PERFORMANCE-DRIVEN DESIGN OF MACHINE-LEARNED INTERATOMIC POTENTIALS

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine-learned interatomic potentials deliver close to density-functional accuracy at a small fraction of the cost, but they are still orders of magnitude more computationally expensive than classical force fields, which keeps most biomolecular simulation out of reach. Much of that cost is not intrinsic to the physics but is due to operations that were designed for model expressiveness and accuracy. We present the SPLine Interatomic Cluster Expansion (SPLICE), an equivariant message-passing architecture designed for inference speed. Its radial basis is a per-channel spline carrying two learned parameters, which has a closed form and is evaluated inside the message-passing kernel. We also decouple the width of the per-edge tensor product from that of the node features, so that the model capacity can grow with the least impact on inference performance, leading to a family of models along a speed and accuracy front. Trained on biomolecular data, the model family spans 0.2 to 4 μs per atom per step on a single NVIDIA H100 80GB GPU, with the fastest models handling more than a million atoms. We evaluate it on conformer energies, on physically motivated benchmarks, and on condensed-phase dynamics of a solvated protein and show on par accuracy with the state-of-the-art models.

1 INTRODUCTION

Machine-learned interatomic potentials (MLIPs) promise the accuracy of density functional theory at a fraction of its cost. However, they still remain far more expensive than the classical force fields. A fixed-charge classical potential costs a few thousandths of a microsecond per atom per step, foundation MLIPs can cost tens to hundreds of microseconds per atom per step (Table 1), four to five orders of magnitude more. For a biomolecular simulation, which is measured in microseconds of trajectory, that difference decides whether a study takes a week or a century. Table 1 presents a qualitative picture, as several measurement choices carry large multiplicative factors. For instance, several papers quote systems of a few thousand atoms, where fast models are dominated by kernel launch overheads, while production runs reach 10^4 to 10^6 atoms, either for large systems or batched inference of multiple systems packed into a single model call. Whether forces are taken as the gradient of the energy or not can be worth $5\times$ for a single model family (Rhodes et al., 2025), but has documented costs in energy conservation (Mazitov et al., 2025). For molecular dynamics (MD), neighbor-list construction can be a significant overhead for a fast model. Precision, compilation and implementation quality differ as well across the available solutions. It is thus important to study the compute cost of various models under the same protocol in order to understand the cost vs accuracy trade off.

Early MLIPs for molecules learned invariant descriptions of a local environment with graph neural networks (Schütt et al., 2017). Later models built equivariance into the representation itself (Batzner et al., 2022), which bought accuracy and data efficiency at a given model size. The MACE family (Batatia et al., 2022) reached high body order cost-effectively by contracting the messages on a node, instead of stacking more layers. Most of what followed has *added* machinery to that picture: convolutions restructured to reach higher angular degrees (Passaro & Zitnick, 2023), attention and transformer blocks (Liao et al., 2024; Qu & Krishnapriyan, 2024), and very large models trained across all of chemistry, including mixtures of experts (Wood et al., 2025). Each step aimed at improved accuracy or generalization, often at increasing cost. Speed has been approached in two ways: One takes the architecture as given and removes something: forces predicted directly by

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.

Table 1: Reported inference cost ranges in the literature, converted to μs per atom per step. As the references use varying protocols and different hardware for inference, below is given only for a qualitative description of the landscape. Sources: [1]Amber developers (2024), [2]Shuang et al. (2026), [3]Li et al. (2026b), [4]Kabylda et al. (2025), [5]Rhodes et al. (2025), [6]Batatia et al. (2022), [7]Wood et al. (2025), [8]Li et al. (2026a).

band	examples	μs/atom
classical fixed-charge MD	*AMBER, GROMACS, OpenMM* [1]	*0.003–0.02*
classical-MD alternative	NEP/UNEP [2], DPA4C [3]	0.03–0.5
fast MLIP	GRACE-FS [2], SO3LR [4], Orb-v3-dir. [5]	1–12
mainstream foundation	Orb-v3-cons. [5], MACE-MPA-0 [6], MatterSim, UMA-S [7]	20–90
large / high accuracy	UMA-M/L [7], SevenNet-MF, NequIP-MP-L, DPA4-Pro [8]	200–2300

a separate head (Rhodes et al., 2025), a restricted body order and radial basis, as in GRACE-FS (Shuang et al., 2026), or a trained model distilled into a smaller one, as in the recent compressed DPA4C family (Li et al., 2026b). Potentials in the NEP family (Shuang et al., 2026) come within an order of magnitude of classical MD on a correspondingly narrow chemical domain. The second approach changes the expensive operations themselves: SO3krates (Frank et al., 2024) pairs a sparse equivariant representation with a self-attention mechanism separating invariant from equivariant information, and removing edge tensor products. Built on it, SO3LR (Kabylda et al., 2025) runs at 3.25 μs/atom on biomolecular systems while also carrying explicit long-range electrostatics.

We take this second route and start from a base equivariant message passing (MP) network, ask what each layer costs, and rewrite the operations that dominate it, while attempting to keep accuracy as close to the starting baseline as possible. As a result, we propose the SPLine Interatomic Cluster Expansion (SPLICE). Main contributions of the work are:

1. A radial basis built from splines with two learned parameters per channel, whose functional form allows it to be fused into the MP kernel, so no per-edge radial feature is written to memory.
2. Modularization of operations in a given layer so that each module can be optimized for best trade-off individually
3. An implementation of these approaches to build a family of models trained on molecular data, benchmarked for speed, accuracy and condensed-phase stability on systems and suites relevant to biomolecular simulation.

2 SPLICE MODEL ARCHITECTURE

SPLICE is an equivariant graph neural network (GNN) based on the MACE family architecture (Batatia et al., 2022). Atoms are nodes and every pair within a cutoff is an edge; the network is a stack of identical layers, each of which passes messages along the edges and then applies a node-local operation: a symmetric contraction of the gathered messages followed by a linear map onto the node features carried to the next layer. A readout maps the final node features to a per-atom energy; forces and stress are the gradients of their sum, so the model is conservative by construction. A more rigorous definition of each block is given in Appendix B.2. Figure 1(a) shows the network and Figure 1(b) one layer of it, compared to a standard implementation: each edge ij carries an interatomic distance r_{ij} and the spherical harmonics $Y^{\ell}(\hat{r}_{ij})$ of its direction. In MACE models, fixed functions of the distance are passed through a multi-layer perceptron (MLP) producing weights for each tensor-product path, a large per-edge tensor that is written to and read from global memory. We replace the MLP with a basis of piecewise cubic splines, one per channel, carrying two learned parameters each: a knot position $e_k r_{\max}$ beyond which the channel vanishes, and a shape angle t_k. Figure 1(c) shows the resulting functions at initialization. Every channel then becomes a closed-form expression in r_{ij}, so nothing needs to be stored: the basis is computed in registers inside the MP kernel, so the per-edge radial tensor that dominates the memory traffic of the original layer never exists. Appendix B.1 gives the closed form of the radial function, and the role each of the two parameters plays. While the radial MLP could itself be fused (as done in our implementation)

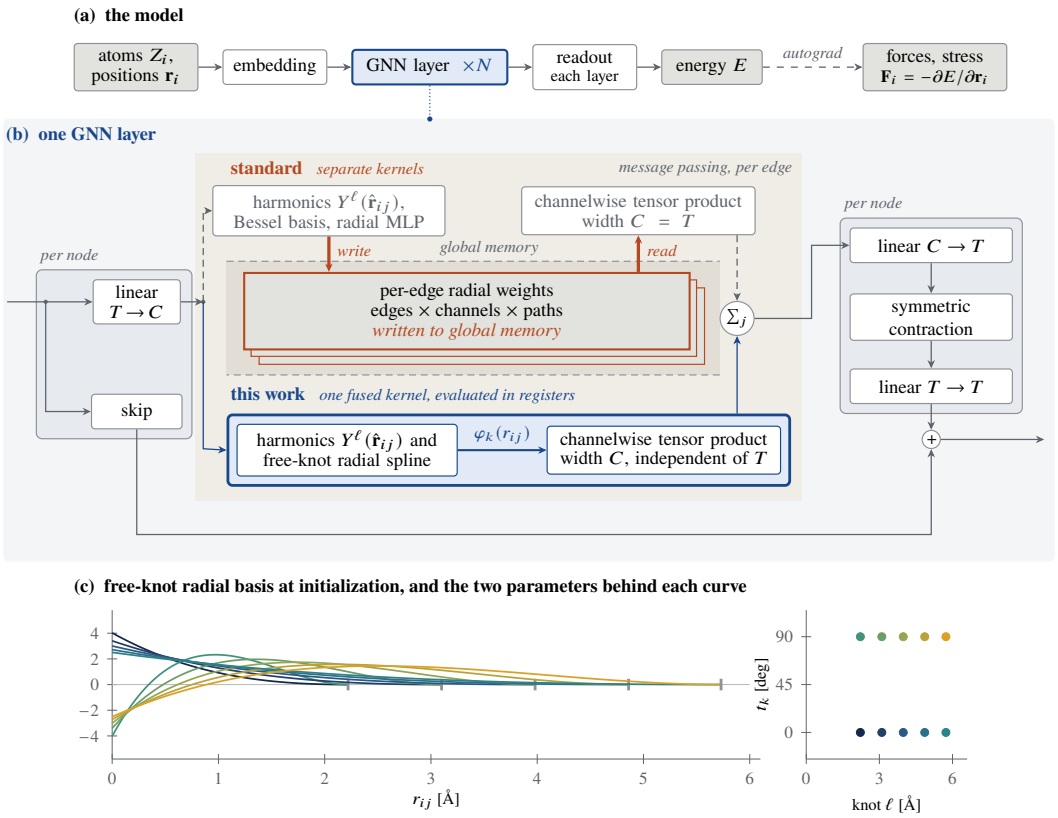

Figure 1: (a) Sketch of the SPLICE model architecture. Every layer emits an atomic energy; forces and stress are gradients of their sum. (b) A single GNN layer. A first linear layer projects node features to width C, then enters the MP operation (shaded band) operating on edges (20-90× the atoms). Common MP implementations (path at the top) use a fixed basis of radial functions, processed with an MLP that writes per-edge weights to memory, while SPLICE (bottom) evaluates spherical harmonics and a learnable radial basis in registers, and fuses both with the tensor product in a single kernel, so no per-edge information is written to device memory. In the common implementations, widths C and T are the same, while in SPLICE they are refactored to be independent. The trunk then lifts the summed message to width T and contracts it, and the skip rejoins at the sum on the right. Appendix B.2 gives the layer as equations. (c) The free-knot radial basis at initialization. Each function is a cubic spline vanishing to first order at its own learned point, "knot", $e_k r_{\max}$, with a learned shape angle t (two trainable scalars per channel, see Appendix B.1). The right panel shows each curve's parameter in the (knot, t_k) plane with the respective color.

it remains a sequence of dense layers over a materialized intermediate, whereas this form removes it entirely.

An architecture designed around inference cost requires an efficient implementation to run it. To ensure that, we use a concurrently developed GPU compiler, Loom (Anonymous, 2026), which fuses fixed-shape functions with indexed reads and reductions into a single kernel. Appendix A describes the implementation. The fused kernels in both SPLICE and the MACE reimplementation benchmarked here are generated with Loom, so that the cost can be compared independently of the impact of the compiler. We take two structurally identical scalar MACE models that differ only in the radial path, one with the standard radial MLP feeding the tensor product and one with the free-knot basis fused into it, and profile both on the same 10 000-atom water box. A forward pass of the standard model takes 6.7 ms of GPU kernel time. Of that, the dense layers of the radial MLP take 2.4 ms and the tensor product 1.9 ms, with a further 1.9 ms evaluating the spherical harmonics and the radial basis that feed them. The fused model replaces all three with one kernel that takes 1.2 ms,

and its whole forward pass is 1.8 ms. The tensor product goes from 28% to 68% of a 3.8× shorter step: the radial path stops being the bottleneck.

Modularity of SPLICE architectures is also noteworthy, as described in Figure 1. The MP operation and the trunk that carries node features between layers are separate blocks with different costs. The tensor product runs over channels and edges, so its cost scales with the number of edges (approx. 20–90× the atoms). Its memory footprint also sets the largest system the model can hold. On the contrary, the trunk operations are dense contractions on node features, which can map onto tensor cores, are cost-effective for their parameter count, and scale efficiently with size. Treating the channel width C and the trunk width T as one number, as the standard formulation does, forces them to move together. Separating them lets the trunk be widened while the tensor product stays small. The same argument applies within a single operation, where the multiplicity of each angular momentum ℓ can be set independently (Appendix I).

3 Training on neutral SPICE2

To compare the accuracy of this new architecture we need a dataset with an established benchmark and an established set of trained baselines. We use SPICE2 (Eastman et al., 2024) as curated and released by InstaDeepAI (Brunken et al., 2025): a neutral-only, heavily filtered subset of 1 734 158 training and 87 764 validation configurations over the 12 elements H, B, C, N, O, F, Si, P, S, Cl, Br and I, covering drug-like molecules, peptides, amino-acid–ligand pairs, dimers drawn from DES370K (Donchev et al., 2021) and solvated systems. The curation is public, and it is also the training set of the reference models in the audit suite, so the comparison is at fixed data.

On that data we train a family of models spanning more than an order of magnitude in inference cost, varying the channel and trunk width, and the angular momentum carried by the trunk. As a controlled comparison, we train the same family with the radial basis replaced by a radial MLP, keeping the rest fixed. Both families use our optimized code, rather than unoptimized implementations, so the architectural comparison is fair. Figure 2 shows the result: the SPLICE family sits below the radial-MLP family, and the separation is large: at matched accuracy SPLICE is 5 to 8× faster. We select four models from this front and use them throughout the paper. They differ in MP channel width C, the maximum order L and multiplicity T of the node irreps in the trunk: **SPLICE-S** (C=32, L=0, T=128), **SPLICE-M** (C=64, L=1, T=256), **SPLICE-L** (C=128, L=2, T=512) and **SPLICE-XL** (C=256, L=2, T=2048). Table 2 reports the speed and maximum capacity of these models, also highlighting the memory efficiency of the architecture (capacity and speed for every model in the family is in Appendix K).

To further validate the capacity of SPLICE models, we evaluate the selected four on MLIPAudit (Wehrhan et al., 2025), a suite of physically motivated benchmarks released by the dataset curators. Figure 3 shows the results on selected tests, compared to several baselines (all tests are normalized to 1). The entries marked "SPICE2" are trained on the same exact split, proving the accuracy of SPLICE models to be in line with other architectures, generally improving with size with SPLICE-XL leading several categories. While the result on water density breaks the size trend, similar poor performance of more expensive models suggests that the perfect score of the smaller models might just be fortunate. When general-purpose models trained on far more data are also included in the comparison (light gray bars in Figure 3), SPLICE models remain behind on some tests, confounding the impact of data with architecture. We show further tests in the suite in Appendix H.

4 Physical and simulation-relevant validation

MLIPs are built to enable useful and stable molecular simulations. However, many models proposed in the literature are found to be not accurate nor stable enough under nominal conditions expected from molecular mechanics force fields (Fu et al., 2023). This section runs three tasks relevant to real applications, on the four selected models, against the held-out SPICE2 validation split (87 764 structures, 4 587 SMILES).

Accurate simulations require reliable deformation energies, the energy cost of distorting a molecule away from its own minimum, which a docking or FEP protocol pays repeatedly. The upper two rows of Figure 4 give the mean percent error per subset on this task. The error falls monotonically with

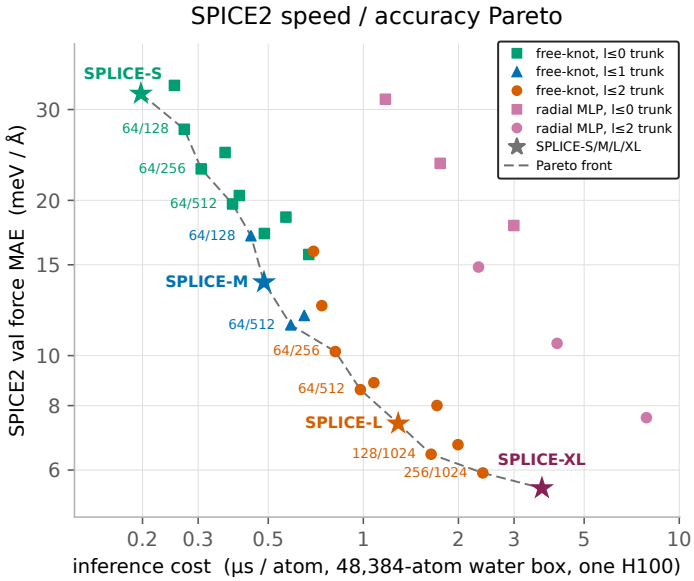

Figure 2: SPLICE models inference speed for family of models trained on the SPICE2 dataset versus accuracy of the models obtained on the SPICE2 validation set. x: inference cost, μs per atom per step, measured on 48 384-atom water box, on a single NVIDIA H100 80GB GPU, using tf32 precision on the equivariant linear layers, channel-sorted weights. y: SPICE2 validation force mean absolute error (MAE), averaged over the eight validation subsets of Brunken et al. (2025). Speed vs accuracy front is explored by changing the hyperparameters (C/T) where C is the channel width of the channelwise TP, and T is the multiplicity of the node-feature irreps. Colors represent different maximum order L. In order to ablate for the impact of the new basis, we reproduce the standard message passing architecture as in MACE (Batatia et al., 2022) (pink triangles). Radial cutoff is 6.0 Å for all models. 6.0 Å. We select four representative configurations, indicated with stars, along the speed vs accuracy front to use in the rest of the work.

Table 2: Inference cost and capacity on one NVIDIA B200 180GB GPU. DPA4C-Mini (Li et al., 2026b) and UMA-S (Wood et al., 2025) for scale; protocol in Appendix J.

model	μs/ atom	max atoms
DPA4C-Mini	0.030	8.92M
SPLICE-S	0.161	4.44M
SPLICE-M	0.494	2.60M
SPLICE-L	1.82	1.27M
SPLICE-XL	4.45	336k
UMA-S	112	142k

model size in every subset, with SPLICE-L and SPLICE-XL within chemical accuracy. Another property is the interaction force, the part of the force on each atom that moves its own fragment rigidly, which is defined and computed for configurations with multiple fragments in Appendix D. The bottom row of Figure 4 reports this error. We also evaluated two community models on these tasks: FeNNix-Bio1-S sits at **SPLICE-L**'s own throughput (see Section 5), while UMA-s-1.1 is a very accurate model and runs 25× slower. At matched speed the margin is large: on the aggregate deformation error **SPLICE-L** is 2.8× better than FeNNix-Bio1-S. At the other end of the spectrum, our large models are as accurate as UMA-S while being considerably faster (additionally, the OMol dataset has a large overlap with this validation set, see Appendix E).

To be useful for molecular simulations, a potential must also be stable in dynamics at a usefully large timestep as well. To assess this, we consider a third task: a condensed-phase trajectory of the tyrosine kinase TYK2 in complex with an inhibitor in explicit water. We run four cells, the apo protein and the holo complex with its bound inhibitor, each in two solvent box paddings, resulting in systems of 35–42k atoms. The assembly carries a net charge of −3 and no neutralising counterions. Dynamics are NVT at 298 K with a Langevin thermostat and a 0.5 fs timestep, run to 100 ps. The holo systems carry the 4GIH co-crystal ligand in its crystal pose. Before the production run: each box is equilibrated by energy minimization, then 100 ps of NVT and 2 ns of NPT at 298 K and 1 bar, with the protein backbone heavy atoms restrained for the whole of it. The models trained on neutral SPICE2 can run these simulations, in some cells for tens of picoseconds, but they do not stay stable: they strip protons from polar and charged residues and then diverge within one to tens of picoseconds. The failure is localized chemistry, not a thermal blow-up, so a run can be monitored as nominal right up to the step where it diverges.

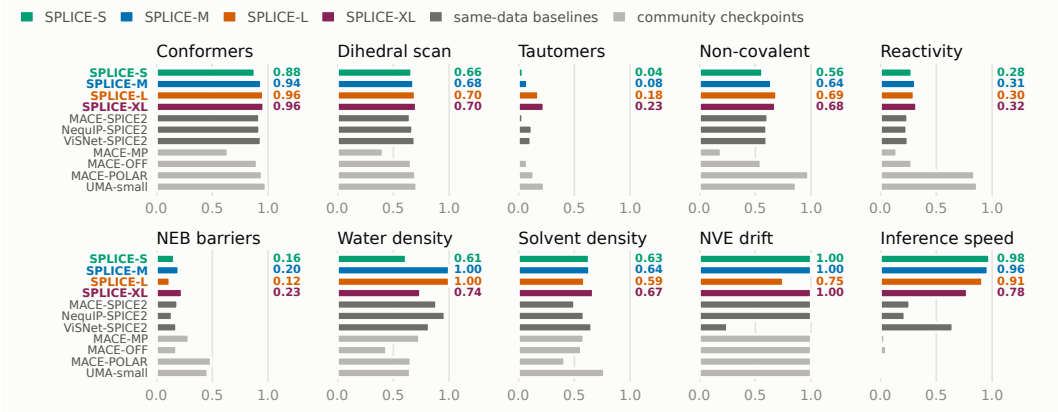

Figure 3: Selected benchmarks from MLIPAudit v0.1.6 (Wehrhan et al., 2025). Each panel is one benchmark; the score is the suite's own and *higher is better*, with 1.0 saturated. Three titles are shortened: *Conformers* is conformer selection, *NEB barriers* the nudged elastic band and *NVE drift* the energy-conservation benchmark. MACE-SPICE2, Nequip-SPICE2 and VisNET-SPICE2 (dark gray) are trained on the same curated SPICE2 dataset as the SPLICE models (Brunken et al., 2025), and their scores are the suite's own published results (Wehrhan et al., 2025). The rest (light gray) are computationally more expensive models trained on different datasets. MACE-POLAR and UMA-small are charge-aware models.

Since the curated SPICE2 removes all charged systems by construction, we reintroduce them in order to obtain a dataset able to handle polarized residues. SPICE2+Charge is the same curated neutral set plus the charged systems of SPICE 2.0.1 (selected on formal charge alone and otherwise unfiltered; see Appendix E for more details). We retrain the four selected models on it, with the total charge entering as a global conditioning input: the molecular charge and spin multiplicity are normalized, passed together through a two-layer perceptron, broadcast to every atom of the graph and added to the scalar channels of the species embedding. We call these models **SPLICE-C-S** to **SPLICE-C-XL**. After retraining, the charge-aware models can complete all four cells to 100 ps at 0.5 fs.

The trajectories are structurally stable: the Cα RMSD against the equilibrated reference stays near 1 Å throughout and never exceeds 1.4 Å, and no backbone, ligand or water bond breaks in any of them. The chemistry that does occur is of one kind: every persistent bond change is a lysine ammonium proton moving onto a neighboring carboxylate, and one particular salt bridge accounts for about half of the trajectories, in both systems and at several widths, without ordering by model size, which reads as a property of the structure rather than of any one model. The broader reading is that these transfers are easy here: several different ammonium–carboxylate pairs hop, so the barrier appears low across this system rather than peculiar to one contact. It is a different behavior from the charge-blind failure above, where the model strips protons and then diverges; here the transfer happens and the trajectory carries on with the notable exception of a rare failure of the C-XL model, starting from this mode. It is worth mentioning that this rare stability issue varies between different trainings of the same model: checkpoints that are indistinguishable on multiple static measures, the validation errors of Section 3 and the conformer and interaction tasks above, can differ in whether their trajectories are stable. A model therefore cannot be qualified for dynamics on static errors alone, nor on the short thermodynamic runs. Since this occurs only on the C-XL model, after tens of ps, it may be that this larger model of over 150M parameters, is potentially underconstrained. The community would greatly benefit from an agreed upon, extensive evaluation suite that is built to detect such stability and accuracy issues.

5 An External Accuracy and Speed Benchmark

Everything so far is measured on splits of our own training corpus, or on a related suite. For a fair comparison to the field, we use the test set proposed by Eastman et al. (Eastman et al., 2024): 800 molecules, scored by the spread of relative conformer energies over all 45 conformer pairs.

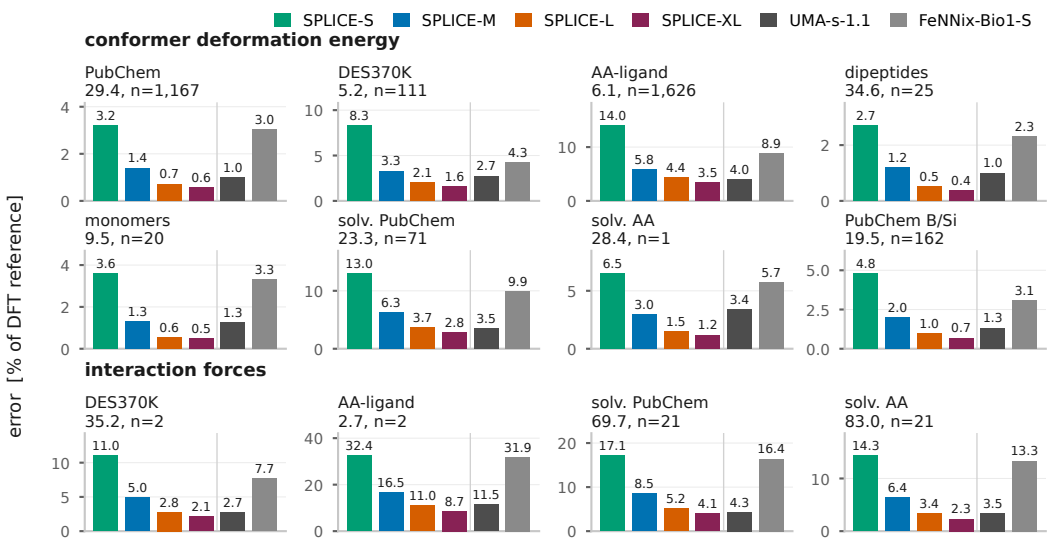

Figure 4: Validation error as a percentage of the DFT reference, one panel per subset: eight conformer-deformation subsets above, four interaction-force subsets below. Each panel is headed by its own DFT reference (kcal/mol above, meV/Å below) and the number of molecules or structures behind it. The comparison to make is *within* a panel: the error falls monotonically from **SPLICE-S** to **SPLICE-XL**. Gray bars are community models run on the same structures through the same harness; FeNNix-Bio1-S has throughput similar to **SPLICE-L**, while UMA-s-1.1 is $25\times$ slower. UMA-s is trained on OMol25, which contains 87.3% of the molecules plotted here, so its bars are in-sample where ours are held out. Full tables, confidence intervals, and another model are reported in Appendix D.

This data is held out of every model's training and is built to be larger and more application-shaped than typical training data (Appendix E). We re-measure the throughput of every published model ourselves so that both axes of Figure 5 come from the same machine under one protocol. However, it is important to note that the speed axis of Figure 5 is the benchmark's own definition, a MD step on a 2 661-atom box, which can be a limiting regime to evaluate a fast model: at that size much of the step is kernel-launch overhead and neighbor-list construction. Appendix J re-measures the same models on a protocol built for that regime: timing a single energy-and-force call on a 98 880-atom box with the neighbor list held outside the timed region. The ordering among the fast models changes materially, with the smaller DPA4C proving considerably faster.

In this test, neutral SPICE2 models are at the accuracy and speed frontier. Panel (d) shows that the mean error is dominated by few conformers: for SPLICE-L the top 5% of molecules are responsible for 65% of the total error. The published models (in gray), mostly trained on OMol25 (Levine et al., 2025), are much more even in accuracy distribution within a subset but also across the charged vs neutral subsets. SPLICE models appear suitable for neutral systems where they run one to two orders of magnitude faster (panel b) with comparable accuracy. The **SPLICE-C**-class models improve on their neutral counterpart by a factor 1.1 to $1.8\times$ on both neutral and charged splits. The neutral gain is not attributable to the charge embedding; it comes from a larger and more chemically diverse corpus. To show the capacity of the architecture, we also train versions of SPLICE on OMol25. Because it is far larger and more chemically diverse than SPICE2, and because long-range interactions matter for the charged species that dominate the remaining error, we make some changes: we add a third GNN layer for the largest models, and convert the linear skip connection layer to be species agnostic, as species-indexed layers would end up dominating the parameter count for this 86 species dataset. Moreover, we pin a fraction of the radial basis functions at the full cutoff distance, so that every layer retains more long-range reach. Finally, we add a total charge embedding as in the SPICE2+Charge models. The resulting models, dubbed **SPLICE-OMol** and numbered 1 to 7 from the fastest to the most accurate, score better on exactly the charged systems where the original SPLICE models were weak. On the all-800 mean they pass MACE-OMOL-0: the most accurate

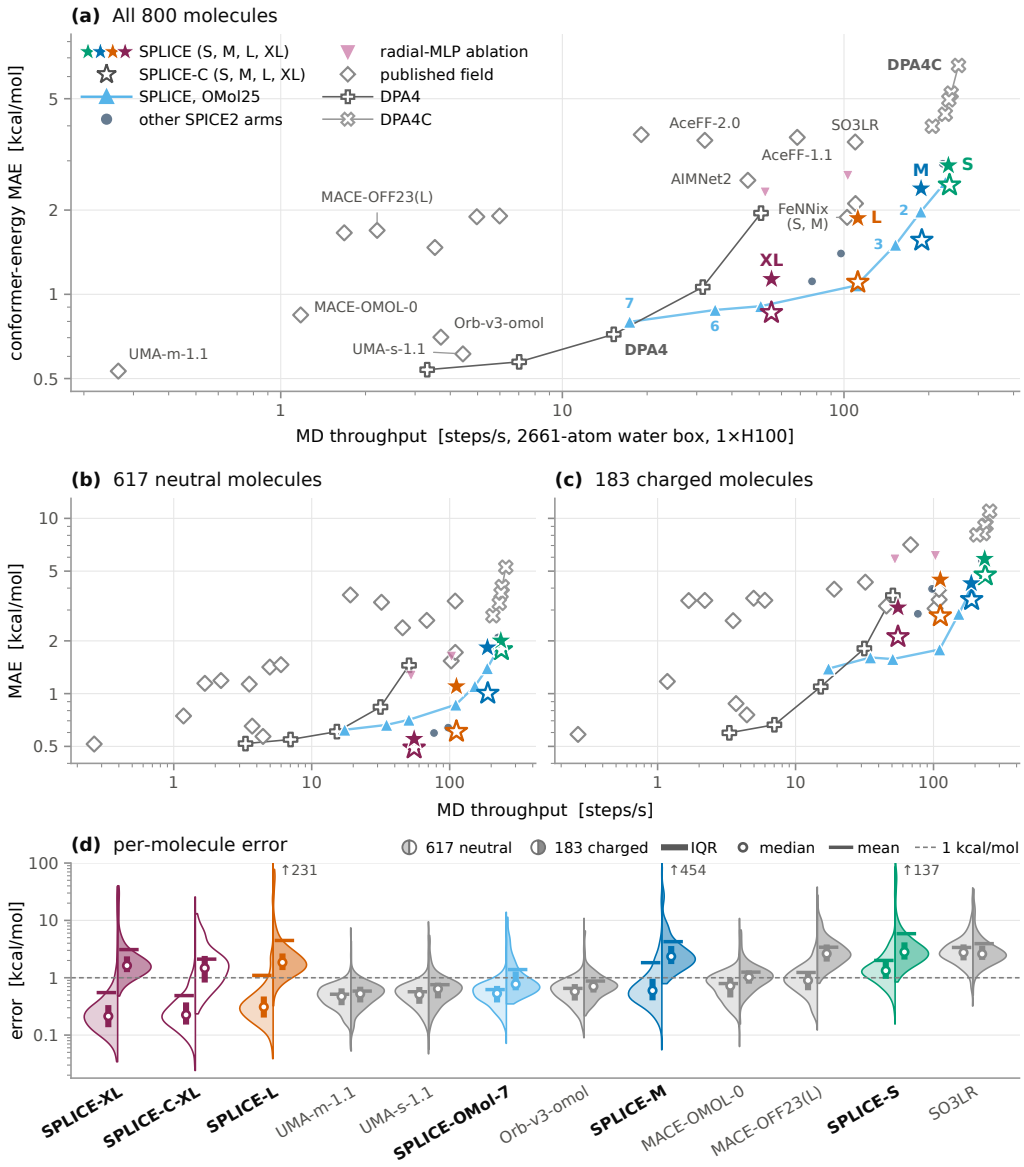

Figure 5: Conformer energies on the SPICE test set of Eastman et al. (2024). (a) all 800 molecules, error against MD throughput (measured on NVIDIA H100 80GB GPUs with the original protocol for all models). Empty diamonds are the originally published models, re-tested by us. Filled stars are the four **SPLICE-*** models trained on neutral SPICE2, *hollow* stars of the same color are their charged twins, **SPLICE-C-***, trained on SPICE2 with charged configurations. Blue triangles are SPLICE models trained on OMol25, numbered **SPLICE-OMol-**N of Table 6 where the ladder has room for a digit. Hollow crosses are the DPA4 family, also trained on OMol25 (Li et al., 2026a;b). (b, c) the same comparison split by neutral vs charged conformers. (d) per-molecule error distribution, split within each violin: neutral molecules on the left half, charged on the right, ordered by the neutral median. The vertical bar is the interquartile range, the dot the median and the dash the mean.

model arm scores 0.797 kcal/mol against its 0.897 while running 14× faster, and three further arms stay within 0.05 of that mark at 30 to 47× speed up.

Given the focus of the models being inference speed, we extend Figure 5 with a number of models from literature. The recently proposed DPA4 family (Li et al., 2026a) sits at the accurate end of the

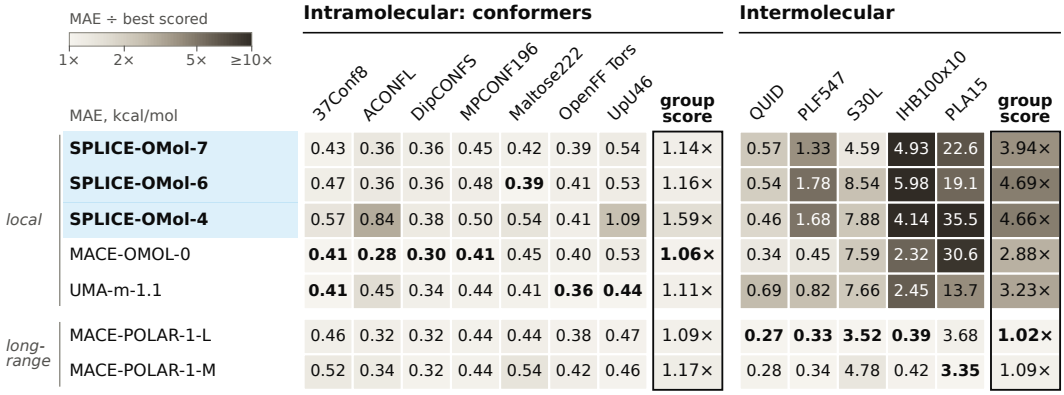

Figure 6: SPLICE-OMol against MACE-POLAR-1 (Batatia et al., 2026) and the local OMol25-trained models on that work's benchmarks. **SPLICE-OMol-7**, **-6** and **-4** are the 512/3072, 256/2048 and three-layer 64/256 arms of Table 6, at 17, 35 and 111 steps/s. Each cell is the MAE (kcal/mol, lower is better); bold marks the best value any model we scored reaches on that benchmark, and the shade is the ratio to it. Those references are the same in Appendix F, so a score here is comparable with one there. The framed group score is the geometric mean of that ratio over the group (1.00× = best on every set). Intermolecular sets are ordered by increasing sensitivity to long-range electrostatics. The periodic sets, the remaining benchmarks and the other four of our arms are in Appendix F.

frontier, and its compressed DPA4C variant (Li et al., 2026b) on the fast end. For a better comparison in this regime, see Appendix J. SPLICE models hold the front between the two. AIMNet2 (Anstine et al., 2025), which is also charge-aware, runs at speed close to **SPLICE-XL**. SO3LR (Kabylda et al., 2025) is an instructive comparison: It expresses explicit long-range electrostatics and yet runs at **SPLICE-L** cost range. While at that throughput, SPLICE models are more accurate, but it should be noted that SO3LR is trained at PBE0+MBD level of theory, unlike the benchmark it is compared to. (see Appendix D). Per-model errors for all benchmark entries are in Appendix G.

Having trained on OMol25, we can also test these models on the suite that was used to evaluate MACE-POLAR-1 (Batatia et al., 2026): a large sets of conformer and interaction energies, scored as the mean absolute error in kcal/mol. We report the most relevant results in Figure 6 (more models, tests, and details in Appendix F). On intramolecular conformers best SPLICE models are in line with the state of the art, with **SPLICE-OMol-4** starting to show deterioration. The intermolecular interaction benchmarks clearly separate MACE-POLAR-1, with explicit long-range electrostatics, from the local models. The smaller models become progressively worse at these tasks, while the generalization to periodic systems, completely out of distribution, shows little correlation to model size (see Appendix F).

6 Conclusion

We have presented SPLICE, a family of MLIPs built around performant operations. The spline radial basis and decoupled widths move the speed–accuracy front: at matched accuracy on SPICE2 the front is 5 to 8× faster than the radial-MLP ablation, and on the external conformer benchmark, they reach the state-of-the-art on neutral molecules while running up to two orders of magnitude faster. These models are not uniformly strong. They carry no explicit long-range electrostatics, and the intermolecular interaction benchmarks show that the missing electrostatics cannot be recovered with increased model size; charged conformers remain their weakest point. Data is as important in model accuracy as the architecture itself: adding charged systems to the training set largely closed an accuracy gap with little change to the network. Separately, the underlying assumptions of the DFT functional ultimately set the lower bound of the error and should be considered in accuracy vs speed trade-offs. Our work shows that the performance-critical operations of a model class can be identified, measured and replaced, without impacting the accuracy significantly.

AI USE STATEMENT

In this work we used generative AI tools, in the form of LLM-based coding assistants, for several tasks. We used them to design and give feedback on measurement and benchmarking methodology, for example in fixing the timing protocol and the real-OOM capacity bisection. We use them to implement methods used for the analysis, evaluation and benchmarking of our models and the training and inference launchers. Agents have also been used extensively to develop the code that support our models, but not to design the architecture itself. We have used assistants to clean and reformat datasets, parse and adapt externally supplied benchmark bundles, to support qualitative and thematic analysis of failure cases, and to interpret results by sifting through data and running basic analyses.

We have not used generative AI tools to generate synthetic data sets, to propose or refine the hypotheses of this work, or to develop its theoretical models or conceptual frameworks: the spline radial basis, the decoupling of the channelwise and trunk widths, and the questions this paper sets out to answer are the authors' own. The remaining required-disclosure tasks, namely formulating mathematical claims, providing critical ingredients for proving them, assisting in the writing of proofs, and translation, are not applicable to this work.

Additionally, we used generative AI tools for creating and modifying figures and tables; creating and editing code; suggesting experimental parameters (mainly in choosing sampling values); drafting parts of this paper and editing it for consistency; searching for and summarizing related literature; and formatting references.

We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI. Every quantitative claim in this paper comes from a recorded run. Speed numbers were measured on the current code base under a single protocol. AI-assisted code that produces a reported number was reviewed by an author, and its output cross-checked against an independent measurement wherever one existed.

REPRODUCIBILITY STATEMENT

We made every effort to include all the information needed to reproduce this work. The architecture is introduced in Section 2 and defined completely in Appendix B.2, which states the forward pass equation by equation with the irreps at every step, together with the radial basis of Appendix B.1; each model is then fixed by its row in the configuration tables of Appendix B, which give the channel and trunk widths, cutoff, correlation order, layer count and the arithmetic precision of each part of the network. The training protocol for each dataset is in Appendix B.4: optimizer, learning-rate schedule, loss weights and batch budgets. The provenance of every training set, the curation applied to it, and its overlap with the evaluation sets are in Appendix E. Every speed, capacity and throughput number in this paper was measured by us under the single protocol given in Appendix C, which fixes the system, the autotune budget, the timing recipe and the bisection to a real out-of-memory failure used for the capacity numbers; published figures for other models are re-measured on our hardware under that same protocol, with the exception of the MLIPAudit baseline scores, which are the suite's own published results and are marked as such. Per-model results are reported in full in Appendices G, H, D and K, and the per-molecule errors behind the conformer benchmark are released alongside the paper, so that any subset of it can be recomputed without re-running a model.

The software stack these models are built on is described at a high level in Appendix A: the package, its operation set, the neighbor list and simulation entry points, and the Loom compiler. Upon acceptance we will release the code used to train and to benchmark these models, together with the SPLICE checkpoints evaluated here and the resolved architecture files that pair with them, so that the reported numbers can be reproduced end to end.

REFERENCES

Amber developers. Amber24: `pmemd.cuda` performance information. https://ambermd.org/GPUPerformance.php, 2024. JAC/DHFR, 23 558 atoms, PME, 4 fs: 1479 ns/day on one H100.

Anonymous. Loom: A GPU compiler for scalar functions over finite index maps. Manuscript; author list withheld for double-blind review, 2026.

Dylan M. Anstine, Roman Zubatyuk, and Olexandr Isayev. AIMNet2: A neural network potential to meet your neutral, charged, organic, and elemental-organic needs. *Chemical Science*, 16(23): 10228–10244, 2025. doi: 10.1039/d4sc08572h.

Ilyes Batatia, Dávid P. Kovács, Gregor N. C. Simm, Christoph Ortner, and Gábor Csányi. MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. *Advances in Neural Information Processing Systems*, 2022.

Ilyes Batatia, William J. Baldwin, Domantas Kuryla, Joseph Hart, Elliott Kasoar, Alin M. Elena, Harry Moore, Mikołaj J. Gawkowski, Benjamin X. Shi, Venkat Kapil, Panagiotis Kourtis, Ioan-Bogdan Magdău, and Gábor Csányi. MACE-POLAR-1: A polarisable electrostatic foundation model for molecular chemistry. *arXiv preprint arXiv:2602.19411*, 2026. URL https://arxiv.org/abs/2602.19411.

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*, 13:2453, 2022. doi: 10.1038/s41467-022-29939-5.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/jax-ml/jax.

Christoph Brunken, Olivier Peltre, Heloise Chomet, Lucien Walewski, Manus McAuliffe, Valentin Heyraud, Solal Attias, Martin Maarand, Yessine Khanfir, Edan Toledo, Fabio Falcioni, Marie Bluntzer, Silvia Acosta-Gutiérrez, and Jules Tilly. Machine learning interatomic potentials: library for efficient training, model development and simulation of molecular systems. *arXiv preprint arXiv:2505.22397*, 2025. InstaDeepAI/SPICE2-curated, HuggingFace rev 8ecd5ea3; mlip v0.1.2 curation.

Alexander G. Donchev, Andrew G. Taube, Elizabeth Decolvenaere, Cory Hargus, Robert T. McGibbon, Ka-Hei Law, Brent A. Gregersen, Je-Luen Li, Kim Palmo, Karthik Siva, Michael Bergdorf, John L. Klepeis, and David E. Shaw. Quantum chemical benchmark databases of gold-standard dimer interaction energies. *Scientific Data*, 8(55), 2021. doi: 10.1038/s41597-021-00833-x.

Peter Eastman, Benjamin P. Pritchard, John D. Chodera, and Thomas E. Markland. Nutmeg and SPICE: Models and data for biomolecular machine learning. *Journal of Chemical Theory and Computation*, 20(19):8583–8593, 2024. doi: 10.1021/acs.jctc.4c00794.

J. Thorben Frank, Oliver T. Unke, Klaus-Robert Müller, and Stefan Chmiela. A euclidean transformer for fast and stable machine learned force fields. *Nature Communications*, 15:6539, 2024. doi: 10.1038/s41467-024-50620-6.

Xiang Fu, Zhenghao Wu, Wujie Wang, Tian Xie, Sinan Keten, Rafael Gomez-Bombarelli, and Tommi Jaakkola. Forces are not enough: Benchmark and critical evaluation for machine learning force fields with molecular simulations. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=A8pqQipwkt.

Ask Hjorth Larsen, Jens Jørgen Mortensen, Jakob Blomqvist, Ivano E. Castelli, Rune Christensen, Marcin Dułak, Jesper Friis, Michael N. Groves, Bjørk Hammer, Cory Hargus, Eric D. Hermes, Paul C. Jennings, Peter Bjerre Jensen, James Kermode, John R. Kitchin, Esben Leonhard Kolsbjerg, Joseph Kubal, Kristen Kaasbjerg, Steen Lysgaard, Jón Bergmann Maronsson, Tristan Maxson, Thomas Olsen, Lars Pastewka, Andrew Peterson, Carsten Rostgaard, Jakob Schiøtz, Ole Schütt, Mikkel Strange, Kristian S. Thygesen, Tejs Vegge, Lasse Vilhelmsen, Michael Walter, Zhenhua Zeng, and Karsten W. Jacobsen. The atomic simulation environment—a Python library for working with atoms. *Journal of Physics: Condensed Matter*, 29(27):273002, 2017. doi: 10.1088/1361-648X/aa680e.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Adil Kabylda, J. Thorben Frank, Sergio Suárez-Dou, Almaz Khabibrakhmanov, Leonardo Medrano Sandonas, Oliver T. Unke, Stefan Chmiela, Klaus-Robert Müller, and Alexandre Tkatchenko. Molecular simulations with a pretrained neural network and universal pairwise force fields. *Journal of the American Chemical Society*, 147(37):33723–33734, 2025. doi: 10.1021/jacs.5c09558.

Dávid Péter Kovács, J. Harry Moore, Nicholas J. Browning, Ilyes Batatia, Joshua T. Horton, Yixuan Pu, Venkat Kapil, William C. Witt, Ioan-Bogdan Magdău, Daniel J. Cole, and Gábor Csányi. MACE-OFF: Short-range transferable machine learning force fields for organic molecules. *Journal of the American Chemical Society*, 147(21):17598–17611, 2025. doi: 10.1021/jacs.4c07099.

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. *arXiv preprint arXiv:2505.08762*, 2025.

Tiancheng Li, Wentao Li, Anyang Peng, Jianming Xue, Linfeng Zhang, Duo Zhang, and Han Wang. DPA4: Pushing the accuracy-cost frontier of interatomic potentials with EMFA SO(2) convolution. *arXiv preprint arXiv:2606.02419*, 2026a.

Tiancheng Li, Jianming Xue, Linfeng Zhang, Duo Zhang, and Han Wang. Universal machine-learning molecular dynamics at the speed of empirical potentials. *arXiv preprint arXiv:2608.19041*, 2026b.

Yi-Lun Liao, Brandon Wood, Abhishek Das, and Tess Smidt. EquiformerV2: Improved equivariant transformer for scaling to higher-degree representations. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=mCOBKZmrzD.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Arslan Mazitov, Filippo Bigi, Matthias Kellner, Paolo Pegolo, Davide Tisi, Guillaume Fraux, Sergey Pozdnyakov, Philip Loche, and Michele Ceriotti. PET-MAD, a lightweight universal interatomic potential for advanced materials modeling. *arXiv preprint arXiv:2503.14118*, 2025.

Saro Passaro and C. Lawrence Zitnick. Reducing SO(3) convolutions to SO(2) for efficient equivariant GNNs. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 27420–27438. PMLR, 2023. URL https://proceedings.mlr.press/v202/passaro23a.html.

Eric Qu and Aditi S. Krishnapriyan. The importance of being scalable: Improving the speed and accuracy of neural network interatomic potentials across chemical domains. In *Advances in Neural Information Processing Systems*, 2024.

Benjamin Rhodes, Sander Vandenhaute, Vaidotas Šimkus, James Gin, Jonathan Godwin, Tim Duignan, and Mark Neumann. Orb-v3: Atomistic simulation at scale. *arXiv preprint arXiv:2504.06231*, 2025.

Samuel S. Schoenholz and Ekin D. Cubuk. JAX, M.D. a framework for differentiable physics. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124016, 2021. doi: 10.1088/1742-5468/ac3ae9.

Kristof T. Schütt, Pieter-Jan Kindermans, Huziel E. Sauceda, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/303ed4c69846ab36c2904d3ba8573050-Abstract.html.

Fei Shuang, Penghua Ying, Kai Liu, Zixiong Wei, Fengxian Liu, Zheyong Fan, Minqiang Jiang, and Poulumi Dey. Benchmarking chemically scalable machine-learning interatomic potentials for large-scale simulations of multicomponent alloys. *Physical Review Materials*, 10(7):073802, 2026. doi: 10.1103/1qc9-ypyb.

Leon Wehrhan, Lucien Walewski, Marie Bluntzer, Heloise Chomet, Jules Tilly, Christoph Brunken, and Silvia Acosta-Gutiérrez. MLIPAudit: A benchmarking tool for machine learned interatomic potentials. *arXiv preprint arXiv:2511.20487*, 2025.

Brandon M. Wood, Misko Dzamba, Xiang Fu, Meng Gao, Muhammed Shuaibi, Luis Barroso-Luque, Kareem Abdelmaqsoud, Vahe Gharakhanyan, John R. Kitchin, Daniel S. Levine, Kyle Michel, Anuroop Sriram, Taco Cohen, Abhishek Das, Ammar Rizvi, Sushree Jagriti Sahoo, Zachary W. Ulissi, and C. Lawrence Zitnick. UMA: A family of universal models for atoms. *Advances in Neural Information Processing Systems*, 38:143528–143564, 2025.

A IMPLEMENTATION

A.1 THE SPLICE PACKAGE

The model, its training loop, an inference calculator, a molecular-dynamics driver and the compiler (Loom) ship together as a single Python package built on JAX (Bradbury et al., 2018). In the SPLICE code, a model is declared as a resolved sequence of typed operations, each carrying its own input and output irreducible representations, its parameter dtype and its arithmetic precision.

The operation set is small and deliberately close to the mathematical description: species and charge embeddings, equivariant linear maps in both a shared and a species-indexed form, spherical harmonics, the radial basis, the channelwise tensor product (MP operation), the symmetric contraction, scalar activations, the readout and energy head, and per-species energy offsets. The two contributions of this work appear as properties of these operations: the radial basis is evaluated inside the MP operation, and the channel and trunk widths are simply different multiplicities on existing operations.

Alongside the model, the package provides a GPU-resident cell neighbor list, an ASE (Hjorth Larsen et al., 2017) calculator, a lower-level force-field entry point for callers that manage their own neighbor list, and a MD driver used for the condensed-phase results of Section 4. Kernels are compiled ahead of time and autotuned under an explicit search budget, with a persistent compile cache, so that the compilation cost is paid once rather than per run; the budget used for each measurement is recorded in Appendix C.

A.2 LOOM

Loom (Anonymous, 2026) is the GPU compiler used to explore and implement the architectural choices studied here. Developed using agentic AI under the authors' direction and review, it accepts fixed-shape JAX functions built from supported scalar operations, together with finite index maps specifying data placement, and generates NVIDIA GPU kernels. Its programming model separates a local formula from indexed reads and additive reductions, with a contiguous axis of independent lanes providing regular parallelism. Neither this nor the compilation strategy are technically restricted to atoms, equivariance, or a particular neural-network architecture, although MLIPs motivated its development and provide the evaluation workloads in the companion paper. In SPLICE, channels supply the independent lanes, and the radial basis, spherical harmonics, and channelwise tensor product are expressed together so that Loom can fuse their evaluation with neighbor gathers and aggregation without materializing the per-edge radial tensor. Programs produced by JAX automatic differentiation pass through the same compiler. Loom combines algebraic simplification with register-aware kernel partitioning and autotuning over execution layouts and traversal strategies; a search budget limits optimization effort to reduce compilation times. Both SPLICE and the MACE reimplementation used in our architectural comparisons use Loom-generated fused kernels, making the compiler a shared backend that does not advantage our architecture. Loom enabled these experiments, but its compiler methods are not claimed as contributions of SPLICE.

B THE SPLICE ARCHITECTURE AND ITS CONFIGURATIONS

This appendix defines the architecture: first the radial basis, then the forward pass, then the configurations used in this paper, and last the training protocol.

B.1 THE FREE-KNOT RADIAL BASIS

A **free-knot radial spline** is a piecewise cubic whose support endpoint is learned. Each channel k carries two unconstrained trainable parameters $a_k, s_k \in \mathbb{R}$. They act through two intermediate quantities, the knot fraction e_k and the shape angle t_k, which are the coordinates used in the main text and in Figure 1(c). The reparameterization enforces every constraint on the basis without bounds on the trainable parameters.

Knot. The knot is $\ell_k = e_k r_{\max}$, where $r_{\max}$ is the global cutoff and

$$e_k = e_{\min} + (1 - e_{\min})\,\sigma(a_k), \qquad \sigma(a) = \tfrac{1}{2}\Big(1 + \tfrac{a}{\sqrt{1+a^2}}\Big), \qquad e_{\min} = 0.1.$$

The floor keeps every support at least $e_{\min} r_{\max}$ long, so no channel can shrink to a function that never sees an edge; the full cutoff $e_k = 1$ is approached only as $a_k \to \infty$. The squashing function σ is free of exponentials and its tails decay polynomially, so a_k keeps a useful gradient far from the origin, where a logistic sigmoid would saturate.

Shape. On $x = r/\ell_k \in [0, 1]$ let

$$B(x) = \sqrt{105}\,x(1-x)^2, \qquad D(x) = \sqrt{3}\,(2 - 7x)(1-x)^2.$$

Both vanish with their first derivative at $x = 1$, they are orthonormal on $[0, 1]$, and together they span every cubic with that property. The rotated pair $P = (B + D)/\sqrt{2}$, $Q = (B - D)/\sqrt{2}$ is orthonormal as well. For $r \geq 0$, define

$$\varphi_k(r) = \begin{cases} \sqrt{r_{\max}/\ell_k}\,\big(\cos t_k\, P(x) + \sin t_k\, Q(x)\big) & \text{if } 0 \leq r < \ell_k, \\ 0 & \text{if } r \geq \ell_k, \end{cases} \qquad t_k = \tfrac{\pi}{4} - \arctan s_k.$$

Equivalently $\varphi_k = \sqrt{r_{\max}/\ell_k}\,(B + s_k D)/\sqrt{1 + s_k^2}$ on the support, the form the kernel evaluates. Thus φ_k has degree three on its support $[0, \ell_k]$ and $\varphi_k(\ell_k) = \varphi_k'(\ell_k) = 0$: it is C^1. Because P and Q are orthonormal,

$$\frac{1}{r_{\max}} \int_0^{r_{\max}} \varphi_k(r)^2\,\mathrm{d}r = 1$$

independently of both the learned knot ℓ_k and the shape t_k: every channel has unit mean square for r uniform on $[0, r_{\max}]$. The angle t_k covers the open half circle $(-\pi/4, 3\pi/4)$, so each shape appears once up to sign; $t_k = \pi/4$ is the pure bump B, and $\pm D$ are reached only as $s_k \to \pm\infty$. The sign, and the overall scale, are supplied by an amplitude g_{pk} that multiplies φ_k separately for every tensor-product path p; it is trained and initialized from $\mathcal{N}(0, 1)$.

Initialization. Initialization is deterministic. The $C/2$ values $\sigma(a_k)$ are spaced uniformly over $[0.3, 0.95]$, which places the knots at $e_k \in [0.37, 0.955]$. Each knot carries two channels at $s_k = +1$ and $s_k = -1$, that is $t_k = 0$ and $t_k = \pi/2$, the orthogonal pair P and Q; in general two channels at one knot are orthogonal exactly when $s_k s_{k'} = -1$. Pinned channels (Appendix B.2) have $e_k = 1$ fixed, while their shape and amplitude are still trained.

Parameter	Domain	Description
a_k	$\mathbb{R}$	Trainable knot parameter; sets e_k through σ.
s_k	$\mathbb{R}$	Trainable shape parameter; sets $t_k = \pi/4 - \arctan s_k$.
g_{pk}	$\mathbb{R}$	Trainable amplitude, one per path and channel.
e_k	$(e_{\min}, 1)$	Knot as a fraction of the cutoff; the knot is $\ell_k = e_k r_{\max}$.
t_k	$(-\pi/4, 3\pi/4)$	Shape angle; mixes P and Q at constant norm.
$e_{\min}$	0.1	Fixed floor on the knot fraction.
$r_{\max}$	$(0, \infty)$	Fixed global cutoff; upper bound on the knot.

Knot evolution during training. Because initialization is deterministic, with two orthogonal shapes at each of a set of evenly spaced knots, the starting configuration is a regular lattice in the (knot, shape) plane and any structure in the trained parameters is learned. Figure 7 shows both, for SPLICE-XL, the widest SPICE2 arm at $C = 256$ channels per layer. Training moves the knots inward. The median support falls from 3.97 Å to 3.18 Å in the first layer and 3.16 Å in the second; a population of channels appears inside 2 Å, where none was initialized; and the channels still reaching past 5 Å thin from 54 to 35 and 23, respectively. This is not particular to one model. Across all 26 SPICE2 arms of Table 4 the median support lands between 2.68 and 3.22 Å, about a tenth of channels end up inside 2 Å, and the second layer retains only 35% of its long-reach channels on average against 52% for the first. We speculate that the larger amount of chemical information available at short range drives the "short-sighted" tendency of the radial basis.

This asymmetry is the argument for the pinned channels of Table 6. Nothing in the loss explicitly rewards a channel for keeping its support wide, but especially for charged systems we expect a large receptive field to be important for an accurate model. Pinning holds a fixed number of channels per layer at the full cutoff, restraining only their knot while leaving shape and amplitude to be trained. This forces the model to retain awareness of the full receptive field for the whole training.

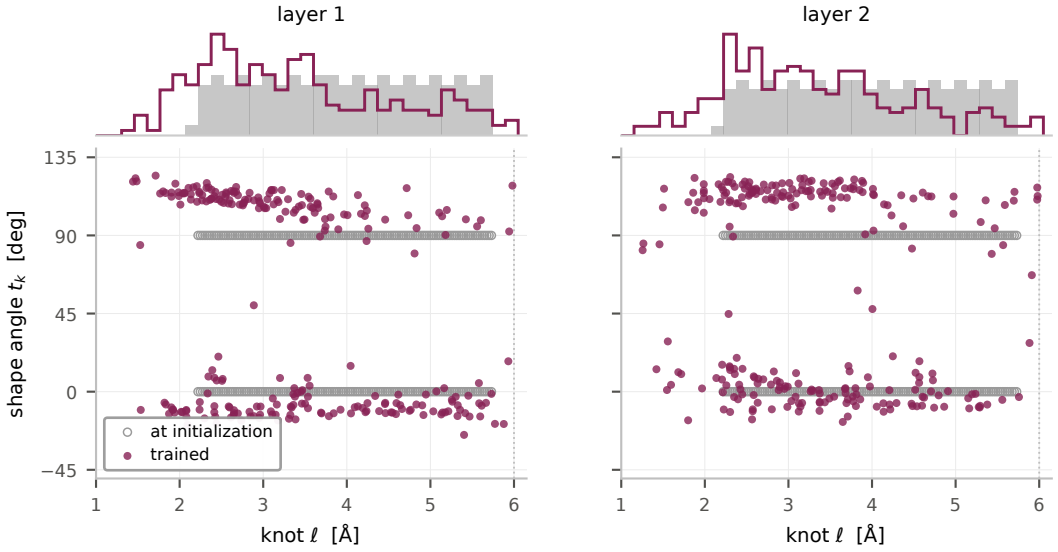

Figure 7: Free-knot radial parameters of SPLICE-XL before and after training, per layer. Each point is one of the 256 channels; gray is the deterministic initialization, color the trained model. The horizontal axis is the knot, the support endpoint of that channel; the vertical axis is the shape angle t_k, which sets how the two cubics are mixed on that support. Both axes are the derived coordinates of Appendix B.1, as in Figure 1(c), not the unconstrained parameters the checkpoint stores. Histograms above give the knot distribution. Initialization places two channels at each of 128 knots with the orthogonal shapes $t_k = 0$ and 90°, which is the pair of gray lines; training broadens the shape and shifts the knots inward, and the shift is larger in the second layer.

B.2 The forward pass

SPLICE is an equivariant message-passing network of the MACE family (Batatia et al., 2022). This subsection states the forward pass as the sequence of operations actually executed, with the irreducible representations (irreps) at every step, so that each model in this paper is completely determined by what follows together with its row in Tables 4–6.

Notation. A structure has N atoms with species $Z_i \in \{1, \dots, N_Z\}$ and positions $\mathbf{r}_i$. Every ordered pair with $r_{ij} = \|\mathbf{r}_{ij}\| < r_c$, where $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, is an edge, and $\mathcal{N}(i)$ is the neighbor set of i. Node features are real $O(3)$ tensors $h_{i,\ell m,k}$ with degree ℓ, order $m = -\ell, \dots, \ell$ and channel

$k = 1, \ldots, W$; we write $W \times (\ell p)$ for W copies of the irrep of degree ℓ and parity p, abbreviate parity as e or o, and stack irreps with $+$, so that $512{\times}0e + 512{\times}1o$ is a width-512 feature carrying scalars and vectors. Every degree appears with its natural parity $p = (-1)^\ell$. The Y^ℓ_m are real spherical harmonics and $C^{\ell_3 m_3}_{\ell_1 m_1 \ell_2 m_2}$ Clebsch–Gordan coefficients. An *equivariant linear* W acts within each degree and mixes only channels, $(Wh)_{\ell m,k'} = \sum_k W^{(\ell)}_{k'k} h_{\ell m,k}$, and carries no bias; a degree present in the output but absent from the input receives zero. A *species-indexed* linear is the same map with one weight tensor per element, selected by Z_i.

Three widths recur, and each is a column of the architecture tables. T is the **trunk** width, carried by the node features from layer to layer; C is the **channel** width at which the per-edge tensor product runs; R is the **readout** width of the last hidden layer (the `_headR` name suffix marks a value other than the default 16). The trunk degree ℓ_T bounds the node features and $\ell_{\max} = 3$ the spherical harmonics, in every model here. The network has L layers: in this work $L = 2$ except where the name carries `_3L` (larger OMol25 models).

Embedding. The initial node features are scalars only, one learned row per element,

$$h^{(0)}_{i,00,k} \;=\; N_Z^{-1/2}\, W^{\mathrm{emb}}_{Z_i k}, \qquad k = 1, \ldots, T. \tag{1}$$

The charge-aware models add one term to (1); it is given at the end of this subsection.

One layer. Layer $t = 0, \ldots, L-1$ maps $h^{(t)}$ to $h^{(t+1)}$ and emits a site-energy contribution. It performs the following seven operations.

(1) Skip. A linear map of the incoming features is set aside for the end of the layer,

$$s^{(t)}_i \;=\; W^{\mathrm{skip},(t)}_{Z_i}\, h^{(t)}_i, \tag{2}$$

which is the residual branch of the layer: it is the only path by which features reach $h^{(t+1)}$ without passing through the neighborhood. It is also where the trunk irreps change, since it maps the irreps entering layer t onto those leaving it. Two variants are used, and they are the *skip* column of Tables 5 and 6: **indexed**, as written in (2), holds one weight tensor per element and costs N_Z times the parameters of a single map; **shared** drops the subscript, $W^{\mathrm{skip},(t)}$, and is used where N_Z is large enough for the indexed form to dominate the parameter count (for the 83-element OMol25 models).

Being equivariant, (2) is block-diagonal in ℓ: a weight block exists only for a degree carried by *both* its input and its output, and no parameters are allocated for the rest. In a two-layer model the first skip reads the scalar-only embedding, so the $\ell > 0$ outputs it is declared with are identically zero; the second skip writes the scalar-only features the energy reads, so its $\ell > 0$ inputs are dropped. So in every two-layer model the skip is a pure scalar map. The middle layer of a three-layer arm carries $\ell > 0$ on both sides, and there the skip does propagate vectors and rank-2 tensors.

(2) Down-projection, $T \to C$. An equivariant linear takes the node features from the trunk width to the channel width at every degree $\ell \le \ell_T$, without mixing degrees,

$$u^{(t)}_i \;=\; W^{\downarrow(t)}\, h^{(t)}_i. \tag{3}$$

(3) Message. For each edge the neighbor's features are combined with the spherical harmonics of the edge direction and with the radial functions, and the result is summed over neighbors:

$$A^{(t),p}_{i,\ell_o m_o,k} \;=\; \epsilon \sum_{j \in \mathcal{N}(i)} g^{(t)}_{pk}\, \varphi^{(t)}_k(r_{ij}) \sum_{m_{\mathrm{in}},\, m_s} C^{\ell_o m_o}_{\ell_{\mathrm{in}} m_{\mathrm{in}}\, \ell_s m_s}\, Y^{\ell_s}_{m_s}(\hat{\mathbf{r}}_{ij})\; u^{(t)}_{j,\ell_{\mathrm{in}} m_{\mathrm{in}},k}. \tag{4}$$

The path index $p = (\ell_{\mathrm{in}}, \ell_s, \ell_o)$ runs over every triple admissible under the Clebsch–Gordan selection rules with $\ell_{\mathrm{in}} \le \ell_T$, $\ell_s \le \ell_{\max}$ and $\ell_o \le \ell_{\max}$. The channel index k is carried through untouched: (4) mixes degrees but never channels, which is what makes its cost proportional to C rather than to C^2, and is the sense in which the tensor product is *channelwise*. The radial functions $\varphi^{(t)}_k$ are the free-knot splines of Appendix B.1, one per channel and shared across paths, and $g^{(t)}_{pk}$ is a trainable per-path amplitude; their product is the per-(edge, path, channel) weight that the MACE-family radial MLP would otherwise emit into memory. Paths are *concatenated* rather than

summed, so the width of $A^{(t)}$ at degree ℓ_o is C times the number of paths reaching ℓ_o. Steps (3) and the evaluation of φ and Y are one fused kernel; the intermediate quantities in (4) never reach device memory.

(4) Path mixing and lift, $C \to T$. A single equivariant linear both mixes the paths of each output degree and returns the width to the trunk,

$$m_i^{(t)} \;=\; W^{\uparrow(t)}\, A_i^{(t)}. \tag{5}$$

Its input carries width $n_\ell C$ at degree ℓ, with n_ℓ the number of paths of (4) reaching ℓ, and its output carries width T at every $\ell \le \ell_{\max}$. Together, the down-projection and the lift confine C to the per-edge region and T to the node-local one; setting $C = T$ recovers the usual single-width formulation.

(5) Symmetric contraction. Products of up to ν copies of the message are formed on each node and contracted back to a node feature,

$$\tilde{h}^{(t)}_{i,\ell m,k} \;=\; \sum_{\nu=1}^{\nu_{\max}} \sum_{\eta \in \mathcal{P}_{\nu,\ell}} w^{(t)}_{Z_i,\eta,k}\; B^{\nu}_{\eta,\ell m}\Big(m^{(t)}_{i,\cdot,k}\Big), \tag{6}$$

where $B^{\nu}_{\eta,\ell m}$ is the η-th element of the basis of permutation-invariant, equivariant polynomials of degree exactly ν in the components of $m^{(t)}_{i,\cdot,k}$ that transform as (ℓm), and $\mathcal{P}_{\nu,\ell}$ enumerates that basis. This is the one non-linearity of the trunk, and the one operation that raises the body order: with $\nu_{\max} = 3$, the *correlation order*, a layer is already four-body. Two properties of (6) are used throughout the paper: it is *channelwise* (i.e. k is not mixed, as in (4)) and its weights are always species-indexed. $w^{(t)}$ has shape $N_Z \times |\mathcal{P}| \times T$ and is the largest single tensor in the OMol25 models. The output is truncated to $\ell \le \ell_T$, which is where the trunk degree is imposed: an $\ell_T{=}0$ arm still *builds* $\ell{=}3$ messages in (4) and keeps only the scalars of (6). The last layer truncates to $\ell = 0$ regardless, since only scalars reach the energy. Correlation order is 3 in every model except where Table 6 marks otherwise.

(6) Linear and residual. A final equivariant linear — $T \to T$, on whatever irreps the contraction emitted — then the branch set aside in (2):

$$h_i^{(t+1)} \;=\; W^{(t)} \tilde{h}_i^{(t)} \;+\; s_i^{(t)}. \tag{7}$$

(7) Readout. Every layer contributes a site energy, not only the last. An intermediate layer uses a linear readout; the final layer narrows to the readout width R, applies SiLU and a second linear:

$$E_i^{(t)} = W^{\mathrm{out},(t)} h_i^{(t+1)} \;\; (t < L-1), \qquad E_i^{(L-1)} = W_2\, \mathrm{silu}\Big(W_1 h_i^{(L)}\Big). \tag{8}$$

Both act on the $0e$ block alone. R is the *readout* column of Table 6 and is 16 in every SPICE2 and SPICE2+Charge model.

Energy, forces and stress. The site energies of all layers are summed with a per-element reference energy,

$$E = \sum_{i=1}^{N} \Big(E_0(Z_i) + \sum_{t=0}^{L-1} E_i^{(t)} \Big), \qquad \mathbf{F}_i = -\frac{\partial E}{\partial \mathbf{r}_i}, \qquad \sigma = \frac{1}{V}\frac{\partial E}{\partial \boldsymbol{\varepsilon}}. \tag{9}$$

E_0 is a learnable per-element table, fitted by least squares on the training corpus and held at zero during training, because the same table is subtracted from every DFT target by the loader; the loss therefore runs on interaction energies, which are four to five orders of magnitude smaller than the totals. It is restored at inference so that the model returns absolute energies. Forces and the stress are derivatives of (9) taken by automatic differentiation, so the model is conservative by construction and no quantity is produced by a separate head.

A worked example. Table 3 runs **SPLICE-L** ($\ell_T{=}2$, $C{=}128$, $T{=}512$, $L{=}2$, $N_Z{=}12$, $r_c{=}6$ Å) through the operations above. The jump from 128 to $384/640/640/512$ between (4) and (5) is the path concatenation, and it is the reason the lift is the widest linear in the model.

Table 3: The forward pass of SPLICE-L, step by step. Widths are multiplicities; "→" separates the irreps in from the irreps out. The per-edge region is (4) alone; everything else is per node. †The skips are written with the irreps they are declared with, but only a degree present on both sides carries a weight block.

	operation	layer 0	layer 1
(1)	embedding	$\to 512{\times}0e$	—
(2)	skip (indexed, $N_Z{=}12$)†	$512(0e) \to 512(0e{+}1o{+}2e)$	$512(0e{+}1o{+}2e) \to 512(0e)$
(3)	down-projection	$512(0e) \to 128(0e)$	$512(0e{+}1o{+}2e) \to 128(0e{+}1o{+}2e)$
(4)	message, $\ell_{\max}{=}3$	$\to 128(0e{+}1o{+}2e{+}3o)$	$\to 384{\times}0e{+}640{\times}1o{+}640{\times}2e{+}512{\times}3o$
(5)	mix and lift	$\to 512(0e{+}1o{+}2e{+}3o)$	$\to 512(0e{+}1o{+}2e{+}3o)$
(6)	symmetric contraction, $\nu_{\max}{=}3$	$\to 512(0e{+}1o{+}2e)$	$\to 512(0e)$
(7)	linear $T \to T$, then residual	$\to 512(0e{+}1o{+}2e)$	$\to 512(0e)$
(8)	readout	$\to 1{\times}0e$	$512(0e) \to 16(0e) \to 1{\times}0e$

Other constants. The per-edge normalization ϵ of (4) is the reciprocal of the average neighbor count of the training corpus ($\epsilon = 0.0441$, i.e. $\langle n \rangle \approx 22.7$, for the SPICE2 models at $r_c = 6$ Å); it is a property of the dataset, read from its card and never trained. The *correlation order* is the $\nu_{\max}$ of (6). *Pinned* channels are those whose radial extent is held at the full cutoff ($e_k = 1$) instead of being learned, while their shape and amplitude are still trained, so that a fixed number of channels per layer always reaches the whole neighborhood; the count is the *pinned* column.

Conditioning on charge and spin. The charge-aware models take the molecular charge Q and the spin multiplicity S of the structure containing atom i, scale each by a factor 10, and add a two-layer perceptron of the pair to the scalar block of (1):

$$h^{(0)}_{i,00,k} \;\mathrel{+}=\; \Big[W_2\,\mathrm{silu}\big(W_1 q_i + b_1\big)\Big]_k, \qquad q_i = \big(Q/10,\; S/10\big), \tag{10}$$

with a hidden width of 64. The factor is not fitted: it is chosen so that both inputs land inside $[-1, 1]$ over the range the training sets actually span (Q from -10 to $+10$, S from 1 to 11), which puts them on the scale the perceptron's normally-initialized weights expect. Because (10) adds this to the existing scalars, no downstream width changes. A structure carrying neither falls back to the neutral singlet, $Q = 0$ and $S = 1$.

Numerical precision. The equivariant linears ((2), (3), (5), (7) and (8)) run their matrix products in TF32. The message (4), the radial basis it evaluates, and the symmetric contraction (6) are held at FP32. The two operations kept at FP32 are those whose intermediates are high-degree polynomials in the inputs, where the reduced mantissa is not safe, and both are fused kernels of ours in which the matrix-multiply units are not the limiting resource.

We measured the cost of reduced precision linears by comparing the outputs of the same network with both precisions on the same structures in one process (40 SPICE test molecules per model). Forces move by 1.18, 0.85 and 0.56 meV/Å RMS for SPLICE-S, -M and -L (which is 1.2, 0.8 and 0.6×10^{-3} of the largest force in each structure), and energies by under 0.25 meV/atom. Added in quadrature to each model's own force error, that is $+0.13\%$, $+0.19\%$ and $+0.29\%$ on the reported MAE. The effect is small but not invisible, and it shrinks with width. The stability of MD runs is unaffected by this numerical precision change, so we believe that the use of TF32 for the linear operations is fully justified.

B.3 The configurations

Tables 4, 5 and 6 list every architecture referred to anywhere in this paper, one table per training set. The short name in the first column is the identifier used in all other tables and figures, so a model can be traced between the accuracy, speed and capacity results without reference to a run tag.

The four models carried through the main text keep the names **SPLICE-S**, **-M**, **-L** and **-XL**. Each has a twin trained on SPICE2+Charge at the same ℓ_T, C, T, layer count and cutoff, named **SPLICE-C-S** to **-C-XL**; the two differ only in the dataset, and in the species count and charge embedding it requires. The OMol25 arms are **SPLICE-OMol-1** to **-7**, numbered by throughput with 1 the fastest; that ladder is not the S/M/L/XL one and does not borrow its letters, and the widths behind each

number are in Table 6. Every remaining name is an ablation arm, built as `L`ℓ_T`_`C`_`T, with a suffix only where one is needed to separate otherwise identical rows: `_MLP` for the radial-MLP ablation arms, `_head`N for a readout width other than the default, `_3L` for a three-layer model and `_Rc5` for the 5 Å cutoff. In the C and T columns a "+" separates degrees and a "/" separates layers, so a single number means the width is uniform across both.

For all models, the spherical harmonics run to $\ell_{\max}$=3 in *all* of them, including those whose trunk carries scalars only, and the radial basis is the free-knot spline unless the name carries `_MLP`. The precision split is the one stated at the end of Appendix B.2.

Each table drops the fields that are constant within its family. Every SPICE2 model has two GNN layers, species-indexed skip connections, a readout width of 16 and no pinned channels, so those columns appear only for the other two data sets. The charge-aware SPICE2+Charge and OMol25 models additionally carry a charge and spin embedding, which the SPICE2 models, being charge-blind, do not. The species dimension follows the corpus: 12 elements for SPICE2, 17 for SPICE2+Charge and 83 for OMol25.

Table 4: SPICE2 architectures. ℓ_T: maximum degree of the trunk node features. C: channel width entering the channelwise tensor product. T: trunk width. r_c: cutoff in Å. All of these models have two layers, species-indexed skips, a readout width of 16, correlation order 3 and no pinned channels.

name	ℓ_T	C	T	r_c	params
SPLICE-S	0	32	128	6	511k
L0_64_64	0	64	64	6	170k
L0_64_64_MLP	0	64	64	6	219k
L0_64_64_Rc5	0	64	64	5	170k
L0_64_128	0	64	128	6	552k
L0_64_256	0	64	256	6	1.96M
L0_64_256_Rc5	0	64	256	5	1.96M
L0_64_512	0	64	512	6	7.32M
L0_128_128	0	128	128	6	635k
L0_128_128_MLP	0	128	128	6	716k
L0_128_256	0	128	256	6	2.12M
L0_128_512	0	128	512	6	7.65M
L0_256_256	0	256	256	6	2.45M
L0_256_256_MLP	0	256	256	6	2.60M
L0_256_512	0	256	512	6	8.30M
L1_64_128	1	64	128	6	651k
L1_64_256+64	1	64	256+64	6	2.06M
SPLICE-M	1	64	256	6	2.19M
L1_64_512+128	1	64	512+128	6	7.53M
L1_128_256	1	128	256	6	2.47M
L2_64_64	2	64	64	6	268k
L2_64_64_MLP	2	64	64	6	370k
L2_64_128	2	64	128	6	764k
L2_64/64+32+16_256	2	64/64+32+16	256	6	2.29M
L2_64_256	2	64	256	6	2.44M
L2_64_512	2	64	512	6	8.56M
L2_64_512+128+96	2	64	512+128+96	6	7.80M
L2_64_512_Rc5	2	64	512	5	8.56M
L2_128_128_MLP	2	128	128	6	1.16M
L2_128_256	2	128	256	6	2.86M
L2_128/128+64+32_512	2	128/128+64+32	512	6	8.75M
SPLICE-L	2	128	512	6	9.38M
L2_128_1024	2	128	1024	6	33.4M
L2_128_1024_Rc5	2	128	1024	5	33.4M
L2_256_256	2	256	256	6	3.68M
L2_256_256_MLP	2	256	256	6	4.03M
L2_256_512	2	256	512	6	11.0M
L2_256_1024	2	256	1024	6	36.7M
SPLICE-XL	2	256	2048	6	132.1M

Table 5: SPICE2+Charge architectures: the charged twin of each main-text model. Columns as in Table 4; all have two layers, a 6 Å cutoff, correlation order 3, species-indexed skip connections, a readout width of 16 and no pinned channels. Each carries about a third more parameters than the SPICE2 model of the same widths, and the difference is the corpus rather than the design: the charged set adds five elements (Li, Na, Mg, K, Ca), which widens every species-indexed tensor, and one charge-embedding layer is added on top. Nothing else differs.

name	ℓ_T	C	T	params
SPLICE-C-S	0	32	128	701k
SPLICE-C-M	1	64	256	2.91M
SPLICE-C-L	2	128	512	12.2M
SPLICE-C-XL	2	256	2048	174.9M

Table 6: OMol25 architectures. Columns as in Table 4; all have a 6 Å cutoff and shared skip connections. †correlation order 2; all other models use 3.

name	ℓ_T	layers	C	T	readout	pinned	params
SPLICE-OMol-1	0	2	64	128	128	16	460k
SPLICE-OMol-2	0	2	128	512	512	32	3.15M
SPLICE-OMol-3	2	2	64	256	256	16	2.23M
SPLICE-OMol-4	2	3	64	256	256	16	3.99M
SPLICE-OMol-5	2	3	128	1536	16	32	51.6M
SPLICE-OMol-6†	2	3	256	2048	16	32	80.2M
SPLICE-OMol-7	2	3	512	3072	256	32	213.6M

B.4 The training protocol

Every model in this paper was trained with the same optimizer and the same loss. The optimizer is Muon (Jordan et al., 2024) on the weight matrices, with the remaining parameters on AdamW (Loshchilov & Hutter, 2019); the loss is a Huber loss on energies and forces with $\delta = 0.01$ on forces; an exponential moving average of the weights with decay 0.999 is kept throughout and every reported number comes from the EMA weights. The learning-rate schedule is a short warm-up, a long hold at the base rate, and a cosine quench to 3% of it, with evaluation continuing past the end of the quench. Batches are formed to a fixed per-GPU budget in atoms, edges and graphs, rather than to a fixed number of structures, so a batch holds as many molecules as fit; the atom budget is the figure quoted below and the edge and graph budgets scale with it.

Within a dataset the recipe is held fixed and only the architecture moves, so the speed-accuracy fronts in this paper vary width and angular momentum at a fixed training protocol. Table 7 gives the three protocols. The differences between them follow the data: OMol25 is far larger than SPICE2, so 22 epochs there are more optimizer steps than 765 on SPICE2, and its force weight is lower because its force distribution is much broader. Two entries in the OMol column are set per model: the atom budget and the epoch count of the large arms were chosen per run and do not follow model size monotonically.

Table 7: Training protocol per corpus. The OMol25 arms of Table 6 split cleanly by trunk width: *small* is **SPLICE-OMol-1** to **-4** ($T \leq 512$), *large* is **SPLICE-OMol-5**, **-6** and **-7** ($T \geq 1536$). Learning rates are the base rate, before the Muon group scaling in the row below. †set per model rather than per corpus: the atom budget is 2048, 4096 and 1024 and the epoch count 22, 15 and 8 for **SPLICE-OMol-5**, **-6** and **-7** respectively. **SPLICE-OMol-7** held the base rate throughout instead of quenching.

	SPICE2	SPICE2	OMol25	
		+Charge	small	large
base learning rate	0.01	0.01	0.005	0.01
Muon group LR scale	3	3	1	3
warm-up fraction	0.02	0.002	0.002	0.002
quench length (steps)	900k	985k	400k	400k
quench final fraction	0.03	0.03	0.03	0.03
epochs	765	765	22	8–22†
atom budget per GPU	4096	4096	8192	1024–4096†
energy weight	10	10	10	10
force weight	10	10	1	1
Huber δ, forces	0.01	0.01	0.01	0.01
EMA decay	0.999	0.999	0.999	0.999
cutoff	6 Å	6 Å	6 Å	6 Å

C MEASUREMENT PROTOCOL

Every speed, capacity and accuracy number we report for our own models was measured by us, and every published model we compare against was re-measured by us on the same hardware under the protocol of the corresponding subsection. We do not quote throughput figures from other papers except in Table 1, which is explicitly qualitative.

Hardware and precision. All measurements are on a single NVIDIA H100 80 GB GPU unless stated otherwise. Precision is set per operation, not globally: TF32 is enabled deliberately and only on the equivariant linear layers, while message passing, the radial basis it evaluates and the symmetric contraction are held at FP32 by design (Appendix B.2).

Inference throughput (μs per atom per step). This is the speed axis of Figure 2 and the μs/atom column of Table 14. The figure quoted for the model family is the cost of one energy-and-force evaluation, reported per atom and per step, on a liquid water box of $48\,384$ atoms. The box is large enough that the models are not dominated by kernel-launch overhead, which is the regime a production simulation runs in and is *not* the regime of the 1000-atom systems most papers quote. The autotuner runs without a search-time cap, so each model is timed at the best kernel configuration the compiler can find for it.

MD throughput (steps per second). This is the speed axis of Figure 5 and the throughput column of Table 11. For the external conformer benchmark, throughput is reported as complete molecular-dynamics steps per second on the $2\,661$-atom water box that benchmark uses, as the median of three replicates. Every model, ours included, is driven through the benchmark's own ASE calculator interface, so the integrator and the harness are common to all of them, and published models are re-measured on the same GPU through their own production inference path. What we supply is a GPU-resident cell neighbor list. That choice matters more than the model does: with the benchmark's default host-side neighbor list, construction is over four fifths of a step and the potential itself under a tenth, so a host-side number measures the harness rather than the model. The published throughputs we compare against were themselves obtained with a GPU neighbor list. This system is small enough that launch overhead is a significant fraction of the step for our faster models, so these numbers are not comparable to the previous paragraph's and are not a like-for-like statement of model cost; they are reported because they are the quantity in which the external benchmark is defined.

What is inside the timed region. The two throughput numbers place the neighbor list differently, which is the main reason they are not comparable. For the μs/atom figure the box is fixed, so the neighbor list is built once outside the timed region and the measurement is the energy-and-force evaluation alone: six warm-up calls, then thirty timed calls. For the MD steps/s figure the timed region is the complete integration step, neighbor-list rebuild included, which is the quantity the external benchmark defines.

Capacity. This is the *max atoms* column of Table 14. Capacity is reported as the largest number of atoms that completes a forward and backward pass, found by bisection until an adjacent pair of sizes is separated by less than about 1%, where the larger of the pair raises a genuine out-of-memory error. We report this rather than a measured memory footprint as a more reliable indicator. Two minor details: all capacity figures are taken with a warm cache, and capacity was measured with the autotuner capped at a 250 ms search budget (whereas the throughput numbers are uncapped).

Deformation energy and interaction force. Both are defined in Appendix D: the deformation error per molecule is referred to that molecule's lowest-DFT-energy conformer, and the interaction force is the rigid-body part of the force on each fragment. Figure 4 reports them as a percentage of the DFT reference of each subset; the absolute values are in Tables 8 and 9.

Validation error. Force errors on the SPICE2 validation set are mean absolute errors weighted by atom count across the eight validation subsets, so that the reported figure is the error on a randomly drawn atom rather than the unweighted mean of eight subset means.

Conformer-energy error. This is the accuracy axis of Figure 5 and the error columns of Table 11. On the external conformer benchmark the metric is computed per molecule as the mean over all conformer *pairs* of $|\Delta E_{\mathrm{ref}} - \Delta E_{\mathrm{pred}}|$; taking pair differences cancels any per-composition constant, so the metric is reference-free and no per-molecule energy offset can be fitted. We report the mean and the median over molecules, as the distribution is heavy-tailed. Errors are reported over all 800 molecules with nothing excluded. Coverage is not identical for every model: SO3LR (Kabylda et al., 2025) supports eight elements and scores 789 of the 800, and five MACE-family models score 799; in each case the mean is taken over the molecules that model covers.

Benchmark-suite scores. These are Figure 3, Figure 9 and Table 12. Scores from the external audit suite are the suite's own, computed through its own code path. Where a model exposes more than one entry point we use the same path the suite's published baselines use. Three of the suite's MD benchmarks are excluded from this paper entirely: the suite computes its metric over whatever frames survive, so a simulation that blew up early can score a perfect 1.000. This affects the suite's published leaderboard exactly as it affects us. The three are named in Appendix H.

Charge in the condensed-phase MD. The TYK2 assembly of Section 4 carries a net charge of -3, but the charge-aware models are run with $Q = 0$ and a singlet spin throughout those trajectories: the conditioning input is per graph and the simulation box is one graph, so a single number would have to stand for solute and solvent together. The charged training data is therefore what the model draws on, not a per-run charge label.

D CONFORMER DEFORMATION AND INTERACTION FORCES, ABSOLUTE ERRORS

The two metrics. If we group the validation structures by molecule (atom-mapped SMILES) and let r be the conformer of lowest *DFT* energy, the deformation error of a molecule with n conformers is

$$\frac{1}{n-1}\sum_{i\neq r}\left|\left(E_i^{\mathrm{pred}}-E_r^{\mathrm{pred}}\right)-\left(E_i^{\mathrm{DFT}}-E_r^{\mathrm{DFT}}\right)\right|,$$

and the tables report the mean of that over molecules (molecules with a single conformer are dropped). The interaction force is the part of an atom's total force that moves its own fragment *rigidly*. Split a structure into fragments; for fragment j of total mass M_j, center of mass $\mathbf{R}_j$ and inertia tensor $\mathbf{I}_j$, write $\mathbf{d}_i=\mathbf{r}_i-\mathbf{R}_j$ and $\mathbf{T}_j=\sum_{k\in j}\mathbf{d}_k\times\mathbf{f}_k$ for the net torque. Then, for each atom i of that fragment,

$$\mathbf{f}_i^{\mathrm{int}} \;=\; \underbrace{\frac{m_i}{M_j}\sum_{k\in j}\mathbf{f}_k}_{\text{net force, shared out by mass}} \;+\; \underbrace{m_i\left(\mathbf{I}_j^{-1}\mathbf{T}_j\right)\times\mathbf{d}_i}_{\text{rigid rotation the torque induces}} \quad .$$

The remainder $\mathbf{f}_i-\mathbf{f}_i^{\mathrm{int}}$ deforms the fragment internally and carries zero net force and zero net torque, so $\mathbf{f}^{\mathrm{int}}$ is exactly the push and twist the fragments exert on one another. It is a projection of the forces of the structure itself and needs no separate isolated-fragment calculation; we follow the construction used by Kovács et al. (2025). The error is $|\mathbf{f}^{\mathrm{int}}_{\mathrm{pred}}-\mathbf{f}^{\mathrm{int}}_{\mathrm{DFT}}|$ averaged over *atoms*, pooled across the subset rather than per structure. Only multi-fragment structures carry an interaction force at all, which is why it is reported for the dimer and solvated subsets alone.

Table 8: Deformation energy error, kcal/mol, on the SPICE2 validation set, mean over molecules, with the error as a percentage of the reference. "DFT" column refers to the mean deformation energy, $\langle|E_i-E_{\mathrm{ref}}|\rangle$, calculated using the DFT reference. Columns average within a molecule before averaging across molecules. Under each value, in brackets, is the 95% bootstrap interval for that model's own mean (2 000 resamples over molecules). Solvated amino acids is a single molecule, so there is nothing to resample over and it carries no interval.

subset	n	DFT gap	SPLICE-S	SPLICE-M	SPLICE-L	SPLICE-XL
pubchem	1 167	29.4	0.95 (3.2%)	0.41 (1.4%)	0.21 (0.72%)	0.16 (0.56%)
			[0.90, 1.01]	[0.38, 0.47]	[0.19, 0.24]	[0.14, 0.22]
des370k	111	5.2	0.43 (8.3%)	0.17 (3.3%)	0.108 (2.07%)	0.081 (1.6%)
			[0.38, 0.50]	[0.15, 0.20]	[0.093, 0.129]	[0.060, 0.117]
amino acid ligand	1 626	6.1	0.84 (14%)	0.36 (5.8%)	0.27 (4.4%)	0.21 (3.5%)
			[0.73, 1.00]	[0.32, 0.41]	[0.20, 0.39]	[0.15, 0.35]
dipeptides	25	34.6	0.94 (2.7%)	0.41 (1.2%)	0.18 (0.51%)	0.14 (0.39%)
			[0.75, 1.22]	[0.31, 0.55]	[0.13, 0.26]	[0.10, 0.21]
monomers	20	9.5	0.34 (3.6%)	0.12 (1.3%)	0.052 (0.55%)	0.047 (0.50%)
			[0.21, 0.49]	[0.08, 0.18]	[0.031, 0.078]	[0.024, 0.087]
solvated pubchem	71	23.3	2.9 (13%)	1.5 (6.3%)	0.87 (3.7%)	0.66 (2.8%)
			[2.6, 3.5]	[1.2, 1.9]	[0.74, 1.02]	[0.55, 0.79]
solvated amino acids	1	28.4	1.8 (6.5%)	0.85 (3.0%)	0.43 (1.5%)	0.33 (1.2%)
pubchem boron silicon	162	19.5	0.93 (4.8%)	0.38 (2.0%)	0.19 (0.98%)	0.13 (0.69%)
			[0.80, 1.08]	[0.33, 0.44]	[0.16, 0.23]	[0.11, 0.16]
all val	3 183	15.9	0.91 (5.7%)	0.40 (2.5%)	0.25 (1.6%)	0.20 (1.2%)
			[0.85, 1.00]	[0.37, 0.43]	[0.22, 0.31]	[0.16, 0.26]

Comparing to SO3LR. SO3LR was run on these structures through the same harness, but it is not shown in Figure 4 and is not compared against in Section 4 because it is not trained against the reference these subsets are scored on. Its training set is computed throughout at PBE0+MBD (the SPICE dipeptide and DES dimer portions of it were recomputed at that level rather than taken at their native ωB97M-D3(BJ)/def2-TZVPPD (Kabylda et al., 2025)). Its error here is therefore due to the gap between two density functionals as much as the gap between two models. The size of that

Table 9: Interaction-force error, meV/Å, against the DFT reference. Under each value, in brackets, is the 95% bootstrap interval for that model's own mean (2 000 resamples over structures).

subset	mol/struct	DFT signal	SPLICE-S	SPLICE-M	SPLICE-L	SPLICE-XL
DES370K	2	35.2	3.87 (11.0%) [3.72, 4.03]	1.75 (4.96%) [1.68, 1.83]	0.97 (2.8%) [0.93, 1.01]	0.74 (2.1%) [0.70, 0.79]
Amino Acid Ligand	2	2.7	0.880 (32.4%) [0.861, 0.899]	0.448 (16.5%) [0.439, 0.459]	0.300 (11.0%) [0.292, 0.307]	0.235 (8.67%) [0.230, 0.241]
Solvated PubChem Set 1	21	69.7	11.9 (17.1%) [11.6, 12.3]	5.9 (8.5%) [5.7, 6.3]	3.7 (5.2%) [3.5, 3.9]	2.83 (4.06%) [2.66, 3.03]
Solvated Amino Acids	21	83.0	11.9 (14.3%) [11.5, 12.3]	5.3 (6.4%) [5.1, 5.5]	2.83 (3.41%) [2.74, 2.93]	1.93 (2.32%) [1.85, 2.00]

Table 10: Community models on the same structures, as absolute errors. Each was run through the same harness on the same validation structures, under the two metric definitions above, so the **SPLICE-L** column reproduces Tables 8 and 9. Figure 4 shows UMA-s-1.1 and FeNNix-Bio1-S only as a percentage of each subset's DFT reference; this is the same data in absolute units, together with SO3LR, which that figure omits. A dash is element coverage, not a failure: SO3LR supports eight elements and does not span the boron/silicon subset. Under each value, in brackets, is the 95% bootstrap interval for that model's own mean, on the same 2 000 resamples as Tables 8 and 9 and read with the same caution: the intervals are marginal, and two of them overlapping does not mean the two models are indistinguishable, since every model is scored on the same structures and a difference between two columns is paired. Deformation on solvated amino acids is a single molecule, so there is nothing to resample over and it carries no interval.

subset	SPLICE-L	UMA-s-1.1	FeNNix-Bio1-S	SO3LR
deformation energy, kcal/mol				
pubchem	0.21 [0.19, 0.24]	0.30 [0.29, 0.31]	0.89 [0.87, 0.92]	3.36 [3.24, 3.50]
des370k	0.11 [0.09, 0.13]	0.14 [0.13, 0.16]	0.22 [0.19, 0.26]	0.97 [0.87, 1.07]
amino acid ligand	0.27 [0.20, 0.39]	0.24 [0.23, 0.26]	0.54 [0.49, 0.63]	1.44 [1.33, 1.59]
dipeptides	0.18 [0.13, 0.26]	0.35 [0.30, 0.40]	0.80 [0.70, 0.92]	2.04 [1.74, 2.42]
monomers	0.05 [0.03, 0.08]	0.12 [0.10, 0.14]	0.32 [0.25, 0.37]	0.53 [0.43, 0.63]
solvated pubchem	0.87 [0.74, 1.02]	0.82 [0.67, 1.02]	2.31 [2.07, 2.56]	7.11 [6.10, 8.21]
solvated amino acids	0.43	0.96	1.63	3.69
pubchem boron silicon	0.19 [0.16, 0.23]	0.26 [0.24, 0.28]	0.60 [0.56, 0.64]	–
interaction force, meV/Å				
des370k	0.97 [0.93, 1.01]	0.96 [0.94, 0.98]	2.72 [2.64, 2.80]	6.64 [6.46, 6.82]
amino acid ligand	0.30 [0.29, 0.31]	0.31 [0.307, 0.314]	0.86 [0.85, 0.88]	1.85 [1.81, 1.88]
solvated pubchem	3.65 [3.46, 3.87]	3.02 [2.94, 3.10]	11.42 [11.17, 11.69]	24.65 [23.89, 25.51]
solvated amino acids	2.83 [2.74, 2.93]	2.87 [2.80, 2.93]	11.06 [10.68, 11.44]	21.98 [21.62, 22.37]

effect is worth recording (see Table 10): over the seven deformation subsets SO3LR covers it reads 5–15× SPLICE-L's error, where FeNNix-Bio1-S reads 2–6× and UMA-s 0.9–2×; both of the latter are trained at or near the reference level. This shows how on a metric of this kind, which functional the labels came from can outweigh the difference between one architecture and another. The same caveat applies, in a slightly lesser form, to the results in Section 5. Additionally, SO3LR supports eight elements, so it is absent from PubChem B/Si and scores 72 895 of the 87 764 structures.

E DATA PROVENANCE AND CONTAMINATION

All SPICE2 models here are trained on the curated release described in Section 3, using the vendor's 95/5 per-SMILES split so that no conformer of a validation molecule appears in training. The models take no formal charge as input.

The conformer test set. The 800 molecules of Eastman et al. (2024) are 200 ligands of 40–50 atoms and 200 of 70–80 from Ligand Expo, 200 random-sequence pentapeptides and 200 dimers from PDB structures. It therefore probes extrapolation in size and composition rather than in element coverage.

On contamination of the conformer benchmark: 32 of the 800 molecules occur in SPICE 2.0.1, and 23 of those survive the curation used here, all of them neutral. No conformer is shared, the minimum RMSD between a test conformer and any training conformer being 0.79 Å. The metric is reference-free, a mean over all 45 conformer pairs, so any per-molecule constant cancels. We report on all 800 molecules because this overlap is field-wide and removing it for us alone would not make the comparison cleaner. The OMol25-trained models carry a wider overlap with the same benchmark, which we measured the same way: OMol25 also contains 32 of the 800, 27 of them in its training split. Those are the 23 above together with nine charged molecules that our curation removes, so the OMol25 models are contaminated on the charged column as well as on the neutral one.

SPICE2+Charge dataset. When adding charged configurations back into the curated SPICE2 dataset we simply added all configurations: the charged half is every structure of SPICE 2.0.1 whose stored SMILES carries a non-zero formal charge; no chemical, geometric or energetic screen was applied, and the only structures removed are the nine molecules that also appear in the conformer benchmark of Section 5, to limit contamination. It is therefore the raw SPICE 2.0.1 charged chemistry.

On contamination of the validation subsets of Section 4: the community models are not held out of these molecules the way ours are. By exact atom-mapped SMILES match, OMol25, which UMA-s is trained on, holds 87.3% of the 4 587 validation molecules, and between 79% and 100% of every individual subset, so UMA-s is in-sample essentially everywhere in Figure 4 and Table 10 where our models are held out. FeNNix-Bio1 is trained on an extension of SPICE2 and so overlaps this split as well; we have not measured by how much. Only molecules are matched, not conformers, so this bounds the overlap from above rather than measuring it exactly. Our own SPICE2 models have zero overlap by construction, the split being per-SMILES.

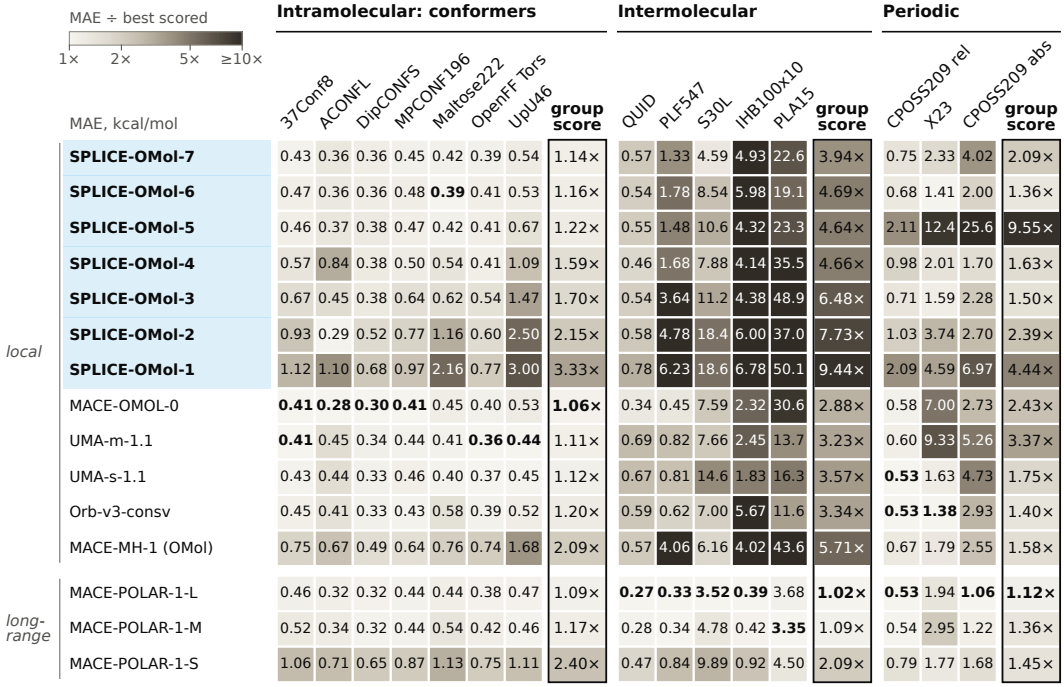

Figure 8: More models on all 15 benchmarks of Batatia et al. (2026): all seven of our OMol arms, largest first, and the eight published models that carry no added dispersion correction. Encoding as in Figure 6, against the same references. The conformer scores fall monotonically with capacity; the intermolecular scores do not improve nearly as fast. The periodic sets extrapolate for every model trained only on OMol25, which contains no periodic structures.

F THE MACE-POLAR-1 SUITE

Every number is a mean absolute error in kcal/mol over one benchmark, from a single run of one harness, on all seven OMol arms. The sets fall into five groups.

- *Intramolecular* conformer energies: 37Conf8 (organic, 259 relative energies), ACONFL (long n-alkanes, 50), DipCONFS (amino acids and dipeptides, 612), MPCONF196 (biomolecular, 196), Maltose222 (222), OpenFF Tors (torsion scans, 1 357) and UpU46 (RNA backbone, 45).
- *Intermolecular* interaction energies: PLF547 (protein–ligand fragments, 547), S30L (supramolecular, 30), IHB100x10 (ionic hydrogen bonds, 1 000) and PLA15 (protein–ligand pockets, 15). These eleven are the sets of Figure 6, and are the ones that work reports for every model it compares.
- *Further non-covalent sets*, reported here only: QUID (ligand–pocket dimers), LNCI16, L14 and vL11 (large and very large complexes, 16, 14 and 11 interaction energies), ion dimers (580 distances across 10 salt curves) and water-cluster separation, split by charge.
- *Conformer sets the source work does not report*: Glucose205, solvated MPCONF196.
- *Periodic*: X23 lattice energies and CPOSS209 polymorphs, absolute and relative.

The periodic sets. OMol25 contains no periodic structures, so every model trained on it is extrapolating. Across our seven arms, capacity predicts accuracy 89% of the time on the molecular sets and 56% of the time here. We report our results for completeness in Figure 8, but do not believe these to be appropriate tests for these models.

Provenance. Peer values are the source work's own comparison models. Four rows are the authors' own corrections and supersede the published values. QUID is scored on the 90 unique ge-

ometries that carry a coupled-cluster reference, ours and the peers' on that same selection. For the ion-dimer curves, each curve is referenced to its own minimum with the same leading points dropped on both sides.

G CONFORMER-ENERGY ERRORS, PER MODEL

Table 11: Conformer-energy error, kcal/mol, per model. Throughput is our own re-measurement at 2 661 atoms on one H100, under the protocol of Appendix C, and is not ranked: it places a model on the cost axis rather than being a score to win. In each error column **bold** is the best value in the table and underline the runner-up. *trained on OMol25.

model	steps/s	all 800		neutral (617)		charged
		mean	med.	mean	med.	mean
this work						
SPLICE-S	235	2.89	1.51	2.00	1.33	5.87
SPLICE-M	188	2.39	0.74	1.83	0.60	4.26
SPLICE-L	112	1.87	0.38	1.10	0.31	4.47
SPLICE-XL	55	1.13	**0.27**	0.55	**0.21**	3.10
this work, charge-aware						
SPLICE-C-S	237	2.46	1.58	1.78	1.34	4.74
SPLICE-C-M	189	1.56	0.79	1.00	0.64	3.43
SPLICE-C-L	112	1.10	0.39	0.61	0.32	2.77
SPLICE-C-XL	55	0.86	0.29	**0.49**	0.23	2.11
SPLICE-OMol-1*	229	2.52	1.71	1.86	1.51	4.75
SPLICE-OMol-2*	187	1.97	1.23	1.40	1.10	3.92
SPLICE-OMol-3*	152	1.50	0.93	1.10	0.84	2.85
SPLICE-OMol-4*	111	1.08	0.76	0.86	0.70	1.79
SPLICE-OMol-5*	51	0.91	0.62	0.71	0.57	1.57
SPLICE-OMol-6*	35	0.88	0.61	0.66	0.55	1.61
SPLICE-OMol-7*	17	0.80	0.58	0.62	0.53	1.39
published field						
UMA-m-1.1*	0.27	**0.53**	0.48	0.52	0.47	**0.59**
UMA-s-1.1*	4.43	0.61	0.53	0.57	0.51	0.76
Orb-v3-omol*	3.70	0.70	0.61	0.65	0.58	0.88
MACE-OMOL-0*	1.18	0.90	0.80	0.79	0.72	1.25
MACE-MH-1	3.53	1.50	1.06	1.16	0.87	2.64
Egret-1	1.68	1.70	0.97	1.19	0.82	3.41
MACE-OFF23(L)	2.20	1.73	1.08	1.24	0.90	3.41
FeNNix-Bio1(M)	103	1.88	1.20	1.54	1.14	3.06
MACE-OFF24(M)	4.97	1.95	1.31	1.47	1.11	3.56
MACELES-OFF	5.99	1.96	1.18	1.52	0.98	3.45
FeNNix-Bio1(S)	110	2.11	1.41	1.72	1.33	3.44
AIMNet2	46	2.55	2.22	2.37	2.15	3.16
AceFF-2.0	32	2.86	2.47	2.69	2.22	3.43
SO3LR	110	3.50	2.69	3.37	2.75	3.93
AceFF-1.1	68	3.64	2.73	2.62	2.28	7.07
MACE-OFF23(S)	19	3.73	1.78	3.67	1.56	3.92

H MLIPAUDIT, ALL BENCHMARKS

Here we report some of the MLIPAudit benchmarks that were held out from the main Figure 3 as they are saturated by most models. Table 12 contains results for all models and tests. Three of the suite's MD benchmarks (`stability`, `folding_stability` and `sampling`) are reported nowhere here, because they score a failed simulation on the frames it managed to produce and record that as a pass, for the published baselines as much as for us; some of our simulations do fail. Inference speed is measured through the same JAX-MD (Schoenholz & Cubuk, 2021) code path the suite itself uses at v0.1.6.

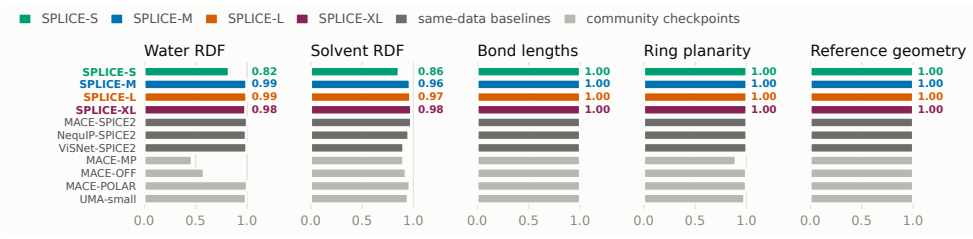

Figure 9: MLIPAudit, the benchmarks held out of Figure 3 because they do not separate the field. The two radial distributions saturate: for `water_radial_distribution` the RMSE term is pinned at 1.0 below a 0.1 threshold, so the score reduces to first-peak position, which is why every SPICE2-family model sits at 0.99–1.00. `water_density` in Figure 3 is the discriminating water result. The remaining three are at or near 1.0 for all ten models.

Table 12: MLIPAudit v0.1.6, all 15 reported benchmarks × 11 models; bold is best-or-tied in the row at the two decimals printed. SPLICE-L's two solvent cells come from an earlier standard-mode run of the same checkpoint, which reproduces the current run to three decimals on the six benchmarks they share.

	this work				same-data (SPICE2)			community			
benchmark	SPLICE-S	SPLICE-M	SPLICE-L	SPLICE-XL	MACE	NequIP	ViSNet	MACE-MP	MACE-OFF	POLAR	UMA-s
bond length distribution	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
conformer selection	0.88	0.94	0.96	0.96	0.92	0.92	0.94	0.64	0.90	0.95	**0.98**
dihedral scan	0.66	0.68	0.70	0.70	0.65	0.67	0.69	0.41	0.66	0.70	**0.71**
inference speed	**0.98**	0.96	0.91	0.78	0.26	0.22	0.65	0.04	0.05	0.01	0.02
noncovalent interactions	0.56	0.64	0.69	0.68	0.61	0.60	0.60	0.19	0.55	**0.98**	0.86
nudged elastic band	0.16	0.20	0.12	0.23	0.19	0.14	0.18	0.29	0.18	**0.49**	0.46
nve energy conservation	**1.00**	**1.00**	0.75	**1.00**	**1.00**	**1.00**	0.25	**1.00**	**1.00**	**1.00**	**1.00**
reactivity	0.28	0.31	0.30	0.32	0.24	0.24	0.25	0.15	0.28	0.84	**0.87**
reference geometry stability	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
ring planarity	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.90	**1.00**	0.99	0.98
solvent density	0.63	0.64	0.59	0.67	0.50	0.59	0.66	0.59	0.56	0.41	**0.77**
solvent radial distribution	0.86	0.96	0.97	**0.98**	**0.98**	0.95	0.90	0.90	0.93	0.96	0.94
tautomers	0.04	0.08	0.18	**0.23**	0.04	0.12	0.11	0.01	0.08	0.14	**0.23**
water density	0.61	**1.00**	**1.00**	0.74	0.89	0.96	0.82	0.73	0.44	0.66	0.65
water radial distribution	0.82	0.99	0.99	0.98	**1.00**	0.99	0.99	0.46	0.58	**1.00**	0.99

I OTHER ABLATIONS: NON-UNIFORM WIDTHS AND CUTOFF

Three levers are measured here that the main text does not use: making the trunk non-uniform in ℓ, tapering the channel width of the tensor product across ℓ, and shortening the cutoff. Figure 10 places both against the family the main text reports.

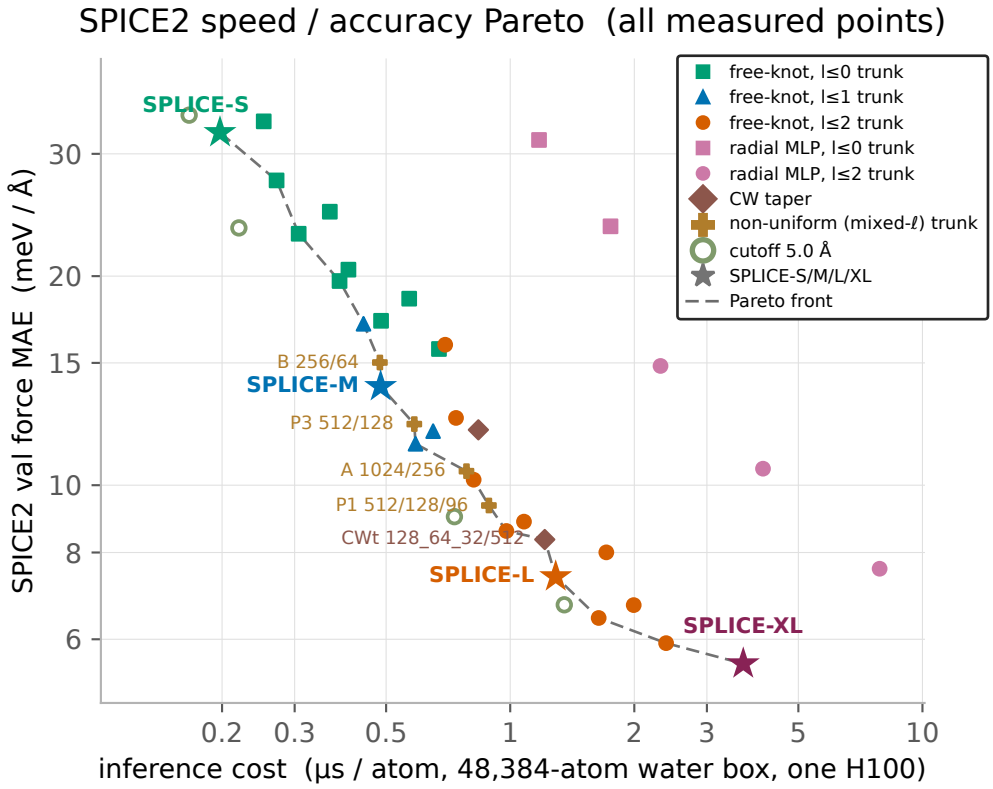

Figure 10: As Figure 2, with every measured point: the non-uniform trunks and the channel-width taper of this appendix, and the 5.0 Å cutoff series. Only the arms of this appendix are labeled; the uniform arms carry the labels of Figure 2. The dashed front is the 6.0 Å uniform front of the main figure.

I.1 A NON-UNIFORM TRUNK

Throughout this work the spherical harmonics run to $\ell_{\max} = 3$ in every model; what the L labels distinguish is the irreps carried by the *node features* between layers. Our uniform arms take that trunk to be either scalars only ($256\times0e$) or full width up to ℓ=2 ($256\times0e + 256\times1o + 256\times2e$). Nothing forces the multiplicity to be the same for every ℓ. One arm makes it non-uniform: the trunk is $256\times0e + 64\times1o$, i.e. it stops at ℓ=1 and gives the vector band one quarter of the scalar width. Its layer-0 symmetric contraction is correspondingly split in two, a 192-channel branch emitting scalars only, and a 64-channel branch emitting scalars and vectors, concatenated back to $256\times0e + 64\times1o$, so the expensive equivariant paths are evaluated on a quarter of the channels.

At fixed channel width 64 and fixed scalar trunk width 256, we can compare different options as shown in Table 13: adding just a small number of vector features has a strong impact in accuracy while costing less performance. This knob does not, however, beat a uniform trunk at matched cost. The mixed-ℓ arm dominates three uniform ℓ=0 arms, but still sits above the uniform front interpolated at its own speed. Three further non-uniform architectures have been explored, as shown in Figure 10. They sit on the uniform front or a little above it, and none of them is clearly below it. The option is therefore real and cheap, but on this dataset it did not seem worth exploring further.

Table 13: One lever, three settings. Channel width 64, scalar trunk width 256 and $\ell_{\max} = 3$ harmonics throughout; only the node-feature irreps differ.

trunk (node-feature) irreps	params	μs/atom	val F (meV/Å)
$256\times0e$	1.96 M	0.307	23.02
$256\times0e + 64\times1o$	2.06 M	0.483	15.02
$256\times0e + 256\times1o + 256\times2e$	2.44 M	0.815	10.18

I.2 A TAPERED CHANNEL WIDTH

Similarly, the channelwise tensor product (4) carries the same channel width C at every degree, and it need not. The two `CWt` arms of Figure 10 taper it across degrees: $64/32/16$ and $128/64/32$ channels at $\ell = 0/1/2$, so that the expensive high-degree paths are evaluated on fewer channels than the scalars. It is the same idea as the non-uniform trunk applied one stage earlier, but in this case it seems to be somewhat less efficient.

I.3 A SHORTER CUTOFF

Every model in this paper uses a 6 Å cutoff, so that a speed–accuracy comparison between two arms varies the architecture and nothing else. The cutoff itself is a cheaper lever than any architectural one. Four arms were retrained at 5 Å and are otherwise identical to their 6 Å twins: they run 17 to 34% faster for 2 to 5% more force error, and three of them land *below* the 6 Å front interpolated at their own speed.

Considering the molecular nature of the dataset, we do not want to draw the conclusion that these models are better, and we have decided not to include these models in the main discussion for a better comparison with existing models. It is however good to remember that this axis exists and could be easy to exploit for certain applications.

J A MEASUREMENT BUILT FOR FAST MODELS

The speed axis of Figure 5 is the external benchmark's own: MD steps per second on a 2 661-atom water box. That definition is good for reproducing the benchmark, but can be problematic when measuring a fast model. At 2 661 atoms a model costing a fraction of a microsecond per atom spends much of each step in kernel-launch overhead and in rebuilding the neighbor list, so the axis compresses precisely the region this paper is about, and it risks measuring neighbor-list implementations instead of the models. This penalizes fast models disproportionately, as can be seen comparing e.g. the DPA4C family between Figure 11 and the original Figure 5.

Figure 11 re-measures the same models on a protocol built for that region, leaving the accuracy axis untouched. One energy-and-force call is timed on a 98 880-atom box on a single NVIDIA B200 180GB GPU, with the neighbor list built once and held outside the timed region, a fixed number of warm-up calls, and one fresh process per model and size, so that neither allocator state nor compilation carries between cells. We have verified that the speed at 98 880 atoms is within 5% of the top speed for all models in this figure.

Figure 11 correctly shows the speed of DPA4C models, while still showing our advantage in the intermediate region. We believe this plot to be complementary to the original one, and more informative in the fast-model regime.

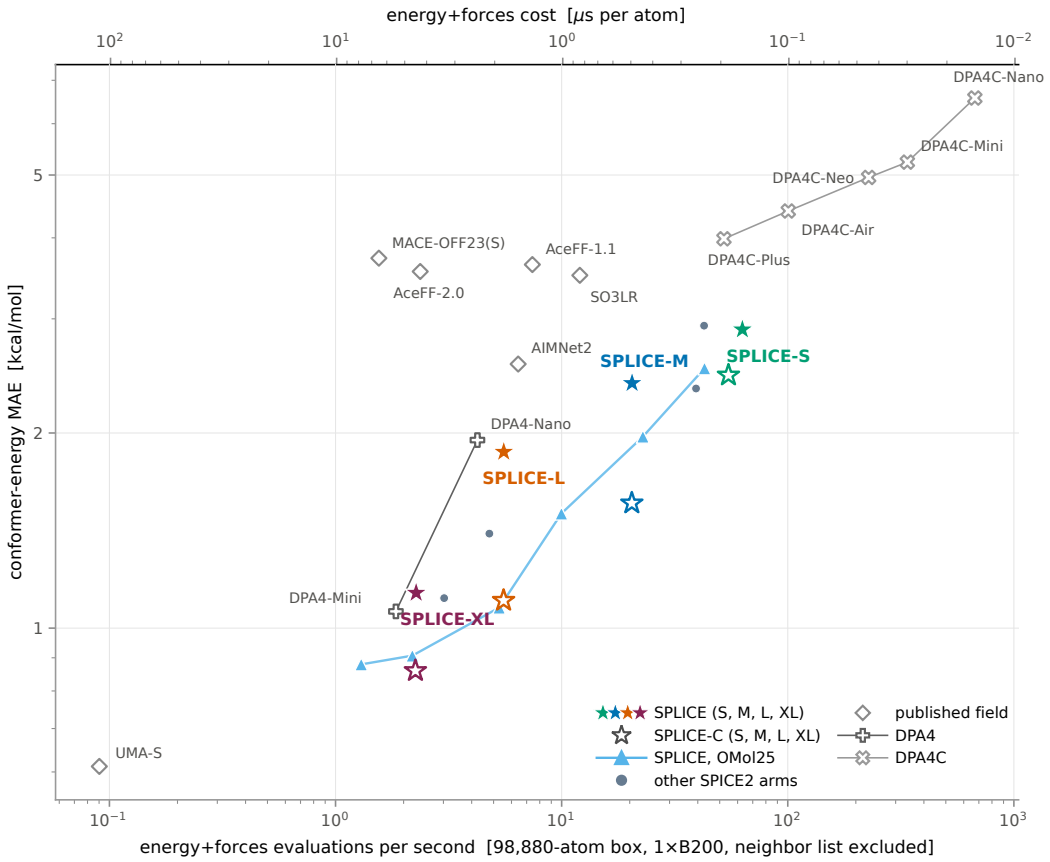

Figure 11: Conformer-energy error against energy-and-force evaluations per second on a 98 880-atom box, one B200, with the neighbor list excluded from the timed region. Faster is to the right, as in Figure 5. The top axis gives the same points as cost per atom, the quantity used elsewhere in this paper; the two are reciprocal at fixed system size, so it runs the other way. Accuracy is the all-800 mean of Figure 5, unchanged. Models absent here are absent because they did not complete a call at this size, not because they were excluded. Most of those absences are GPU memory, at this size or a smaller one. FeNNix-Bio1 is limited by its neighbor list rather than by the GPU.

K THE MODEL FAMILY: ALL MEASURED POINTS

Table 14 gives the numbers behind Figures 2 and 10, with the capacity of each arm alongside them. The three quantities that decide whether a model is usable for a given job, how accurate it is, what a step costs and how large a system it holds, are reported together rather than in separate appendices, and every row is keyed by the names of Appendix B.

Table 14: Every measured arm of the SPICE2 family, ordered by inference cost. μs/atom is one energy-and-force step per atom on the $48\,384$-atom water box; val F is the atoms-weighted force MAE in meV/Å and val E the energy MAE in meV/atom, both on the SPICE2 validation set; max atoms is the largest system that completes a step on one NVIDIA H100 80GB GPU. **Bold** marks the four models carried through the main text. †these two arms cannot hold the $48\,384$-atom box on an 80 GB card and were timed on a smaller one ($24\,000$ and $12\,288$ atoms).

name	μs/atom	val F	val E	max atoms
L0_64_64_Rc5	0.166	34.11	1.811	3 594 240
SPLICE-S	0.198	32.20	1.650	2 073 600
L0_64_256_Rc5	0.220	23.47	1.206	1 505 280
L0_64_64	0.252	33.41	1.681	2 995 200
L0_64_128	0.271	27.47	1.388	1 976 832
L0_64_256	0.307	23.02	1.132	1 497 600
L0_128_128	0.365	24.76	1.249	1 824 768
L0_64_512	0.386	19.68	0.916	774 144
L0_128_256	0.405	20.43	1.002	1 354 752
L1_64_128	0.441	17.08	0.926	1 797 120
L1_64_256+64	0.483	15.02	0.766	1 354 752
SPLICE-M	0.485	13.87	0.705	1 397 760
L0_128_512	0.486	17.25	0.796	760 320
L0_256_256	0.569	18.56	0.893	1 119 744
L1_64_512+128	0.585	12.25	0.605	746 496
L1_64_512	0.589	11.47	0.579	746 496
L1_128_256	0.650	11.96	0.625	1 277 952
L0_256_512	0.672	15.71	0.751	709 632
L2_64_64	0.696	15.93	0.928	1 689 600
L2_64_512_Rc5	0.732	9.01	0.523	718 848
L2_64_128	0.739	12.50	0.702	1 497 600
L1_64_1024+256	0.783	10.48	0.518	331 776
L2_64_256	0.815	10.18	0.644	1 228 800
L2_64/64+32+16_256	0.837	12.01	0.630	1 300 992
L2_64_512+128+96	0.888	9.35	0.489	737 280
L2_64_512	0.979	8.59	0.473	709 632
L2_128_256	1.080	8.86	0.479	829 440
L0_64_64_MLP	1.175	31.41	1.588	139 776
L2_128/128+64+32_512	1.213	8.35	0.437	677 376
SPLICE-L	1.289	7.39	0.377	677 376
L2_128_1024_Rc5	1.353	6.73	0.403	301 056
L2_128_1024	1.639	6.44	0.335	301 056
L2_256_256	1.710	8.00	0.433	539 136
L0_128_128_MLP	1.751	23.59	1.145	86 016
L2_256_512	1.993	6.72	0.357	456 192
L2_64_64_MLP	2.315	14.85	0.800	55 296
L2_256_1024	2.389	5.92	0.315	290 304
L0_256_256_MLP	2.994	17.88	0.968	46 080
SPLICE-XL	3.676	5.53	0.368	147 456
L2_128_128_MLP†	4.104	10.56	0.565	23 040
L2_256_256_MLP†	7.872	7.58	0.399	12 288