Description of the Allegro Model Used in High-pressure CO-O_2 Machine-Learned Interatomic Potential

Reetam Paul,[1] Jonathan C. Crowhurst,[1] and Stanimir A. Bonev[1]

[1] *Physics Division, Lawrence Livermore National Laboratory, Livermore, CA 94550, USA*

1. DATASET & NEIGHBOUR CONSTRUCTION

The model is trained on 126 500 *ab-initio* frames obtained from isothermal Born–Oppenheimer molecular dynamics runs of 432-atom CO-O_2 at 300 K from ambient pressure through 100 GPa, as well as 632-atom two-phase simulations of recovered product at ambient conditions. Each snapshot is saved in an `.extxyz` file with energies (eV), per-atom forces (eV Å^{-1}) and full virial stresses (kbar). A single consolidated trajectory `raw_data/vasp_frames.extxyz` is referenced in the YAML. Frames are split deterministically: 90% for training and 10% for validation using `data.seed = 696969`.

A global cut-off $r_{\max} = 4.10$ Å is chosen from the first minimum of the radial-distribution function $g(r)$ of the trajectory. Neighbor lists are built once per epoch with `NeighborListTransform`. The choice of $r_{\max} = 4.10$ Å is considered apt because of the inclusion of compressed high-pressure snapshots in the AIMD trajectory, which leads to an adequate number of neighbors:

$$\text{Mean number of neighbors} \pm \text{std} : 22.00 \pm 1.41$$
$$\text{Min / Max number of neighbors} : \ 18\,/\,27.$$

Empirically, local ML interatomic potentials converge well when the mean first-shell coordination lies in the 12–18 range for covalent and ionic materials [1–3]. The histogram of forces, energies, and stresses of the training dataset can be seen in Figs. 1-3 below.

2. ALLEGRO ARCHITECTURE

Allegro [4] is an $E(3)$-equivariant message-passing network. Nodes represent atoms with scalar ($l = 0$) and tensor ($l > 0$) features; edges carry relative position vectors $\mathbf{r}_{ij}$. The model used here is kept intentionally small (two interaction layers) to minimize MD overhead but still exceeds a traditional NequIP (1x32) in accuracy.

TABLE I. MLIP training hyperparameters.

Hyper-parameter	Setting

Radial basis	`num_bessels: 8,`
	`polynomial_cutoff_p: 6`

Scalar (F_s)/tensor (F_t) width	`num_scalar_features: 64,`
	`num_tensor_features: 32`

Interaction layers	`num_layers: 2`
Maximum angular momentum	`l_max: 2`
Channel coupling	`tp_path_channel_coupling: true`
Parity equivariance	`parity: false`
Activation	`*_nonlinearity: silu` for scalar, Allegro, and read-out MLPs

Radial–chemical embedding. For each edge, the radial kernel is $g(r_{ij}) = \sum_{n=1}^{8} c_n\, j_0\big(z_n r_{ij}/r_{\max}\big)\, f_{\text{cut}}(r_{ij})$, i.e. the distance r_{ij} between atoms i and j is expanded in a set of 8 Bessel basis functions $R_n\,(r_{ij})$ and multiplied by the polynomial cut-off $f_{\text{cut}}\,(r)$. The result is an 8 -dimensional *radial vector*. Thus, each neighbor carries

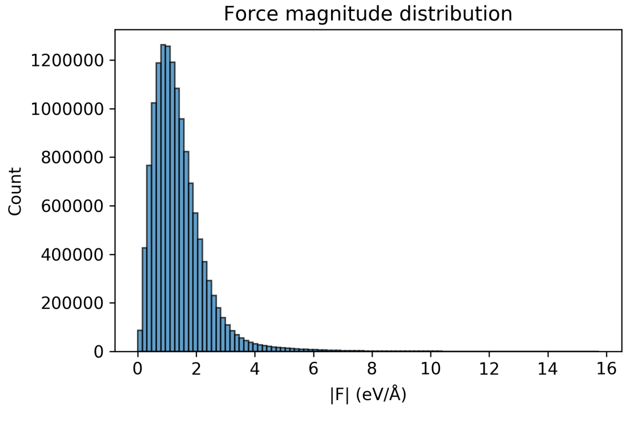

FIG. 1. Force histogram of training data.

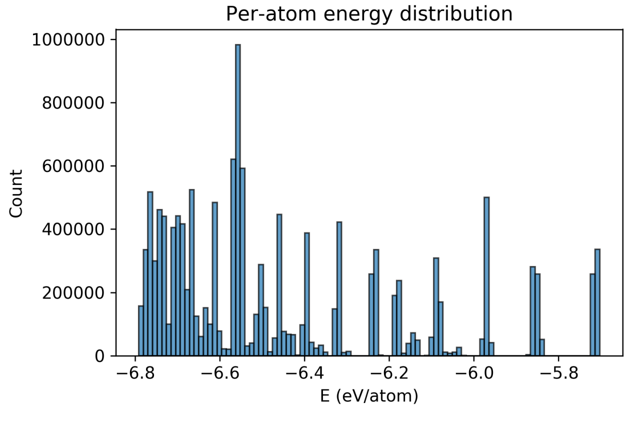

FIG. 2. Energy histogram of training data.

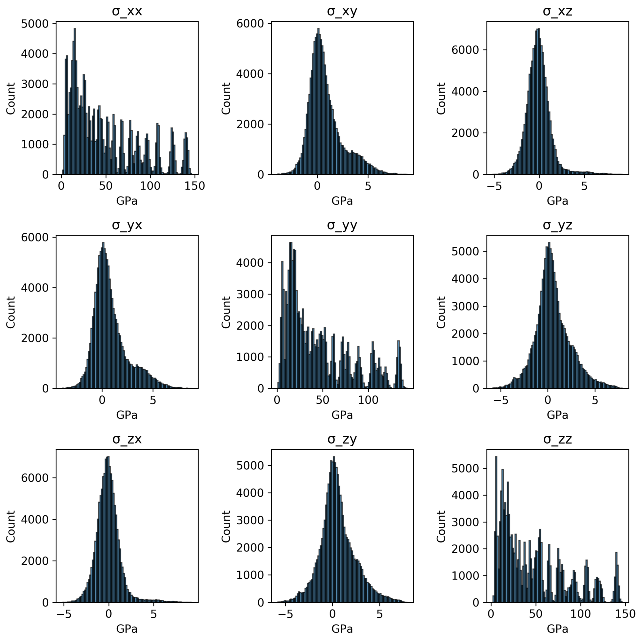

FIG. 3. Stress histogram of training data.

a 8-dimensional learnable embedding vector that identifies its chemical species. Such two pieces are pasted together, giving a 16-dimensional edge feature. A linear layer then projects that 16-vector into the model's scalar width $F_s = 64$, producing the edge scalar features used by the first Allegro interaction block.

Interaction layer. Each of the two Allegro blocks (`num_layers: 2`) first aggregates edge messages as $\mathbf{m}_i = \sum_{j\in\mathcal{N}(i)} \phi(\mathbf{r}_{ij})$, where the edge function ϕ is built from the 8 Bessel basis (`num_bessels: 8`) and cutoff polynomial of order 6 (`polynomial_cutoff_p: 6`) as listed in Table I. The aggregated vector is then combined with the atom's current features $\mathbf{h}_i \in \mathbb{R}^{F_s}$ ($F_s = 64$ is `num_scalar_features: 64`) through the $E(3)$-equivariant tensor-product convolution $\mathrm{TP}(\mathbf{h}_i, \mathbf{m}_i)$. Applying the global activation (SiLU, see Table I) followed by a channel mixing matrix $\mathbf{W}_s \in \mathbb{R}^{F_s\times F_s}$ yields

$$\mathbf{h}_i' = \mathbf{W}_s\, \sigma\big(\mathrm{TP}(\mathbf{h}_i, \mathbf{m}_i)\big).$$

Because `tp_path_channel_coupling` is enabled, scalar and tensor channels ($F_s = 64$ and $F_t = 32$) are interwoven inside the TP path, enlarging the receptive field without increasing the network depth.

Read-out. The final hidden vector of each atom ($\mathbf{h}_i^{(2)}$ from the second interaction layer) is *scored* by a single learned weight vector $\mathbf{w}_\varepsilon$, that maps 64-dimensional features to a single predicted site energy:

$$\varepsilon_i \;=\; \mathbf{w}_\varepsilon^{\mathsf{T}}\, \mathbf{h}_i^{(2)} \qquad \longrightarrow \qquad E_{\text{total}} \;=\; \sum_i \varepsilon_i.$$

In other words, a one-line dot product gives the *site energy* ε_i, and the total energy is the simple sum over all atoms. Forces are obtained automatically by analytical gradients,

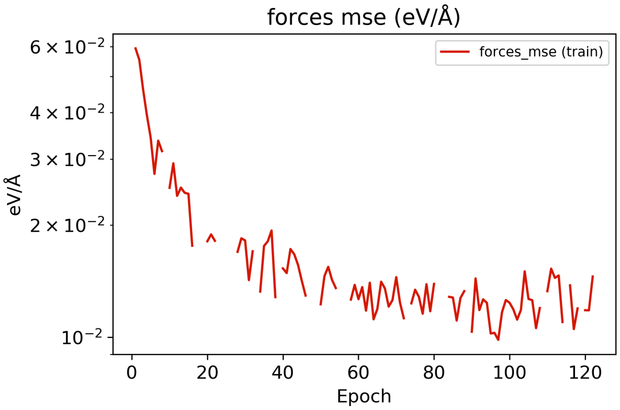

FIG. 4. Forces mean square error versus epoch.

$$\mathbf{F}_i \;=\; -\nabla_{\mathbf{R}_i} E_{\text{total}},$$

so no extra network branch is required.

3. LOSS, OPTIMISATION AND CONVERGENCE

The composite loss function for energy, forces, and stress errors is

$$\mathcal{L} = \mathcal{L}_E + w_F \mathcal{L}_F + w_\sigma \mathcal{L}_\sigma, \tag{1}$$

$$\mathcal{L}_E = \frac{1}{N}\sum_k (E_k^{\text{pred}} - E_k^{\text{DFT}})^2, \tag{2}$$

$$\mathcal{L}_F = \frac{1}{3N}\sum_{k,i} |F_{k,i}^{\text{pred}} - F_{k,i}^{\text{DFT}}|^2, \tag{3}$$

with $w_F = 10$ (`forces: 10.0`), $w_\sigma = 0$ (stresses excluded during training), N is the number of training configurations in a mini-batch. Thus, energies and forces are trained together, but forces are made ten times more important. Mean square error (MSE) curves (Fig. 4, Fig. 5) show smooth decrease, with training ending at epoch 122. This is because the loss function failed to improve by at least 1.0×10^{-4} (`min_delta: 1.0e-4`) for 30 epochs (`patience: 30`).

4. NORMALISATION & PHYSICAL PRIORS

Before the optimiser sees a single gradient, three small but critical pre-processing steps—each governed by a YAML key—make the loss well-behaved and the units consistent.

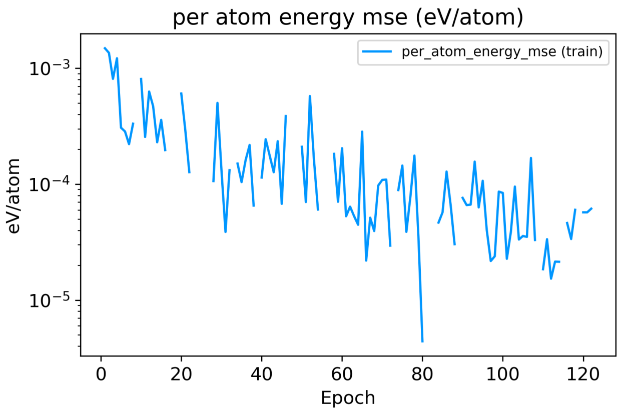

FIG. 5. Per atom energy mean square error versus epoch.

4.1 Neighbour Normalisation

With `neighbor_normalization: true` the raw site energy ε_i^* is divided by the *average* first-shell count,

$$\varepsilon_i = \frac{\varepsilon_i^*}{\langle n_{\mathrm{nbr}}\rangle}, \qquad \langle n_{\mathrm{nbr}}\rangle \approx 22.0.$$

This makes the model insensitive to the exact cut-off $r_{\max} = 4.10\,\text{Å}$: tighten or loosen $r_{\max}$ and the energy scale stays consistent.

4.2 Per-species Shifts

The YAML lists fixed offsets

```
per_species_offset:
  C:  +0.009431    # eV
  O:  -0.027169    # eV
trainable_per_type_shift: false
```

so every carbon gets a $+9.43\,\mathrm{meV}$ and every oxygen a $-27.17\,\mathrm{meV}$ shift. These are DFT atomic energies of one C or one O atom in a 15Å × 15Å × 15Å cell. These constants remove the large, element-specific baseline energy, letting the network focus on *bond-level* deviations. The flag `trainable_per_type_shift: false` freezes them during training.

4.3 Initial Loss Scaling

Because `compute_stats: true`, Allegro first measures the training-set force RMS and rescales the energy term so that at epoch 0 the two pieces of the loss

$$\mathcal{L}_E + w_F\,\mathcal{L}_F \quad \text{with} \quad w_F = 10$$

start on comparable magnitudes. Without this step the larger force errors would swamp the energy term until late in training.
Neighbour normalisation and per-species shifts act like unit-conversion constants: once measured from the dataset they never change, which speeds up convergence and guarantees the model remains reproducible if you retrain later.

5. DEPLOYMENT WORKFLOW

The model is compiled with

```
nequip-compile \
        --input-path   best.ckpt   \
        --output-path  pot.nequip.pth \
        --mode         torchscript \
        --device       cuda \
        --target       pair_allegro
```

producing a single `pot.nequip.pth` .

CONCLUSION

We share a LAMMPS-compatible Allegro MLIP (https://github.com/paulRqsg/COO2_MLIP) for high-pressure CO and O_2 mixtures. The two-layer network reaches $82.7\,\mathrm{meV}\,\text{Å}^{-1}$ MAE for forces and $5.39\,\mathrm{meV\,atom}^{-1}$ MAE for energies on an independent test set.

[1] S. Batzner, J. Musaelian, L. Sun *et al.*, E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials, *Nat. Commun.* **13**, 2453 (2022).

[2] P. Raj and R. Car, Optimal neighbor cutoffs for atom-centered neural network potentials, *J. Chem. Theory Comput.* **16**, 6254–6265 (2020).

[3] O. T. Unke, S. Chmiela, H. E. Sauceda *et al.*, Machine learning force fields, *Chem. Rev.* **121**, 10142–10186 (2021).

[4] S. Batzner *et al.*, *Nat. Commun.* **13**, 2453 (2022).