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

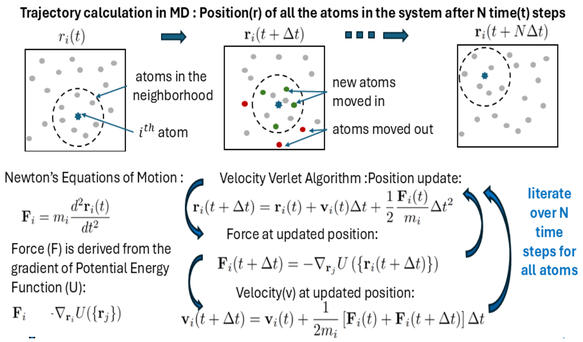
Udari De Alwis, Maria Cecilia Barrera, Thomas J. Ashby, IMEC, Leuven, Belgium

IMPACT — Integrating Machine learning, Performance and Advanced Compute for Therapeutics

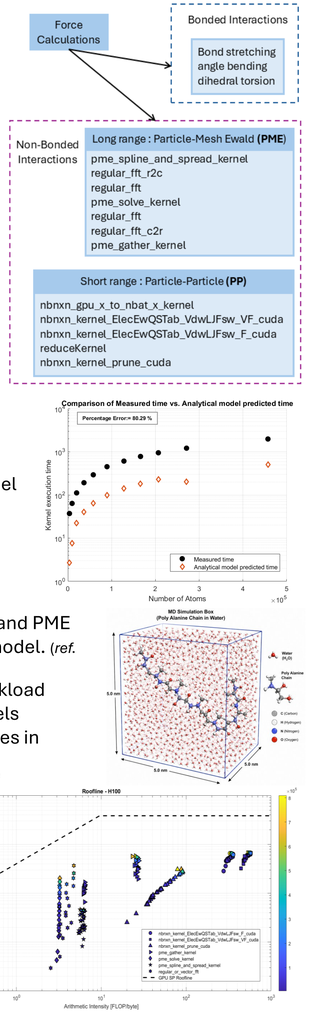
Abstract: Molecular dynamics simulations increasingly rely on GPU acceleration, yet predicting application performance across diverse molecular systems and GPU architectures remains challenging due to complex interactions between simulation parameters, runtime behavior, and hardware characteristics. We present a hierarchical, reasonably explainable surrogate model that integrates molecular descriptors, GROMACS simulation parameters, and kernel-level GPU performance counters to learn architecture-aware representations of Particle–Particle and Particle Mesh Ewald computations. A teacher–student learning framework distills detailed profiling knowledge into a lightweight predictive model that requires only system and GPU descriptors at inference, enabling accurate, interpretable, and portable performance prediction across GPU architectures.

Molecular Dynamics Kernel Performance

- Molecular Dynamics (MD) is a computational technique that simulates the temporal evolution of atoms in molecular systems by numerically solving Newton's equations of motion, providing valuable insights into molecular structure, dynamics, stability, and interactions that support drug discovery and molecular design.



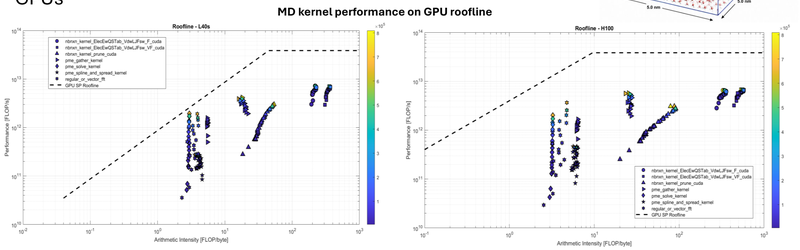
- Bonded interactions generally consume a smaller fraction of the total computation time compared to short-range (PP) and long-range (PME) forces.*(ref. fig. 'Force Calculations')*
- PP and PME typically dominate the computational cost of MD in simulation software like GROMACS[1].



- Existing roofline based analytical models cannot accurately represent complexity of the MD kernels, while full architecture simulators are prohibitively expensive for large-scale evaluation. (*ref. fig.: 'Comparison of measured and analytical model predicted time'*)
- We developed a data-driven surrogate model that leverages application features and GPU profiling information to accurately predict runtime and characterize hardware bottlenecks across multiple GPU architectures.

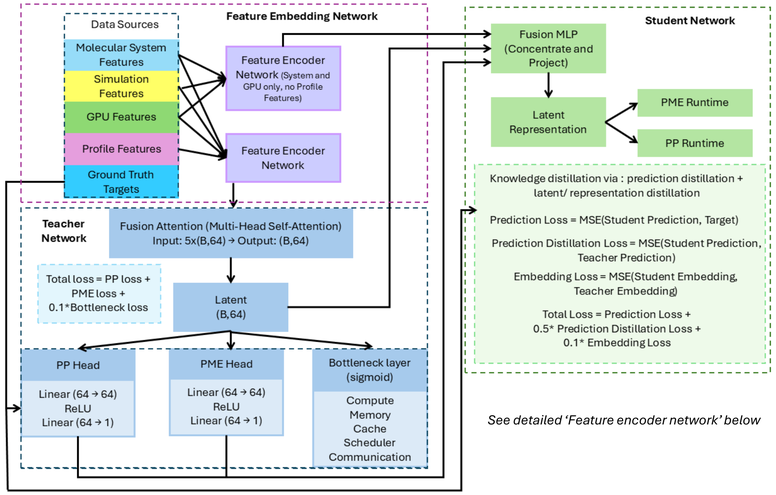
- Protein [2] in water systems are used as scale up workloads in GROMACS (v2025.4) to investigate the kernel performance. (*ref. fig.: 'MD simulation box'*)
- We profiled this workload across H100, A100, L40 and V100 GPUs using Nvidia Nsight Compute (NCU)[3] and Nsight Systems (NSys)[4]

- The PP kernels tend to fall in the compute bound region and PME kernels fall in the memory bound region of the roofline model. (*ref. fig.: 'MD kernel performance on GPU roofline'*)
- The compute and memory utilization grows with the workload size, and for majority of the workload sizes, the MD kernels underutilize the available compute and memory resources in GPUs



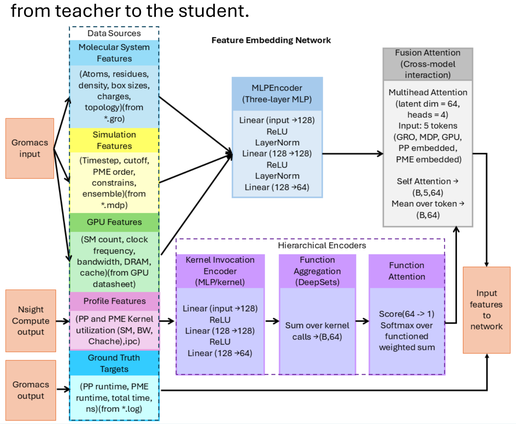
Hierarchical Hardware Feature Learning

- The rapid evolution of GPU architectures makes application performance modeling increasingly difficult using traditional analytical approaches.
- Recent research [5-7] has explored machine learning approaches for predicting GPU application performance. While these approaches achieve high prediction accuracy for regular and largely static AI workloads, they are not directly applicable to MD, whose performance relies on heterogeneous kernel executions over dynamic data.
- This motivates the development of a data-driven surrogate model that leverages application features and GPU profiling information to accurately predict runtime and characterize hardware bottlenecks across multiple GPU architectures.

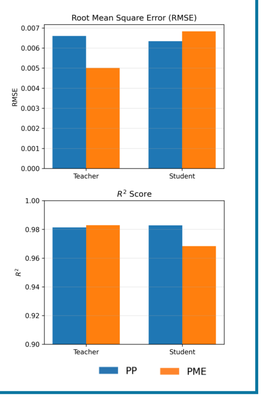


See detailed 'Feature encoder network' below

- The Teacher network combines molecular system features, simulation parameters, GPU architectural descriptors, and NCU kernel profiles.
- The Student network uses the same features as the Teacher network, except for the NCU profiles. They are omitted to enable direct prediction from easily calculated features without needing to do expensive profiling runs.
- Both models predict PP and PME execution times, and achieve high R^2(>0.96) with low RMSE; the Student closely matches the Teacher, with slightly better PP performance but somewhat higher PME error. (*ref. fig.: 'R^2 and RMSE of Teacher and Student models'*)
- Prediction distillation loss, embedding loss contributes to distil the architectural knowledge from teacher to the student.



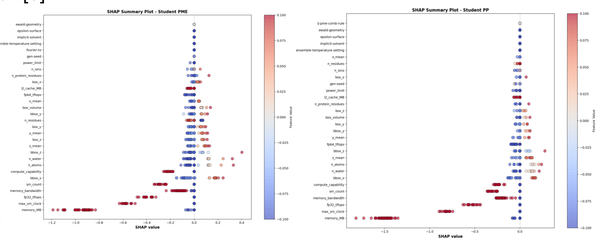
R^2 and RMSE of Teacher and Student models



GPU Architectural Understanding and Generalization

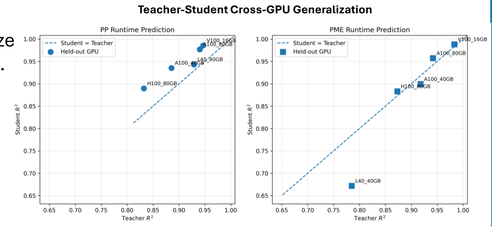
- SHAP (SHapley Additive exPlanations) measures how much each input feature contributes to the model's prediction [8].
- The results demonstrate cross-GPU generalization, and that the student model can learn transferable relationships between molecular workloads and GPU architectures

SHAP summaries of the Student model



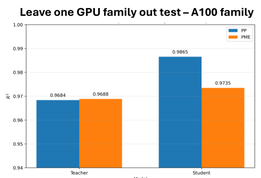
- GPU memory capacity is by far the most influential hardware feature for both PP and PME, indicating that the model strongly associates the available memory capacity with GPU performance for these workloads. (*ref. fig.: 'SHAP summaries of the Student model'*)
- GPU compute characteristics : `max_sm_clock`, `fp32_tflops`, `memory_bandwidth`, and `sm_count` are the next most important hardware features, showing that the models learns meaningful relationships between GPU computational resources and kernel runtime.
- Workload/system features such as `n_atoms`, `n_water`, `bounding_box_dimensions`, and `box_volume` have moderate importance, reflecting the dependence of kernel execution time on problem size and spatial configuration.
- For knowledge distillation, this provides evidence that the student has learned hardware workload relationships rather than relying solely on the GROMACS configuration.
- Leave-one-GPU-out testing demonstrates the ability to generalize in both teacher and student models.
- The student model consistently matches teacher PP prediction accuracy, demonstrating strong transfer of architecture-aware representations. (*ref. fig.: 'Teacher-Student cross GPU generalization'*)
- PME prediction generalizes slightly less well compared to PP prediction.

Teacher-Student Cross-GPU Generalization



- For example, the model achieves $R^2 = 0.987$ for PP, $R^2 = 0.973$ for PME on A100 GPU family, when the whole family is omitted from training and used only for testing. (*ref. fig.: 'Leave one GPU family out test'*)
- This indicates the transferable relationships between workload characteristics and GPU architectural behaviour.

Leave one GPU family out test – A100 family



Conclusion and Future Work

- This work introduced a hierarchical, reasonably explainable surrogate model that integrates molecular descriptors, GROMACS simulation parameters, application-level timing information, and kernel-level GPU performance counters.
- This model can be used to estimate the run times and hardware utilization of GPUs in local/cloud computing services running MD workloads.
- Future extensions : Integration of domain decomposition and communication model for MD to extend model beyond single GPU, Generalization across other types of GPUs.

IMPACT, imec .prospect project :

- Advancing discovery and development of novel therapeutics
- Multidisciplinary partnerships with Biotech companies
- IMPACT is funded by the imec.prospect initiative

References

[1] M. J. Abraham,et al, SoftwareX,2015.
[2] H. M. Berman,et al,NAR,2000.
[3] "NVIDIA Nsight Compute Documentation." [Online].
[4] "NVIDIA Nsight Systems Documentation." [Online].
[5] Y. He, et al, SC, 2021.
[6] N. Bhendawade, et al., ISCA, 2024.
[7] X. Zhao, et al.,ISPASS, 2024.
[8] S. M. Lundberg,et al,NeurIPS,2017.

connect with us: Udari.DeAlwis@imec.be, Tom.Ashby@imec.be