Data-Driven GPU Performance Model for Molecular Dynamics in Computational Drug Discovery

Udari De Alwis, Maria Barrera, Thomas J. Ashby
IMEC, Leuven, Belgium
Email: {udari.dealwis,mariacecilia.barreramoreno, tom.ashby}@imec.be

***Abstract*—Molecular dynamics (MD) simulations increasingly rely on GPU acceleration to meet the computational demands of large biomolecular systems. However, predicting application performance across diverse molecular systems and heterogeneous GPU architectures remains challenging due to complex heterogeneous MD kernels with varying compute and memory demands. This work presents a hierarchical, explainable surrogate model that integrates molecular descriptors, simulation parameters and GPU architectural features to learn architecture-aware representations of MD execution on GPUs.**

***Index Terms*—molecular dynamics, GPU performance modelling, Gromacs, knowledge distillation, explainable machine learning**

I. Introduction

Molecular dynamics (MD) simulations play a central role in computational drug discovery by enabling the investigation of molecular structure, dynamics, and interactions at atomic resolution. As simulation complexity increases, GPU acceleration has become essential for achieving practical performance. However, predicting MD performance remains difficult because execution depends on molecular properties, simulation parameters, runtime behavior, and GPU architecture. Conventional analytical models, including Roofline analysis, provide only coarse estimates and cannot capture the heterogeneous execution characteristics of modern MD workloads.

This work proposes a hierarchical data-driven surrogate model that combines molecular descriptors, Gromacs simulation parameters, GPU architectural information, and NVIDIA Nsight Compute (NCU) kernel metrics. A Teacher–Student learning strategy transfers detailed profiling knowledge into a lightweight model capable of accurate inference without requiring kernel-level profiling.

II. Background and Motivation

MD applications execute diverse GPU kernels with different computational intensity, memory-access behavior, synchronization requirements, and cache utilization. Overall execution time therefore depends on the interaction between workload characteristics and hardware architecture.

Existing analytical models simplify GPU execution using arithmetic intensity and peak hardware performance, making them unsuitable for representing heterogeneous MD kernels. Likewise, most learning-based performance models target deep learning workloads with comparatively regular execution patterns. Consequently, there remains a need for a portable performance model capable of accurately predicting MD execution on previously unseen GPU architectures.

This research is funded by the imec.prospect project IMPACT.

III. Proposed Methodology

The proposed framework employs a hierarchical Teacher–Student neural network. Molecular descriptors are extracted from Gromacs `.gro` files, simulation parameters from `.mdp` files, application timing from Gromacs `.log` files, and GPU architectural descriptors from hardware specifications.

Kernel-level execution metrics are collected using NCU profiling. Instead of representing these measurements as a flat feature vector, kernel invocations are organized hierarchically according to the Gromacs execution structure for non bonded interaction calculation step. Individual kernel invocations are encoded into latent representations, aggregated at the function level using DeepSets pooling, and combined through an attention mechanism to generate compact Particle–Particle (PP) and Particle Mesh Ewald (PME) embeddings.

These hierarchical embeddings are fused with molecular, simulation, and GPU features through a multi-head attention module to produce a shared latent representation used to predict PP and PME execution times. A bottleneck layer further encourages the latent space to encode interpretable hardware-performance concepts, including computation, memory behavior, cache utilization, scheduling efficiency, and communication overhead.

During training, the Teacher network utilizes the complete hierarchical profiling representation together with system and GPU features. Knowledge distillation transfers this information to a Student model that requires only molecular descriptors, simulation parameters, and GPU characteristics during inference, substantially reducing deployment cost.

IV. Evaluation and Results

The framework was evaluated using protein–water Gromacs systems executed on NVIDIA V100, A100 (40 GB and 80 GB), H100, and L40 GPUs. Both Teacher and Student models achieved high prediction accuracy for PP and PME execution times. Despite operating without kernel-level profiling during inference, the Student closely matched Teacher performance, demonstrating the effectiveness of the distillation strategy.Both models achieve high $R^2 > 0.96$ with low RMSE.

Hardware portability was assessed using leave-one-GPU-out and leave-one-GPU-family-out experiments. When the complete NVIDIA A100 family was excluded during training, the Student achieved approximately $R^2 = 0.98$ for PP prediction and $R^2 = 0.97$ for PME prediction, indicating strong generalization to previously unseen hardware. The learned hierarchical representation also provides interpretability by identifying the contribution of PP and PME components through attention weights and bottleneck features.

V. Conclusion

This work presents a hierarchical Teacher–Student surrogate model for GPU-accelerated molecular dynamics performance prediction. By integrating molecular descriptors, simulation parameters, GPU architectural information, and hierarchical kernel-level profiling, the framework provides accurate, interpretable, and portable prediction of PP and PME execution times. Knowledge distillation removes the dependence on expensive runtime profiling while maintaining strong accuracy across previously unseen GPU architectures, providing a practical foundation for architecture-aware performance modeling in computational drug discovery.