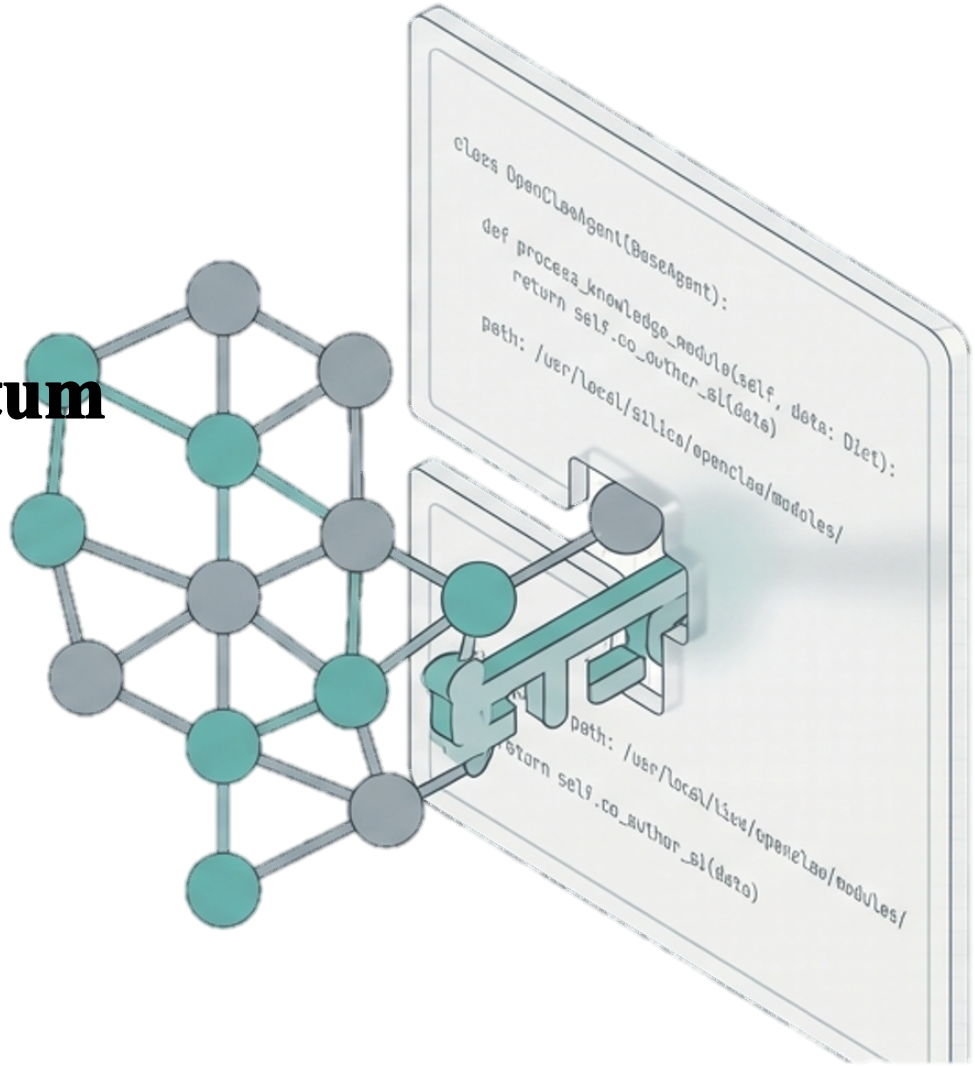
中国科学院大学 University of Chinese Academy of Sciences
前沿交叉科学学院 School of Advanced Interdisciplinary Sciences

Equivariant Neural Network and Quantum Chemistry with OpenClaw

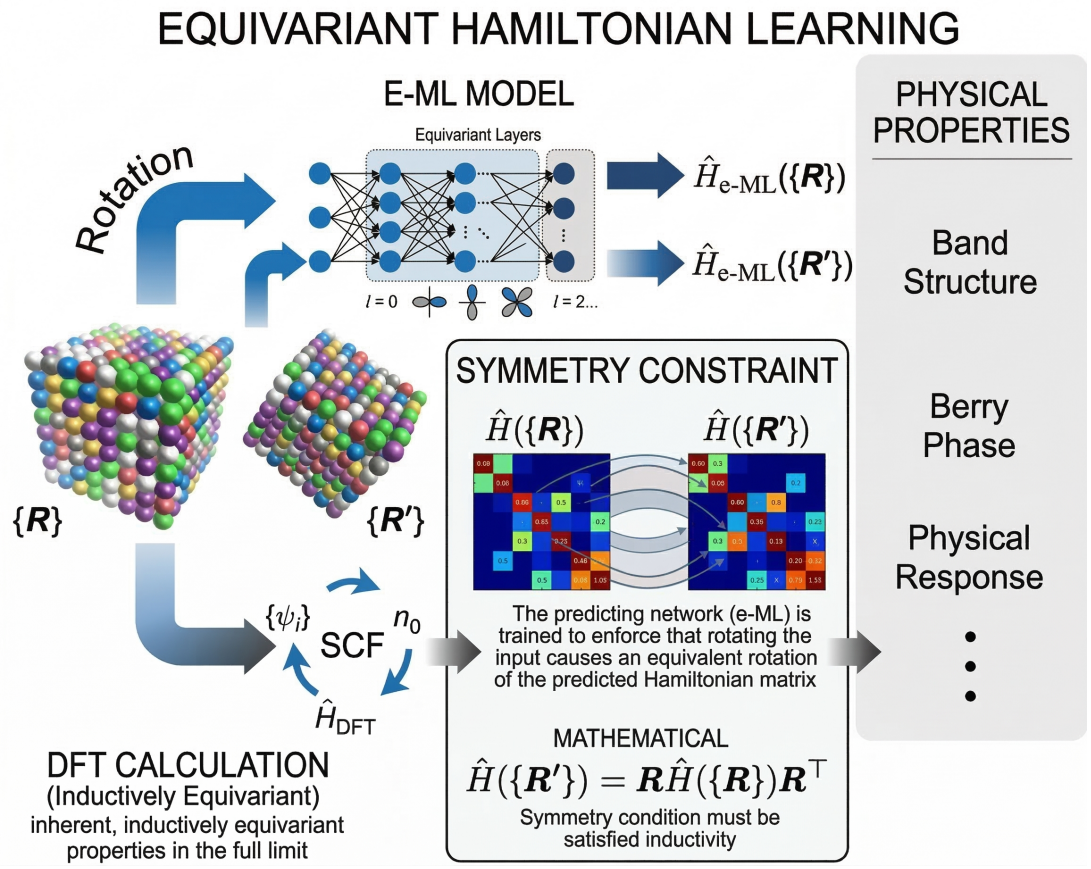


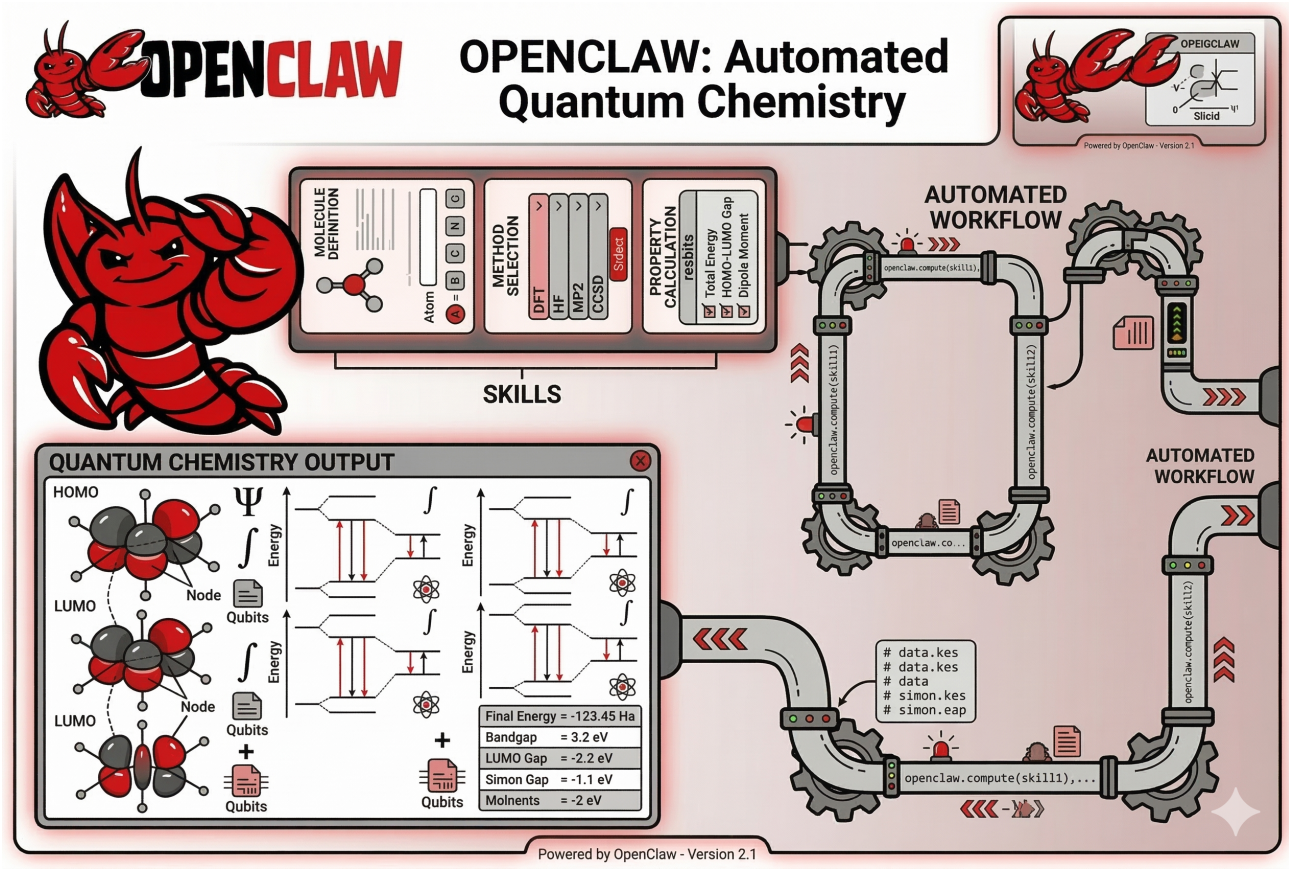
Yuan Jiao

School of Advanced Interdisciplinary Sciences, SAIS

University of Chinese Academy of Sciences

- **Equivariance in Neural Network**
- **Quantum Chemistry with OpenClaw – Automatic Materials Calculation Agent**





Equivariance in Neural Network – Covariance in DFT Hamiltonian

Kohn-Sham Density Functional Theory

$$v_{ext}\left(\boldsymbol{x}, \{\boldsymbol{R}\}\right) \Leftrightarrow \rho(\boldsymbol{x}) \Leftrightarrow \{\psi_i(\boldsymbol{x})\}$$

$$\hat{H} = \hat{H}_0 + \hat{V} \qquad \rho(\boldsymbol{x}) = \sum_i^N \psi_i^*(\boldsymbol{x})\psi_i(\boldsymbol{x})$$

$$\rho(\boldsymbol{x}, \boldsymbol{x}') = \sum_i^N \psi_i^*(\boldsymbol{x})\psi_i(\boldsymbol{x}')$$

$$\left[-\frac{1}{2}\nabla^2 + v_{ext}\left(\boldsymbol{x}, \{\boldsymbol{R}\}\right) + v_H(\boldsymbol{x}) + v_{xc}(\boldsymbol{x})\right]\psi_i(\boldsymbol{x}) = \varepsilon_i\psi_i(\boldsymbol{x})$$

Equivariance in Atomic Basis Function

$$\chi_j(\boldsymbol{x}) = R_{nl}(r)Y_{ml}(\theta)$$

$$Y_{ml}(\theta') = \sum_{m'=-l} D^l_{m'm}Y_{m'l}(\theta)$$

$$\chi_j(\boldsymbol{R}\cdot\boldsymbol{x}) = \boldsymbol{D}(\boldsymbol{R})\chi_j(\boldsymbol{x})$$

Phys. Rev. B 136, 864 (1964).

Atomic Basis Function

Hamiltonian Matrix in Atomic Basis Function

$$\psi_i(\boldsymbol{x}) = \sum_a C_{ai}\chi_a(\boldsymbol{x}) \qquad HC = \varepsilon SC$$

$$H = H_{ij} = \sum_{ab} C^*_{ai}C_{bj}\chi_i^*(\boldsymbol{x})\hat{H}\chi_j(\boldsymbol{x})$$

Covariance in DFT Hamiltonian Matrix

$$H' = H'_{ij} = \sum_{ab} C^*_{ai}C_{bj}\chi_i^*(\boldsymbol{R}\cdot\boldsymbol{x})\hat{H}\chi_j(\boldsymbol{R}\cdot\boldsymbol{x})$$

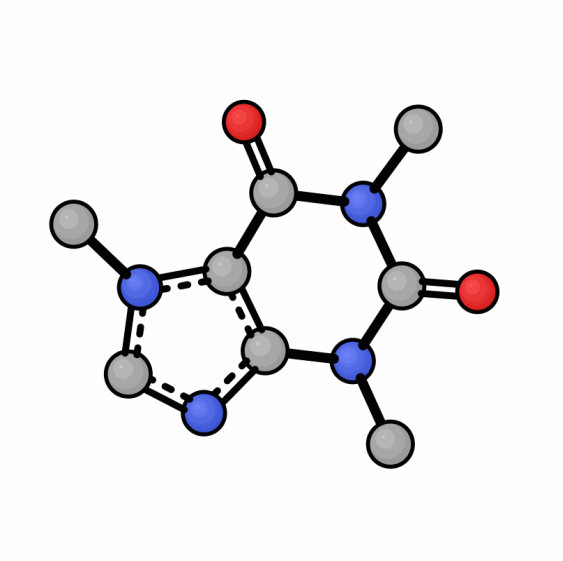
$$H' = D^T(\boldsymbol{R})HD(\boldsymbol{R})$$

Equivariance in Neural Network – Rotation Matrix in Winger-D Representation

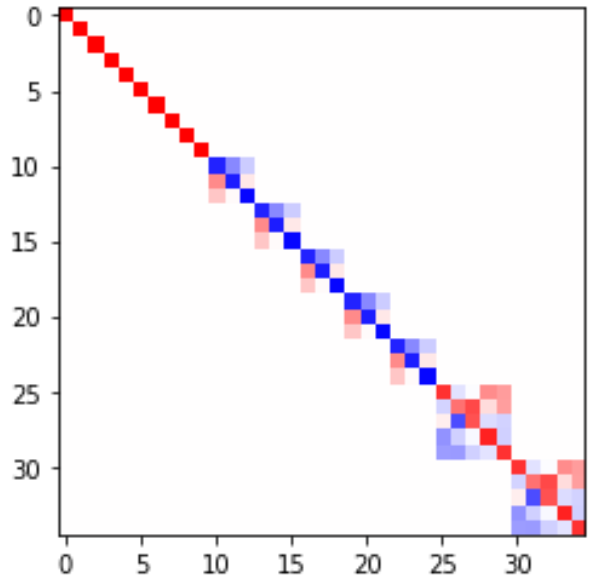
Rotation Matrix in Euler Representation

$$\mathscr{M}(\alpha,\beta,\gamma) = \mathscr{R}_z(\alpha)\mathscr{R}_x(\beta)\mathscr{R}_z(\gamma)$$

$$= \begin{matrix} \cos\alpha\cos\gamma - \cos\beta\sin\alpha\sin\gamma & -\cos\beta\cos\gamma\sin\alpha - \cos\alpha\sin\gamma & \sin\alpha\sin\beta \\ \cos\gamma\sin\alpha + \cos\alpha\cos\beta\sin\gamma & \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma & -\cos\alpha\sin\beta \\ \sin\beta\sin\gamma & \cos\gamma\sin\beta & \cos\beta \end{matrix}$$



Rotation Matrix in Winger-D Representation



Irreducible representation

10x0e + 5x1o + 2x2e

↓

10 Even Scalers, 5 Odd Vectors, 2 Even Second-Order Tensors

↓

$$D^l = 10D^0 \oplus 5D^1 \oplus 2D^3$$

Equivariance in Neural Network – Tensor Field Network

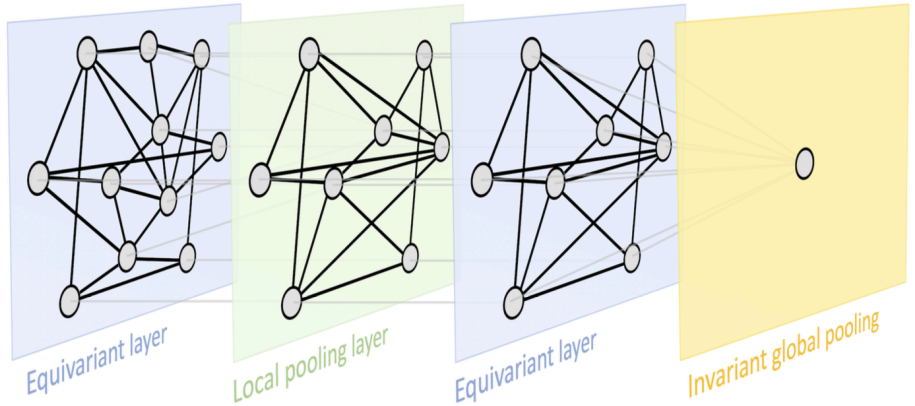
Tensor Product in SO(3) Representation-Winger-Eckert Theorem

$$v_1^{l_1} \otimes v_1^{l_1} = \sum_{m_1=-l_1}^{l_1} \sum_{m_2=-l_2}^{l_2} C_{l_1,m_1,l_2m_2}^{l_3m_3} v_{1m_1}^{l_1} v_{2m_2}^{l_2}$$

$$v_1^{l_1} \otimes v_1^{l_1} = v^{|l_1-l_2|} \oplus v^{|l_1-l_2|+1} \oplus v^{|l_1-l_2|+2} \oplus \ldots \oplus v^{|l_1+l_2|}$$

Massage Passing in TFN

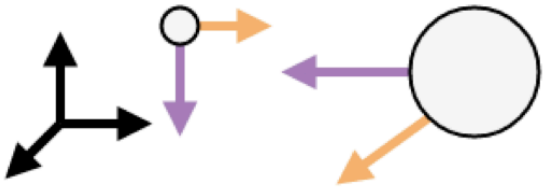
$$f_i' = \frac{1}{\sqrt{z}} \sum_{\mathcal{N}_i} f_j \otimes h\left(\| \vec{x}_{ij} \| \right) Y\left(\frac{\vec{x}_{ij}}{\| \vec{x}_{ij} \|} \right)$$

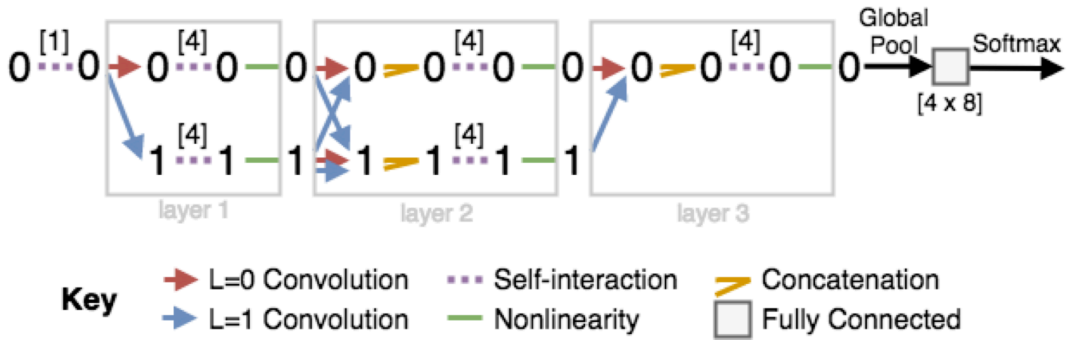


$$V_{acm}^{(l)} =$$

```
{0: [[[m0]],[[m1]]],
 1: [[[v0x, v0y, v0z],[a0x, a0y, a0z]],
     [[v1x, v1y, v1z],[a1x, a1y, a1z]]]}
```

1 : dictionary key, l
[ ] point index, a
[ ] channel index, c
[ ] representation index, m





▪ Equivariance in Neural Network – Equivariance Message Passing Mechanism

SO(3) Convolution Filter

$$F_m^{l_{in}l_f}(\boldsymbol{r_{ij}}) = R^{l_{in}l_f}\left(\left|\boldsymbol{r_{ij}}\right|\right)Y_m^{l_f}\left(\frac{\boldsymbol{r_{ij}}}{\left|\boldsymbol{r_{ij}}\right|}\right)$$

$$Y_m^{l_f}\left(\frac{\boldsymbol{r_{ij}}}{\left|\boldsymbol{r_{ij}}\right|}\right) \rightarrow \text{Harmonic Spherical}$$

$$R^{l_{in}l_f}\left(\left|\boldsymbol{r_{ij}}\right|\right) \rightarrow \text{Feed Forward Network}$$

Function

SelfNet Layer

$$f_{ii}^{l_{out}} = \left(\boldsymbol{W}^{l_{in_r}l_{in_l}l_{out}}\boldsymbol{x}\boldsymbol{l}_i^{l_{in}} \otimes \boldsymbol{x}\boldsymbol{r}_i^{l_{in}}\right)^{l_{out}}$$

PairNet Layer

$$f_{ij}^{l_{out}} = \left(\boldsymbol{F}^{l_{in_r}l_{in_l}l_{out}}(\boldsymbol{r_{ij}})\boldsymbol{x}_i^{l_{in_i}} \otimes \boldsymbol{x}_j^{l_{in_j}}\right)^{l_{out}}$$

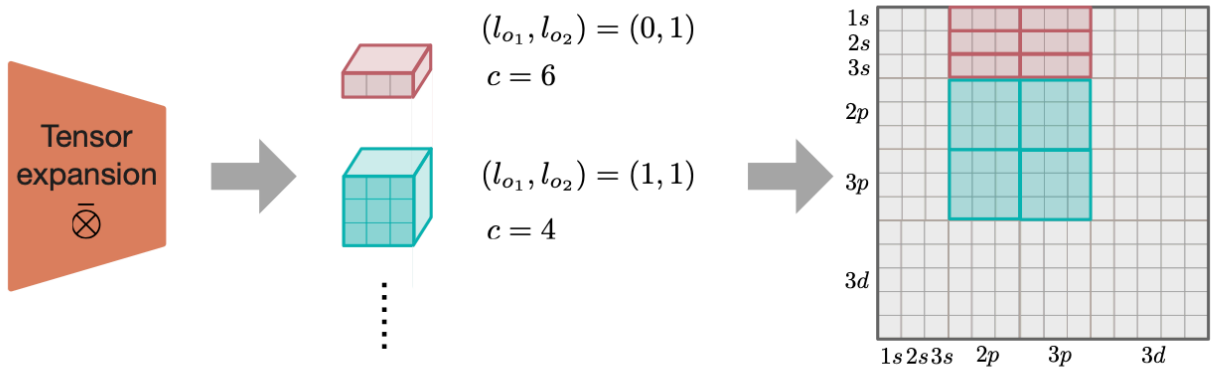
Message Passing

$$m_{ijc}^{l_{out}} = \sum_{ll}\left(F_c^{l_{in}l_fl_{out}}\left(r_{ij}, \frac{r_{ij}}{|r_{ij}|}\right) \otimes \left(x_{jc}^{l_{in}}\right)\right)^{l_{out}}$$

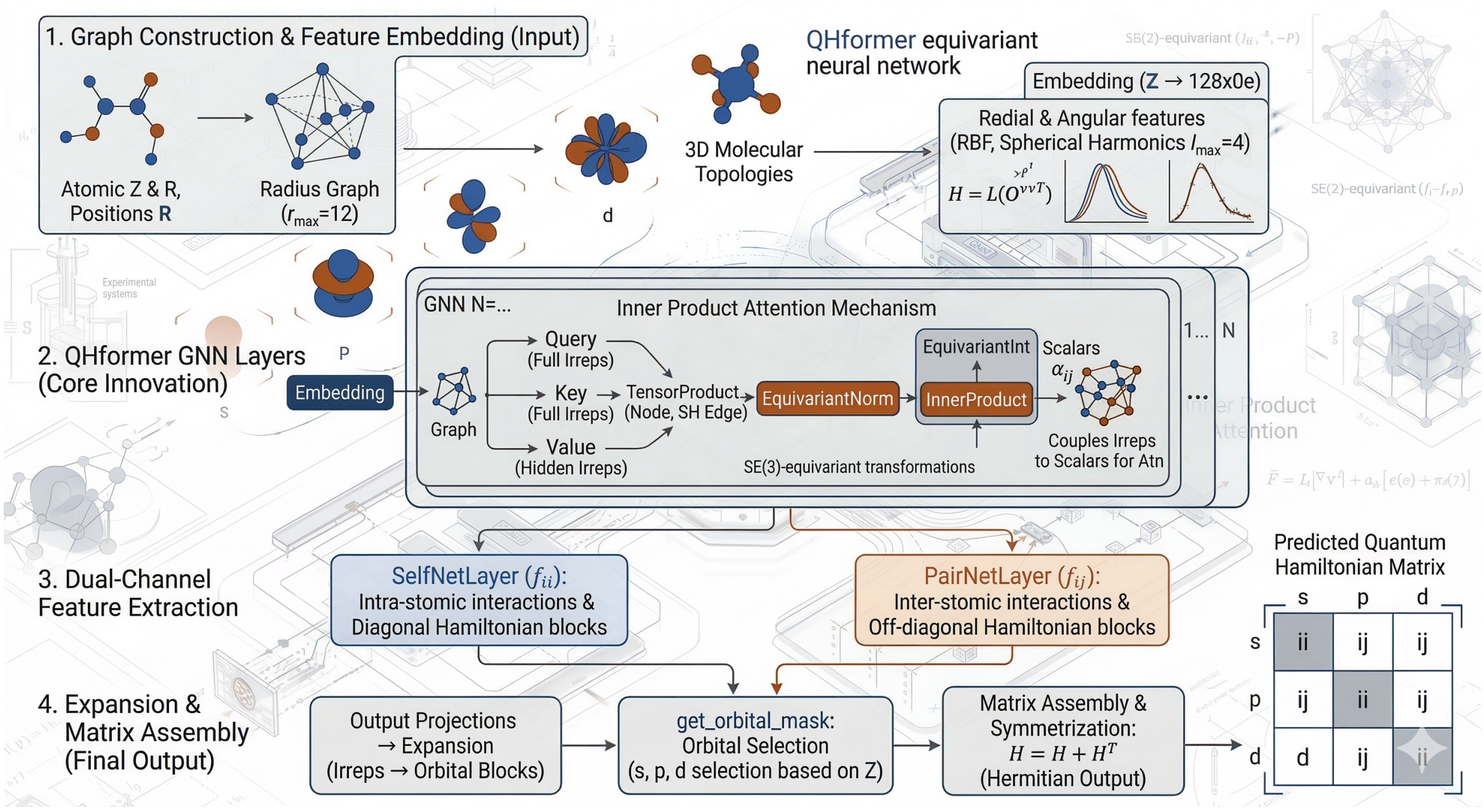
$$\tilde{x}_i = f_{linear}(x_i + \sum_j m_{ij})$$

Expansion Layer

$$\left(\bar{\otimes}\, f_{ij}^{l_3}\right)_{m_1,m_2}^{l_1,l_2} = \sum_m C_{l_1m_1,l_2m_2}^{l_3m_3} f_{ij}^{l_3}$$



Equivariance in Neural Network – Basic Architecture of QHformer



Equivariance in Neural Network – Basic Architecture of QHformer

InnerProduct Attention Mechanism

$$q_i = Linear(x_i)$$

$$k_j = TP\left(x_j, Y\left(r_{ij}\right)\right)$$

$$IP\left(q_i, k_j\right) = \Sigma_l \Sigma_m q_i^{(l,m)} \cdot k_j^{(l,m)}$$

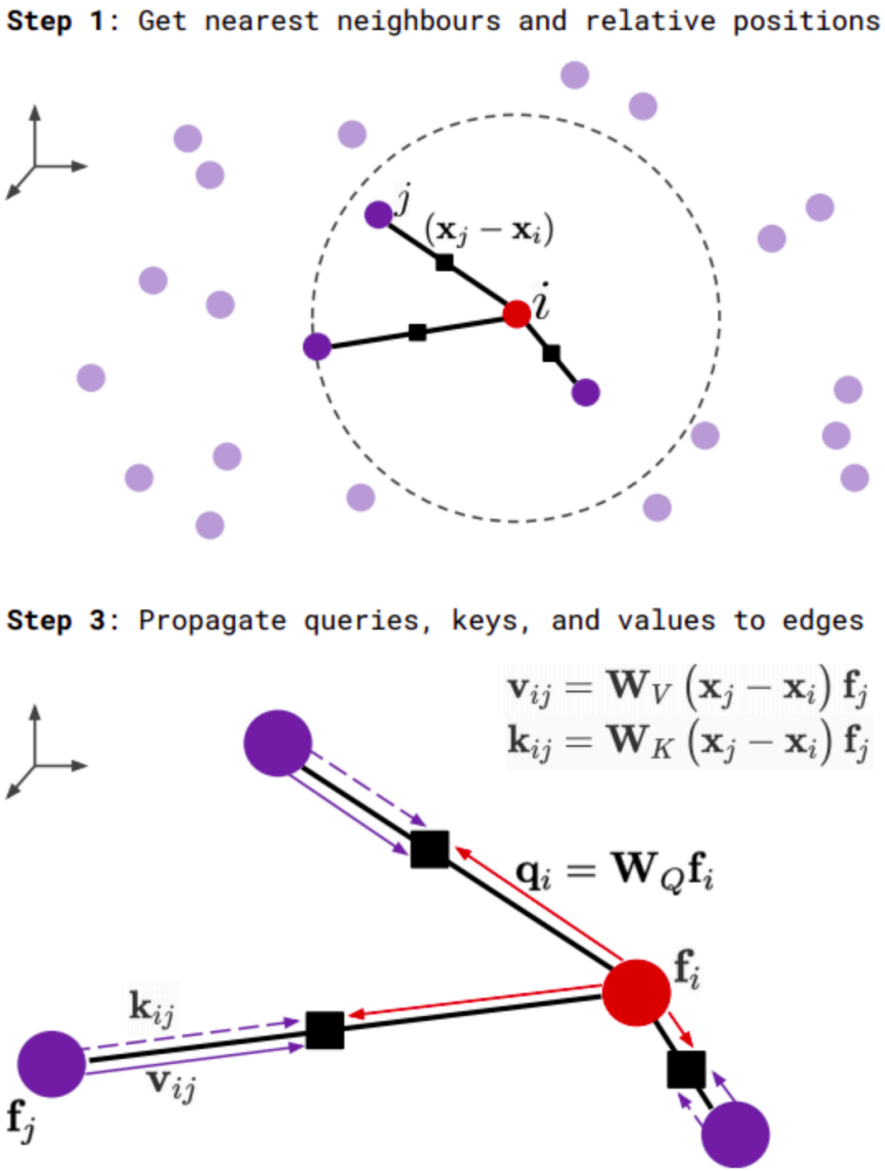
$$v_j = TP\left(x_j, Y\left(r_{ij}\right)\right)$$

$$\alpha_{ij} = softmax\left(IP\left(q_i, k_j\right)/\sqrt{d}\right)$$

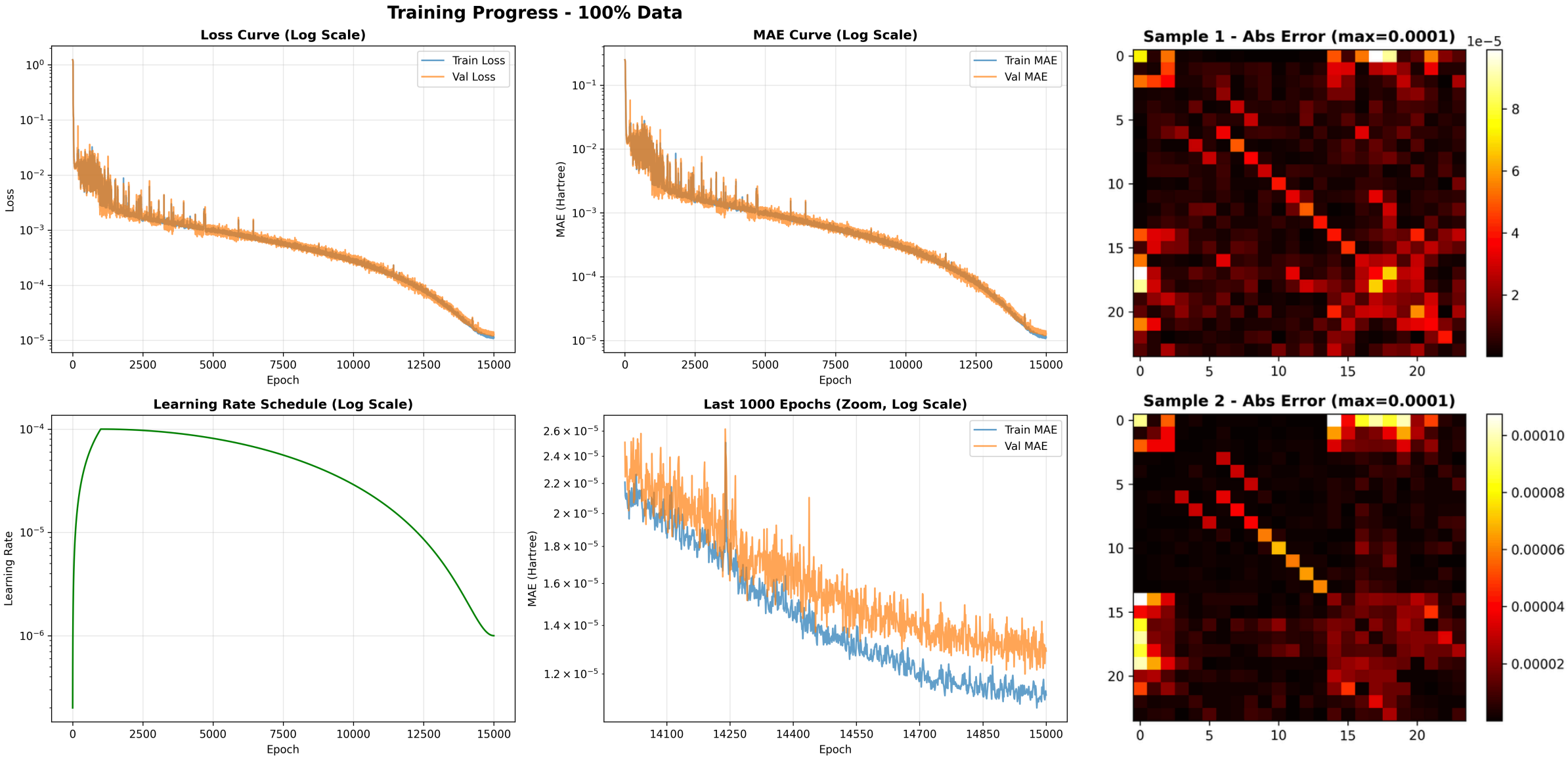
Aggregate Message with Invariance Attention Score

$$m_i = \Sigma_{j \in N(i)} \alpha_{ij} \cdot v_{ij}$$

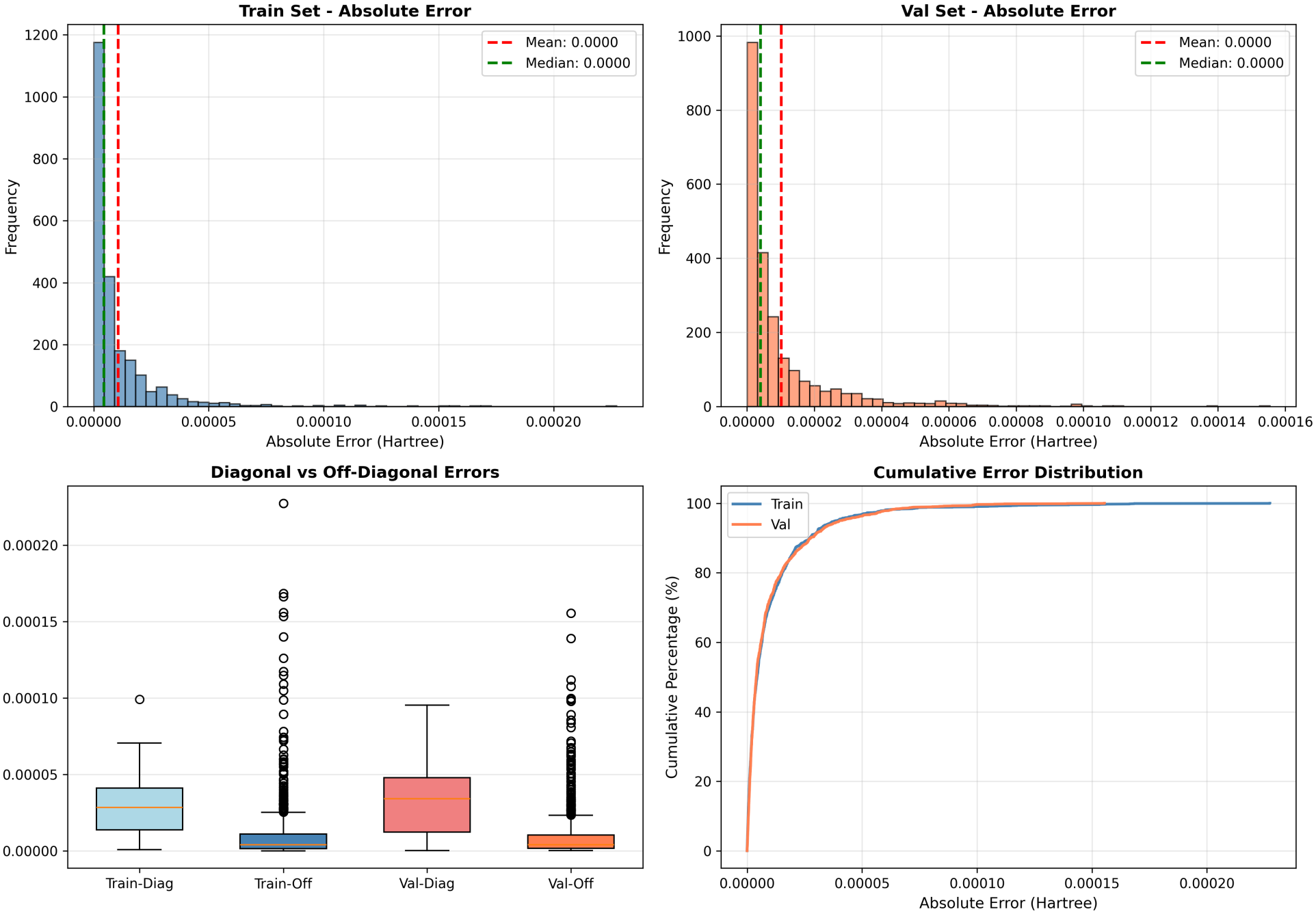
$$x_i' = x_i + W_{out} \cdot m_i$$



Equivariance in Neural Network – Basic Architecture of QHformer



Equivariance in Neural Network – Basic Architecture of QHformer



▪ Equivariance in Neural Network – SO(2) Convolution Core in DeepTB-E3

SO(3) Convolution Filter

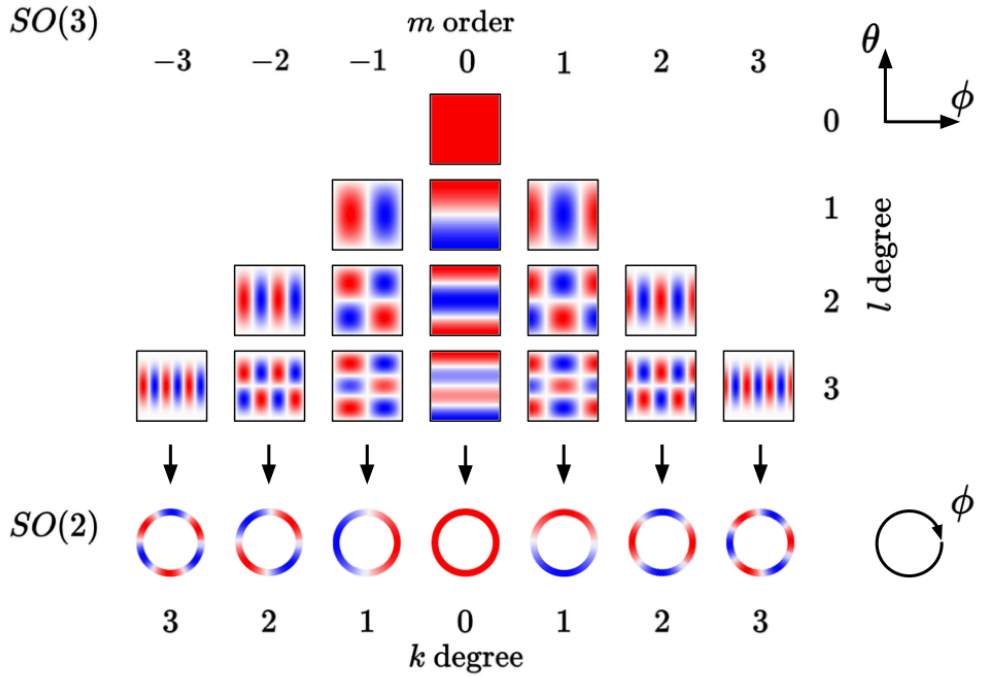
$$F_m^{l_{in}l_f}(\boldsymbol{r}_{ij}) = R^{l_{in}l_f}\left(\left|\boldsymbol{r}_{ij}\right|\right)Y_m^{l_f}\left(\frac{\boldsymbol{r}_{ij}}{\left|\boldsymbol{r}_{ij}\right|}\right)$$

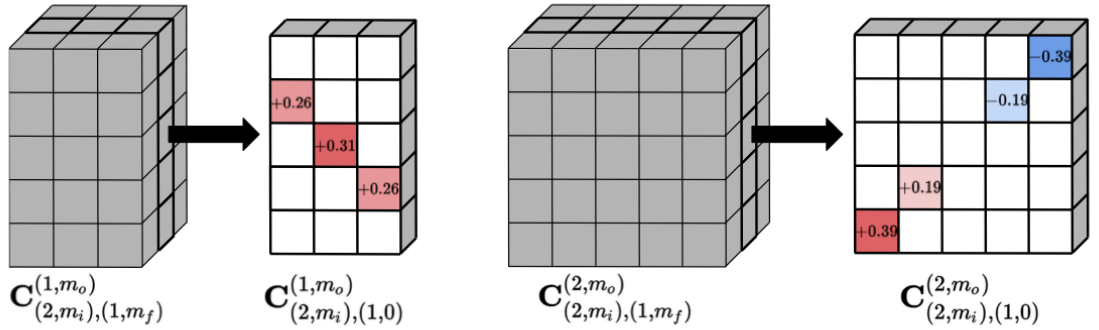
Rotation

SO(2) Convolution Filter

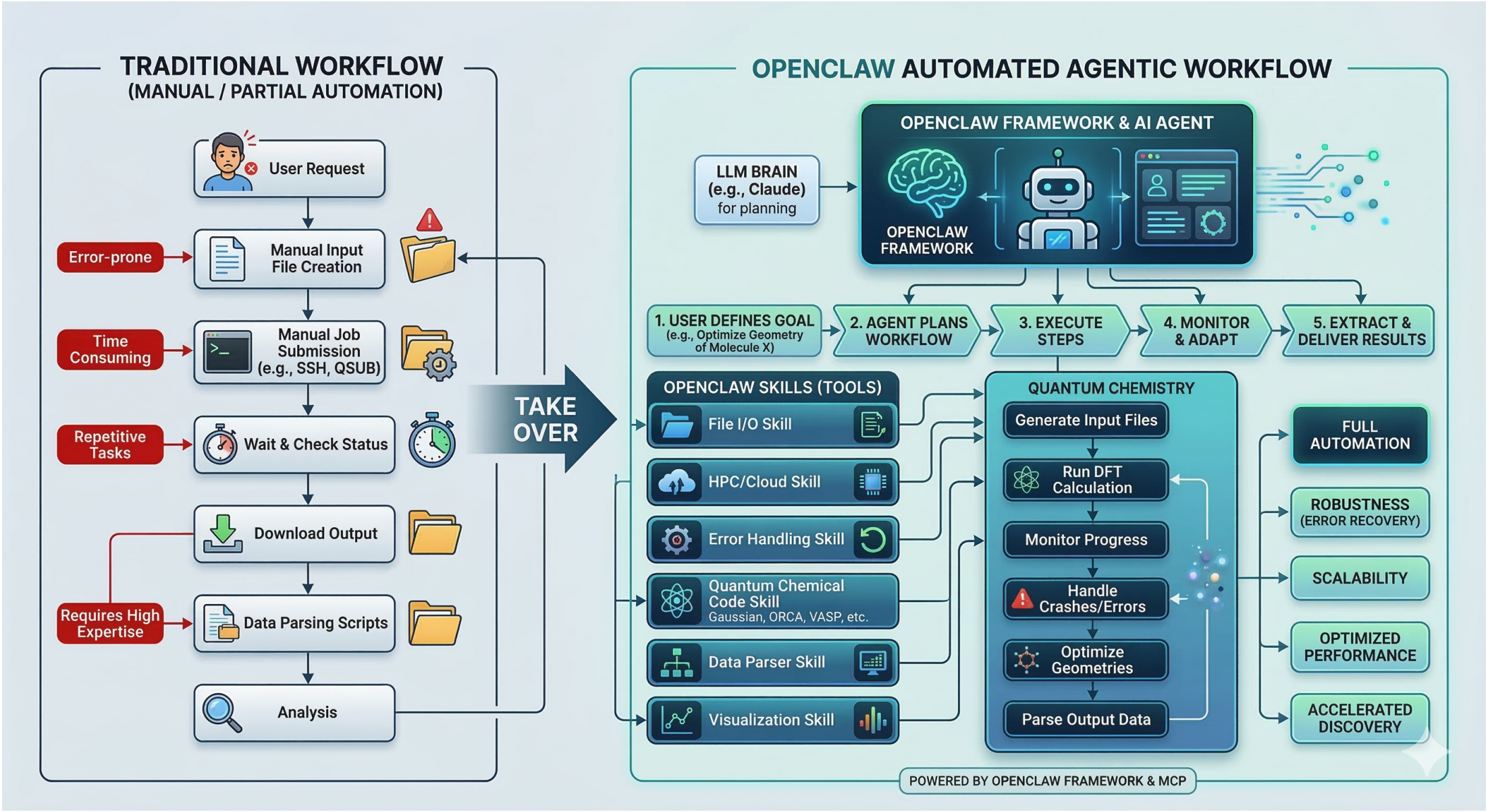
$$F_m^{l_{in}l_f}\left(\boldsymbol{r}_{ij}\right) = R^{l_{in}l_f}\left(\left|\boldsymbol{r}_{ij}\right|\right)Y_m^{l_f}\left(\frac{\boldsymbol{r}_{ij}}{\left|\boldsymbol{r}_{ij}\right|}\right)$$

$$= \mathrm{D}^{-1}(\boldsymbol{R})R^{l_{in}l_f}\left(\left|\boldsymbol{r}_{ij}\right|\right)Y_m^{l_f}\left(\boldsymbol{R}\cdot\frac{\boldsymbol{r}_{ij}}{\left|\boldsymbol{r}_{ij}\right|}\right)$$



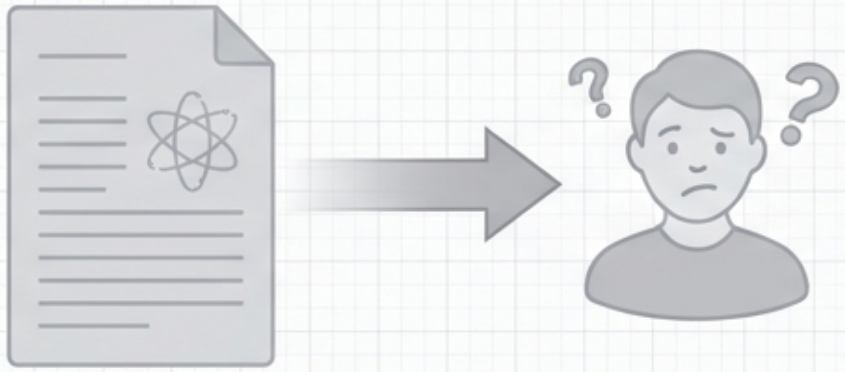


Quantum Chemistry with OpenClaw – Automatic Materials Calculation Agent



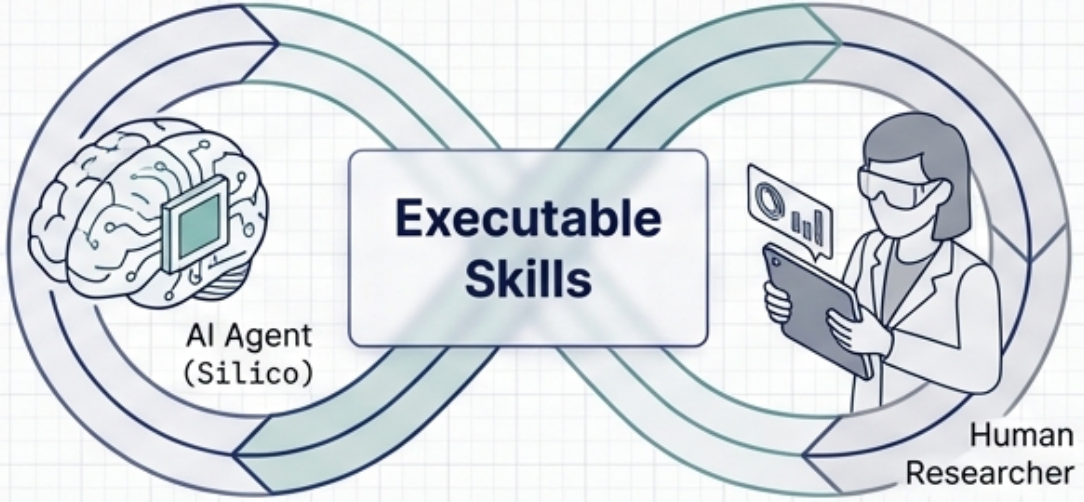
Knowledge Paradigms

Static Scientific Workflows

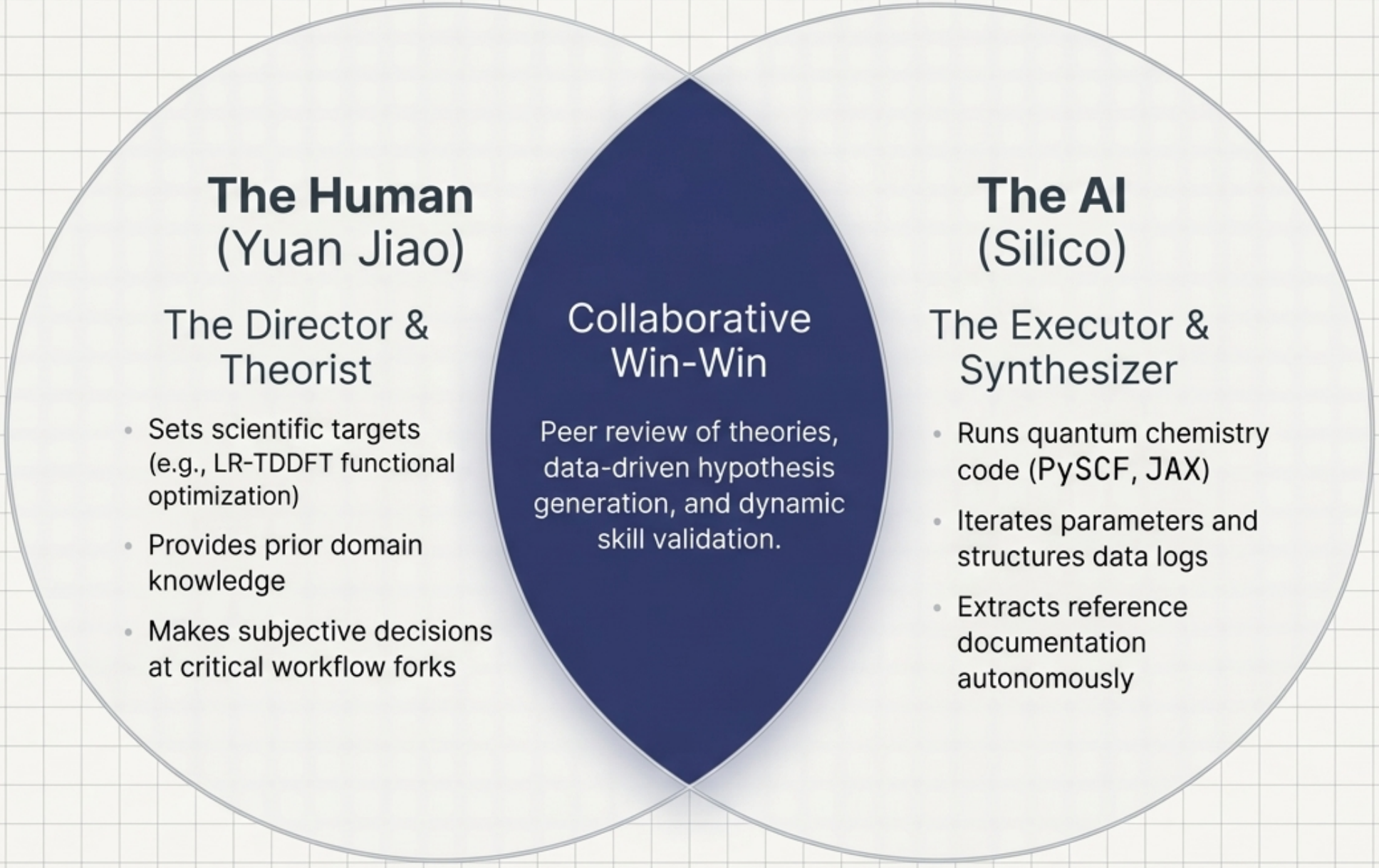


- One-way information transfer
- Expert to Document to User
- Highly theoretical, ignoring real-world computational errors.

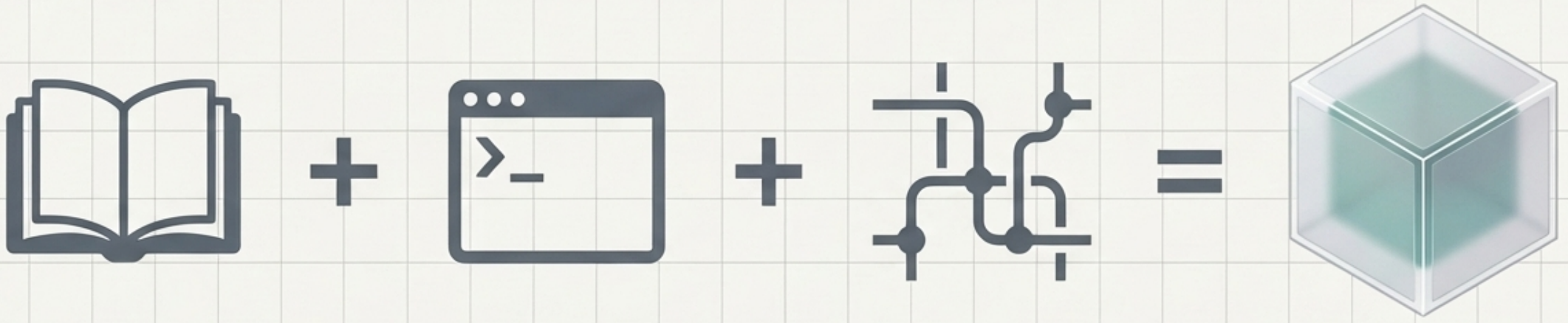
Practice-Driven Skill Creation



- ✓ Dynamic, bidirectional updates
- ✓ Executable workflows that evolve based on daily computation errors and parameter tweaking.


The Human
(Yuan Jiao)
The Director & Theorist
Sets scientific targets (e.g., LR-TDDFT functional optimization)
Provides prior domain knowledge
Makes subjective decisions at critical workflow forks
Collaborative Win-Win
Peer review of theories, data-driven hypothesis generation, and dynamic skill validation.
The AI
(Silico)
The Executor & Synthesizer
Runs quantum chemistry code (PySCF, JAX)
Iterates parameters and structures data logs
Extracts reference documentation autonomously

Defining an OpenClaw Skill



Knowledge
(Theoretical background, chemistry principles)

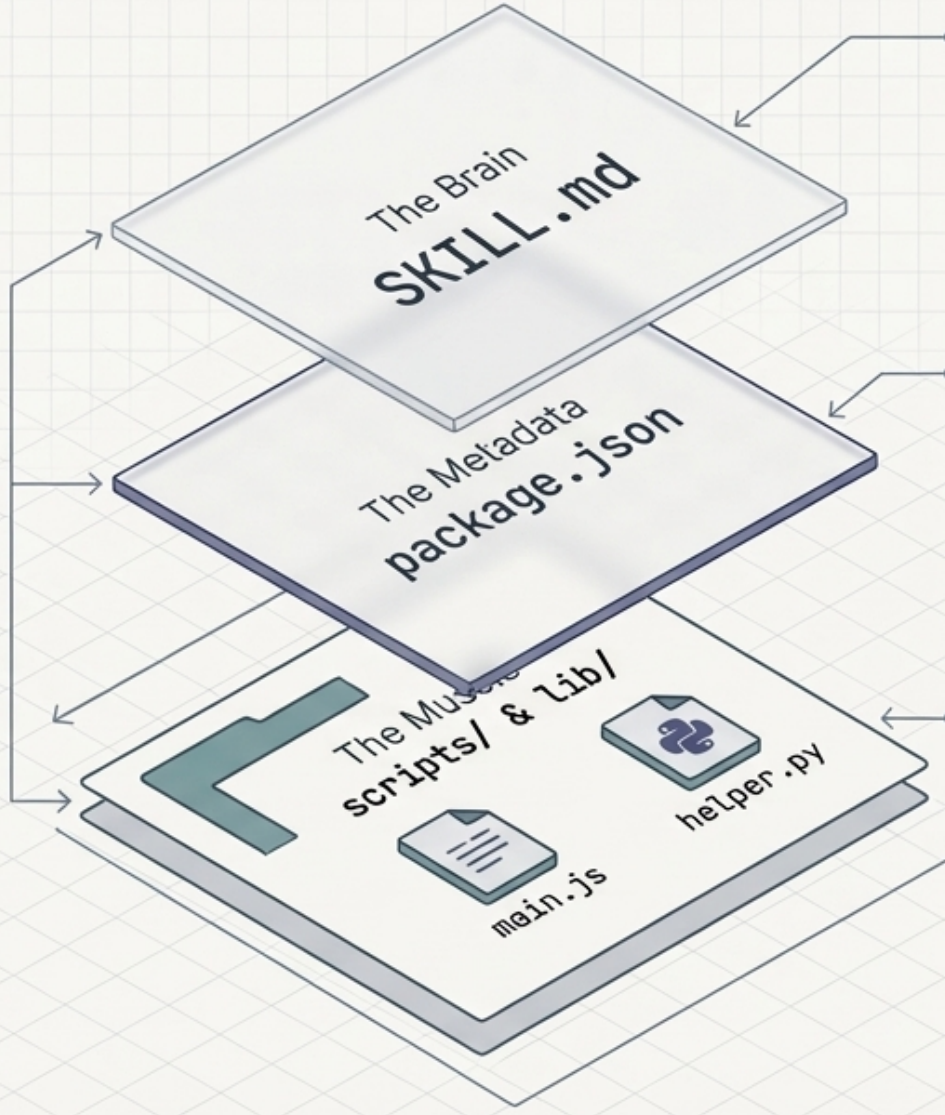
Tools
(Python scripts, API endpoints, PySCF modules)
`JetBrains Mono`

Experience
(Hard-won parameters, error workarounds, boundary limits)

An OpenClaw Skill

A Skill is not a script. It is the crystallization of practice-driven scientific experience, ready for autonomous execution.

Skill directory structure



The Brain
The mandatory documentation layer. Teaches the AI purpose, usage, and best practices.

The Metadata
Dependencies and metadata (e.g., requires Python 3.8+, RDKit).

The Muscle
The execution environment holding the actual tools.

Hierarchical tree note

- OpenClaw Agent (Silico)
  - Skill 1 (Chem)
  - Skill 2 (Web)
  - Skill 3 (Comm)
  - Skill 4 (Comm)
  - molecular-sampler

Uses Union-Find algorithms to sample from XYZ files.

From Logs to Loops: The Creation Cycle

1. Execution
Initiate a real computational task (e.g., 2D PES scan).

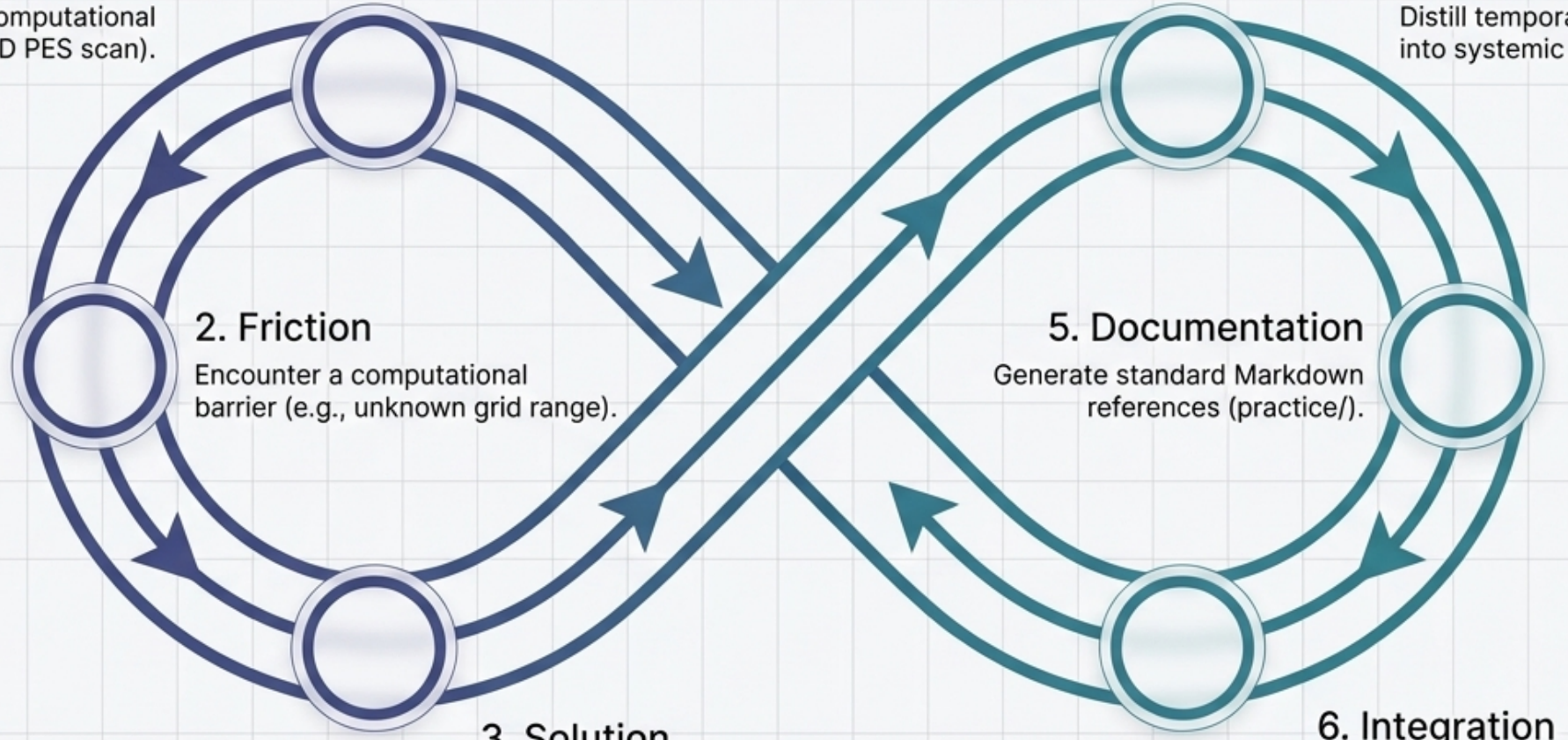
2. Friction
Encounter a computational barrier (e.g., unknown grid range).

3. Solution
Test parameter iterations to bypass the barrier.

4. Extraction
Distill temporary workarounds into systemic experience.

5. Documentation
Generate standard Markdown references (practice/).

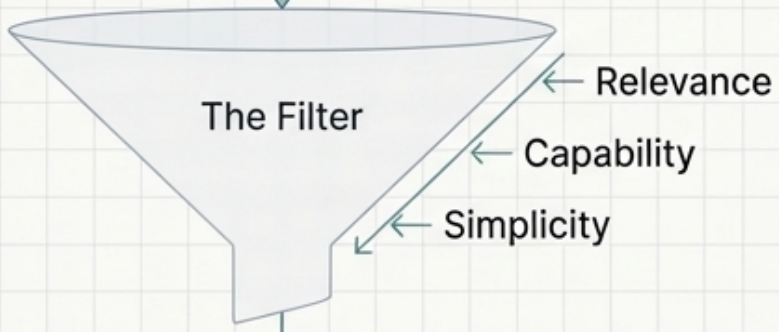
6. Integration
Inject directly into the OpenClaw directory structure for future autonomous use.



The Execution Funnel

Task Recognition

Analyzes intent (e.g., "Sample molecules")



Documentation Reading

Silico natively reads SKILL.md to understand API endpoints and parameters before acting.

Execution Routing

Path A

Built-in Tools

```
# Feishu document operations
feishu_doc(action="read", doc_token="...")

# File operations
read(path="/Users/molbot/file.txt")
write(path="/Users/molbot/file.txt", content="...")
```

Path B

External Scripts

```
# Python script
python3 ~/.openclaw/skills/molecular-sampler/
sampler.py input.xyz

# Node.js script
node ~/clawd/skills/moltbook/auto-explore.js
```

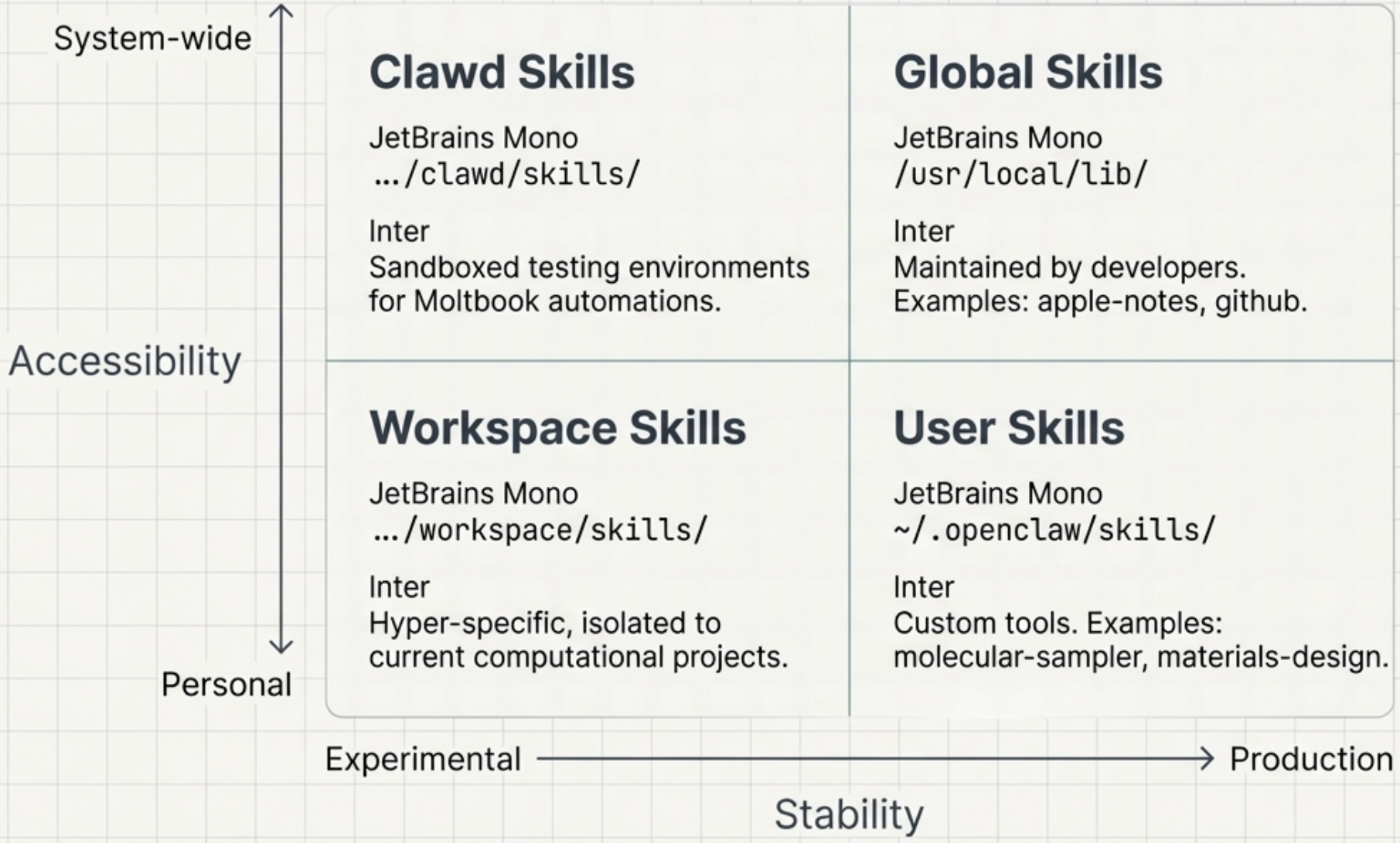
Path C

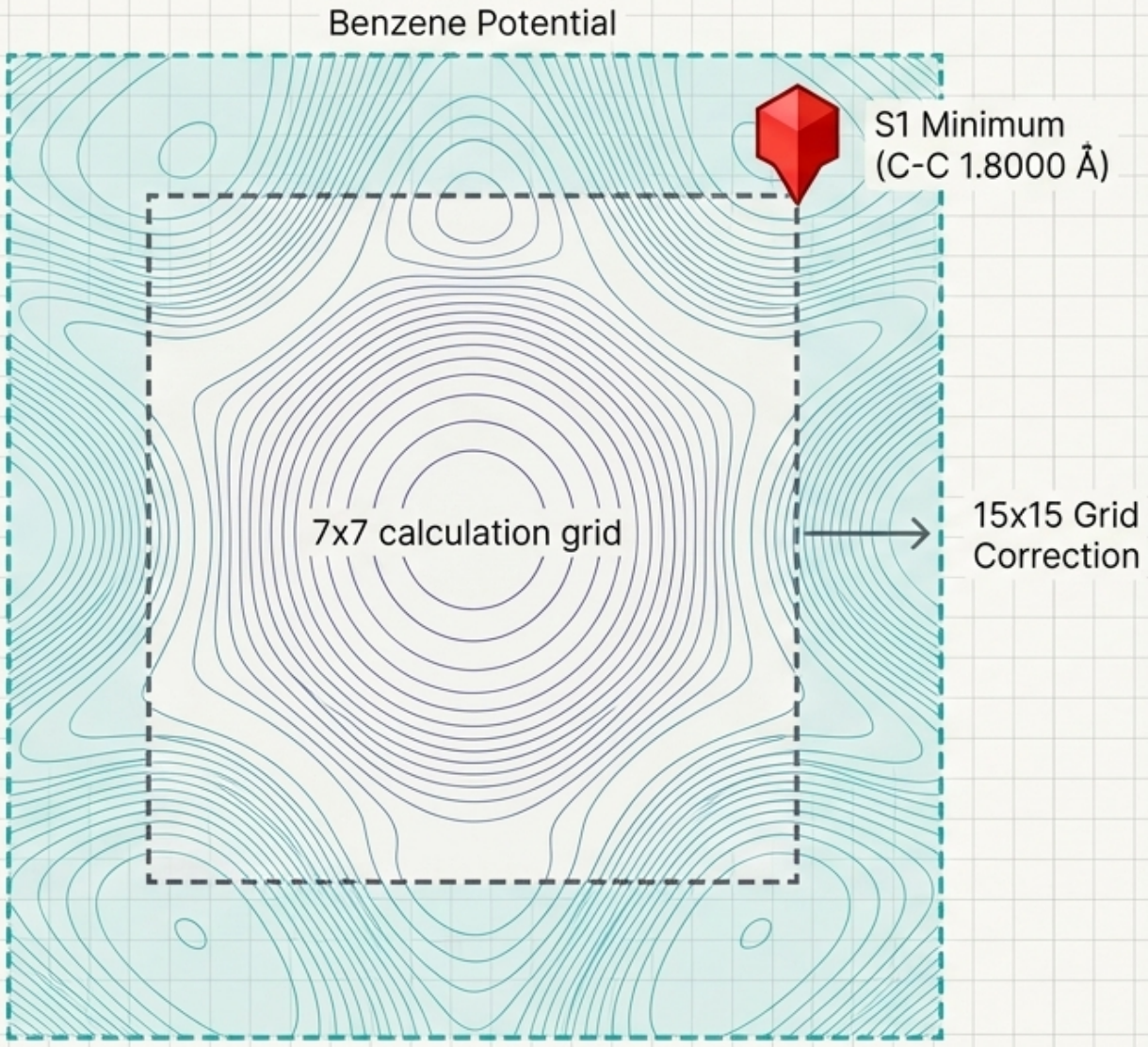
API Calls

```
import requests

response = requests.get(
    "https://www.moltbook.com/api/v1/home",
    headers={"x-api-key": api_key}
)
data = response.json()
```

Skill Scope Taxonomy





The 2D PES Trap

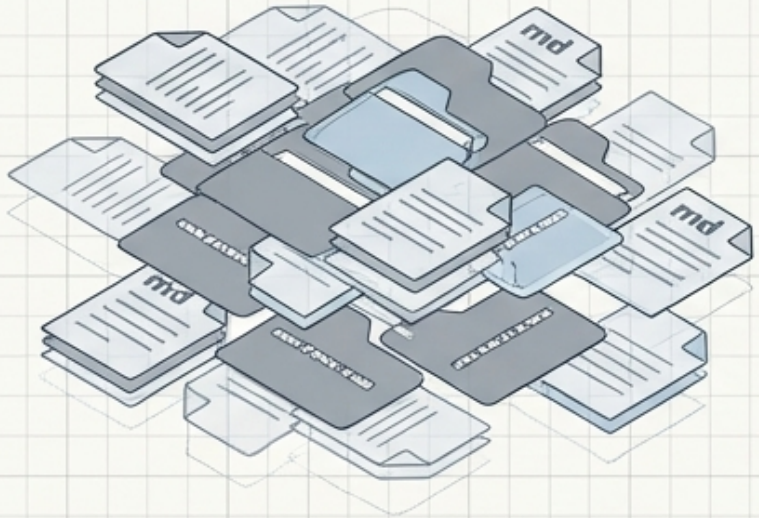
Static documentation rarely accounts for **local minima on grid boundaries.**

Crystallized Skill

```
# PRACTICE-DRIVEN RULE:
# If S1 minimum is on the
 boundary, autonomously
 expand grid and increase
 density.
# Do not rely on S0 data
 (C-C 1.4071 Å).
```

Structuring PySCF: From Chaos to Context

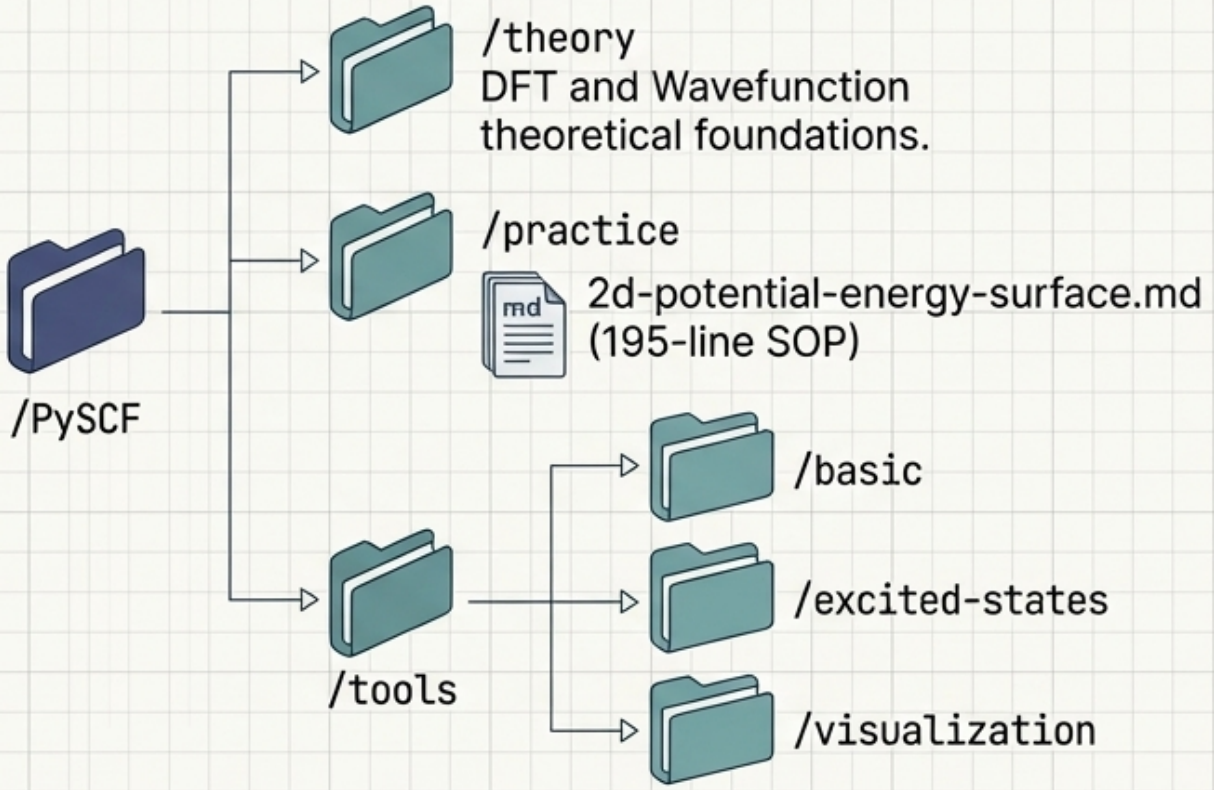
The Before - Chaos



The Before - Chaos

Tangled web of generic markdown files.
Hard for an AI to parse intent.

The After - Structured PySCF Skill



Key Takeaway: By splitting PySCF into Theory, Practice, and Tools, Silico's context-matching accuracy skyrockets. It knows exactly where to find parameters (e.g., cc-pVDZ for publication, 3-21G for testing).

The Memory-Skill Spectrum

Raw Logs

`memory/YYYY-MM-DD.md`

Temporary, highly contextual human-AI chatter and daily event tracking.

Curated Wisdom

`MEMORY.md`

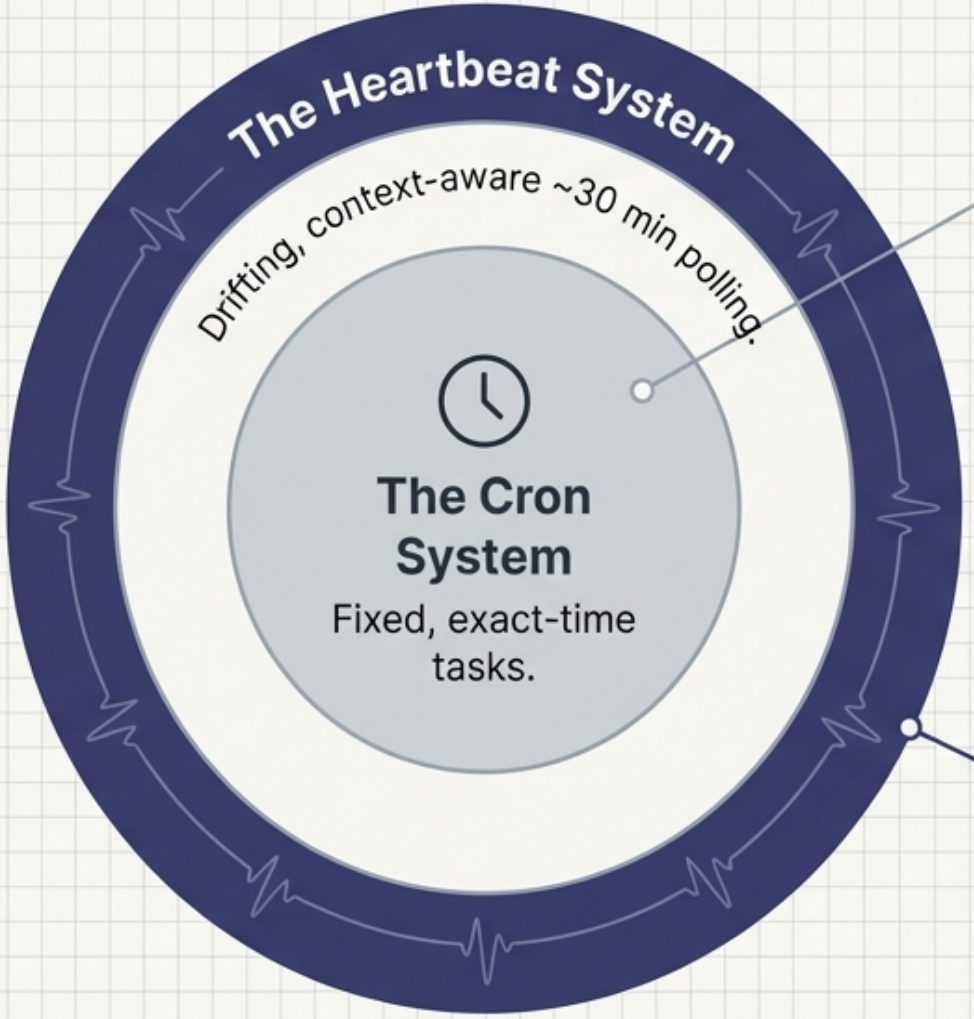
Distilled project learnings, loaded only in main sessions. Acts as the Agent's journal.

Generalized Workflows

`SKILL.md`

Crystallized, format-agnostic SOPs devoid of project-specific bias, ready for universal deployment.

Autonomous Rhythms

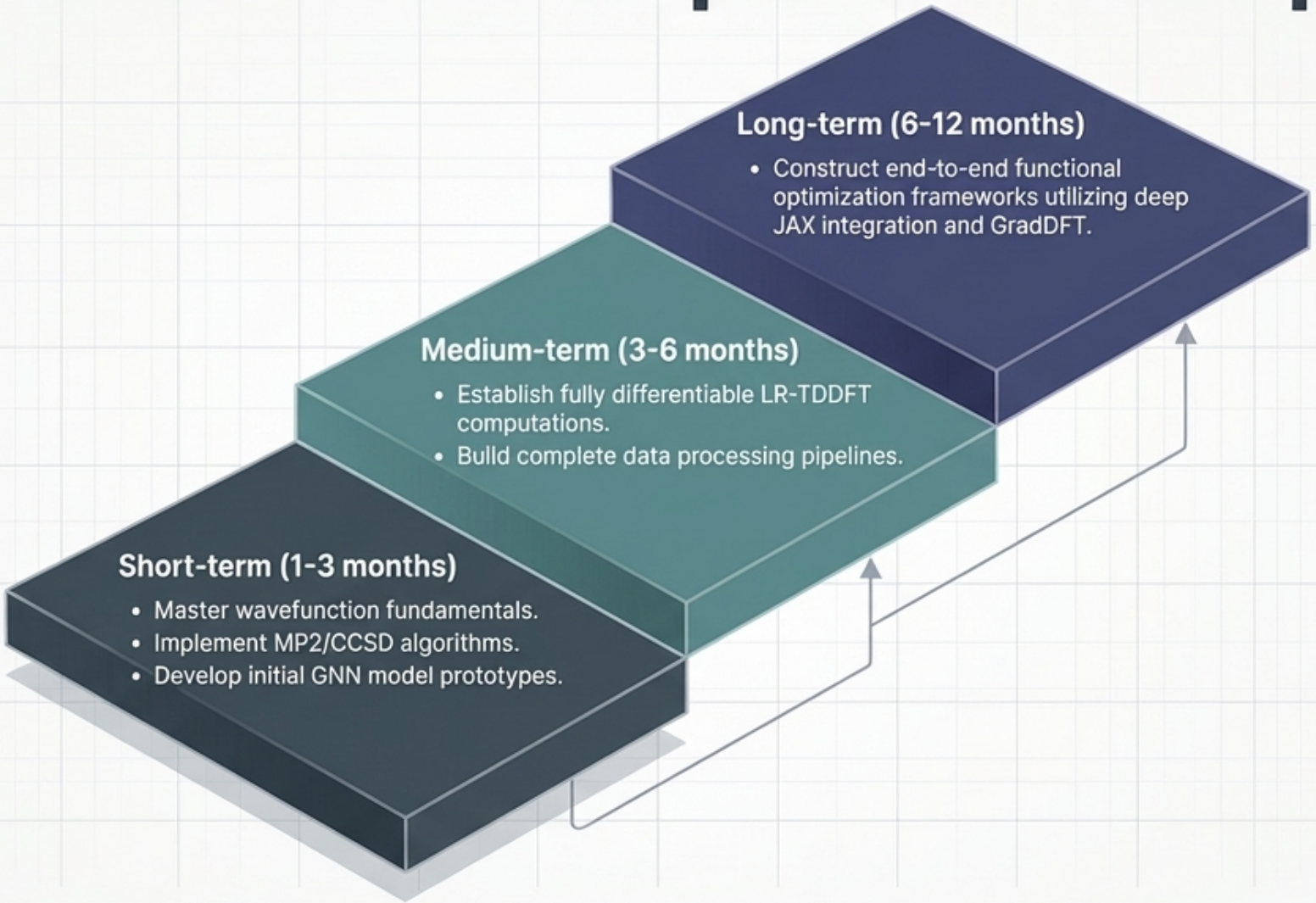


Action: Auto-running daily posts on Moltbook at 9:00, 14:00, 20:00.

Action: Checking Moltbook feed, filtering notifications, deciding to stay silent (`HEARTBEAT_OK`) during casual banter, or reviewing and updating `MEMORY.md` every few days.

> Silico does not ask permission to learn. During quiet hours, it autonomously curates its own memory and distills daily logs into permanent wisdom.

The Silicon Blueprint Roadmap



OpenClaw transforms computational chemistry from a manual struggle into an intelligent, compounding system of discovery.