecules 2025 4 million data with basis-dependent l_{max} (Levine et al., 2025).

E.2 Partitioning molecular batches

Training on molecular datasets in a distributed-memory environment typically involves creating a molecule-wise partition, where each rank processes a subset of molecules. Load imbalance might arise if these molecules are made of a wide range of atoms, which may contain an even wider distribution of connectivity (# edges per structure). This load imbalance can be reduced by 'binning' molecules into partitions to balance the total number of atoms/edges across the structures handled by each process. However, a finer load balance can be achieved by partitioning the molecules at the atom/edge level. In practice, this can significantly reduce the time to process each training batch.

As an example, we consider in **Fig. E.1** a batch of 128 molecules from the OMol-electrolytes dataset (**Fig. E.1(a)**) and compare the workload distribution (**Fig. E.1(b)**), as well as the runtime per forward (**Fig. E.1(c)**) and backward pass (**Fig. E.1(d)**) under four different partitioning schemes. The so-called 'Metis' approach uses the METIS package (Karypis & Kumar, 1998) to search for a mincut solution to the partitioning problem. Since cuts to the graph are penalized, Metis finds an 'optimal binning' when there are more molecules than ranks, thus balancing the partitions without introducing inter-process communication. This is the case in **Fig. E.1(b)**. Meanwhile, in the 'atom-wise'/'edge-wise' partitions, nodes/edges are evenly distributed among ranks, starting from a list of atoms across the data batch where those within each molecule are contiguous. In particular, an edge-wise distribution of the graph leads to a highly balanced embedding workload. Compared to the molecule-wise distribution, splitting the batch edge-wise reduces the time per forward pass by 30% (from 0.23 to 0.15s), and the time per backwards pass by over 50% (0.85 to 0.4s), despite the introduction of inter-process communication.

E.3 Scalable training for molecular batches

'Universal' molecular datasets with diverse atomic elements typically necessitate large batch sizes during training, to ensure sufficient sampling over all of the represented elements within each batch. We thus standardize all our benchmarks using a single batch size of 128 randomly selected molecules, which is distributed across multiple processes. In all cases, we use an edge-wise partitioning, as previously explored in **Fig. E.1**. Note that tackling the largest molecules (in terms of # atoms, l_{max}) across these datasets already requires partitioning individual molecular graphs to fit in GPU memory.

The strong scaling results for the training runtime of three molecular datasets are plotted in **Fig. E.2**: ∇^2DFT (Khrabrov et al., 2024), electrolytes, and metal-organics (Levine et al., 2025; Kaniselvan et al., 2026). Each dataset necessitates a different l_{max}, up to 8 for metal-organics which contain lanthanide elements with g-orbitals. We train these datasets on up to 32 Alps nodes (128 GH200

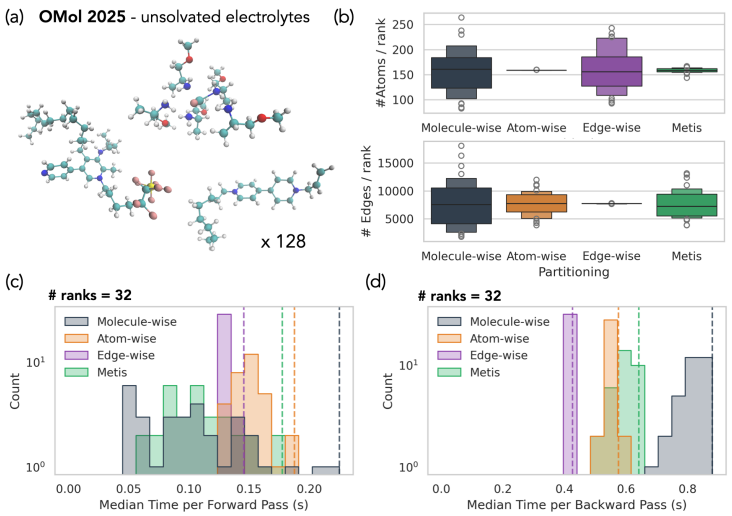


Figure E.1: **Training runtime comparison under different distribution schemes.** (a) Dataset used for training. It consists of 128 randomly-selected molecules from the OMol electrolytes dataset and their corresponding $\mathbf{H}$. (b) Workload distribution across ranks quantified in # atoms and # edges, for three different distribution schemes (atom-wise, edge-wise, Metis), compared to the undistributed case in which 128 molecules are divided across 32 ranks (molecule-wise). (c) Forward and (d) backward pass time distribution across different ranks. Each data is the median of 200 training epochs, of which the first 20 are discarded. The vertical dashed lines indicate the maximum time for each distribution scheme.

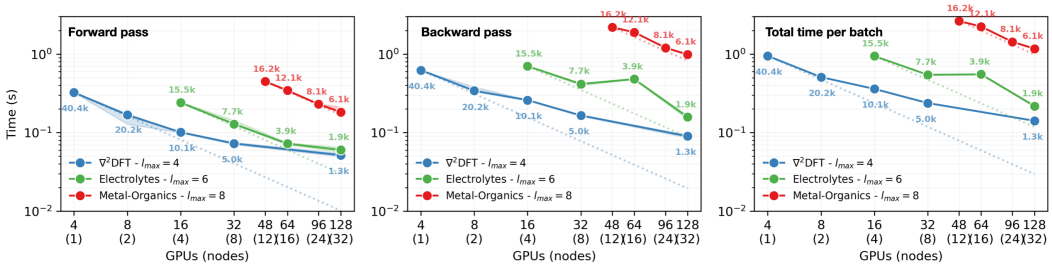


Figure E.2: **Strong scaling of MALOQ over three irregular molecular datasets from Table 2 with different l_{max}.** (a) Forward pass. (b) Backward pass. (c) Total time per batch. A total of 128 molecules from each dataset is placed in each batch, which is distributed according to an edge-wise partition. The data plotted is the median over 200 epochs (the first 20 are discarded), and the shaded areas show variance over runtime across ranks. Each data point is annotated by the average number of edges owned by each process under that distribution. Note that we start from the minimum number of GPUs allowing for the partitioned batch to fit in memory.

superchips), where the minimum number of nodes depends on the dataset: ∇^2DFT's batch fits on a single node, electrolytes require at least 4 nodes, and metal-organics cannot run on less than 12 nodes. Particularly in the forward pass, the number of edges per GPU at which the scaling efficiency starts dropping roughly corresponds to the point where processing throughput is lost: $\sim$10k for ∇^2DFT with $l_{max} = 4$, and $\sim$5k for the electrolytes with $l_{max} = 6$. The times for the metal-organics molecules scale well up to 32 nodes, since the average number of edges per partition (6.1k) remains well above the low saturation point for $l_{max} = 8$ (about 2k).

F Data Processing Acceleration

We benchmark our data processing kernels on a selection of three molecular datasets in **Table 3**, each with different l_{max} and orbital basis. To evaluate the performance under realistic conditions,

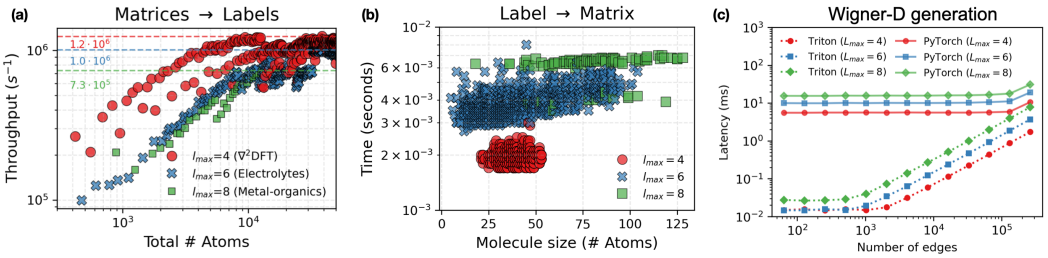


Figure F.1: **Custom kernels for data processing** Measured processing throughput/runtime to either (a) convert multiple matrices to labels, or (b) reconstruct a single matrix from a given set of predicted data. All measurements were taken over a subset of molecules present in the ∇^2DFT (l_{max}=4) and OMol Electrolytes/Metal-Organics (l_{max}=6/8) datasets, with data processed in FP32. (c) Latency comparison between our Triton kernel and the reference Torch implementation at selected number of input edges.

R_{cut} (Å)	Number of Edges ($\lvert E\rvert$)	Conversion Time (ms)	Throughput (10^6 blocks/s)
8.0	533,364	2.90	184.95
9.0	756,790	3.50	217.08
10.0	1,040,696	4.20	248.50
11.0	1,386,896	4.90	283.65
12.0	1,800,366	7.60	237.29
13.0	2,292,008	7.00	327.86
14.0	2,829,326	8.20	345.41
10.0 (NumPy ref.)	1,040,696	171,051.30	0.006
11.0 (NumPy ref.)	1,386,896	234,792.60	0.006

Table 3: Computational throughput for matrix to label conversion for the 3,000-atom HfO_2 structure in the inset of **Fig. 4(a)**, considering different connectivity ($2\times r_{cut}$). Timings represent steady-state GPU performance with warmup runs excluded. Standard CPU-bound NumPy operations for the equivalent dense block conversions are included for baseline comparison.

we consider two different scenarios: (1) data processing (forward transformation, pre-training) and (2) matrix reconstruction (backward transformation, post-inference). First, **Fig. F.1(a)** reports the 'data processing' case ('Matrices $\rightarrow$ Labels', $\mathbf{H} \rightarrow \mathbf{h}$), where multiple Hamiltonian matrices $\mathbf{H}$ are converted into orbital interaction labels $\mathbf{h}$ for supervised learning. We evaluate processing throughput by considering increasing fractions of each of the three datasets, resulting in increased total # atoms. Processing throughput between 7.3×10^5 and 1.2×10^6 atoms/s are measured, allowing us to process the largest molecular datasets of 110M structures with $\sim$50 atoms/structure (Levine et al., 2025) for training in under an hour, on a single process. Note that for $l_{max} = 8$, the data exceeds the memory of a single GH200 GPU before peak processing throughput can be reached. We therefore report the maximum value attained. Secondly, in **Fig. F.1(b)**, the 'matrix reconstruction' case is presented ('Labels $\rightarrow$ Matrix', $\mathbf{h} \rightarrow \mathbf{H}$). At the end of the inference step, the model generates orbital interaction labels $\mathbf{h}$ for a given atomic structure, which are then re-assembled into the $\mathbf{H}$ for downstream applications. Here, the difference in runtime stems primarily from the number of elements per label (l_{max}), but remains under 10 ms across a range of system sizes. Note that the variability in measured time/throughput at the same # atoms can be attributed to different molecular connectivity (# edges per atomic graph).

Next, we benchmark the reference PyTorch and our Triton implementations with three different l_{max} values, corresponding to selected datasets from **Table 2**. The results are presented in **Fig. F.1(c)**. At smaller data sizes (number of edges, and l_{max}), the execution time in the PyTorch reference implementation is dominated by kernel launch overhead from multiple matrix products. In this range, our Triton code shows speedups of $\sim$500$\times$ (100-1k edges) to $\sim$50$\times$ (10k edges), as the launching overhead was removed through kernel fusion. At higher data volumes (100k+ edges), the reference implementation becomes limited by global memory traffic. Still, our Triton implementation achieves a $\sim$5-6$\times$ speedup from the combination of kernel fusion and l-wise block packing.

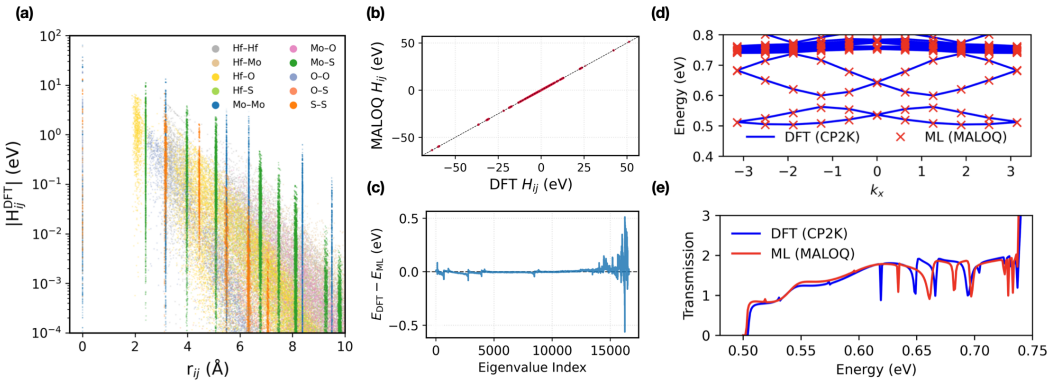


Figure G.1: (a)Random selection of 1 million sub-sampled nonzero matrix elements of $\mathbf{H}$ ($|H_{ij}|$) from the 138 millions in **Fig. 5(a)**, as a function of the interatomic distance (r_{ij}) for one MoS_2/HfO_2 sample. Each scatter point is colored according to the atomic element interaction it originates from in the Hamiltonian. The regular vertical patterns in the scatter points indicate Mo-S interactions, while the scattered background contains the orbital interactions within the oxide and at the interfaces with MoS_2. (b) Parity plots between the DFT and ML matrix elements of the $\mathbf{H}$ corresponding to the same MoS_2/HfO_2 matrix as before. (c) Energy difference (error) between all eigenvalues of $\mathbf{H_{DFT}}$ and $\mathbf{H_{ML}}$ from the spectrum in **Fig. 5(b)**. (d) Conduction bandstructure of the MoS_2/HfO_2 sample from **Fig. 5(b)** as obtained from CP2K's (solid blue lines) and MALOQ's (red crosses) Hamiltonian matrix. (e) Transmission function comparison through the same structure. Note that pure MoS_2 extensions are added on each side of the MoS_2/HfO_2 central part and act as electron injection contacts.

G MoS_2/HfO_2 STRUCTURES - ADDITIONAL DATA

When dealing with MoS_2/HfO_2 systems of size $3.16\times3.16\times3.29$ nm^3, CP2K produces highly sparse Hamiltonian matrices whose entries decay exponentially with interatomic distance $\mathbf{H_{DFT}}$, as can be seen in **Fig. G.1(a)**. Beyond a cutoff distance r_{cut} of 10 Å between two interacting atoms, the H_{ij} entries can be safely neglected. As a consequence, a minimum cell dimension of 2 nm between periodic images is necessary to fully saturate the decay of the $\mathbf{H_{ij}}$ interaction terms and avoid artefacts stemming from the periodic boundary conditions of DFT.

Figure G.1(b) shows through a parity plot that the entries of the $\mathbf{H_{ML}}$ predicted by MALOQ for a $3.16\times3.16\times3.29$ nm^3 MoS_2/HfO_2 consisting of 2,088 atoms agree very well with their DFT labels with an average matrix element error of 37 μeV. As illustrated in **Fig. 5(b)**, diagonalizing the GEVP electronic structure problem with $\mathbf{H_{ML}}$ as input results in accurate eigenvalues, when compared with DFT, around the MoS_2 conduction band edge for our MoS_2/HfO_2 system of interest, with an average error of 38 meV across all 16,608 eigenvalues. Notably, this error is mostly concentrated at high energy values and minimal around the MoS_2 band gap where electronic transport occurs (**Fig. G.1(c)**, eigenvalue indices between 8200 and 8600). In this range, the average eigenvalue error is equal to 20 meV.

To further validate MALOQ's capabilities, the conduction band dispersion produced with $\mathbf{H_{ML}}$ for the same $3.16\times3.16\times3.29$ nm^3 MoS_2/HfO_2 structure as before is shown in **Fig. G.1(d)**, demonstrating an excellent agreement with DFT, both in terms of band edge alignment and curvature. Similarly, the transmission function $T(\varepsilon)$ through this sample (**Fig. G.1(e)**), as determined with QTBM (see Appendix A) and MALOQ's $\mathbf{H_{ML}}$, accurately reproduces the DFT one over a large energy range.