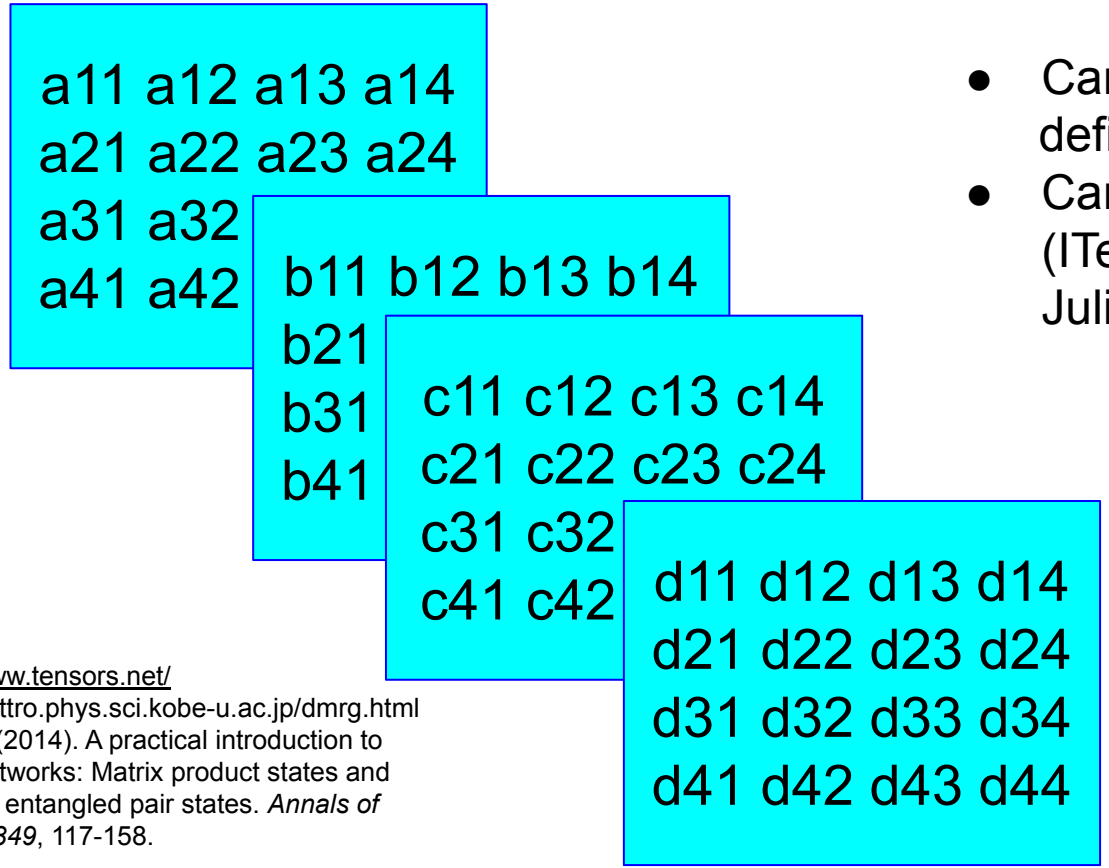
Tensor Networks as a tool to improve LAMMPS simulation potentials

Daniel Castillo Castro
Núcleo de Física, Matemática y Estadística (U. Mayor)
QUDIT, Facultad de Física (UC Chile)
dacastillo@uc.cl

Tensors: Generalization of scalars and vectors

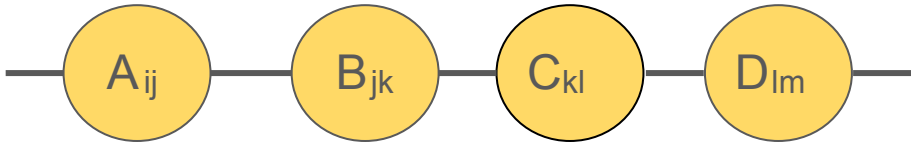


- Can simplify some concepts definition.
- Can be coded using C++ (ITensor), Python (TenPy), Julia (TensorOperations)...

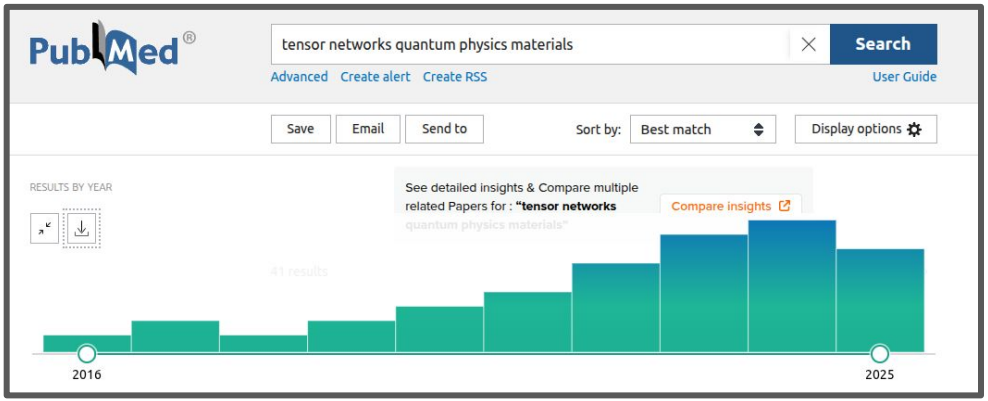
- https://www.tensors.net/
- http://quattro.phys.sci.kobe-u.ac.jp/dmrg.html
- Orús, R. (2014). A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of physics*, *349*, 117-158.

Where can be applied tensors in Physics

- Succinct representation of physical quantities (ex: Stress Tensor, Moment of Inertia, Stress/energy tensor).
- Representation of frameworks of space and time (ex: General Relativity)



Tensor networks: Simplification of matrix products.



Recent interest of the use of tensor networks for studying materials.

Montangero, S., Montangero, E., & Evenson. (2018). *Introduction to tensor network methods* (p. 107). Switzerland, Cham: Springer International Publishing.

Example: Visualization and methods for Quantum Mechanics

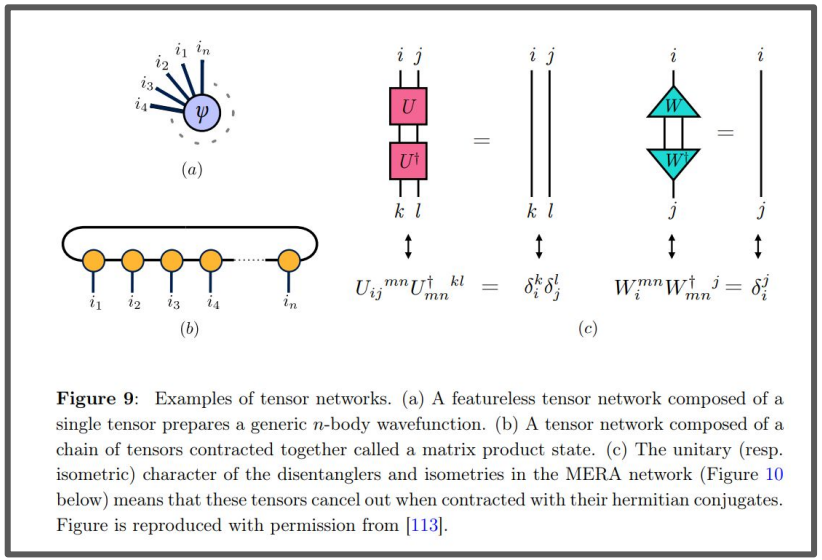
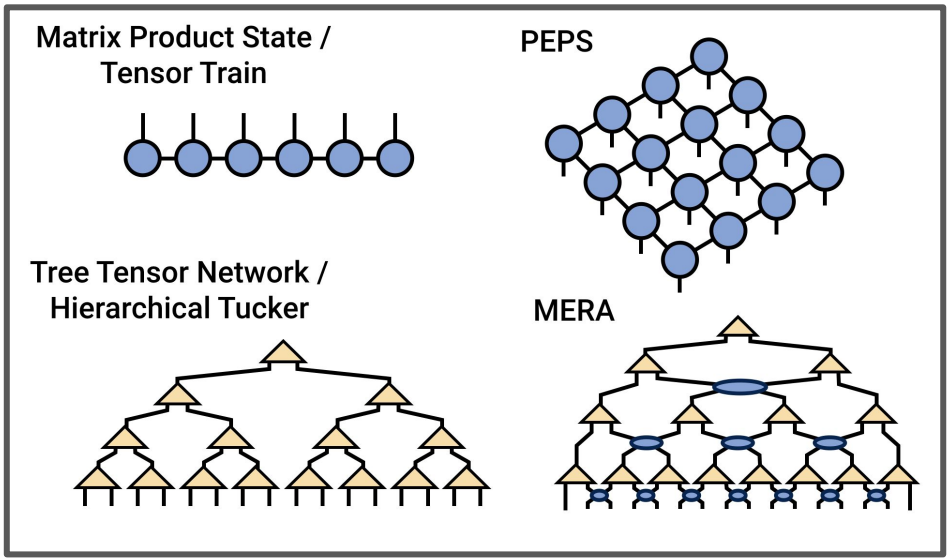


Figure 9: Examples of tensor networks. (a) A featureless tensor network composed of a single tensor prepares a generic n-body wavefunction. (b) A tensor network composed of a chain of tensors contracted together called a matrix product state. (c) The unitary (resp. isometric) character of the disentanglers and isometries in the MERA network (Figure 10 below) means that these tensors cancel out when contracted with their hermitian conjugates. Figure is reproduced with permission from [113].

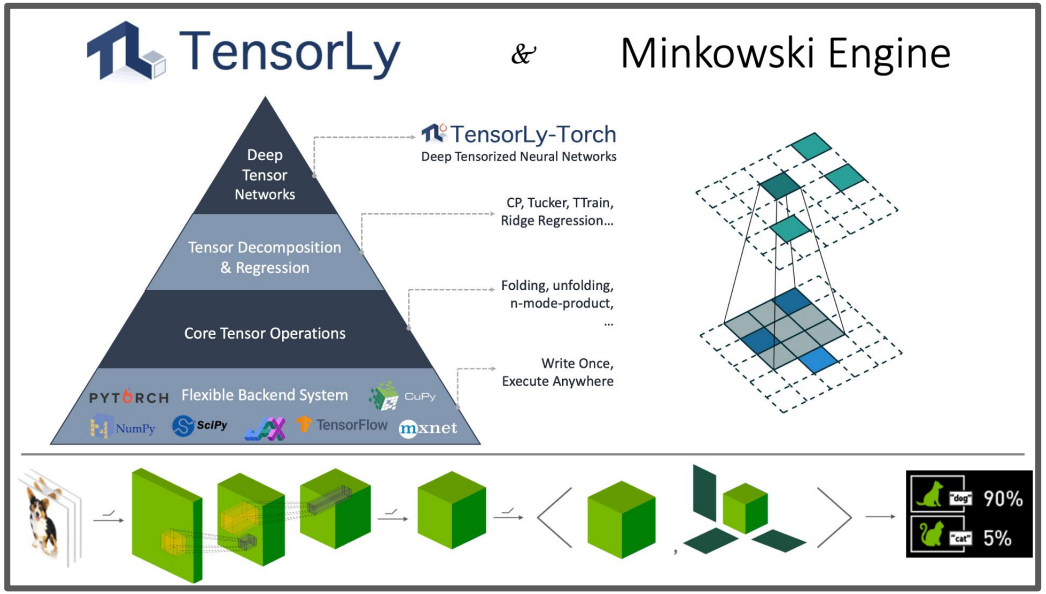


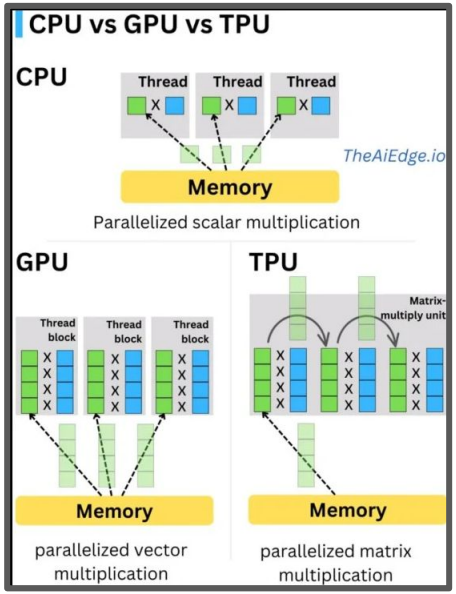
Can these tools be an adequate tool to improve molecular simulations?

- Bowen, Chen *et al* (2022). Quantum Information in Holographic Duality *Rep. Prog. Phys.* 85 046001
- Berezutskii, A., Liu, M., Acharya, A., Ellerbrock, R., Gray, J., Haghshenas, R. & Alexeev, Y. (2025). Tensor networks for quantum computing. *Nature Reviews Physics*, 1-13.

Tensors are useful as computing framework too!

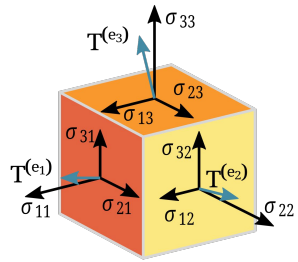
- Codes could be optimized by TN: faster and cheaper calculations.

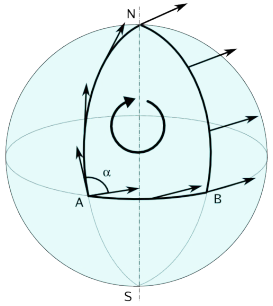


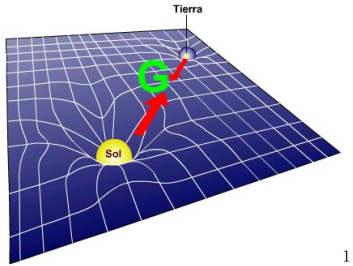


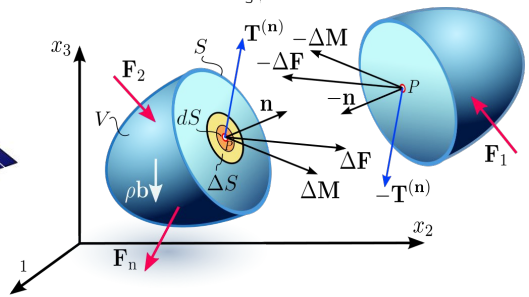
- Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. *SIAM review*, *51*(3), 455-500.
- Zhao-Yu Han, Jun Wang, Heng Fan, Lei Wang, Pan Zhang (2018) *Unsupervised Generative Modeling Using Matrix Product States*, *Phys. Rev. X* **8**, 031012
- https://developer.nvidia.com/blog/nvidia-research-tensors-are-the-future-of-deep-learning/

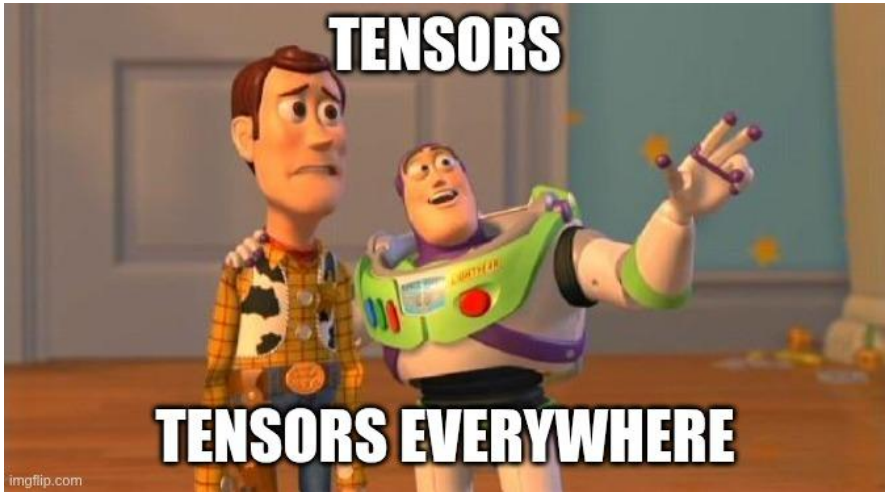
Tensor Networks (TN): An ultrapowerful tool.

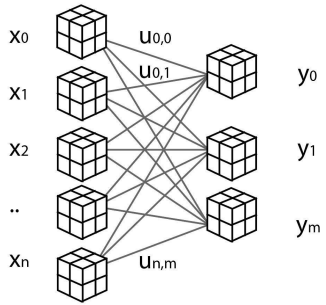


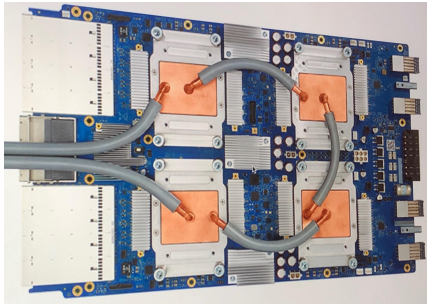






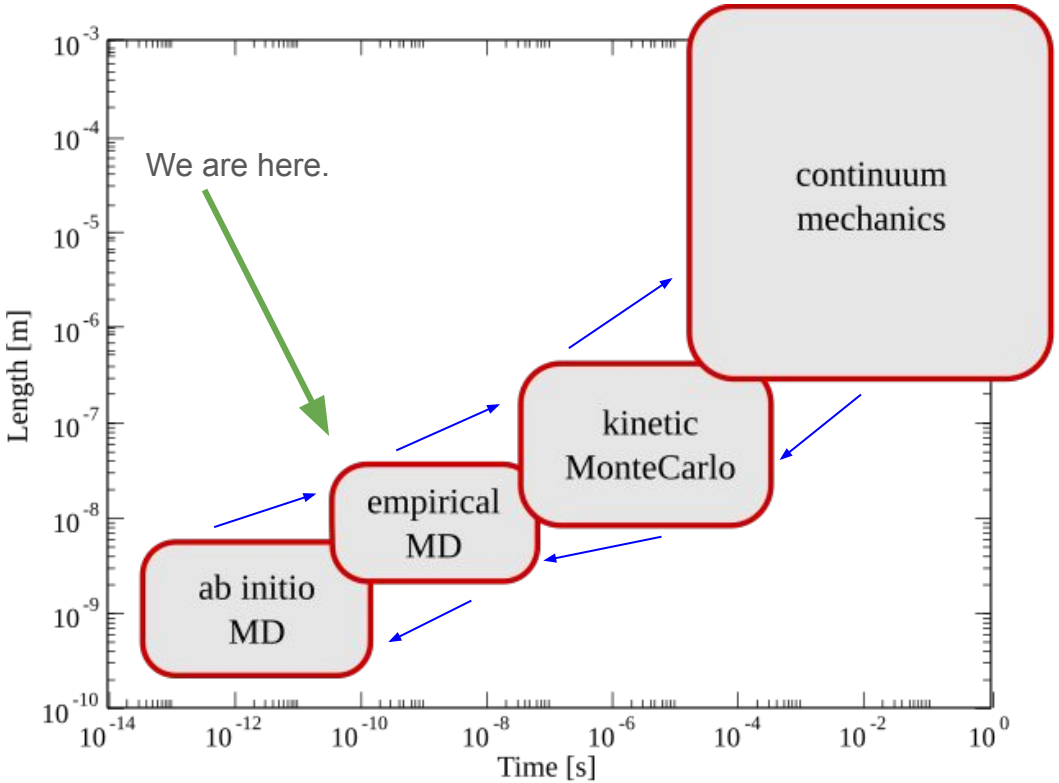






- Tensors in different branches of physics.
- Tensors in machine learning
- Tensors for Quantum Simulation?

Quantum simulation of materials: scaling as a challenge.



Phonon-induced non-equilibrium dynamics in a single solid-state spin A Norambuena, D Tancara, VF Chomali Castro, D Castillo-Castro Quantum Science and Technology		2025
Plasticity in diamond nanoparticles: dislocations and amorphization during loading and dislocation multiplication during unloading F Aquistapace, D Castillo-Castro, RI González, N Amigo, ... Journal of Materials Science 59 (12), 4788-4809	14	2024
Nanoporous amorphous carbon with exceptional ultra-high strength D Castillo-Castro, F Correa, E Aparicio, N Amigo, A Prada, J Figueroa, ... Nanomaterials 13 (8), 1429	9	2023

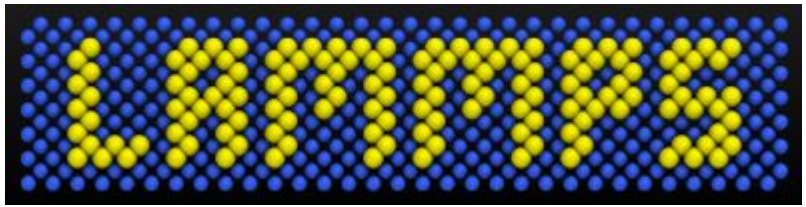
After 5 years approx. in MD for different materials:

- How to scale from one formalism to another?
- Can Tensors contribute to the results exchange?

Proposal: Tensor Networks+Empirical potentials

- DMRG for few atoms (< 100)
- Ground state energy calculations

- Use of results for build potential
- Application of machine learning



- Testing of result for more atoms
- Testing: E(r), RDF, stress/strain…

block2

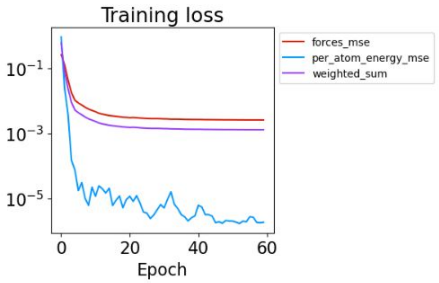
DMRG Calculations

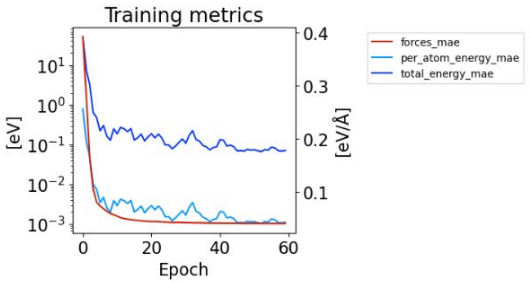
```
!rm sitraj.xyz
atoms_Si=""
#file_object = open('sitraj_new.xyz', 'w')
with open('sitraj.xyz', 'w') as file_object:
    for n1 in range(len(X)):
        fcc_atoms = generate_fcc_lattice(X[n1], 1, 1, 1)
        atoms_Si=""
        for j in range(len(fcc_atoms)):
            atoms_Si +=f" Si {fcc_atoms[j][0]} {fcc_atoms[j][1]} {fcc_atoms[j][2]}"
        # Define the molecular system
        mol = gto.M(atom=atoms_Si, basis='ccpvtz')
        mf = scf.RHF(mol).run(conv_tol=1E-14)
        # Get the necessary integrals for DMRG
        ncas, n_elec, spin, ecore, h1e, g2e, orb_sym = itg.get_rhf_integrals(mf,ncore=2, ncas=26, g2e_symm=8)
        driver = DMRGDriver(scratch="./Si_tmp")
        driver.initialize_system(n_sites=ncas, n_elec=n_elec, spin=spin, orb_sym=orb_sym)
        mpo = driver.get_qc_mpo(h1e=h1e, g2e=g2e, ecore=ecore, iprint=0)
        ket = driver.get_random_mps(tag="KET", bond_dim=100, nroots=1)
        bond_dims = [100] * 4 + [200] * 4 + [300] * 4 + [400] * 4
        noises = [1e-2] * 4 + [1e-3] * 12 + [0]
        thrds = [1e-6] * 10
        energy = driver.dmrg(mpo, ket, n_sweeps=28, bond_dims=bond_dims, noises=noises,
        thrds=thrds, iprint=1, twosite_to_onesite=20)
        #print('DMRG energy (variational) = %20.15f' % energy)
        print(len(fcc_atoms), file=file_object)
        the_array = 'Lattice="12 0.0 0.0 0.0 12 0.0 0.0 0.0 12" Properties=species:S:1:pos:R:3:forces:R:3 energy='+str(energy)
        print(the_array, file=file_object)
        for j in range(len(fcc_atoms)):
            print(f"Si {fcc_atoms[j][0]} {fcc_atoms[j][1]} {fcc_atoms[j][2]} {fx} {fy} {fz} ", file=file_object)
```

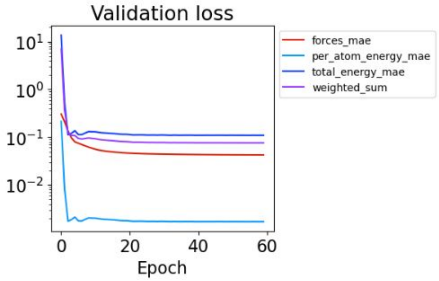
Potential building

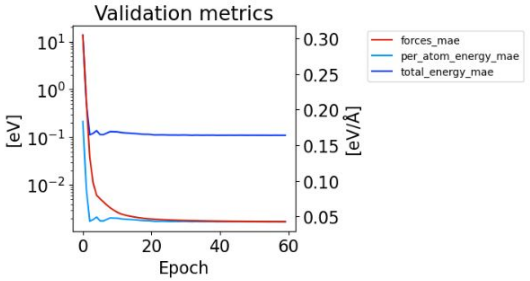
```
## TRAIN NEQUIP ##
!rm -rf ./results
!mkdir ./results
!nequip-train -cn config.yaml
```

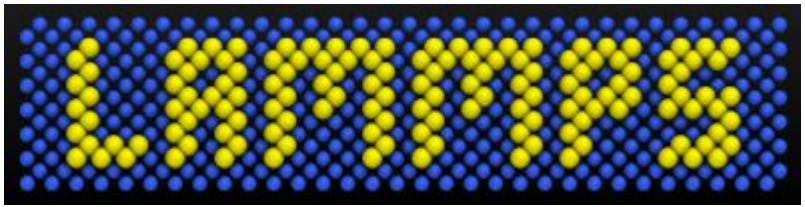
```
# ===========
#     RUN
# ===========
run: [train, test]   #Define train and test subsets
cutoff_radius: 5.0
chemical_symbols: [Si]
model_type_names: ${chemical_symbols}


# ============
#     DATA
# ============
data:
  _target_: nequip.data.datamodule.ASEDataModule
  seed: 456              # dataset seed for reproducibility
  split_dataset:
    file_path: ./sitraj.xyz
    train: 0.8
    val: 0.1
    test: 0.1
  transforms:
    # data doesn't usually come with a neighborlist -- this tranforms prepares the neighborlist
    - _target_: nequip.data.transforms.NeighborListTransform
      r_max: ${cutoff_radius}
    - _target_: nequip.data.transforms.ChemicalSpeciesToAtomTypeMapper
      chemical_symbols: ${chemical_symbols}
  train_dataloader:
    _target_: torch.utils.data.DataLoader
    batch_size: 5
    num_workers: 5
    shuffle: true
  val_dataloader:
    _target_: torch.utils.data.DataLoader
    batch_size: 10
    num_workers: ${data.train_dataloader.num_workers}  # we want to use the same num_workers -- variable interpolation helps
  test_dataloader: ${data.val_dataloader}  # variable interpolation comes in handy again
```









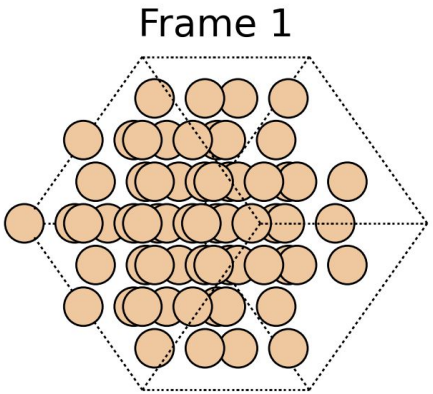


DM testing of potentials

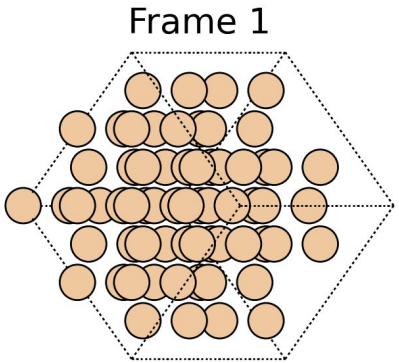
```
# @title Prepare files and run LAMMPS
from ase.io import read, write
example_atoms = read('./sitraj_old.xyz', index=0)
write('./si.data', example_atoms, format='lammps-data')

lammps_input = """
units   metal
atom_style atomic
dimension 3

# set newton on for pair_allegro (off for pair_nequip)
newton off
boundary p p p
read_data ../si.data

# if you want to run a larger system, simply replicate the system in space
#replicate 2 2 2

# allegro pair style
pair_style  nequip
pair_coeff  * * ../deployed.nequip.pth Si

mass 1 28.0855

velocity all create 300.0 1234567 loop geom

neighbor 1.0 bin
neigh_modify delay 5 every 1

timestep 0.001
thermo 10

dump dump_pos all custom 200 dump.xyz id type x y z

# nose-hoover thermostat, 300K
fix  1 all nvt temp 300 300 $(100*dt)

# compute rdf and average after some equilibration
comm_modify cutoff 7.0
compute rdfall all rdf 1000 cutoff 5.0
fix 2 all ave/time 1 2500 5000 c_rdfall[*] file si.rdf mode vector

# run 5ps
run 5000
"""
!rm -rf ./lammps_run
!mkdir lammps_run
with open("lammps_run/si_rdf.in", "w") as f:
    f.write(lammps_input)

!cd lammps_run/ && lammps_nequip -in si_rdf.in
```

Results:

- The results for Si atoms are not satisfactory… yet.
- It will be needed improvements on the DMRG algorithm, add new algorithms for forces calculations and considering geometry simplifications.



NequIP



NequIP+DMRG

Conclusion

- New methods could be used to make improvements on LAMMPS simulations, contributing with the connection with other languages (Quantum Chemistry, Tensor Networks…)
- To reach better results, it will be needed to explore another TN methods to calculate forces per atom and elaborate the code to favor scalability.
- My main project: Make these improvements to convert this work in a library related with LAMMPS package.

Thank You