SAND2022-12230C

Sandia National Laboratories

SNAP and Beyond: Machine Learning Interatomic Potentials in LAMMPS

Aidan Thompson
Center for Computing Research,
Sandia National Laboratories,
Albuquerque, New Mexico

"Multiscale Modeling of Matter under Extreme Conditions," Görlitz, Germany, September 14, 2022

SAND2022-xxxx C

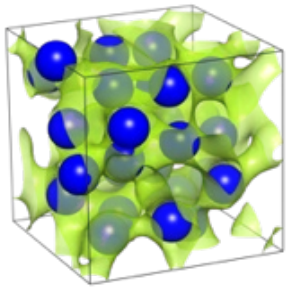
U.S. DEPARTMENT OF ENERGY NNSA

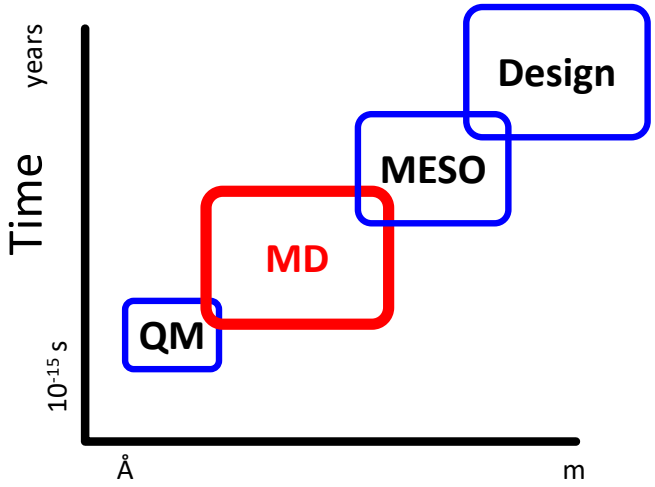
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

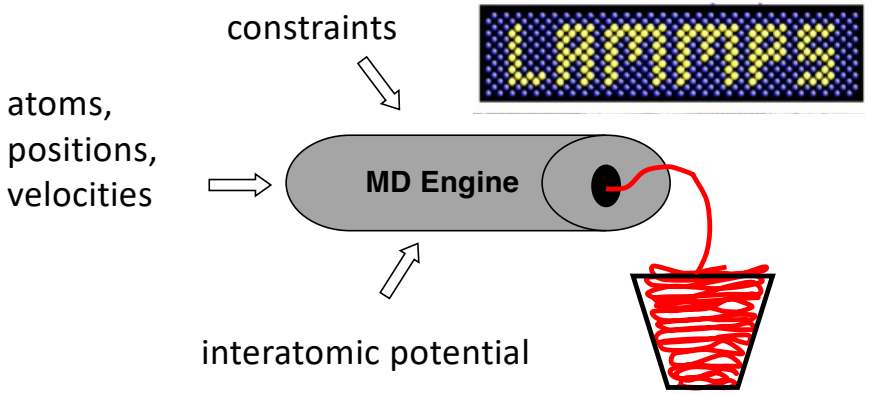
What is Molecular Dynamics Simulation?

Quantum Molecular Dynamics

- Input: ion positions
- Output: Electronic structure
- Energy, forces, stress
- Sample simple bulk properties
- Expensive
- O(N^3) scaling
- N ~ hundreds

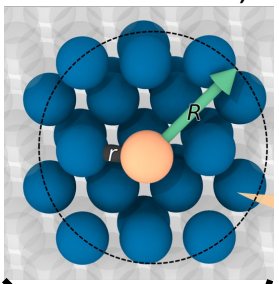


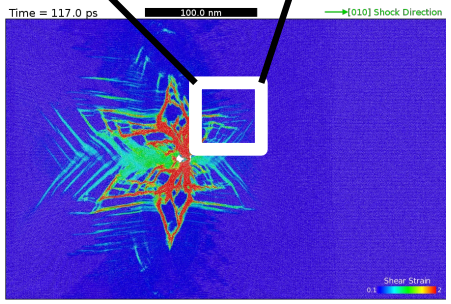




Classical Molecular Dynamics

- No electrons
- Interatomic potential
- Energy, forces, stress
- Describes complex properties and processes
- O(N) scaling
- N ~ millions, billions





What is LAMMPS?

Large-scale Atomic/Molecular Massively Parallel Simulator
https://www.lammps.org

- Focus on materials modeling: soft matter or solids
- Particle simulator at varying length and time scales
  electrons ⇒ atoms ⇒ CG ⇒ mesoscale ⇒ continuum
- Spatial-decomposition of domain for MPI parallelism
- Support for GPUs & OpenMP via Kokkos (portability)
- Can be coupled to other scales: QM, kMC, FE, CFD, ...
- Open source, available on GitHub, GPL or LGPL
- Ecosystem with hundreds of developers

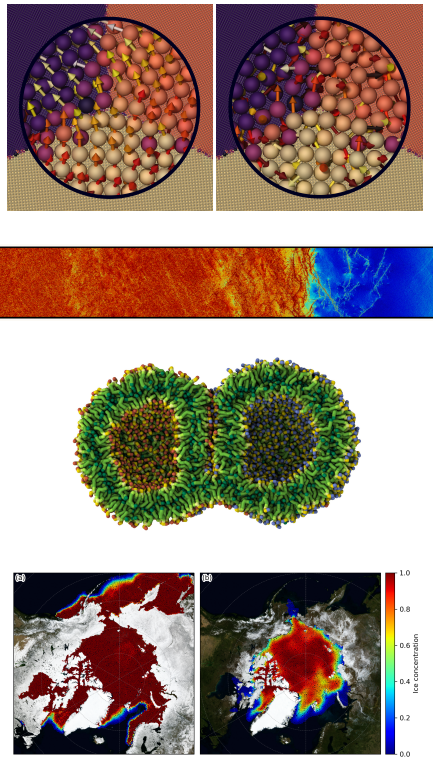
ELSEVIER

Computer Physics Communications
Available online 22 September 2021, 108171
In Press, Journal Pre-proof

Feature article

LAMMPS - A flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales

Aidan P. Thompson [a], H. Metin Aktulga [b], Richard Berger [c], Dan S. Bolintineanu [a], W. Michael Brown [d], Paul S. Crozier [a], Pieter J. in 't Veld [e], Axel Kohlmeyer [c], Stan G. Moore [a], Trung Dac Nguyen [f], Ray Shan [g], Mark Stevens [a], Julien Tranchida [a], Christian Trott [a], Steven J. Plimpton [a]



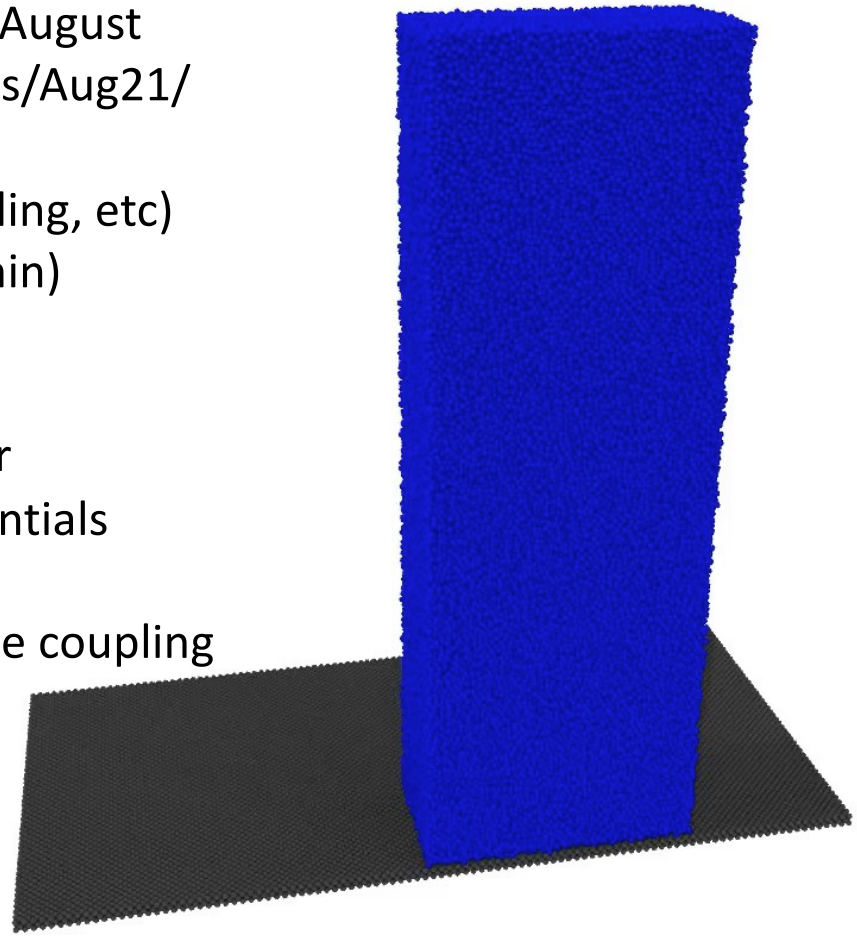
Four images for the journal cover visualizing representative LAMMPS simulations of particle-based material models on a range of length and time scales. a) atomic magnetic spin configurations b) 2B atom simulation with SNAP carbon ML potential c) coarse-grained molecular simulation of biological vesicles fusing d) DEMSI simulation of arctic sea ice

New in LAMMPS

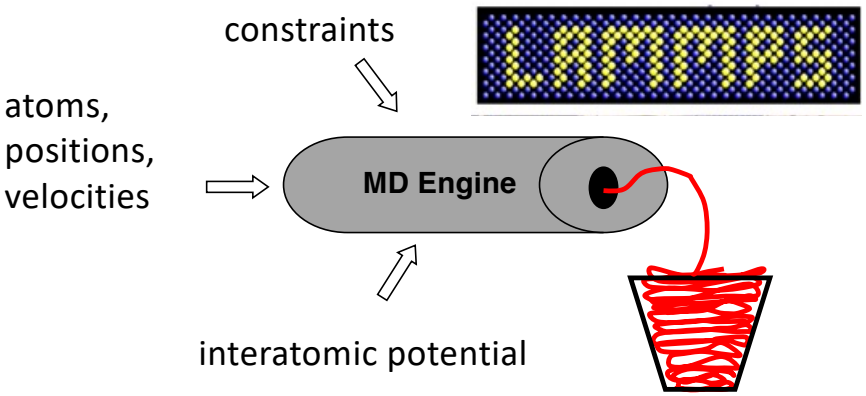
RHEO package, Joel Clemmer, Dan Bolintineanu, Thomas O'Connor (Sandia)
https://download.lammps.org/workshops/Aug21/day3/joel-clemmer.pdf

- New features from virtual LAMMPS workshop in August
  - website: https://www.lammps.org/workshops/Aug21/
  - 4-hour hands-on tutorial, 6 1-hour breakouts
    4 invited talks on fun topics (ML, Covid modeling, etc)
  - 20 short talks (15 min), 20 lightning talks (3 min)
  - recordings of all talks available on website

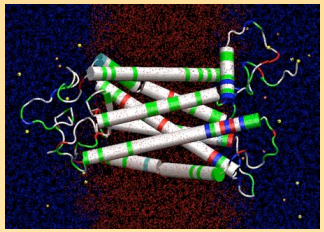
- Python + LAMMPS: call either one from the other
- Inter-operability with OpenKIM database of potentials
- Machine-learning interatomic potentials
- Support for MDI (MolSSI Driver Interface) for code coupling

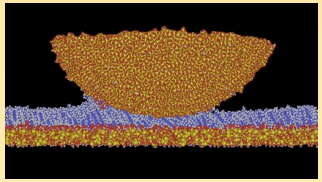


What is LAMMPS?



What is LAMMPS?





Initial positions and velocities

Interatomic potential

- Biomolecules
- Polymers (soft materials)
- Materials science (hard materials)
- Mesoscale to continuum

Large-scale Atomic/Molecular Massively Parallel Simulator

⇨

Positions and velocities at later times

Thanks to ~~Aidan Thompson~~ Mike Chandross

SNAP Training Workflow

https://github.com/FitSNAP/FitSNAP

Model Form

- Energy of atom i expressed as a basis expansion over K components of the bispectrum (B_k^i)

$$E^i_{SNAP} = \boldsymbol{\beta} \cdot \mathbf{B}^i + \frac{1}{2}(\mathbf{B}^i)^T \cdot \boldsymbol{\alpha} \cdot \mathbf{B}^i$$

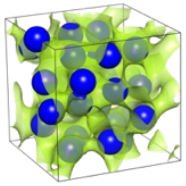
Regression Method

- $\boldsymbol{\beta}$ vector fully describes a SNAP potential
- Decouples MD speed from training set size

$$\min(||\mathbf{w} \cdot D\boldsymbol{\beta} - T||^2 - \gamma_n \, ||\boldsymbol{\beta}||^n)$$

Weights | Set of Descriptors | DFT Training | Regularization Penalty

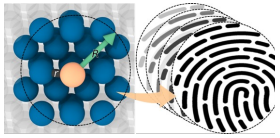
energies
forces
stress tensors



Hyperparameter Optimization (SOGA Genetic Algorithm)

optimize hyper-parameters

DFT Training Data



DAKOTA

FitSNAP.py

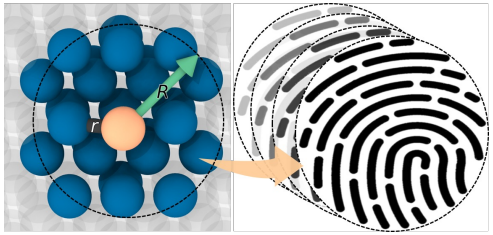
LAMMPS

summit

energy and force errors
material property objective functions

Bispectrum Components as Descriptor

- Neighbors of each atom are mapped onto unit sphere in 4D
  $3D\ Ball\colon (r,\theta,\phi), r < R_{cut} \implies 4D\ Sphere\colon (\theta_0,\theta,\phi), \theta_0 = \frac{r}{R_{cut}}\pi$
- Expand density around each atom in a basis of ***4D hyperspherical harmonics***,
- Bispectrum components of the 4D hyperspherical harmonic expansion are used as the geometric descriptors of the local environment
  - Preserves universal physical symmetries
  - Rotation, translation, permutation
  - Size-consistent (extensible)



$$u^j_{m,m'} = U^j_{m,m'}(0,0,0) + \sum_{r_{ii'_j}<R_{cut}} f_c(r_{ii'})w_i U^j_{m,m'}(\theta_0,\theta,\phi)$$

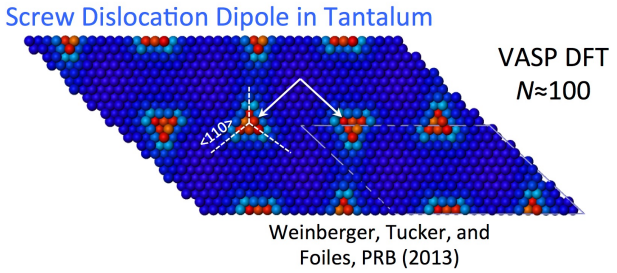
$$B_{j_1,j_2,j} = \sum_{m_1,m'_1=-j_1}^{j_1} \sum_{m_2,m'_2=-j_2}^{j_2} \sum_{m,m'=-j} (u^j_{m,m'})^* H^{jmm'}_{j_1 m_1 m'_1 \atop j_2 m_2 m'_2}\ u^{j_1}_{m_1,m'_1} u^{j_2}_{m_2,m'_2}$$

Symmetry relation: $$\frac{B_{j_1,j_2,j}}{2j+1} = \frac{B_{j,j_2,j_1}}{2j_1+1} = \frac{B_{j_1,j,j_2}}{2j_2+1}$$

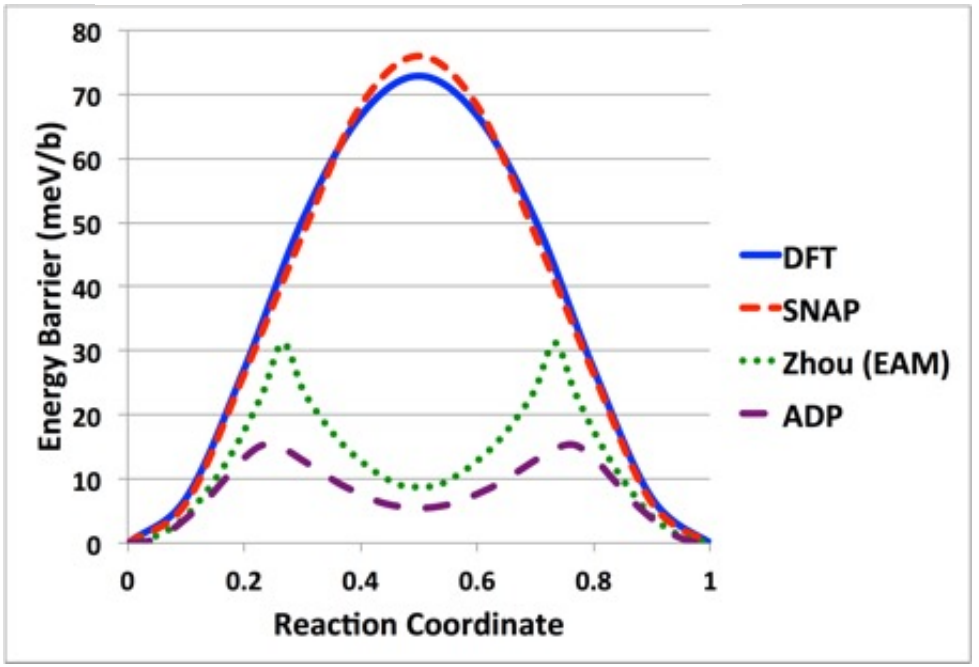
SNAP, SOAP, Steinhardt, Behler, and many other descriptors are all related to the atomic cluster expansion (Ralf Drautz *Phys.Rev.B* 2019)

SNAP Tantalum

- Training data:
  - Energy, force, stress
  - 363 configurations
  - Deformed crystals phases
  - Generalized stacking faults
  - Surfaces
  - Liquid
- Peierls barrier is the activation energy to move a screw dislocation
- Not included in training data
- SNAP **post-diction** agrees well with DFT calculations



Weinberger, Tucker, and Foiles, PRB (2013)



A. P. Thompson , L.P. Swiler, C.R. Trott, S.M. Foiles, and G.J. Tucker, *J. Comp. Phys.,* **285** 316 (2015) .

SNAP Applications

SNL Involved, Independent

System	Year	Usage	Origin	N_{DoF}	$N_{Training}$	Descriptors
Ta	2014	Dislocation motion	SNL, Thompson	31	363	Linear
InP	2015	Radiation damage, defects	SNL, Thompson	31	665	Linear
WBeHe	2017	Plasma facing materials	SNL, Wood	56	25,052	Linear
Mo	2017	Phase diagram prediction	UCSD, Ong	31	1000	Linear
Actinides	2018	Shock, phase transitions	SNL/LLNL	56	20,000	Quadratic
NiMo	2018	Phase diagram prediction	UCSD, Ong	31	2,000	Linear
LiN	2019	Super-Ionic Conductor	UCSD, Ong	31	3,000	Lin+Charge
★ Various	2020	Accuracy/Cost comparison	UCSD/SNL	10-130	1,000	Lin, Quad
InP	2020	Radiation damage, defects	SNL, Cusentino	241	1,000	EME
AlNbTi	2020	High entropy alloy design	SNL, Tranchida	1596	7,250	Quadratic
Si	2020	Neural network SNAP	UNLV, Zhu	1596	>5,000	NN
Al	2021	Predicting electron density	SNL, Ellis	91	30	NN
Fe	2021	Magnetic phase transition	SNL, Nikolov	1596	683	Quad+Spin

SNAP Applications

SNL Involved, Independent

(more in the literature, not an exhaustive list)

System	Year	Usage	Origin	N_{DoF}	$N_{Training}$	Descriptors
WBeHN	-	Plasma facing materials	SNL, Cusentino	56*	>40,000	Linear
C	-	Planetary impacts, shock	USF, Willman	1596	30,000	Quadratic
C, V	2021	Metal plasmas	SNL, Wood	1596	10,000	Quadratic
MoNbTaTi	-	HEA alloy design	SNL, McCarthy	-	>5,000	EME
GeSe	-	Vitrification	UCD, Sievers	-	>5,000	EME
LiMoS	-	Li-ion batteries	UConn, Dongarre	-	>5,000	-
SiGeSnPb	-	Thermoelectric materials	GWU, Li	-	>5,000	-
W	-	Model form selection	LANL/SNL	-	330,000	NN

So what should you train a ML-IAP on?

How do you recognize failures (poor extrapolations)?

- Growing evidence that SNAP is a *general use* material model form, unlike any interatomic potential used in MD to date
- SNAP model training software now incorporated in Materials Design Inc. products

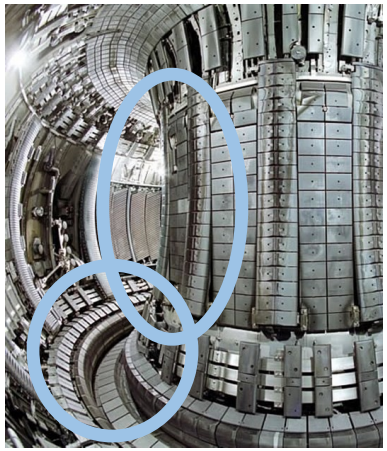
Materials for Fusion Energy

Cusentino, Wood

- Difficult to develop materials to handle extreme conditions within tokamak
- Large heat loads of 10-20 MW/m^3
- High particles fluxes of ~10^{24} $m^{-2}s^{-1}$ of mixed ion species (H/He/Be/N etc.)
- Complex chemical/physical processes

Beryllium First Wall

Tungsten Divertor

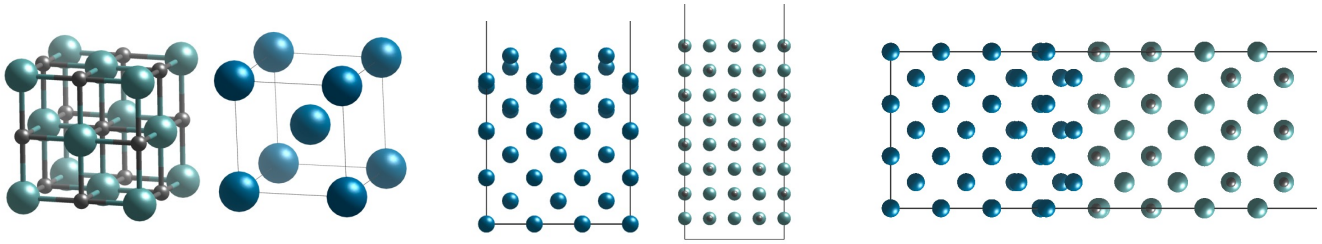


Joint European Torus (JET)

- Molecular dynamics is well suited to study initial implantation, diffusion, material deformation, and surface response of plasma facing materials
- Need accurate interatomic potentials
- We have published potentials for:
  - Helium/Tungsten
  - Beryllium/Tungsten
- In progress:
  - Hydrogen/Tungsten
  - Nitrogen/Tungsten
  - ZrC/Tungsten
- Eventually: H/He/N/ZrC/W

Dispersoid-strengthened Tungsten

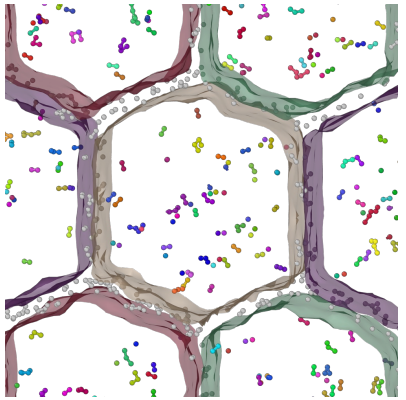
- The W-ZrC SNAP potential is trained on ~8,000 structures including bulk, surfaces, interfaces, as well as ab initio molecular dynamics.



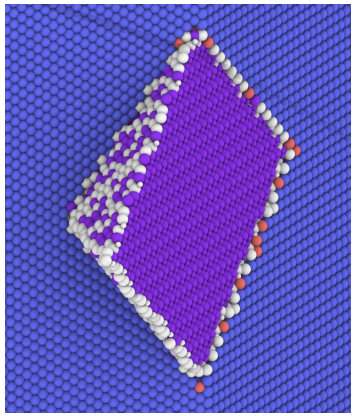
Bulk Surfaces Interfaces

- A genetic algorithm is used to optimize hyper-parameters for low force, energy, and material property errors.
- Potentials are narrowed down based on material properties like bulk modulus (GPa) and surface energy (eV/Å^2) and stability in NVT simulations.

Sikorski, Cusentino, Wood



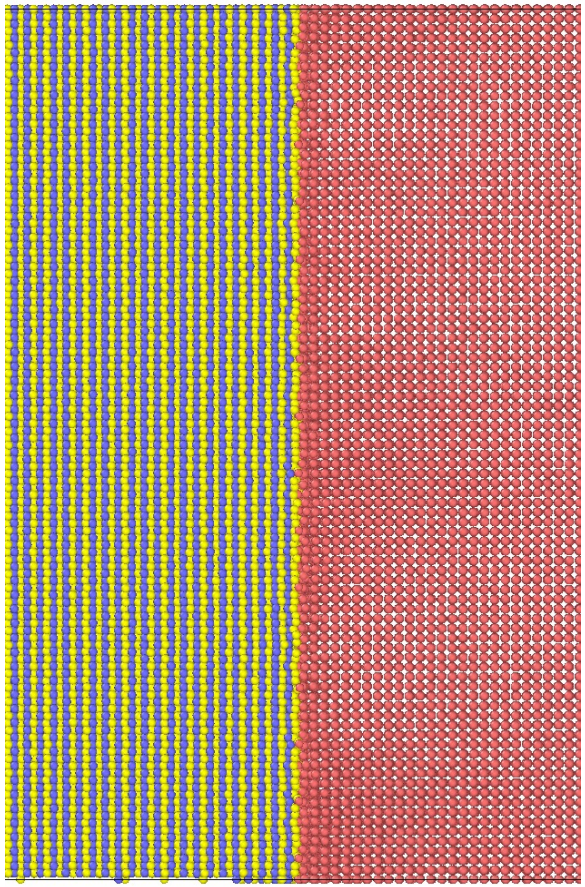
Hydrogen uptake in polycrystalline tungsten



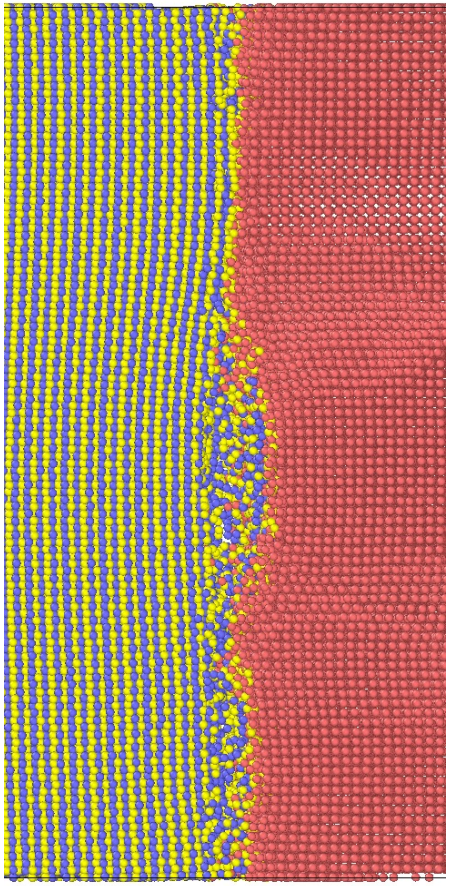
Zirconium carbide dispersoid nanoparticle in tungsten

Saikat C-terminated interface during tensile test @ 300K

Just before fracture

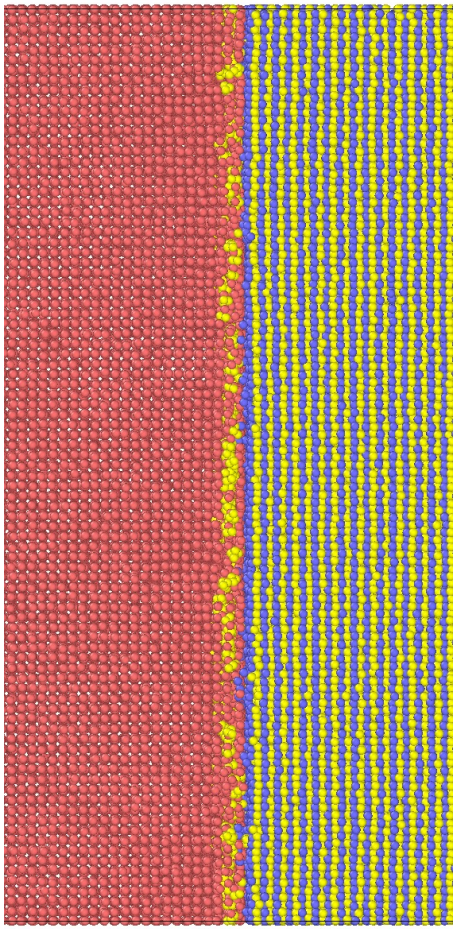


At fracture

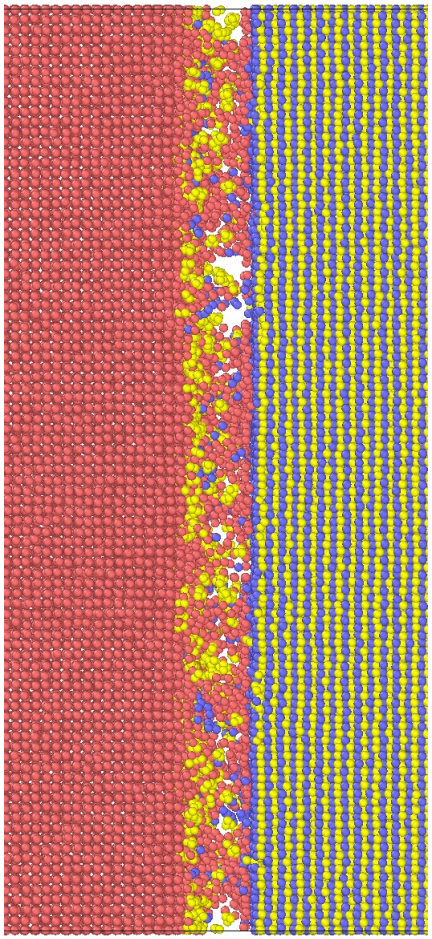


Saikat C-terminated interface during tensile test @ 2500K

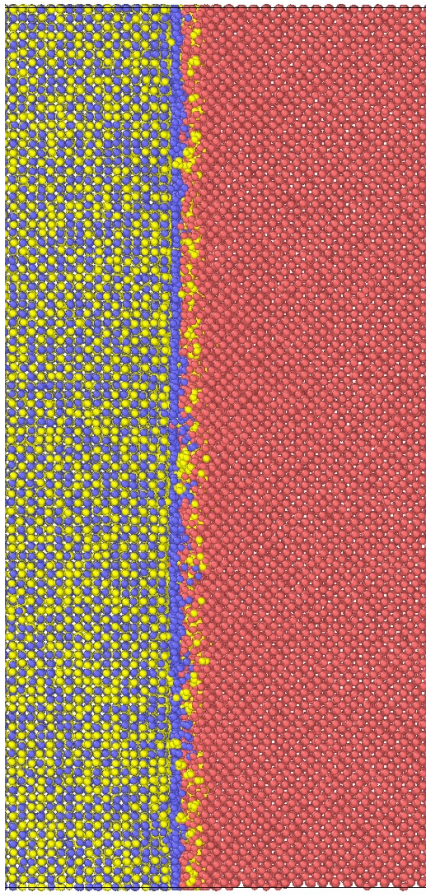
Just before fracture

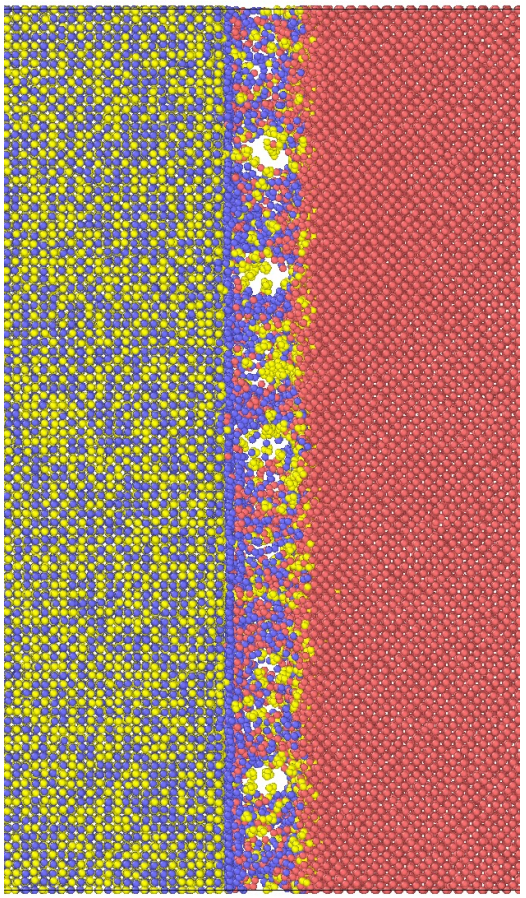


At fracture



Zhang interface during tensile testing @ 2500 K



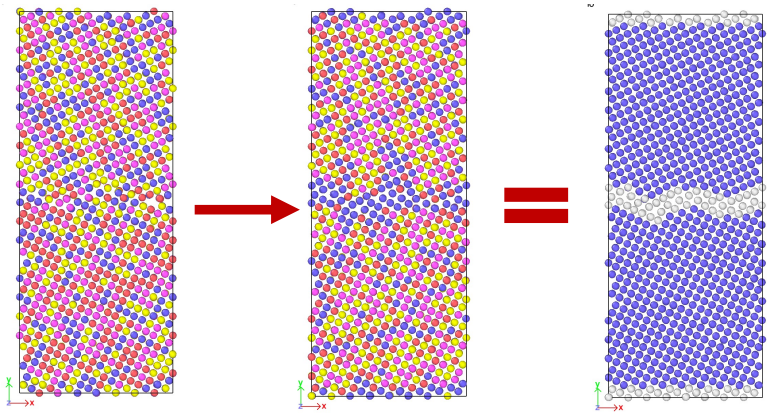


Other Advanced Materials

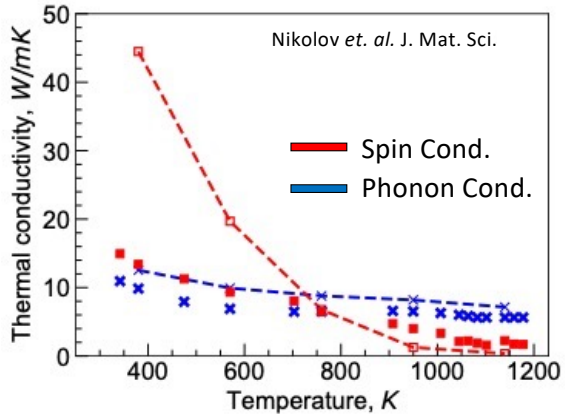
McCarthy, Nikolov, Wood

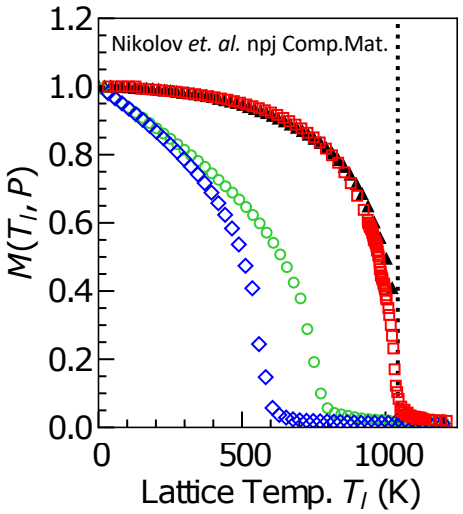
Chemical, Functional Complexity

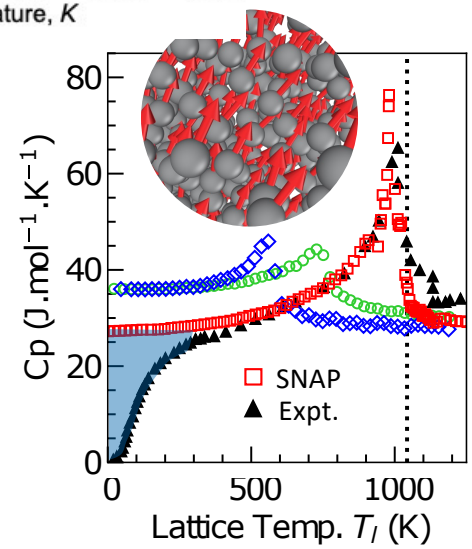
- Many materials are absent from MD literature due to viability/availability of IAP
- (Right) SNAP ML-IAP with explicit treatment of magnetic spin dynamics
- (Below) MoNbTaTi HEA displaying short range chemical order near grain boundaries



RSS, 0 K MC Anneal



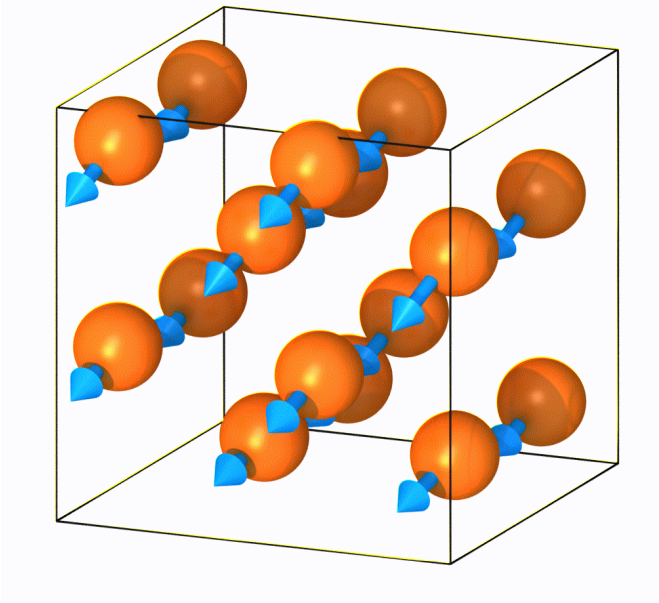




Magnetic Materials Simulations

Chemical, Functional Complexity

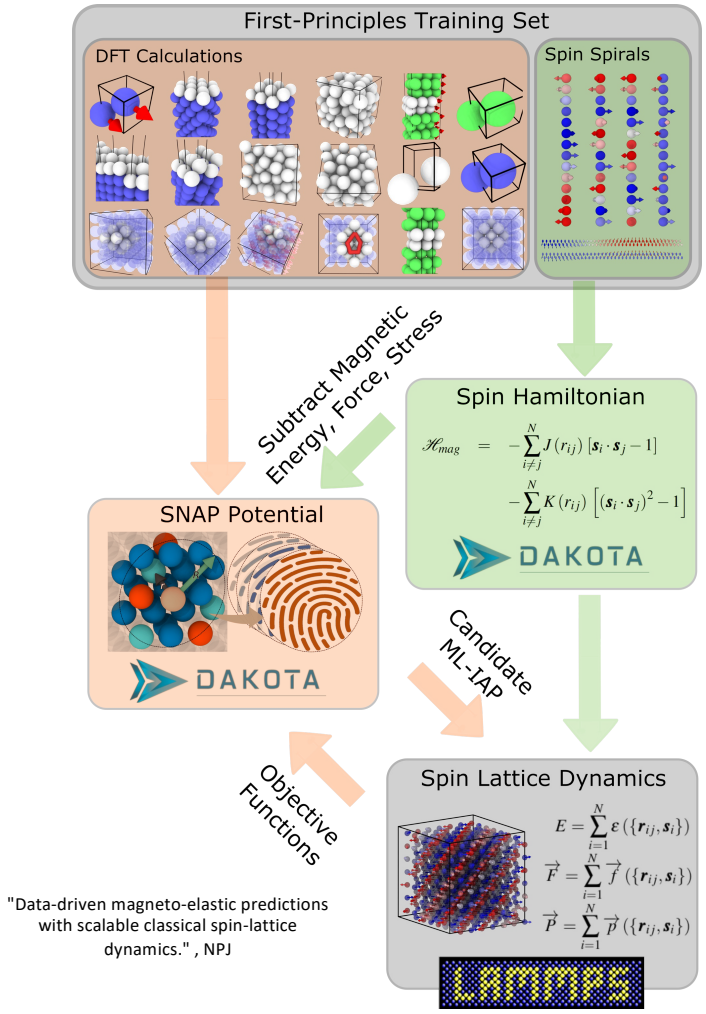
- Electronic structure or Spin-Lattice simulations have been the only options for many decades.
- Former lacks scalability, latter lacks any real dynamic or finite temperature effects



- Molecular Dynamics
  - Atoms interact via nonmagnetic interatomic potential, $U(\mathbf{R})$
  - $\mathcal{H}_{MD} = \sum_i \frac{\mathbf{p}_i^2}{2m} + U(\mathbf{R})$
- Spin Dynamics
  - Atoms interact via exchange Hamiltonian
  - $\mathcal{H}_s = -\sum_{i,j}^{N} J_{ij}(\mathbf{R})\left[\vec{\boldsymbol{s}_i} \cdot \vec{\boldsymbol{s}_j} - 1\right] - \sum_{i,j}^{N} K_{ij}(\mathbf{R})\left[\left(\vec{\boldsymbol{s}_i} \cdot \vec{\boldsymbol{s}_j}\right)^2 - 1\right]$
  - **Momentumless** damped dynamics (Landau–Lifshitz–Gilbert)
- Molecular-Spin Dynamics
  - $\mathcal{H}_{MSD} = \mathcal{H}_l + \mathcal{H}_s = \sum_i \frac{\mathbf{p}_i^2}{2m} + U(\mathbf{R}) - \sum_{i,j}^{N} J_{ij}(\mathbf{R})\left[\vec{\boldsymbol{s}_i} \cdot \vec{\boldsymbol{s}_j} - 1\right] - \sum_{i,j}^{N} K_{ij}(\mathbf{R})\left[\left(\vec{\boldsymbol{s}_i} \cdot \vec{\boldsymbol{s}_j}\right)^2 - 1\right]$
  - Parallel symplectic time integration implemented in LAMMPS
  - Energy and magnetization conservation ~ O(dt^2) for NVE
  - Tranchida et al., J. Comp. Phys. (2018)

Magnetic Materials Simulations

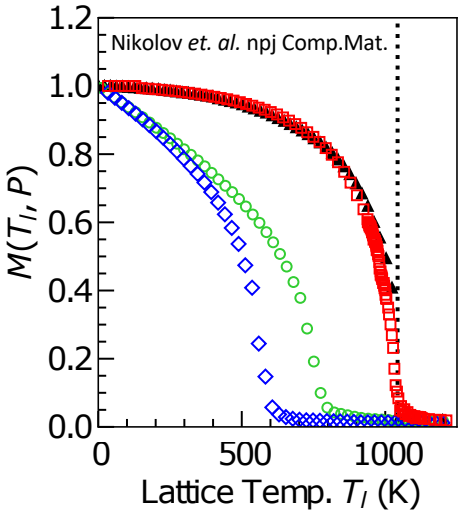
Tranchida, Nikolov, Wood

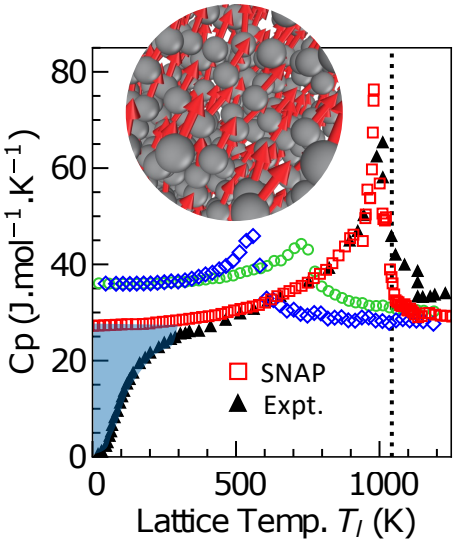


"Data-driven magneto-elastic predictions with scalable classical spin-lattice dynamics." , NPJ

Iron

- Transformational capability to study magnetic materials at the grain scale
- Explicit treatment of spin dynamics captures the second order phase transition at Curie temperature



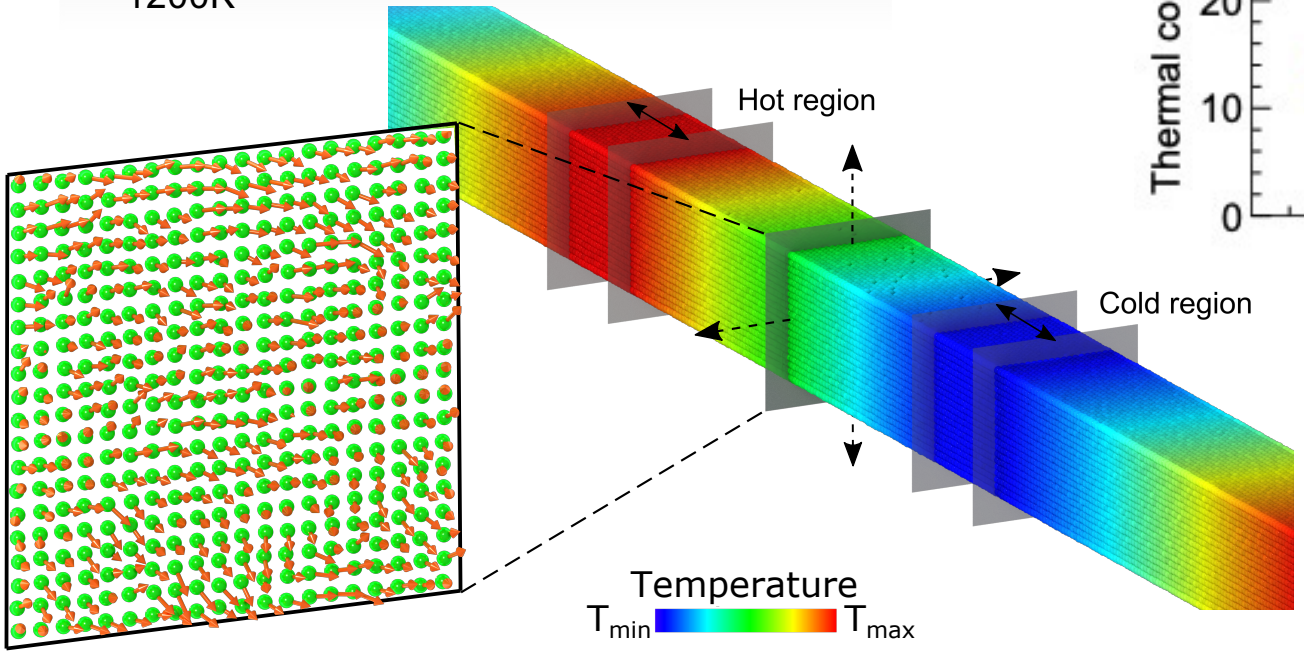


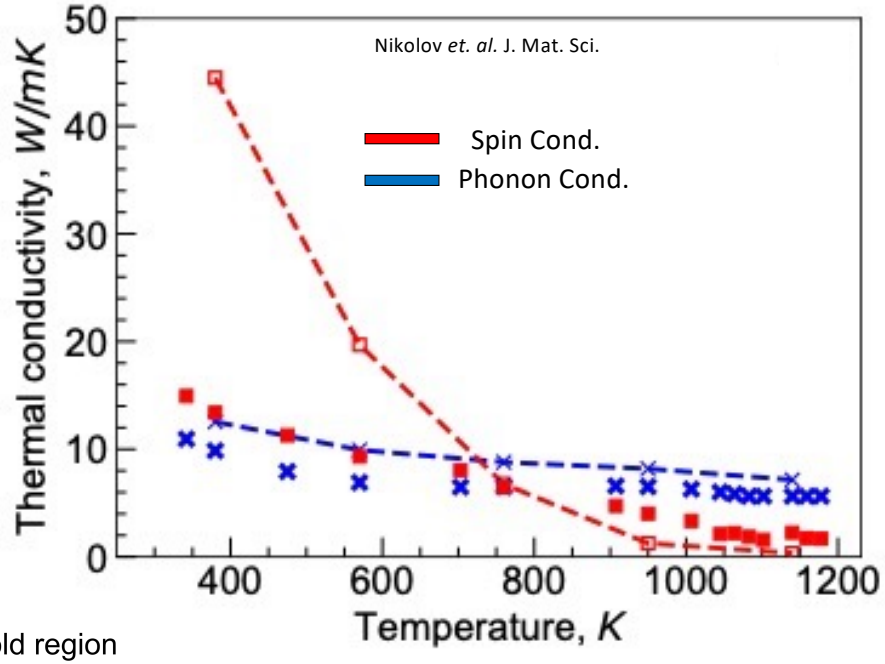
Magnetic Materials Simulations

Tranchida, Nikolov, Wood

Finite Temperature Magnetism

- Hot/cold regions are spaced 28.8 nm apart
- Thermal gradient established by setting hot region to $T_{max} = 1.08T_{min}$,T_{min} : 300 - 1200K





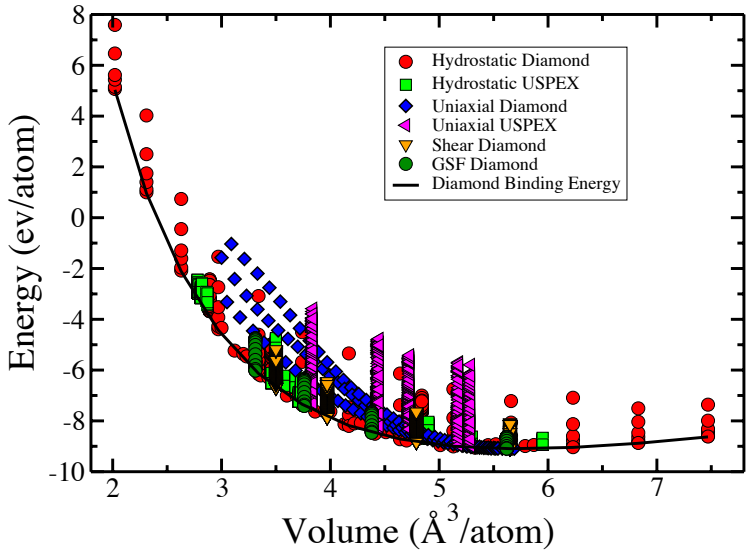
- Magnon-phonon scattering significantly reduces conductivity
- Magnons more conductive than phonons where $T < 0.5T_{Curie}$

High Pressure Carbon

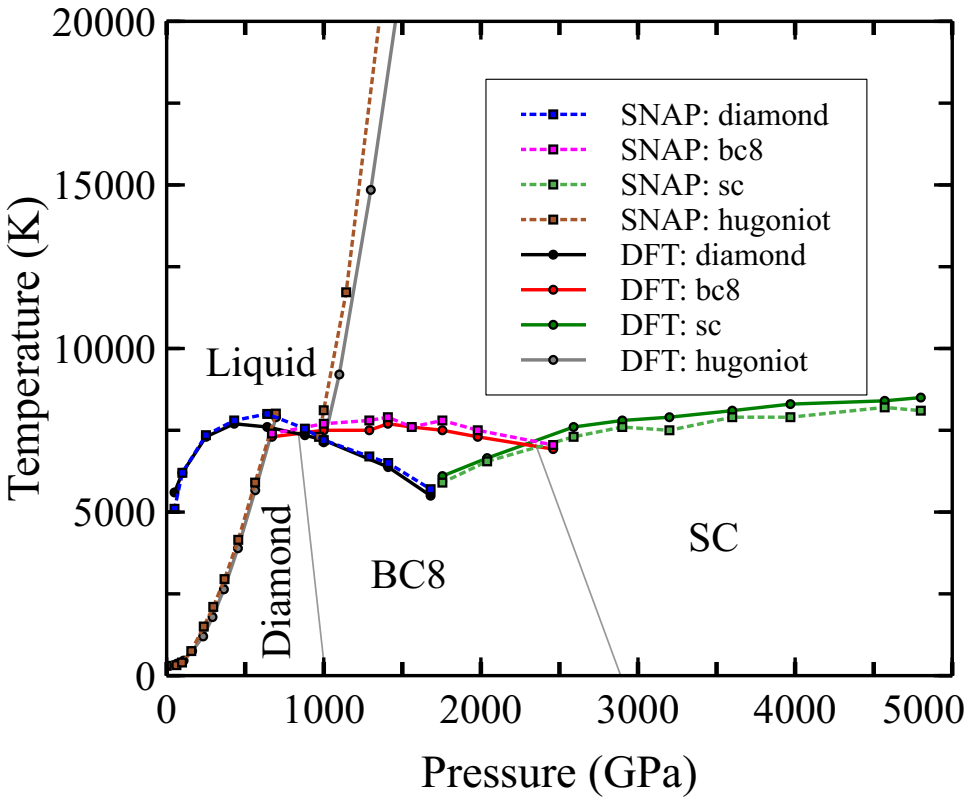
Wood, **Oleynik**, Willman, Nguyen, U. of South Florida

Training Data

Multiple phases of Carbon from 0-4TPa, 0-15,000K



Matches DFT melt curves (two-phase method) for cubic diamond, BC8, and simple cubic phases up to 5 TPa

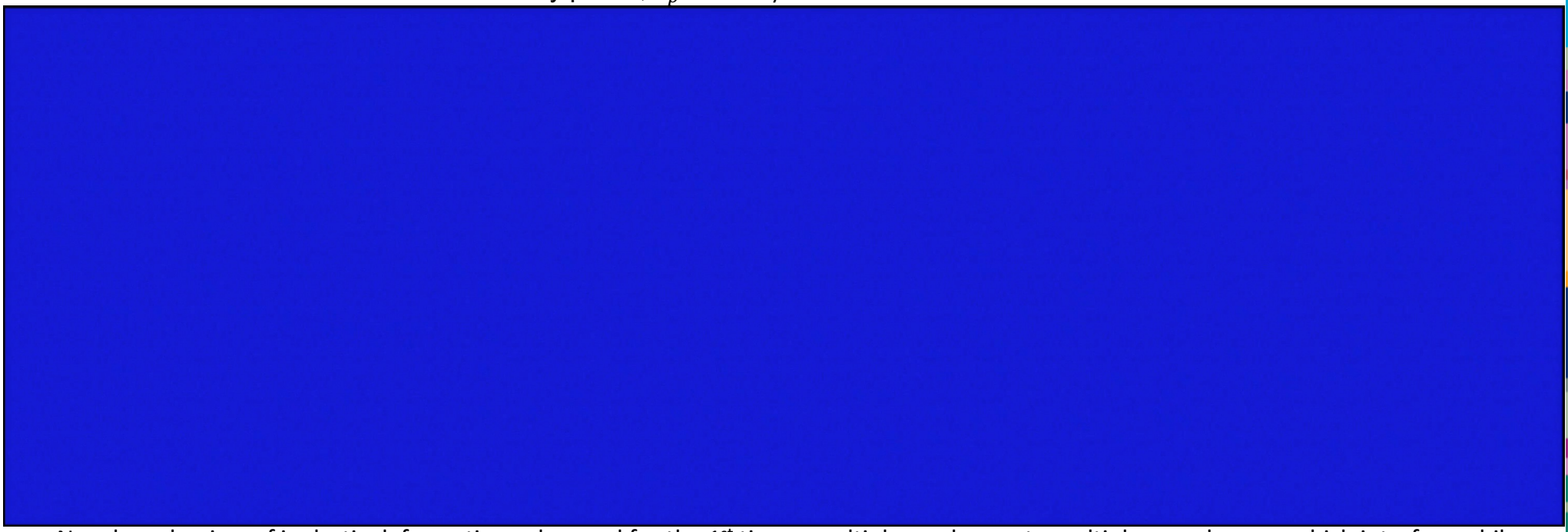


Accuracy and Scale

doi.org/10.1145/3458817.3487400

Oleynik group, U. of South Florida

- 2.6 billion atom diamond sample, 0.5×1.5 μm
- Shock wave in <110> direction initiated by piston, $v_p = 7\ km/s$.

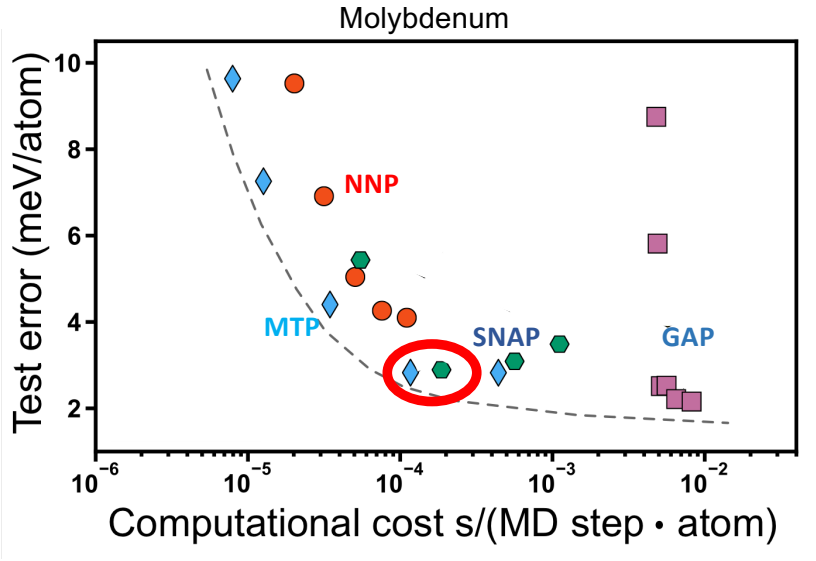


- Novel mechanism of inelastic deformations observed for the 1st time – multiple cracks create multiple sound waves which interfere while propagating towards the elastic front

How Does SNAP Compare?

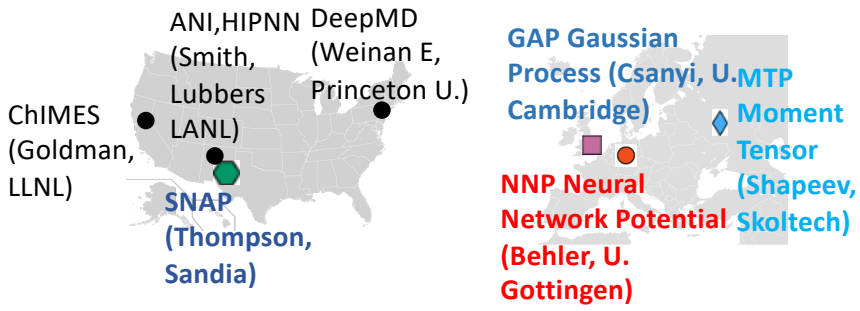
"Performance and Cost Assessment of Machine Learning Interatomic Potentials" Zuo, Chen, Li, Deng, Chen, Behler, Csányi, Shapeev, Thompson, Wood, and Ong. J.Phys.Chem A. 2020.

SNAP provides a good tradeoff between accuracy and performance



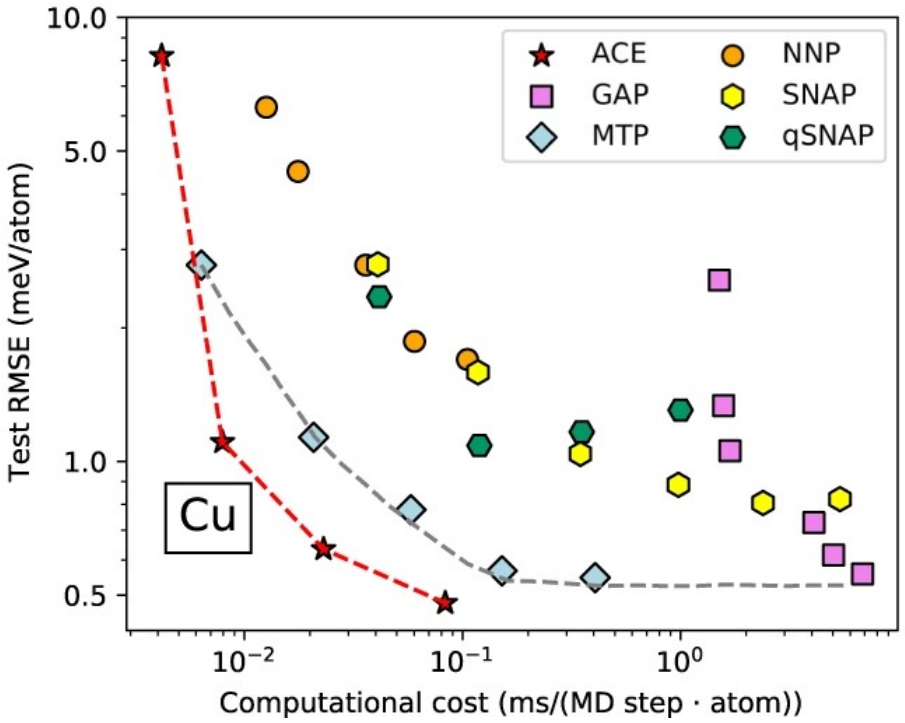
- SNAP is competitive with the best approaches world wide
- In a 2020 independent study of 4 leading approaches (left), quadratic SNAP achieved good cost/accuracy balance on all 6 elements
- **Also showed best stability in extrapolation**
- We continue to evolve the training algorithms for more robust prediction
- Algorithmic improvements have greatly boosted performance on pre-exascale hardware

World Map of Leading ML Potentials

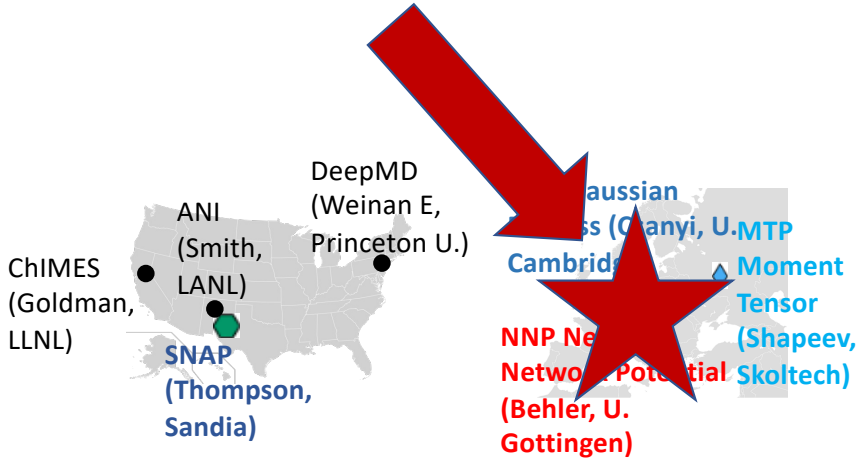


Maybe ACE is Better?

"Performant implementation of the atomic cluster expansion (PACE)," Lysogorskiy et al., npj Comput. Mater. 7 (2021) 97



Atomic Cluster Expansion (Ralf Drautz, ICAMS, Germany)



New Descriptor: Atomic Cluster Expansion (ACE)

Atomic Cluster Expansion (ACE) 2,3,...,N-body **irreducible** scalar invariants

Drautz, Phys.Rev.B, 2019
Willatt, Musil, Ceriotti, J.Chem.Phys. 2019
Seko, Togo, Tanaka, Phys.Rev.B 99, 2019

- Local environment expanded in atomic basis
- Generates very general set of 2, 3, ...N-body irreducible scalar invariants
- Superset of many previous descriptors (SNAP, GAP, MTP, BP)
- For example, SNAP bispectrum components can be expressed in the ACE form
- Available in LAMMPS as the ML-PACE package
- Ported to KOKKOS package for GPUs (Stan Moore, Sandia)
- Prototype training in FitSNAP (James Goff, Sandia)

$$A_{i\mu nlm} = \sum_j R_{nl}^{\mu_j \mu_i}(r_{ji}) Y_{lm}(\hat{\boldsymbol{r}}_{ji})$$

Radial basis

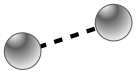
Angular basis

$$\boldsymbol{B}^{(N)}_{\mu_i \boldsymbol{\mu nl L} L_R} = \begin{pmatrix} \boldsymbol{l} \\ \boldsymbol{L} \end{pmatrix} L_R \Big)_N \prod_{k=1}^{N} \boldsymbol{A}_{i\mu_k n_k l_k}$$

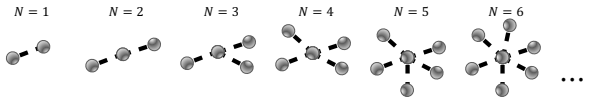
ACE models: powerful but challenging to define analytically
Current basis of ACE descriptors is overcomplete

Powerful descriptors, but with linear dependence

Single bond basis



Form a complete, orthogonal N-bond tensor product basis

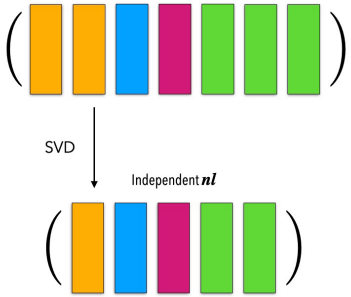


Impose invariance w.r.t. rotations and permutations

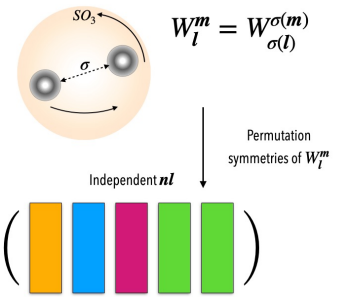
Use symmetry arguments to select unique descriptors *a priori*

Imposing invariance w.r.t. rotations and permutations introduces linear dependencies

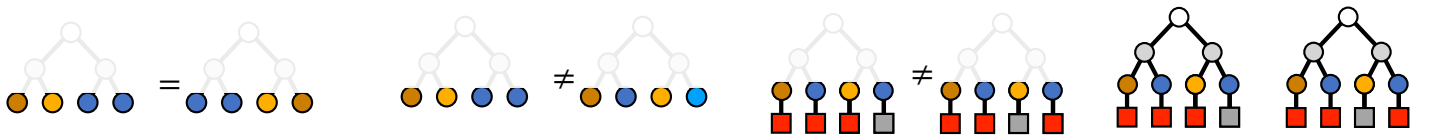
Semi-numerical methods



Permutation-adapted method



Unique descriptor labels, nl
Using symmetries of binary trees



Find equivalent $\boldsymbol{l}$ trees	Find unique $\boldsymbol{l}$ trees	Find unique $\boldsymbol{nl}$ trees	Apply valid internal nodes
$G_N = syl_2(S_N) + \cdots$ Grow group of automorphisms (equivalent tree permutations)	Symmetric orbits of unique $\boldsymbol{l}$ tree - encoded into a Young Tableau	Sort n_i, l_i tuples in the same symmetric orbits	Iterated triangle conditions $\Delta\ (l_i, l_j, L_k)$ Parity conditions

Results and demonstration

Descriptor labels for a single element system

- Exhaustive descriptor counts for $n_{max} = 4$, $l_{max} = 4$
- Significant reduction in the number of basis functions to evaluate compared to numerical methods
- Symmetry reduction becomes more significant at larger ranks (scales with the size of G_N)

Descriptor counts

N	$deg./N$	# All S_N	lexico.	# PA-RPI	lex/PA-RPI
4	2	3	3	1	3.0
	3	270	76	42	1.8
	4	2330	786	605	1.3
	8	19712	7088	4308	1.6
5	2	6	6	2	3.0
	3	1111	338	152	2.2
	4	28960	7907	3665	2.2
	8	505856	120172	35768	3.4
6	2	15	15	1	15.0
	3	19882	2241	330	6.8
	4	577578	85692	14512	5.9
	8	15704064	1993592	175800	11.3

For many practical ACE applications, this provides a significant reduction in computational cost

MLIAPs Available in LAMMPS

Native LAMMPS

- **ML-SNAP**

LAMMPS Interfaces

- **ML-HDNNP**: Singraber, N2P2, Behler-Parrinello Descriptors, ANN Potentials
- **ML-QUIP**: Bartok, Csanyi, GAP Potentials, SOAP Descriptors
- **ML-PACE**: Lysogorskiy, Drautz, Atomic Cluster Expansion
- **ML-RANN**: Dickel, NN potential with fast fingerprints
- **KIM**: Tadmor, many ML potentials: DUNN, hNN, PANNA
- **USER-DEEPMD**: Zhang, E, Car, Deep Network Potentials
- **USER-MLIP**: Shapeev, Moment Tensor Potentials
- **USER-MLIP**: Seko, Machine Learning Potential Repository
- **USER-PINN**: Mishin, Physically informed neural network potential
- **USER-ANI**: Barros, Smith, Lubbers, ANI ANN Potentials
- **USER-AENET**: Artrith, Behler-Parrinello Descriptors, ANN Potentials
- **FLARE++:** Sparse Gaussian process, ACE

:

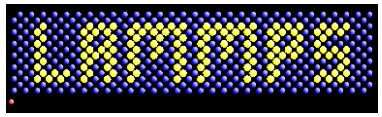
:

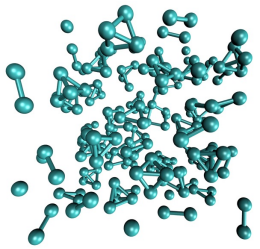
LAMMPS MLIAP Interface: Overview

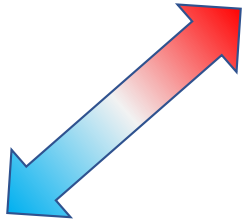
MLIAP Approach

- Provide a common API for many methods
- Decompose ML potential into three independent objects:

  `Descriptor Model Data`
- `Descriptor` generates local fingerprint for each atom
- `Model` computes energy as function of descriptors
- `Data` **handles LAMMPS interface**
- `Data` also handles intermediate quantities e.g. gradients
- `Descriptor` and `Model` insulated from LAMMPs and each other
- **Allows mix-and-matching of Models and Descriptors**





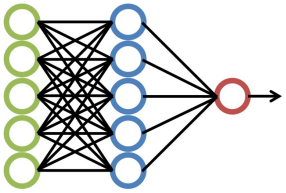


Pair MLIAP

Descriptor

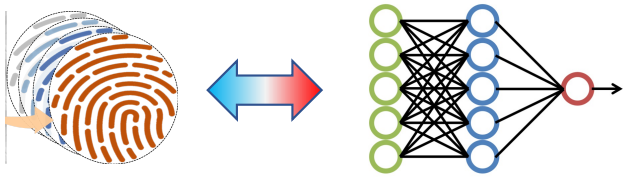
Inputs	***Outputs***
Atoms	*Energy*
Elements	*Force*
Neighbors	*Stress*

Data



Model

LAMMPS MLIAP Interface:



Completed

- All SNAP variants
- Python and PyTorch models now enabled
- SO(3) Descriptors (Qiang Zhu, UNLV)
- Native NN Energy Model (Qiang Zhu, UNLV)
- PyTorch models
- Python models
- Entropy maximization (Danny Perez, LANL)

Force (Running MLIAP Simulation)

```
descs = Desc.getDescs(atoms)
modelGrads = Model.getGrads(descs)
forces = Desc.getForces(modelGrads)
```

https://lammps.sandia.gov/doc/pair_mliap.html

Coming Soon!! (LAMMPS + FitSNAP)

- ACE Descriptors
- Non-linear models (PyTorch)
- HIPPYNN (Lubbers, LANL)

Force Gradient (Training MLIAP)

$$\sum_{j}^{N_{atoms}} \sum_{k}^{K_{components}} \frac{\partial D_{j,k}}{\partial \boldsymbol{r}_i} \frac{\partial^2 f(D_{j,k})}{\partial D_{j,k} \partial \theta} = \frac{\partial F_i}{\partial \theta}$$

Algorithm 1: N_{PARAMS} x $N_{DESCRIPTORS}$

```
descs = Desc.getDescs(atoms)
gradGrads = Model.getGradGrads(descs)
forceGrads = Desc.getForceGrads(gradGrads)
```

$$\frac{\partial^2 f(D_{j,k})}{\partial D_{j,k} \partial \theta}$$

Algorithm 2: N_{NEIGHS} x $N_{DESCRIPTORS}$

```
descGrads = Desc.getDescGrads()
forceGrads = Model.getForceGrads(descGrads)
```

$$\frac{\partial D_{j,k}}{\partial \boldsymbol{r}_i}$$

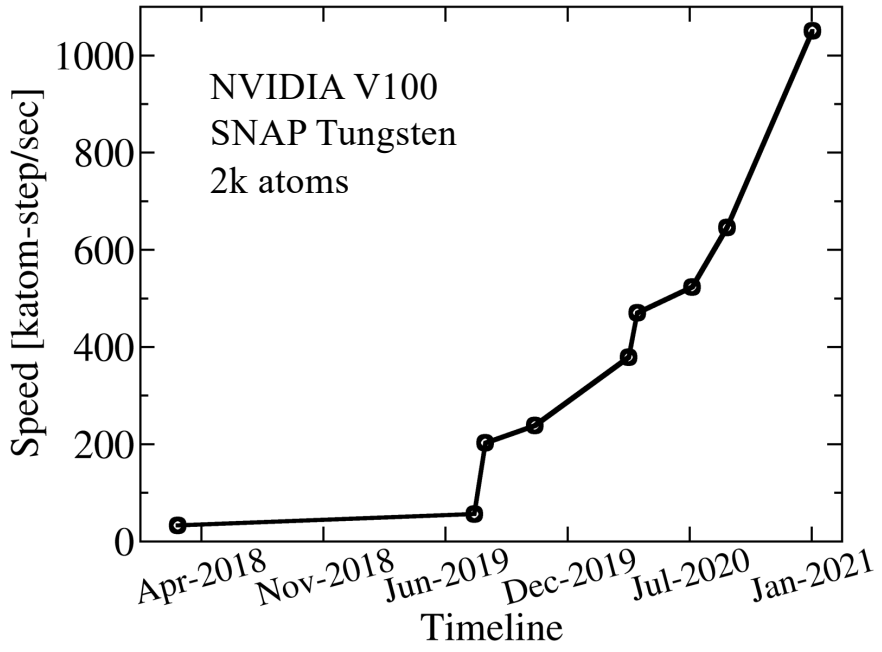
Algorithm 3: Autodiff on Loss Function

SNAP GPU Performance

GPU Performance Optimization

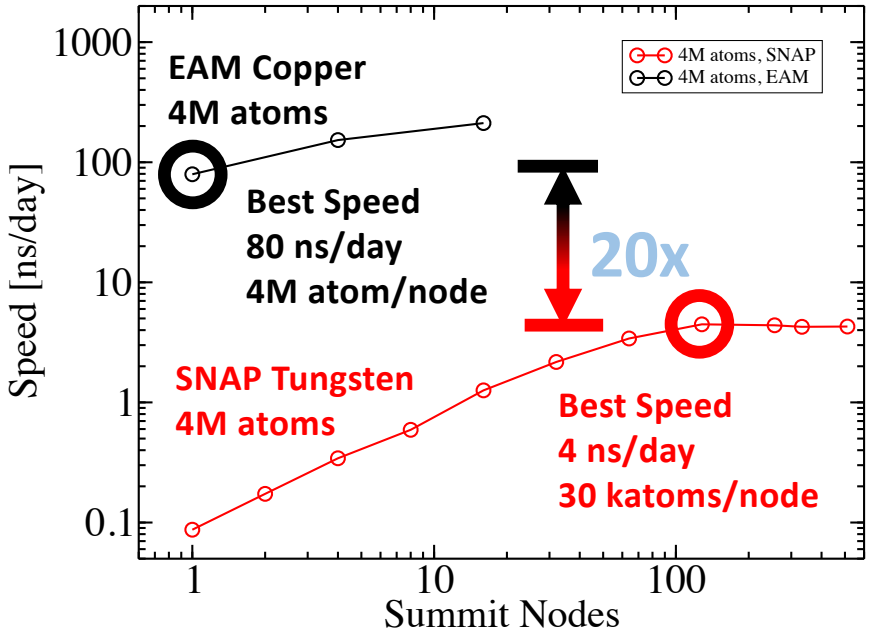
- Highly collaborative effort involving: Sandia, LANL, NERSC, NVIDIA, several hackathons and a lot of experimentation
- Created stripped-down proxy code (TestSNAP)
- Completely rewrote TestSNAP to reduce flops and memory
- Explored many different GPU strategies, using OpenACC, CUDA, and Kokkos
- break up the force kernel into sub-kernels and pushing atom/neighbor parallelism into the sub-kernels
- Ported best implementation back to production code with Kokkos
- Further improvements in memory access

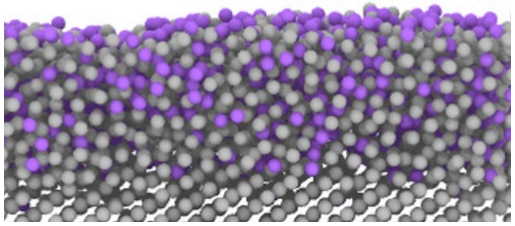
- GPU Performance Timeline
  Gayatri, Moore, Weinberg et al. (2020)
  https://arxiv.org/abs/2011.12875
- ~50x improvement over baseline



SNAP GPU Performance

- Excellent strong scaling on Summit
- Leadership DOE Computing Platform (ORNL)
- 4608 nodes, 6 NVIDIA v100s/node, 200 petaFlops
- Comparison of EAM and SNAP Simulation Speed vs. Summit Node Count



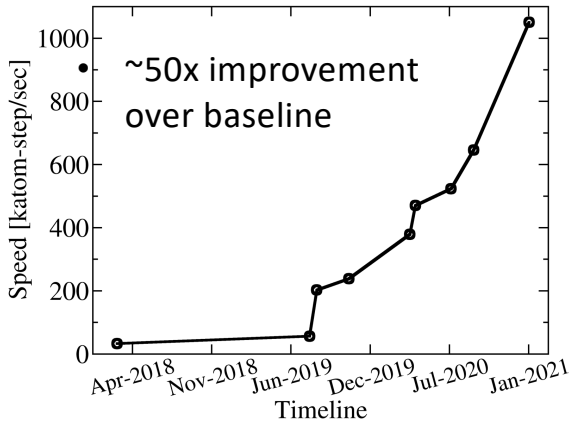


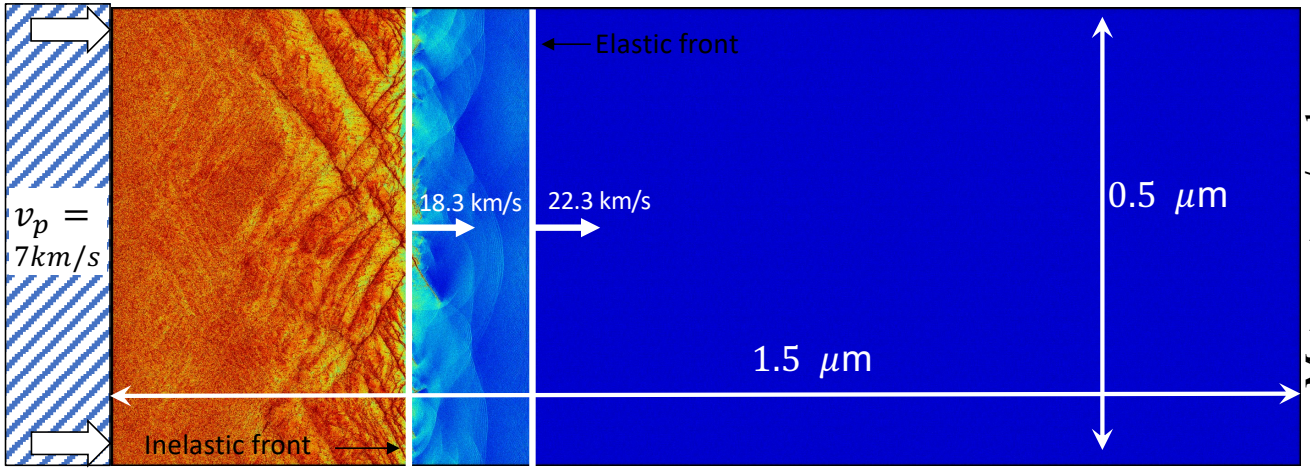
Exascale ML-MD

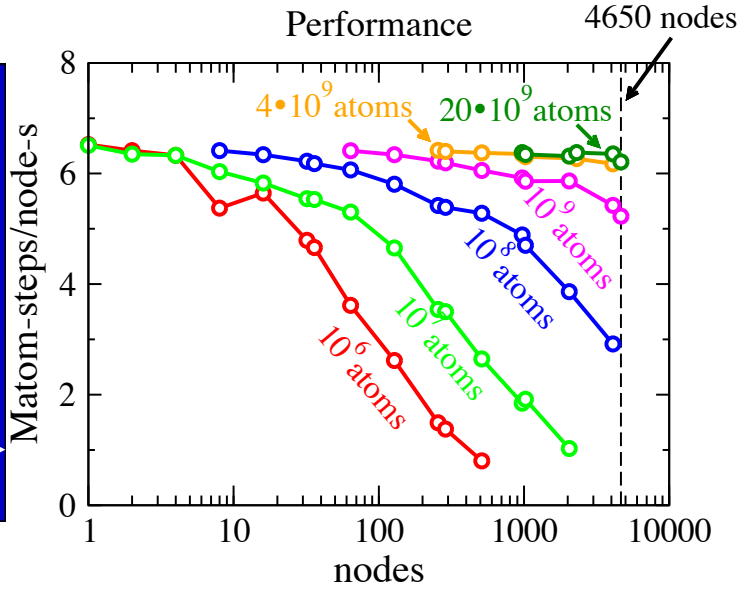
Willman, Moore, and many more

Gordon Bell Finalist

- Production simulations on entire Summit machine
- Simulation rate 25x greater than DeepMD
- Team from USF, Sandia, NERSC, NVIDIA, KTH : doi.org/10.1145/3458817.3487400







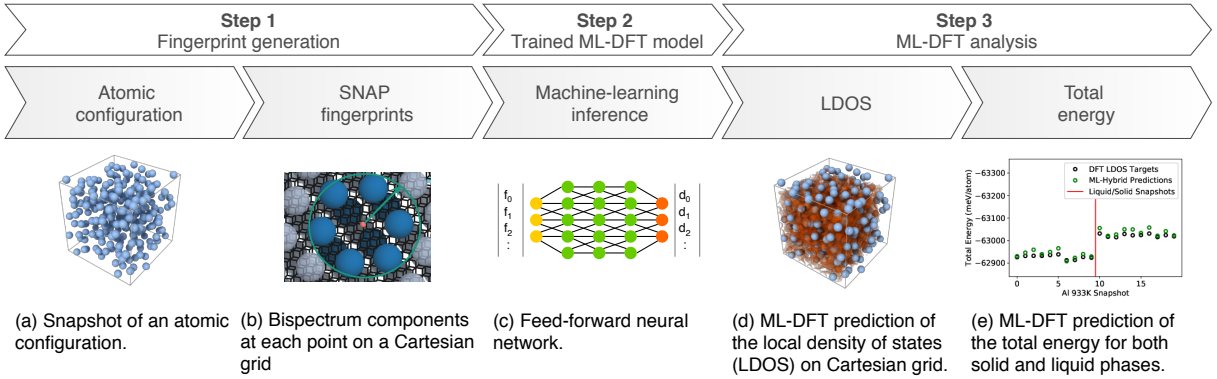
SNAP ML for LDOS, Electron Density, and Energy

Materials Learning Algorithms

QUANTUMESPRESSO

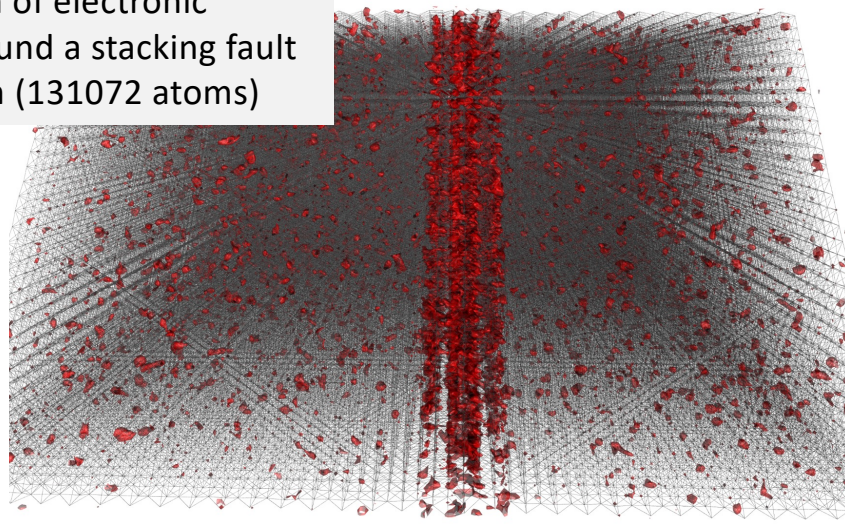
LAMMPS

CASUS
CENTER FOR ADVANCED SYSTEMS UNDERSTANDING



(a) Snapshot of an atomic configuration. (b) Bispectrum components at each point on a Cartesian grid (c) Feed-forward neural network. (d) ML-DFT prediction of the local density of states (LDOS) on Cartesian grid. (e) ML-DFT prediction of the total energy for both solid and liquid phases.

Localization of electronic density around a stacking fault in beryllium (131072 atoms)



- https://github.com/mala-project/mala
- Ellis et al. “Accelerating finite-temperature kohn-sham density functional theory with deep neural networks.” Phys. Rev. B (2021)
- Highly parallelizable
- Coming soon! Force inference in LAMMPS

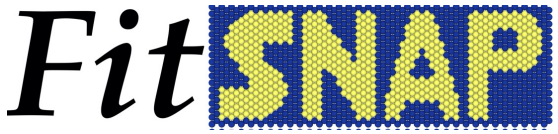
Summary

- Machine-learning is revolutionizing atomistic simulation
- It has enabled the best of both worlds: quantum accuracy and classical scale
- Many challenges remain:

  Robustness: 1-in-a-billion bad force predictions can ruin an entire simulation

  On-the-fly accuracy estimate: hard, because no QM query on large-scale

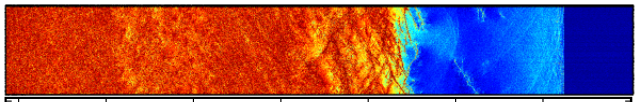
  Active learning: smart training data generation



https://github.com/FitSNAP/FitSNAP

Acknowledgements

FitSNAP



LAMMPS

Steve Plimpton

Axel Kohlmeyer (Temple U.)

- Stan Moore
- Rahul Gayatri (NERSC)
- Evan Weinberg (NVIDIA)

SNL

- Mitch Wood
- Mary Alice Cusentino
- Svetoslav Nikolov
- Habib Najm
- Khachik Sargsyan
- David Montes de Oca Zapiain
- Ember Sikorski
- Megan McCarthy
- James Goff
- Logan Williams
- Drew Rohskopf

USF

- Ivan Oleynik
- Jon Willman
- Kien Nguyen

MALA

Materials Learning Algorithms

- Siva Rajamanickam
- Normand Modine
- Attila Cangi (CASUS)
- Lenz Fiedler (CASUS)