# Local Environment Matrix Construction in DeepMD-HALMD Integration

Sandip Kumar Sah

## Local Environment Matrix Construction

Once the neighbor list is built, the local atomic environments are encoded into a structured representation known as the **environment matrix** or **R-matrix**. This matrix serves as the direct input to the descriptor (embedding or filter network) in DeepMD version 2, where each central atom $i$ is represented by its set of neighboring atoms $j$ within a cutoff distance $r_c$.

### Mathematical Definition

For each atom $i$, the environment matrix $\mathbf{R}^{(i)}$ is constructed with one row per neighbor $j$. Each row contains the inverse distance and scaled direction components:

$$\mathbf{R}_j^{(i)} = \left[ \frac{1}{r_{ij}} \quad \frac{r_{ij,x}}{r_{ij}^2} \quad \frac{r_{ij,y}}{r_{ij}^2} \quad \frac{r_{ij,z}}{r_{ij}^2} \right], \quad r_{ij} = \|\mathbf{r}_j - \mathbf{r}_i\|.$$

Thus, $\mathbf{R}^{(i)}$ has dimensions $(N_{\text{neigh}}, d+1)$, where $d$ is the spatial dimension (typically 3).

The inverse-distance and scaled coordinates ensure that the representation is invariant under translation, rotation, and permutation of atoms of the same type.

### Smooth Cutoff Function

To guarantee that both the potential energy and forces are continuous at the cutoff radius, DeePMD employs a smooth switching function $s(r_{ij})$ between the inner and outer cutoffs $r_{\text{on}}$ and $r_c$. For $r \in [r_{\text{on}}, r_c]$:

$$x = \frac{r - r_{\text{on}}}{r_c - r_{\text{on}}}, \qquad s(r) = x^3(-6x^2 + 15x - 10) + 1,$$

and the function is defined piecewise as:

$$s(r) = \begin{cases} 1, & r \le r_{\text{on}}, \\ x^3(-6x^2 + 15x - 10) + 1, & r_{\text{on}} < r < r_c, \\ 0, & r \ge r_c. \end{cases}$$

This smooth polynomial ensures that $s(r)$ and its derivatives vanish smoothly at $r_c$, maintaining the continuity of both the potential and the forces.

## Weighted Environment Matrix

The switching function $s(r_{ij})$ is applied multiplicatively to all elements of each row in the environment matrix:

$$\tilde{\mathbf{R}}_{ij} = s(r_{ij}) \begin{bmatrix} \frac{1}{r_{ij}} & \frac{r_{ij,x}}{r_{ij}^2} & \frac{r_{ij,y}}{r_{ij}^2} & \frac{r_{ij,z}}{r_{ij}^2} \end{bmatrix}.$$

The resulting matrix $\tilde{\mathbf{R}}$ is the smooth, weighted representation of the local environment used as the input to the embedding (filter) neural network to generate atomic descriptors $\mathcal{D}_i$.

## Normalization and Type Grouping

To handle multiple atomic species (e.g., binary alloys), neighbors are grouped by atomic type, and a fixed number of neighbors per type is selected according to a vector $\texttt{sel} = [n_1, n_2, \ldots, n_T]$, where $T$ is the number of chemical species. The total number of rows is therefore

$$N_{\text{neigh}} = \sum_{t=1}^{T} n_t.$$

This ensures descriptor consistency across all atom types and a fixed input dimensionality to the neural network.

Before feeding into the neural network, each element of $\tilde{\mathbf{R}}_{ij}$ is normalized using precomputed mean and standard deviation parameters $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ obtained during training:

$$R_{ij}^{(\text{norm})} = \frac{\tilde{R}_{ij} - \mu_{ij}}{\sigma_{ij}}.$$

**Shape of the Normalization Statistics:** The tensors $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ have the same shape as the environment matrix, i.e.

$$\boldsymbol{\mu}, \boldsymbol{\sigma} \in \mathbb{R}^{N_{\text{neigh}} \times (d+1)}.$$

Each row corresponds to a specific neighbor slot (ordered by atom type and distance), and each column corresponds to one of the $(d+1)$ coordinate components. These values are extracted directly from the DeepMD frozen model (`frozen_model.pb`) and are used to reproduce the same normalization behavior during inference in HALMD.

**Species-Dependent Normalization:** In multi-component systems such as alloys, the normalization parameters $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ are not shared globally. Instead, DeepMD-kit v2 maintains distinct normalization tensors for each *central atom species*. That is, for a system with $T$ atomic species, the model stores

$$\boldsymbol{\mu}^{(s)}, \boldsymbol{\sigma}^{(s)} \in \mathbb{R}^{N_{\text{neigh}}^{(s)} \times (d+1)}, \quad s = 1, 2, \ldots, T,$$

where $s$ indexes the central atom type. Each central atom type (e.g., Cu or Ag) uses its corresponding $(\boldsymbol{\mu}^{(s)}, \boldsymbol{\sigma}^{(s)})$ when normalizing its own environment matrix. This design reflects the fact that atoms of different species experience distinct local environments and statistical distributions of interatomic distances. Consequently, the normalization constants differ per species, ensuring that the descriptor scales are consistent within each element type but remain unbiased across the alloy system.

**Interpretation of `sel`:** The selection vector `sel` defines how many neighbors are considered for each atomic species. For example, for a binary alloy with species A and B, `sel` $= [60, 40]$ would allocate the first 60 rows of the environment matrix to type A neighbors and the next 40 rows to type B neighbors. This per-type grouping ensures consistent indexing and species-dependent feature ordering across the model.

## Summary of Construction Pipeline

1. Compute relative displacement vectors $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$ using extended coordinates (including ghost atoms).
2. Calculate distances $r_{ij} = \|\mathbf{r}_{ij}\|$ for all neighbors within $r_c$.
3. Apply smooth switching function $s(r_{ij})$ to ensure differentiability at $r_c$.
4. Construct weighted environment matrix $\tilde{\mathbf{R}}_{ij}$ with $\frac{1}{r_{ij}}$ and $\frac{\mathbf{r}_{ij}}{r_{ij}^2}$ components.
5. Normalize $\tilde{\mathbf{R}}_{ij}$ using training statistics $(\boldsymbol{\mu}^{(s)}, \boldsymbol{\sigma}^{(s)})$ corresponding to the central atom species $s$.
6. Sort and group rows by atomic species according to `sel`.

## Interpretation

The resulting environment matrix $\mathbf{R}^{(i)}$ captures both radial and angular information of the local environment around atom $i$ in a smooth, differentiable form. For alloys, the normalization is species-aware, preserving the statistical consistency of descriptor inputs across heterogeneous chemical environments. This representation serves as the foundational input to the neural embedding (filter) network, enabling the model to infer atomic energy contributions and compute consistent forces within molecular dynamics simulations.