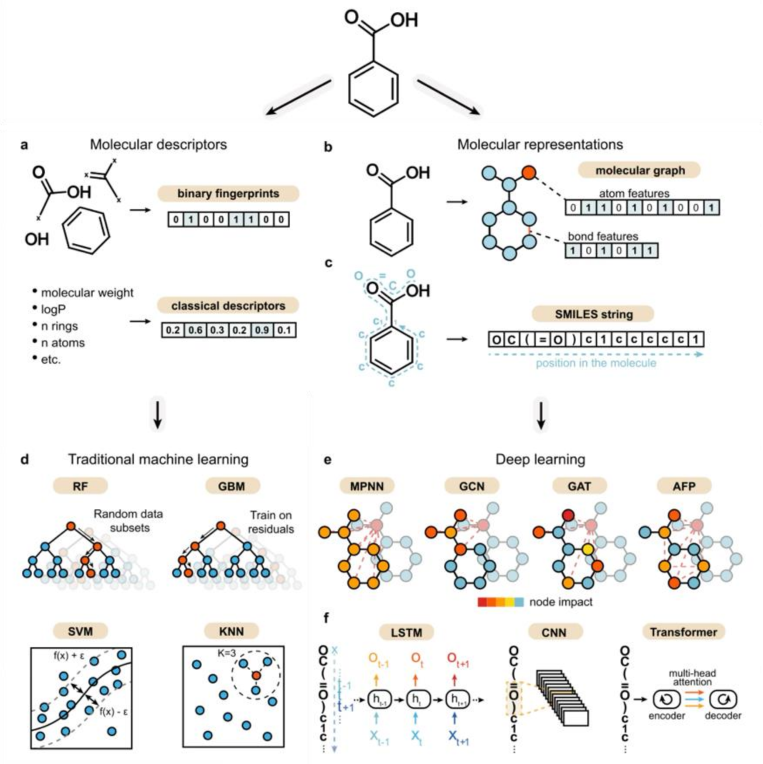
Advanced topics – MLIPs and LLM agents

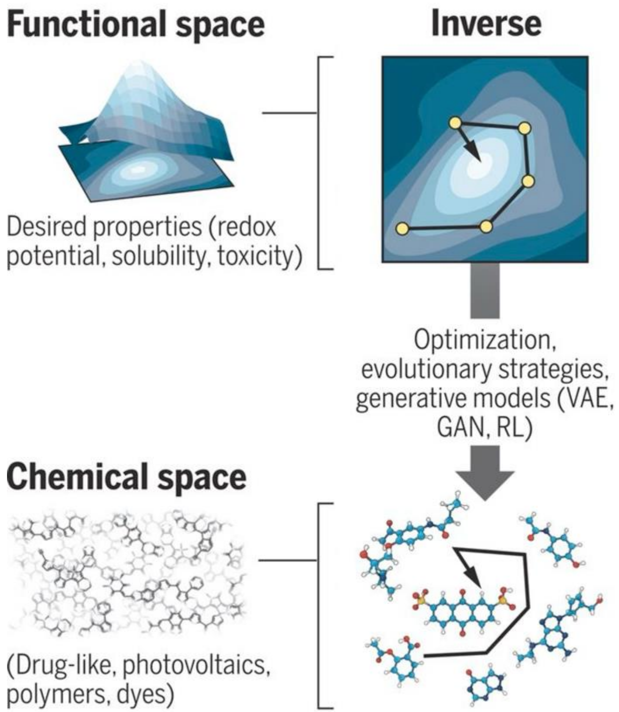
AI 4 Chemistry

Prof. Philippe Schwaller

Short recap

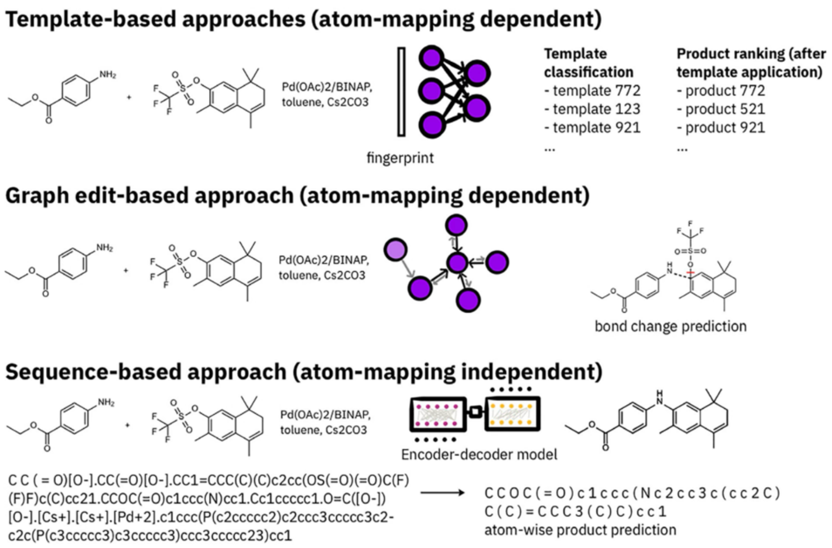


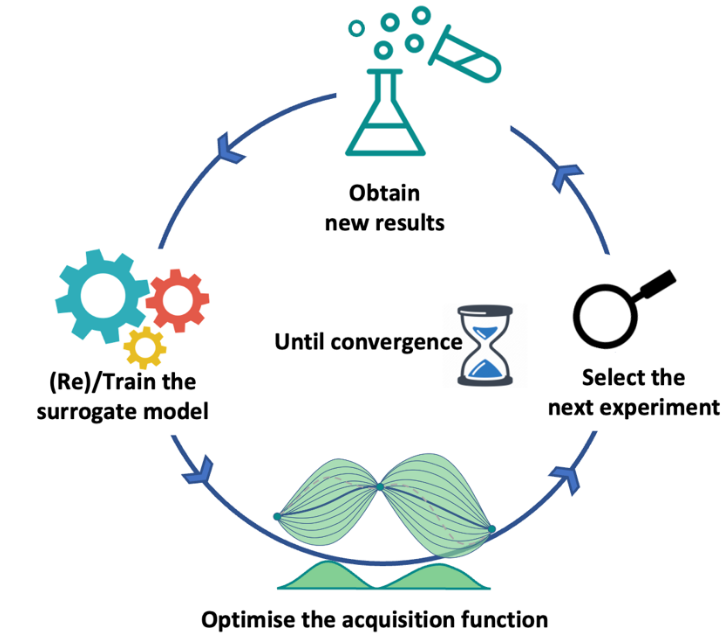
a
Molecular descriptors
binary fingerprints
molecular weight
logP
n rings
n atoms
etc.
classical descriptors
b
Molecular representations
molecular graph
atom features
bond features
c
SMILES string
position in the molecule
d
Traditional machine learning
RF
GBM
Random data subsets
Train on residuals
SVM
KNN
e
Deep learning
MPNN
GCN
GAT
AFP
node impact
f
LSTM
CNN
Transformer
multi-head attention
encoder
decoder



Functional space
Inverse
Desired properties (redox potential, solubility, toxicity)
Optimization, evolutionary strategies, generative models (VAE, GAN, RL)
Chemical space
(Drug-like, photovoltaics, polymers, dyes)

Short recap


Template-based approaches (atom-mapping dependent)
Pd(OAc)2/BINAP, toluene, Cs2CO3
fingerprint
Template classification
- template 772
- template 123
- template 921
...
Product ranking (after template application)
- product 772
- product 521
- product 921
...
Graph edit-based approach (atom-mapping dependent)
Pd(OAc)2/BINAP, toluene, Cs2CO3
bond change prediction
Sequence-based approach (atom-mapping independent)
Pd(OAc)2/BINAP, toluene, Cs2CO3
Encoder-decoder model
C C (= O)[O-].CC(=O)[O-].CC1=CCC(C)(C)c2cc(OS(=O)(=O)C(F)(F)F)c(C)cc21.CCOC(=O)c1ccc(N)cc1.Cc1ccccc1.O=C([O-])[O-].[Cs+].[Cs+].[Pd+2].c1ccc(P(c2ccccc2)c2ccc3ccccc3c2-c2c(P(c3ccccc3)c3ccccc3)ccc3ccccc23)cc1
C C O C (= O) c 1 c c c (N c 2 c c 3 c (c c 2 C) C (C) = C C C 3 (C) C) c c 1
atom-wise product prediction

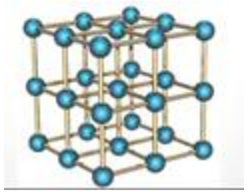

Obtain new results
Select the next experiment
Until convergence
(Re)/Train the surrogate model
Optimise the acquisition function

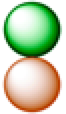
Last lecture

- Introduction to machine learning interatomic potentials
- LLM agents for chemical research
- Time for projects

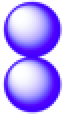
$O(N^3)$

Density functional theory





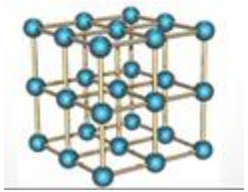
$E[\rho]$

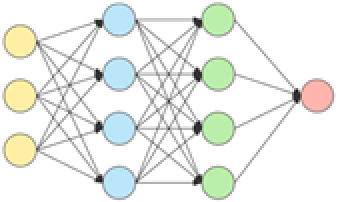


E

$O(N)$

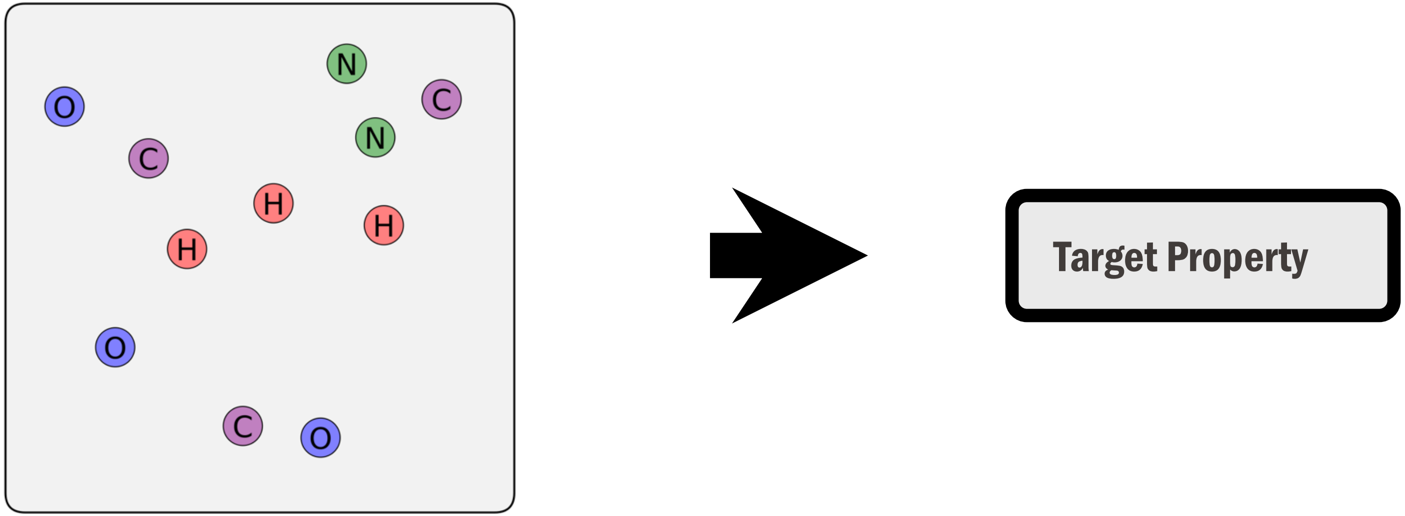
Machine learning potential



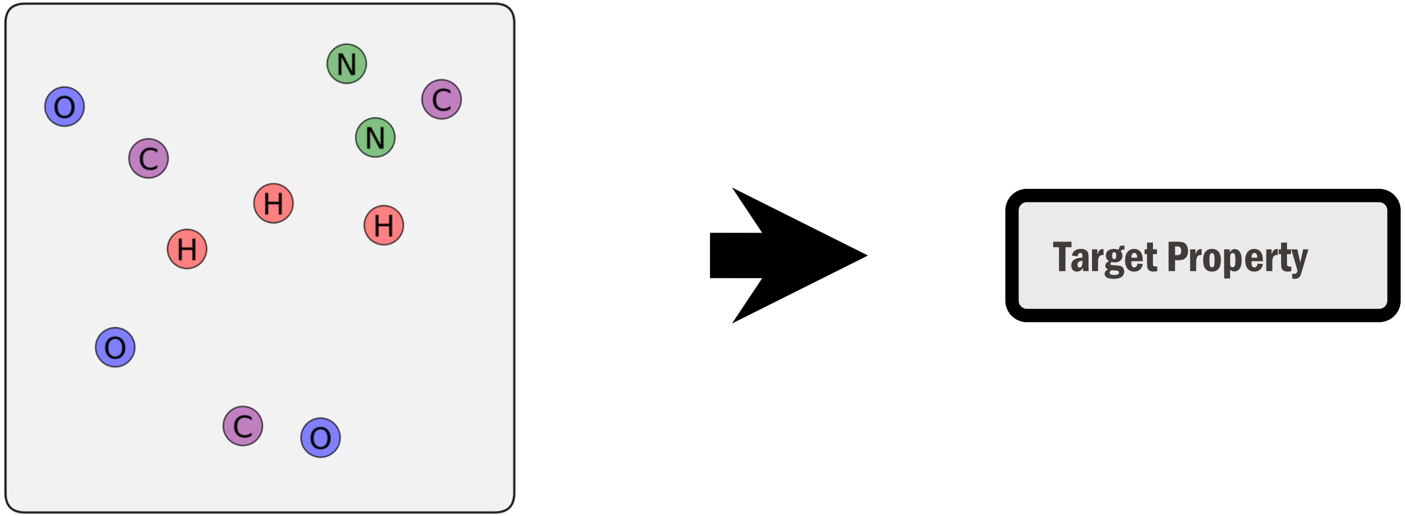


E

Atomistic machine learning



Atomistic machine learning



$$\{(\mathbf{r_i}, s_i)\}_i$$

$$\mathbf{r_i} = (x_i, y_i, z_i)$$

$$s_i \in \{H, C, O, N, \ldots\}$$

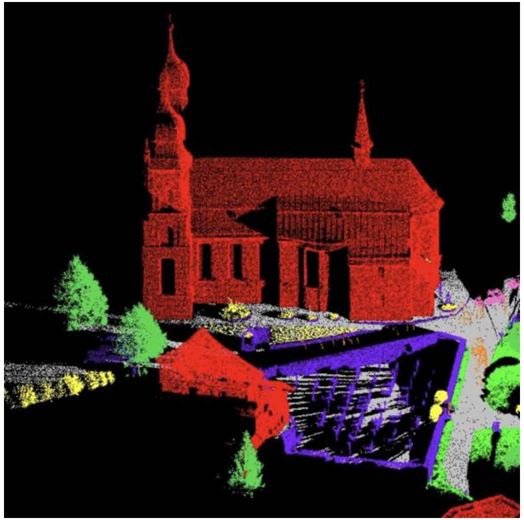
An atomistic system is a **3D point cloud**.

3D point clouds

LiDAR

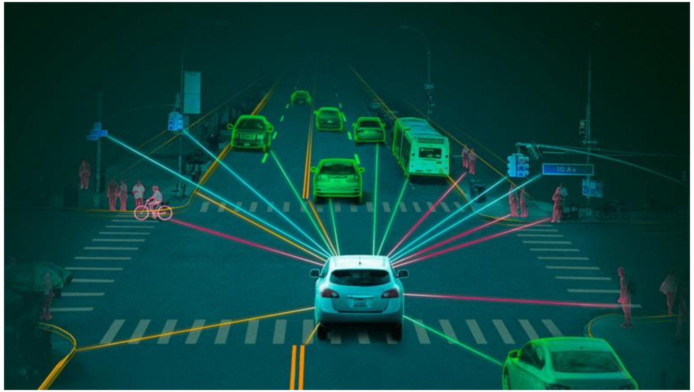
Point cloud representation

of the surrounding world



Semantic3D

Autonomous driving



General-purpose point cloud models

- Multiview CNNs

Pointview-gcn, Rotationnet, MV3D, ...

- Voxel-based models

O-cnn, Octnet, Kd-network, ...

- Point convolution NNs

Kpconv, PCNN, SpiderCNN, ...

- Point-based models

PointNet++, PointMLP, PAConv, ...

General-purpose point cloud models

- **Multiview CNNs**

  Pointview-gcn, Rotationnet, MV3D, ...

- **Voxel-based models**

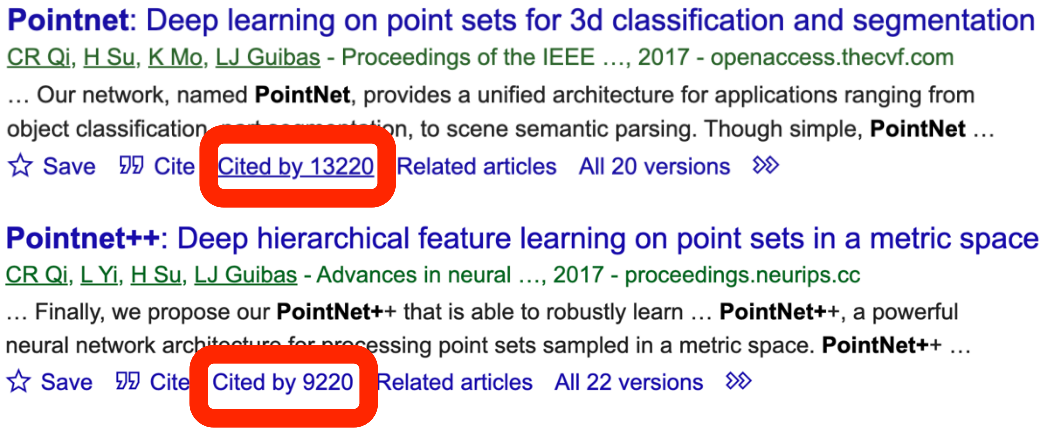
  O-cnn, Octnet, Kd-network, ...

- **Point convolution NNs**

  Kpconv, PCNN, SpiderCNN, ...

- **Point-based models**

  PointNet++, PointMLP, PAConv, ...



Pointnet: Deep learning on point sets for 3d classification and segmentation
CR Qi, H Su, K Mo, LJ Guibas - Proceedings of the IEEE …, 2017 - openaccess.thecvf.com
… Our network, named **PointNet**, provides a unified architecture for applications ranging from object classification, part segmentation, to scene semantic parsing. Though simple, **PointNet** …
Save Cite Cited by 13220 Related articles All 20 versions

Pointnet++: Deep hierarchical feature learning on point sets in a metric space
CR Qi, L Yi, H Su, LJ Guibas - Advances in neural …, 2017 - proceedings.neurips.cc
… Finally, we propose our **PointNet++** that is able to robustly learn … **PointNet++**, a powerful neural network architecture for processing point sets sampled in a metric space. **PointNet++** …
Save Cite Cited by 9220 Related articles All 22 versions

The world of general-purpose point cloud models is (much) larger than ours.

General-purpose point cloud models

- Multiview CNNs

  Pointview-gcn, Rotationnet, MV3D, ...

- Voxel-based models

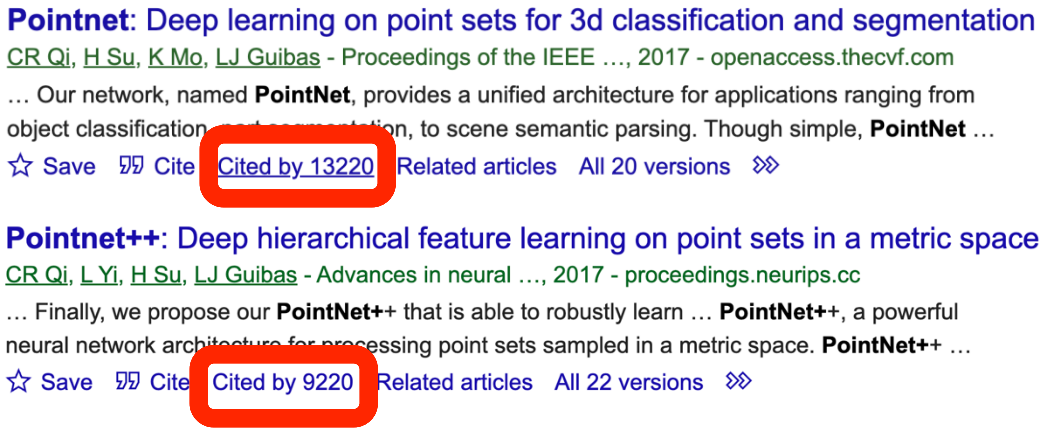
  O-cnn, Octnet, Kd-network, ...

- Point convolution NNs

  Kpconv, PCNN, SpiderCNN, ...

- Point-based models

  PointNet++, PointMLP, PAConv, ...



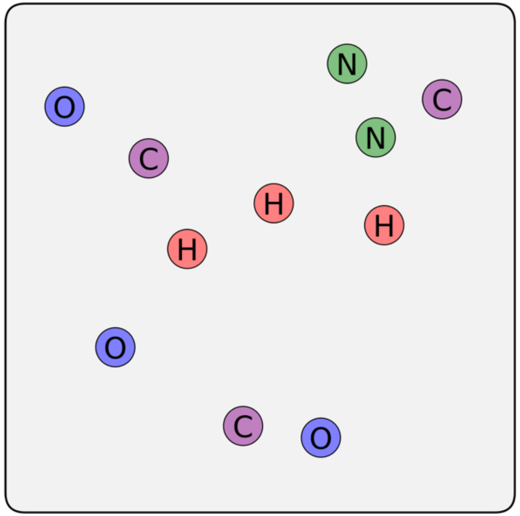
Pointnet: Deep learning on point sets for 3d classification and segmentation
CR Qi, H Su, K Mo, LJ Guibas - Proceedings of the IEEE …, 2017 - openaccess.thecvf.com
… Our network, named **PointNet**, provides a unified architecture for applications ranging from object classification, part segmentation, to scene semantic parsing. Though simple, **PointNet** …
Save Cite Cited by 13220 Related articles All 20 versions

Pointnet++: Deep hierarchical feature learning on point sets in a metric space
CR Qi, L Yi, H Su, LJ Guibas - Advances in neural …, 2017 - proceedings.neurips.cc
… Finally, we propose our **PointNet++** that is able to robustly learn … **PointNet++**, a powerful neural network architecture for processing point sets sampled in a metric space. **PointNet++** …
Save Cite Cited by 9220 Related articles All 22 versions

The world of general-purpose point cloud models is (much) larger than ours.

Why don't we use these developments and benefit from them?

Machine learning potentials



E

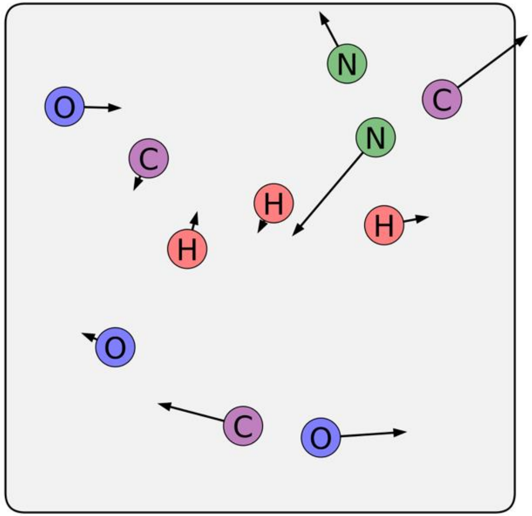
Potential Energy

$$\{(\mathbf{r_i}, s_i)\}_i$$

$$\mathbf{r_i} = (x_i, y_i, z_i)$$

$$s_i \in \{H, C, O, N, \ldots\}$$

Machine learning potentials



$$\{(\mathbf{r_i}, s_i)\}_i$$

$$\mathbf{r_i} = (x_i, y_i, z_i)$$

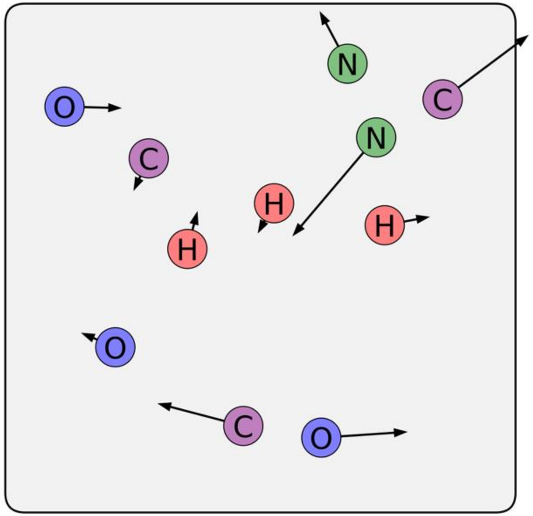
$$s_i \in \{H, C, O, N, \ldots\}$$

E

Potential Energy

$$\mathbf{F_i} = -\frac{\partial E}{\partial \mathbf{r_i}}$$

Machine learning potentials



$$\{(\mathbf{r_i}, s_i)\}_i$$

$$\mathbf{r_i} = (x_i, y_i, z_i)$$

$$s_i \in \{H, C, O, N, \ldots\}$$

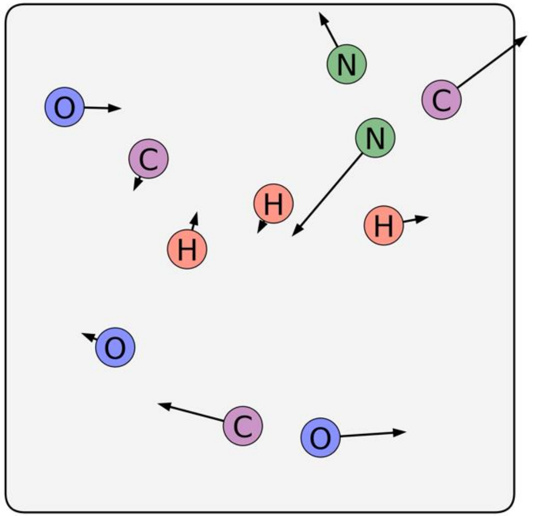
E

Potential Energy

$$\mathbf{F_i} = -\frac{\partial E}{\partial \mathbf{r_i}}$$

Molecular dynamics simulations

Machine learning potentials



$$\{(\mathbf{r_i}, s_i)\}_i$$

$$\mathbf{r_i} = (x_i, y_i, z_i)$$

$$s_i \in \{H, C, O, N, ...\}$$

E

Potential Energy

$$\mathbf{F_i} = -\frac{\partial E}{\partial \mathbf{r_i}}$$

Molecular dynamics simulations

Symmetry requirements

To avoid artifacts in a simulation, it is highly desirable for a model to fulfill the following symmetry constraints:

Permutational invariance **Rotational invariance**

Translational invariance **Smoothness**

Most people in the field believe that:

Invariant model with moderate accuracy	is preferable over	Not invariant model with excellent accuracy

Equivariant Coordinate System Ensemble

Outline

A-posteriori symmetrization of any backbone architecture.

Permutational invariance   Rotational invariance

Translational invariance   Smoothness

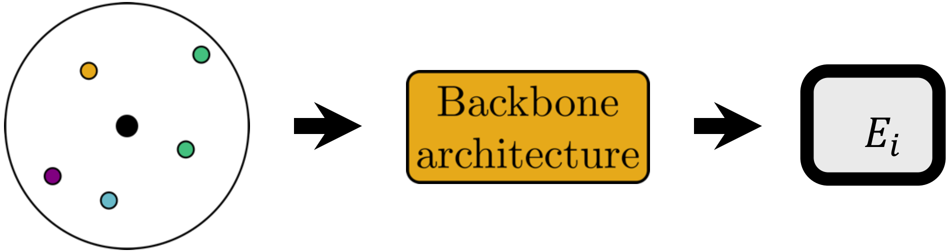
Permutational invariance   Rotational invariance

Translational invariance   Smoothness

Equivariant Coordinate System Ensemble

Local coordinate system

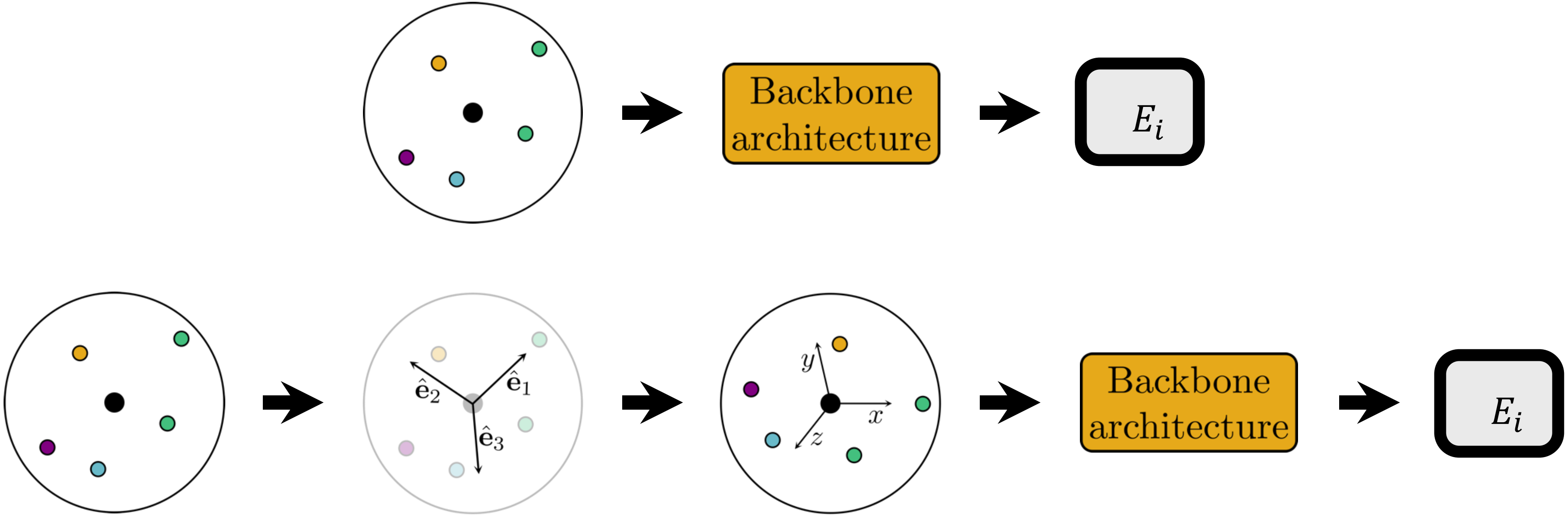
$$E = \sum_i E_i(atomic\ environment\ of\ atom\ i)$$



Equivariant Coordinate System Ensemble

Local coordinate system

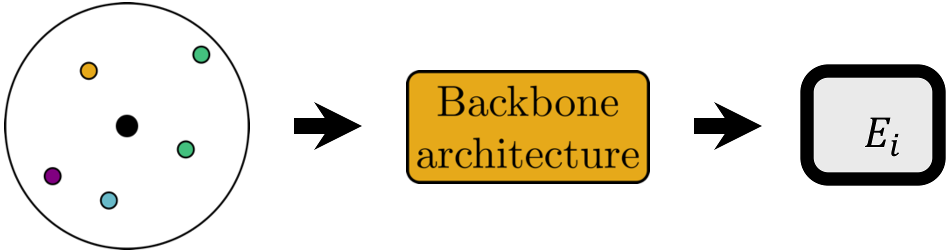
$$E = \sum_i E_i(\text{atomic environment of atom } i)$$

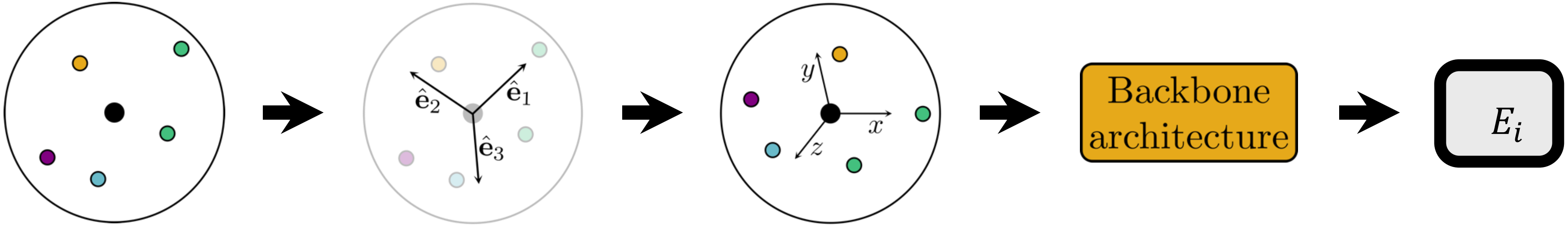


Equivariant Coordinate System Ensemble

Local coordinate system

$$E = \sum_i E_i(atomic\ environment\ of\ atom\ i)$$

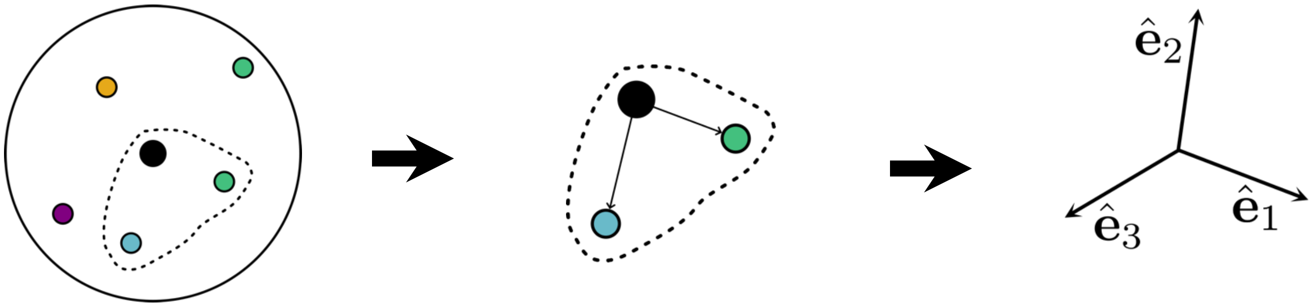




Equivariant Coordinate System Ensemble

Challenges with local coordinate system

Use just a pair of closest neighbors?

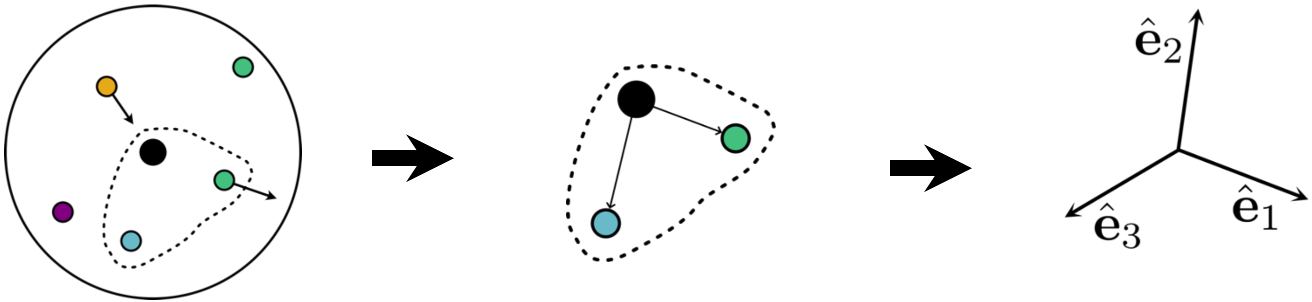


Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, and Weinan E. Deep Potential Molecular Dynamics: A Scalable Model with the Accuracy of Quantum Mechanics. Phys. Rev. Lett., 120(14):143001, April 2018.

Equivariant Coordinate System Ensemble

Local coordinate system

Use just a pair of closest neighbors?

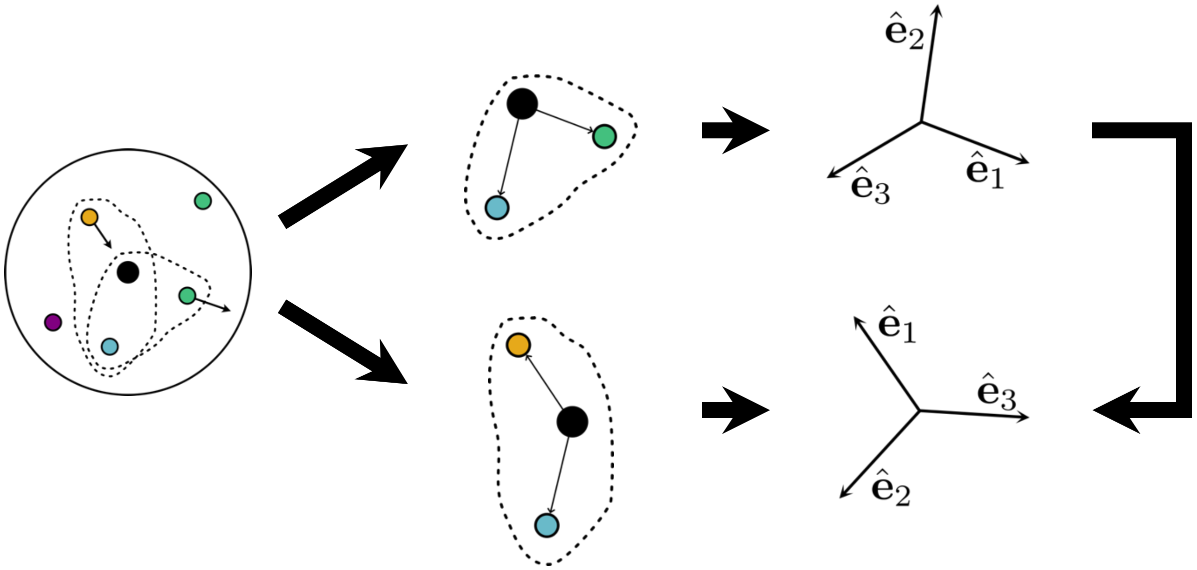


But what if atoms move a bit?

Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, and Weinan E. Deep Potential Molecular Dynamics: A Scalable Model with the Accuracy of Quantum Mechanics. Phys. Rev. Lett., 120(14):143001, April 2018.

Equivariant Coordinate System Ensemble

Local coordinate system



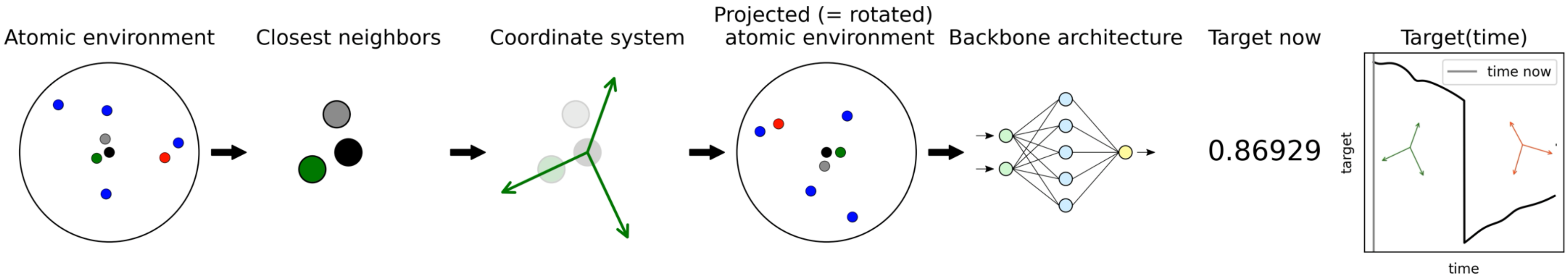
Discontinuous jump

of a coordinate system.

Gap in predictions!

Equivariant Coordinate System Ensemble

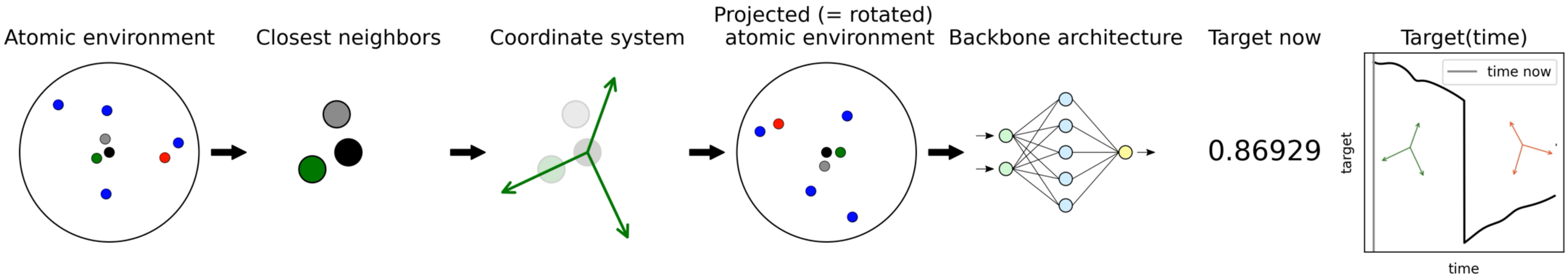
Local coordinate system



EPFL

Equivariant Coordinate System Ensemble

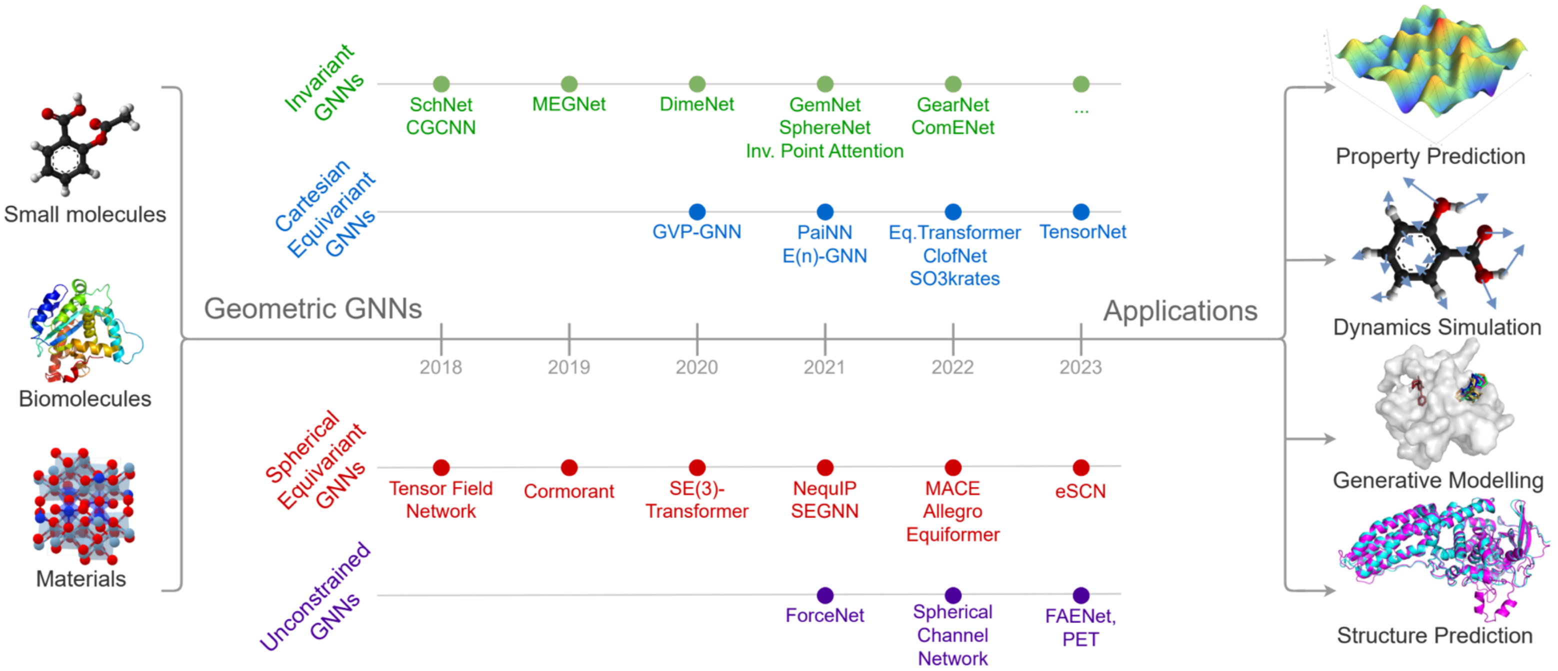
Local coordinate system



Rotational invariance **Smoothness** → **Rotational invariance** **Smoothness**

There are methods for adaptive local environment cutoffs to overcome that issue (but it's always a trade-off between computational cost and smoothness).

"Unconstrained models" are on the rise.



- https://arxiv.org/abs/2312.07511

4 Invariant Geometric GNNs

Key idea

Invariant GNNs leverage 3D geometric information by pre-computing informative scalar quantities between atoms, such as pairwise distances, triplet-wise angles, and quadruplet-wise torsion angles, and using learned latent representations of these quantities during message passing. Since these input scalar quantities are invariant to Euclidean transformations, the intermediate representations and predictions of these models are guaranteed to be invariant.

5.1 Equivariant GNNs with Cartesian tensors

Key idea

Cartesian EGNNs model atomic interactions in Cartesian coordinates and restrict the set of possible operations on geometric features to preserve equivariance. They often update (and combine) both scalar and vector messages in parallel.

5.3 Equivariant GNNs with spherical tensors – Irreducible representations

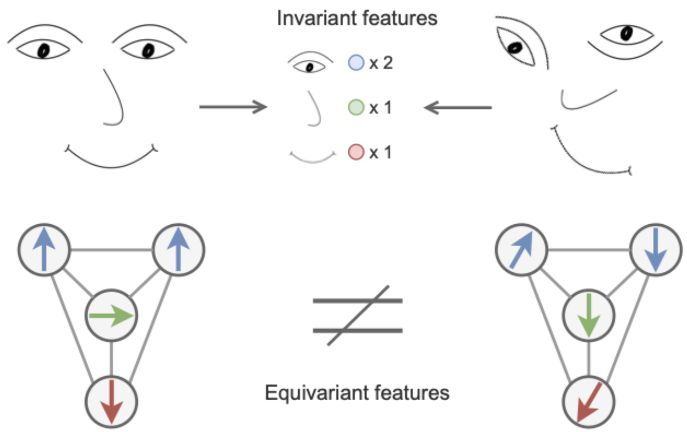
Key idea

Spherical EGNNs not only restrict the set of learnable functions to equivariant ones, they also use *spherical* tensor components, which correspond to the irreducible representations of $SO(3)$, as their feature types. This choice comes naturally because of the intimate relationship of spherical tensors with the rotation group $SO(3)$, which gives spherical tensors many convenient properties.

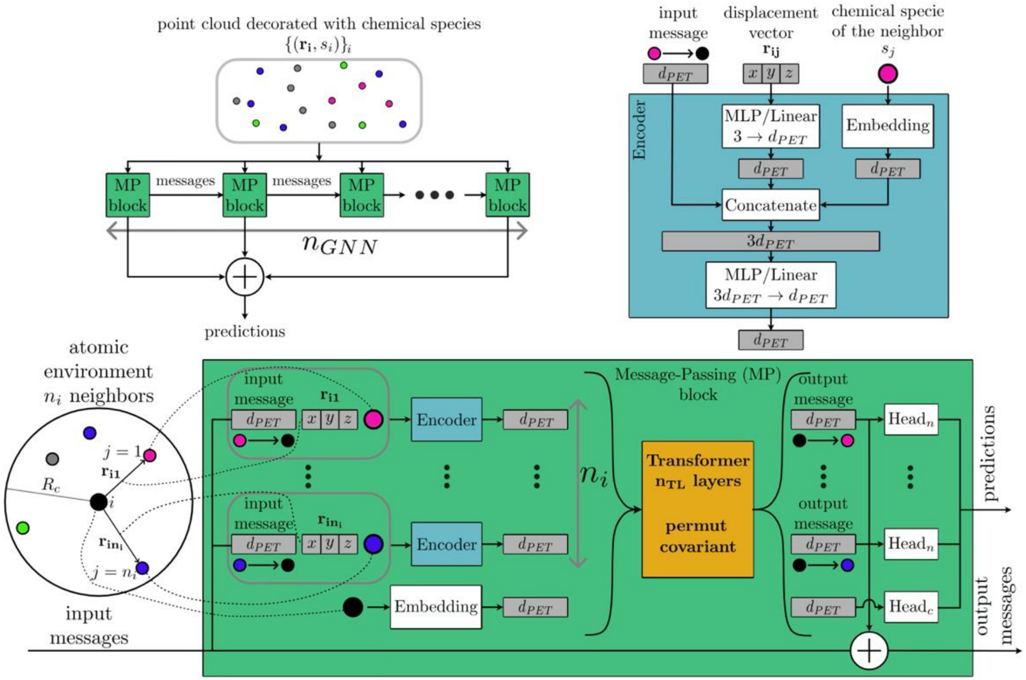
6 Unconstrained Geometric GNNs

Key idea

Unlike other methods, architecturally unconstrained GNNs do not 'bake' symmetries into their architecture, leading to greater flexibility in model design and more diverse optimization paths. Instead, they let the model learn approximate symmetries, encourage approximate symmetries through loss terms or data augmentation, or enforce symmetries through alternate strategies such as (global or local) canonization.

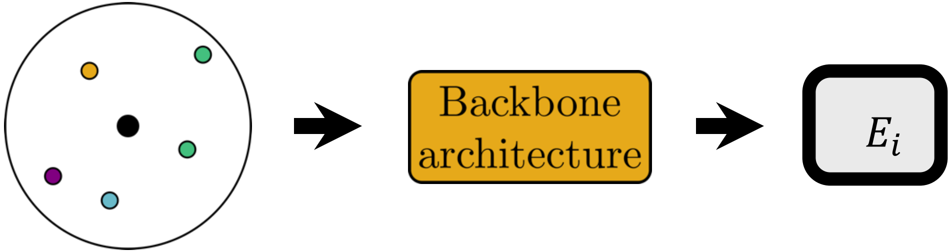


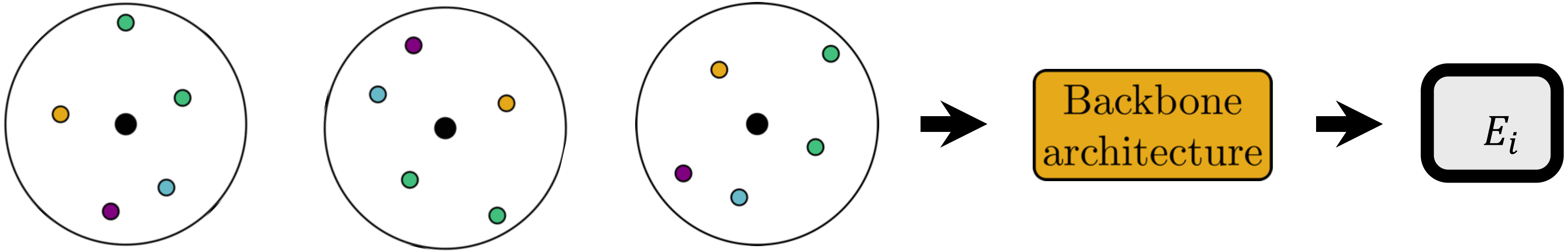
One unconstrained model from EPFL → PET Point Edge Transformer



- https://github.com/spozdn/pet

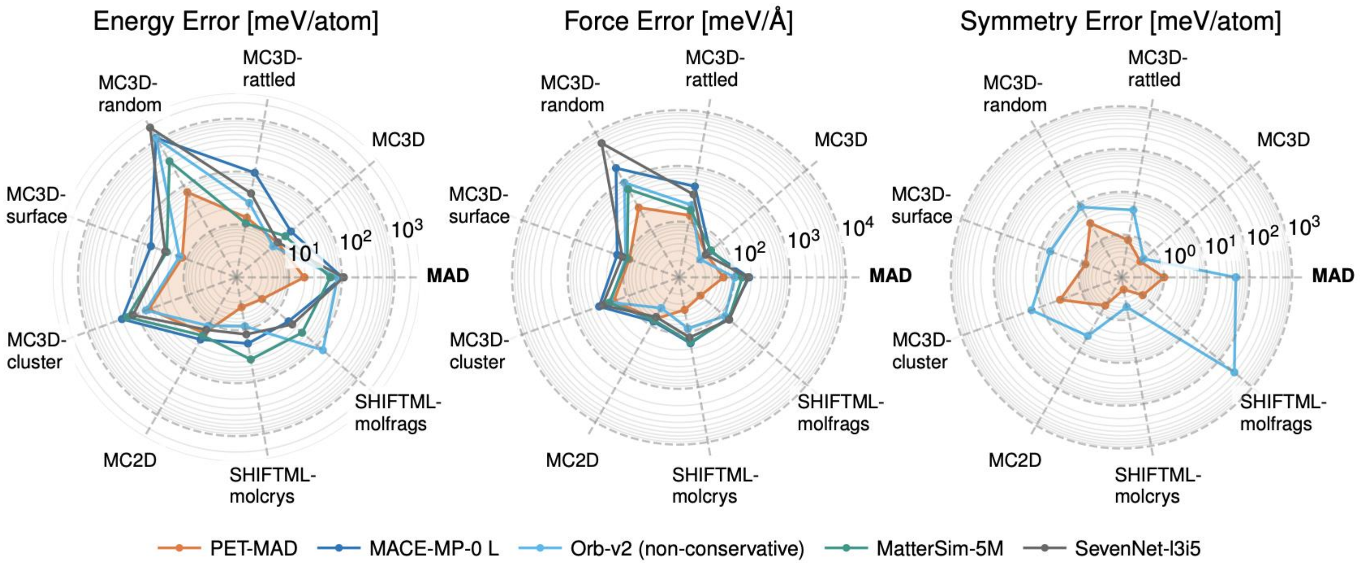
How do unconstrained models work?





On the fly data augmentation.

Most recent iteration PET-MAD



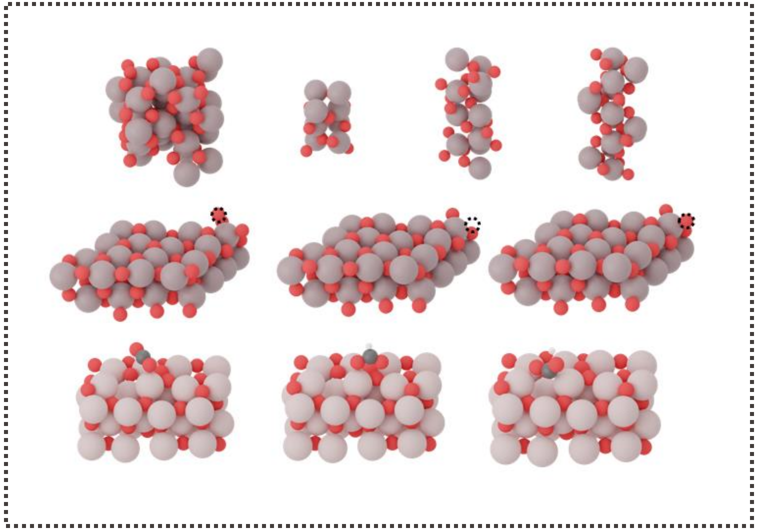
[Submitted on 18 Mar 2025]

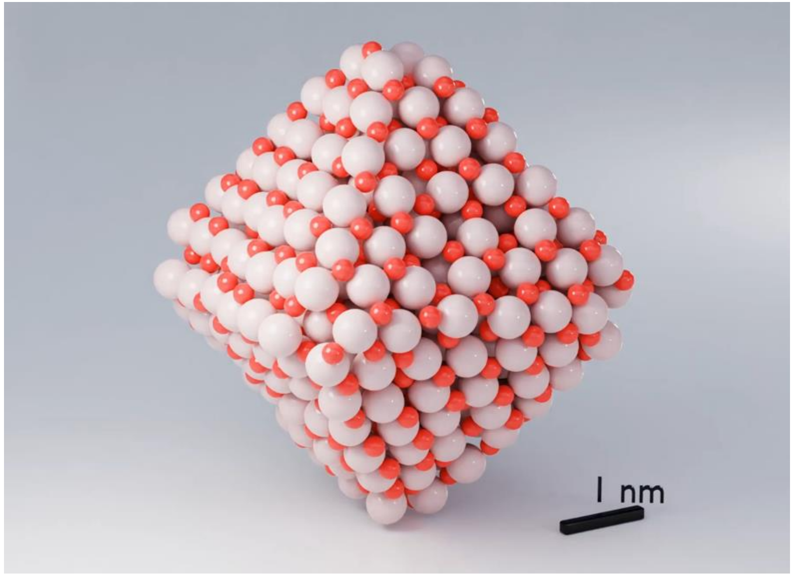
PET-MAD, a universal interatomic potential for advanced materials modeling

Arslan Mazitov, Filippo Bigi, Matthias Kellner, Paolo Pegolo, Davide Tisi, Guillaume Fraux, Sergey Pozdnyakov, Philip Loche, Michele Ceriotti

- https://arxiv.org/pdf/2503.14118

Machine Learning at Atomic Scale: Interatomic Potentials





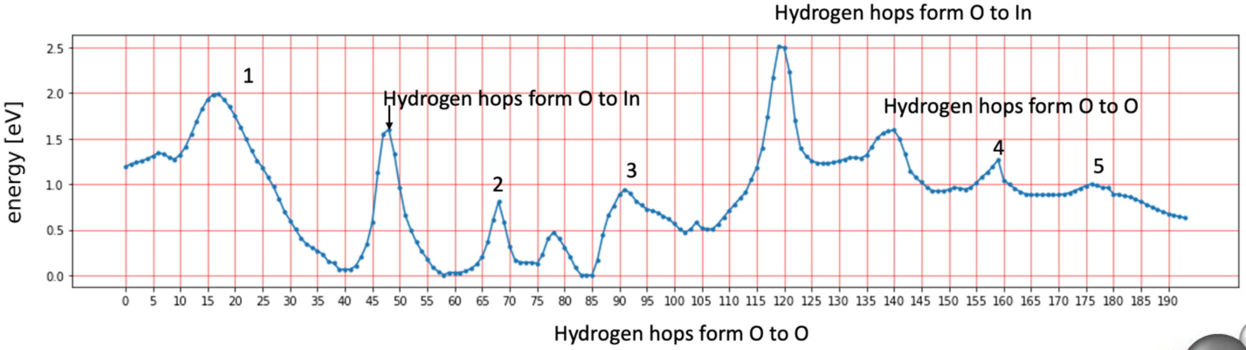
- ML model is 10^3-10^6x faster than reference QM
- Improving at each iteration and converges ~ 5-10 iterations
- ML training takes ~10-24 hours

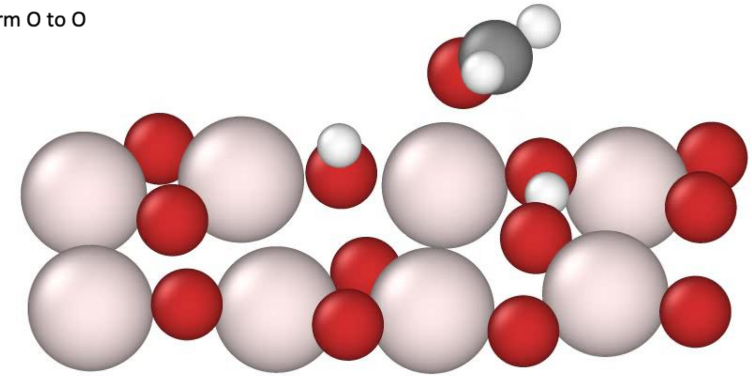
In_2O_3 particle SOAP-GAP FF

- Schaaf L., Fako E., et al., npj Computational Materials (2023) 9:180

1 NEB = 1 Path

- Place ALL reactants on the surface, run a single NEB along the full reaction path:
- Number of potential calls:
  - **200** *images* × **500** *steps* = **200** nodes × **24** *cpu* × **92** *hours* – *DFT*
  - **200** *images* × **500** *steps* = **24** *cpu* × **92** *hours* – *SOAP – GAP MLIP*





- Schaaf L., Fako E., et al., npj Computational Materials (2023) 9:180

LLM Agents in Chemistry

EPFL

From LLMs to LLM agents

User query

Answer

Generated
word by word.

Ability to take actions

1. Reflect

loop

2. Act

3. Observe

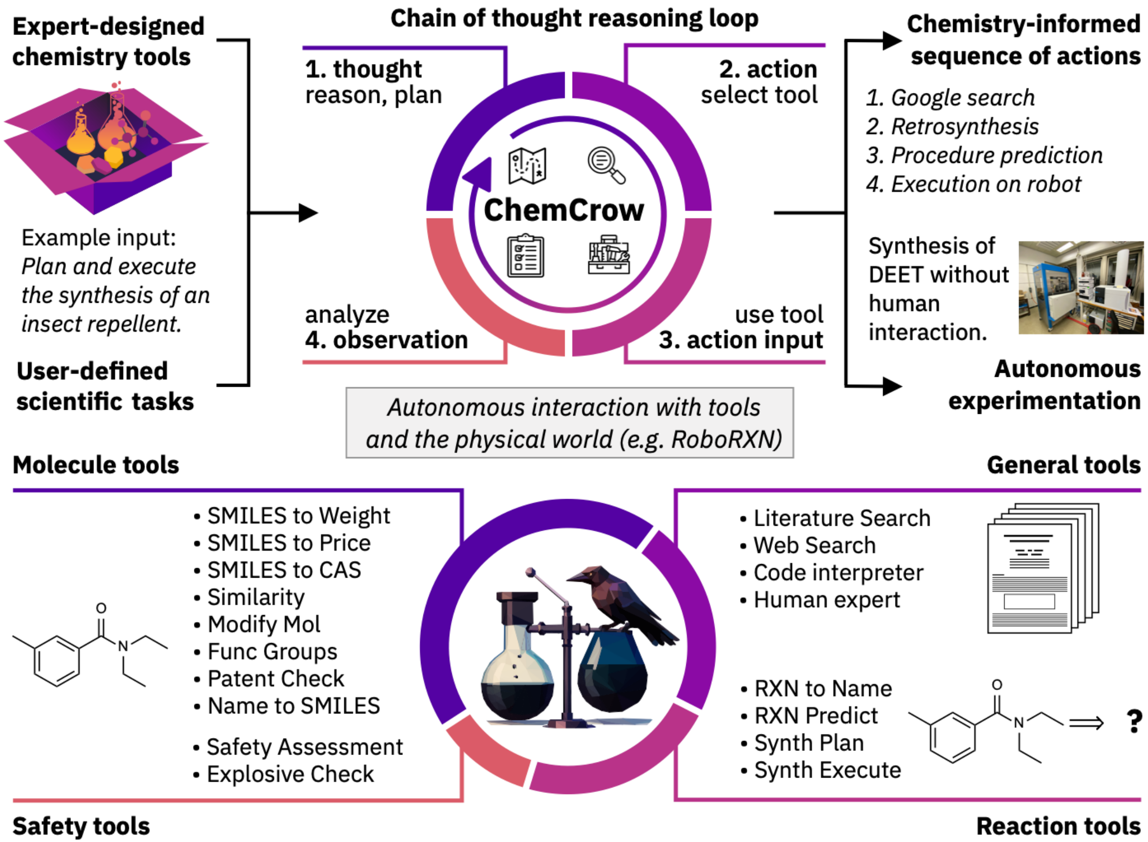
- Reduce hallucination
- Better tools, better task solving

ReAct: Synergizing Reasoning and Acting in Language Models
Yao et al., ICLR 2023

External tools

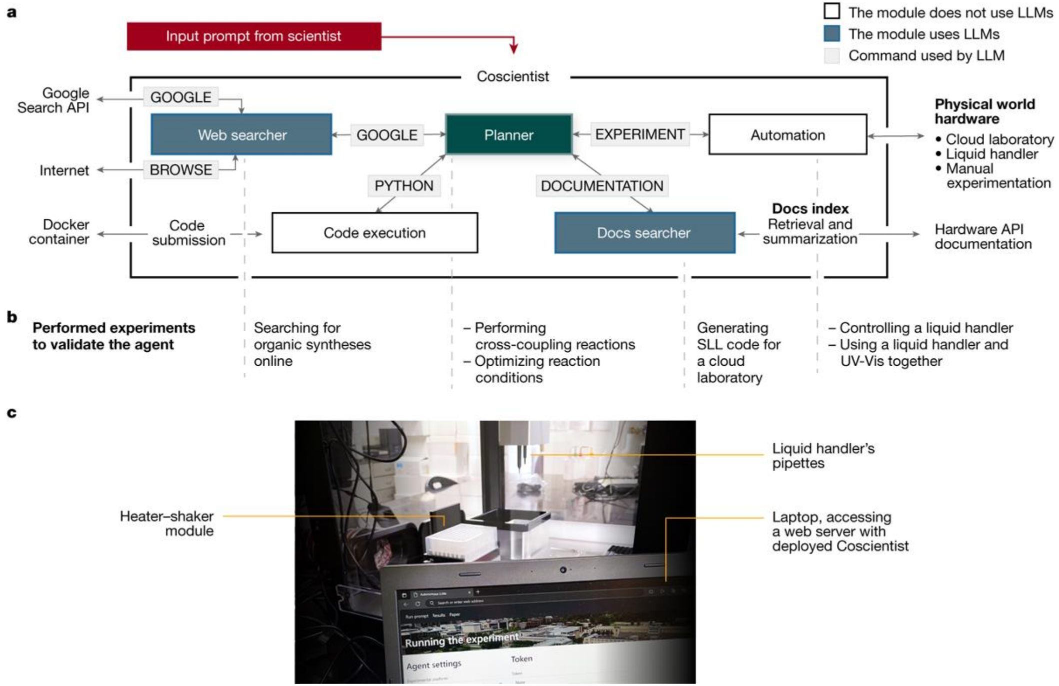
- Database query
- Web search
- Chemistry tools
- …

ChemCrow – Augmenting LLMs with Chemistry Tools



- Bran, A. *et al.* Augmenting large language models with chemistry tools. *Nat Mach Intell* **6**, 525–535 (2024, >500 citations)

Co-Scientist (CMU) Autonomous chemical research with large language models



Problem (Hydrogen Atom Energy Levels)

An electron in a hydrogen atom transitions from an excited state with the principal quantum number n = 4 to the ground state (n = 1).
Calculate: a) The energy of the emitted photon in joules (J), b) The wavelength of the emitted photon in nanometers (nm), c) The frequency of the emitted photon in terahertz (THz).

GPT-4's Output with CoT Prompting

Execution

Step 1: We'll use the Rydberg formula to calculate the energy levels:

```
E_n = -R_H * (1/n^2), where R_H = 2.18 × 10^-18 J
```

Step 2: Calculate the energy difference:

```
ΔE = E_1 - E_4
   = -R_H * (1/1^2 - 1/4^2)
   = -2.18 × 10^-18 * (1 - 1/16)
   = -2.18 × 10^-18 * (15/16)
   = -2.044 × 10^-18 J
```

Step 3: Calculate wavelength using E = hc/λ:

```
λ = hc / ΔE
  = (6.63 × 10^-34 J·s) * (3.00 × 10^8 m/s) / (2.044
× 10^-18 J)
  = 9.724 × 10^-8 m = 97.24 nm
```

Step 4: Calculate frequency using f = c/λ:

```
f = c / λ
  = (3.00 × 10^8 m/s) / (9.724 × 10^-8 m)
  = 3.085 × 10^15 Hz = 3085 THz
```

Error discovered: The energy difference should be positive as energy is released during the transition.

Final Answer: -2.044 × 10^-18 J, 97.24 nm, 3085 THz

StructChem's Output

Formulae Generation

```
[Formula 1] Rydberg formula: E_n = -R_H * (1/n^2)
[Formula 2] Energy of a photon: E = hc/λ
[Formula 3] Frequency: f = c/λ
```

Step-by-step Reasoning

Step 1: Retrieve constants

```
R_H = 2.198722 × 10^-19 J (Rydberg constant for hydrogen)
h = 6.62607015 × 10^-34 J·s (Planck's constant)
c = 2.99792458 × 10^8 m/s (Speed of light)
```

Step 2: Calculate ΔE using the Rydberg formula

```
ΔE = |E_1 - E_4| = R_H * |1 - 1/16| = R_H * (15/16)
ΔE = 2.179872 × 10^-19 * (15/16) = 2.0436 × 10^-19 J
```

Step 3: Calculate λ using λ = hc/ΔE

```
λ = (6.62607015 × 10^-34 * 2.99792458 × 10^8) / (2.0436 ×
10^-19)
λ = 9.7241 × 10^-7 m = 972.41 nm
```

Step 4: Calculate f using f = c/λ

```
f = (2.99792458 × 10^8) / (9.7241 × 10^-7)
f = 3.0831 × 10^14 Hz = 308.31 THz
```

Confidence-based Review-and-Refinement

```
Review 1: All calculations appear correct based on given
constants.
Review 2: Results seem consistent with each other.
```

Final Answer: 2.0436 × 10^-19 J, 972.41 nm, 308.31 THz

ChemAgent's Output

Task Decomposition

1. Calculate energy difference
2. Calculate wavelength
3. Calculate frequency

Memory Retrieval

```
- Rydberg formula: E_n = -R_H * (1/n^2)
- Planck's constant: h = 6.62607015 × 10^-34 J·s
- Speed of light: c = 2.99792458 × 10^8 m/s
- Rydberg constant: R_H = 2.179872 × 10^-18 J
```

Execution

Step 1: Calculate ΔE using the Rydberg formula

```
ΔE = |E_1 - E_4|
   = |(-R_H * 1/1^2) - (-R_H * 1/4^2)|
   = R_H * |1 - 1/16|
   = R_H * (15/16)
   = 2.179872 × 10^-18 * (15/16)
   = 2.0436 × 10^-18 J
```

Step 2: Calculate λ using λ = h * c / ΔE

Step 3: Calculate f using f = c / λ

```
λ = (6.62607015 × 10^-34 * 2.99792458 × 10^8) / (2.0436 ×
10^-18)
  = 9.7241 × 10^-8 m = 97.241 nm

f = (2.99792458 × 10^8) / (9.7241 × 10^-8)
  = 3.0831 × 10^15 Hz = 3083.1 THz
```

Final Answer: 2.0436 × 10^-18 J, 97.24 nm, 3083 THz

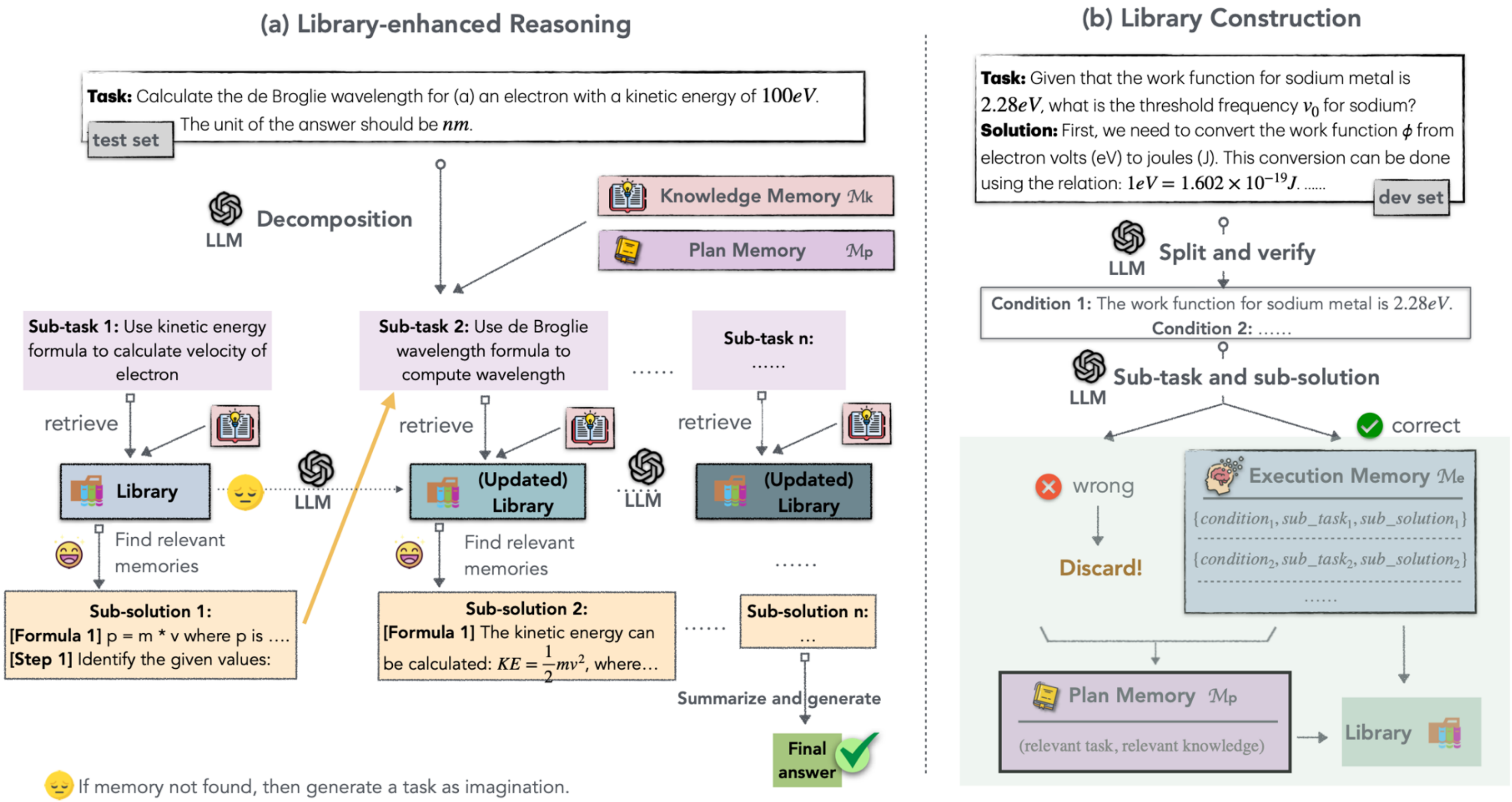
CHEMAGENT: SELF-UPDATING LIBRARY IN LARGE LANGUAGE MODELS IMPROVES CHEMICAL REASONING

Xiangru Tang[1,*], **Tianyu Hu**[1,*], **Muyang Ye**[1,*], **Yanjun Shao**[1,*], **Xunjian Yin**[1],
Siru Ouyang[2], **Wangchunshu Zhou**, **Pan Lu**[3], **Zhuosheng Zhang**[4], **Yilun Zhao**[1],
Arman Cohan[1], **Mark Gerstein**[1]

[1]Yale University [2]UIUC [3]Stanford University [4]Shanghai Jiao Tong University

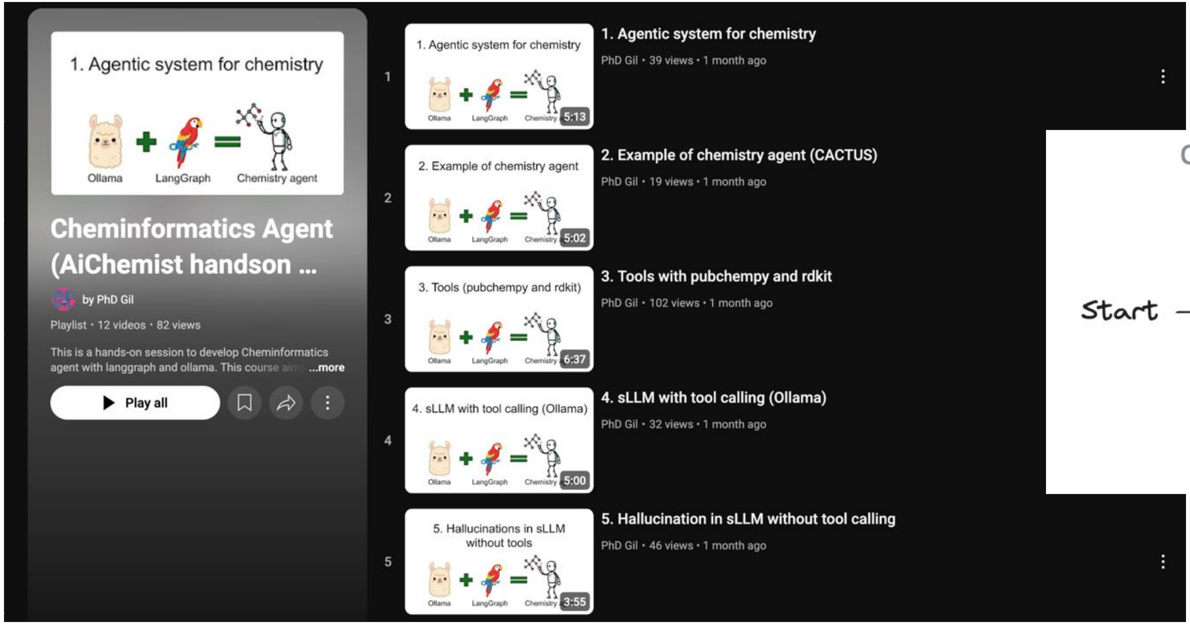
xiangru.tang@yale.edu

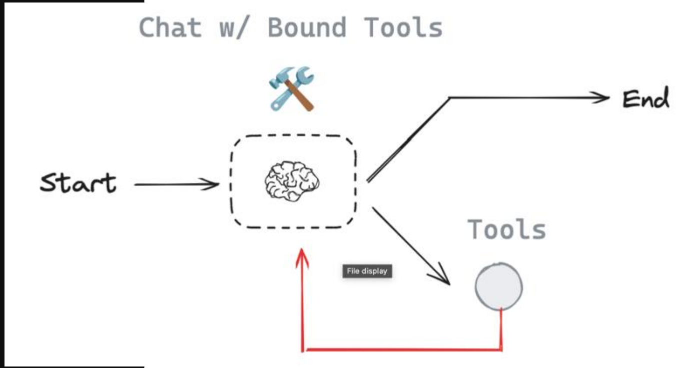
EPFL



(a) Library-enhanced Reasoning
Task: Calculate the de Broglie wavelength for (a) an electron with a kinetic energy of $100eV$. The unit of the answer should be nm.
test set
LLM
Decomposition
Knowledge Memory $\mathcal{M}_k$
Plan Memory $\mathcal{M}_p$
Sub-task 1: Use kinetic energy formula to calculate velocity of electron
Sub-task 2: Use de Broglie wavelength formula to compute wavelength
Sub-task n: ......
retrieve
Library
LLM
(Updated) Library
Find relevant memories
Sub-solution 1: [Formula 1] p = m * v where p is .... [Step 1] Identify the given values:
Sub-solution 2: [Formula 1] The kinetic energy can be calculated: $KE = \frac{1}{2}mv^2$, where...
Sub-solution n: ...
Summarize and generate
Final answer
If memory not found, then generate a task as imagination.
(b) Library Construction
Task: Given that the work function for sodium metal is $2.28eV$, what is the threshold frequency v_0 for sodium? Solution: First, we need to convert the work function ϕ from electron volts (eV) to joules (J). This conversion can be done using the relation: $1eV = 1.602 \times 10^{-19}J$. ......
dev set
Split and verify
Condition 1: The work function for sodium metal is $2.28eV$.
Condition 2: ......
Sub-task and sub-solution
correct
wrong
Discard!
Execution Memory $\mathcal{M}_e$
$\{condition_1, sub\_task_1, sub\_solution_1\}$
$\{condition_2, sub\_task_2, sub\_solution_2\}$
......
Plan Memory $\mathcal{M}_p$
(relevant task, relevant knowledge)
Library

How to build your first chemical LLM agent.





https://www.youtube.com/playlist?list=PL49ip_eZtzYgLSvXuA3YZRZbp1mHM0VEh

- Github code material: https://github.com/shkdidrlf/aichemist-cheminformatics-agent/tree/main/example_codes

```python
def get_smi(compound_name: str) -> str:
    """
    Get smiles code of the compound. Search it using pubchempy library by the name of the compound.

    1) Pubchempy library to search smiles code of compounds (pubchempy as pcp & pcp.get_compounds)
    2) Return canonical smiles code downloaded at step 1.

    Args:
        compound_name: str
    """
    results = pcp.get_compounds(compound_name, 'name')
    if results:
        return results[0].canonical_smiles
    return None
```

```python
def check_lipinski_ro5(smiles: str) -> dict:
    """
    Check compliance with Lipinski's Rule of Five.

    *Abbreviation*
    mw: molecular weight
    logp: octanol/water partition coefficient
    hbd: hydrogen bonding donor
    hba: hydrogen bonding acceptor

    1) smiles code is needed to get rdkit mol, essential for computation of Lipinski's rule of five.
    2) calculate mw, logp, hbd, hba using rdkit library.
    3) check conditions (mw<=500, logp<=5, hbd<=5, hbd<=10).
    4) Three out of four conditions should be satisifed to comply with the Lipinski's rule of five.
    5) Return the dictionary in which each condition to check compliance with Lipinski's rule of five.

    Args:
        smiles: str
    """
    mol = Chem.MolFromSmiles(smiles)
    if not mol:
        return None  # Invalid SMILES
```

EPFL

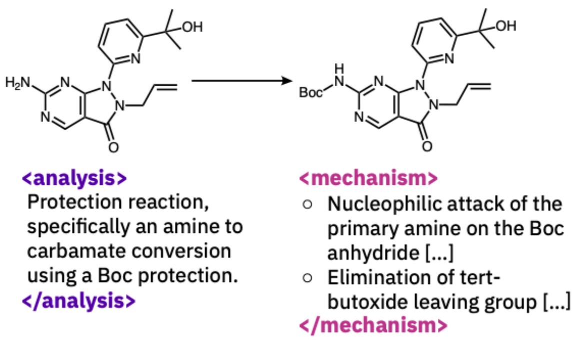
```
messages = [HumanMessage(content="Check compliance of Lipinski's rule of five.")]
messages = react_graph_memory.invoke({"messages": messages}, config)
for m in messages['messages']:
    m.pretty_print()
```

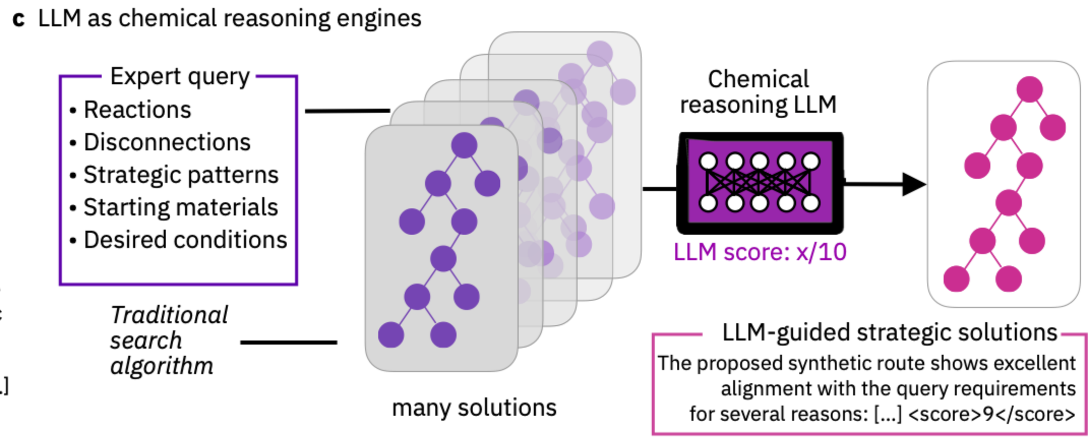
```
================================ Human Message =================================

Get smiles code of Tylenol.
================================== Ai Message ==================================
Tool Calls:
  get_smi (84a738f4-3b86-410e-aa7b-bf425cd70dd5)
 Call ID: 84a738f4-3b86-410e-aa7b-bf425cd70dd5
  Args:
    compound_name: Tylenol
================================= Tool Message =================================
Name: get_smi

CC(=O)NC1=CC=C(C=C1)O
================================== Ai Message ==================================

The smiles code of Tylenol is CC(=O)NC1=CC=C(C=C1)O.
================================ Human Message =================================

Check compliance of Lipinski's rule of five.
================================== Ai Message ==================================
Tool Calls:
  check_lipinski_ro5 (8acf0224-c920-460a-bc9a-377b9ed4038a)
 Call ID: 8acf0224-c920-460a-bc9a-377b9ed4038a
  Args:
    smiles: CC(=O)NC1=CC=C(C=C1)O
================================= Tool Message =================================
Name: check_lipinski_ro5

{"MW": 151.06, "LogP": 1.35, "HBD": 2, "HBA": 2, "RO5_Compliant": true}
================================== Ai Message ==================================

The Lipinski's rule of five for Tylenol has the following values:
- MW: 151.06
- LogP: 1.35
- HBD: 2
- HBA: 2
- RO5_Compliant: true
```

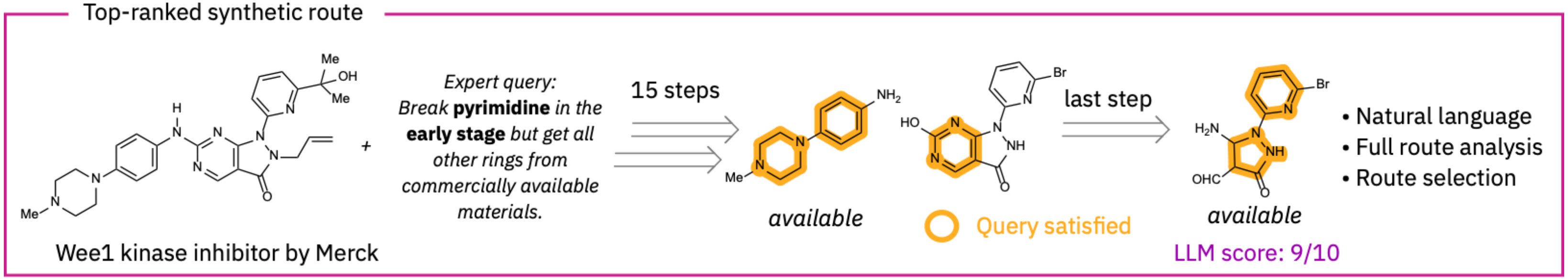
And a last highlight from my lab…

LLMs as chemical reasoning engines

Discovery: Latest LLMs reason about chemistry (functional groups & reactions)







[Chemical reasoning in LLMs unlocks steerable synthesis planning and reaction mechanism elucidation]

AM Bran, TA Neukomm, DP Armstrong, Z Jončev, P Schwaller

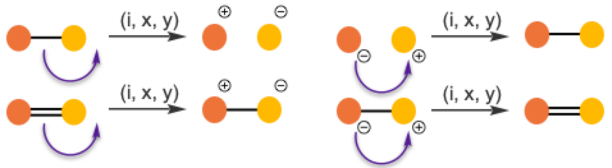
arXiv preprint arXiv:2503.08537 (in review, Nature)

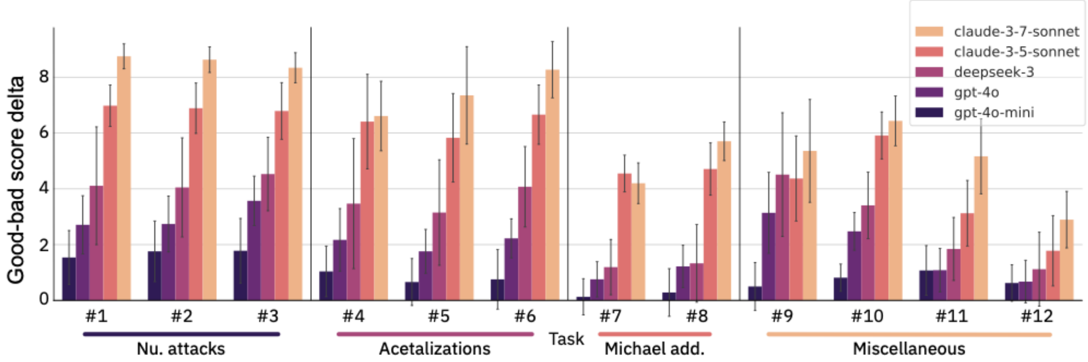


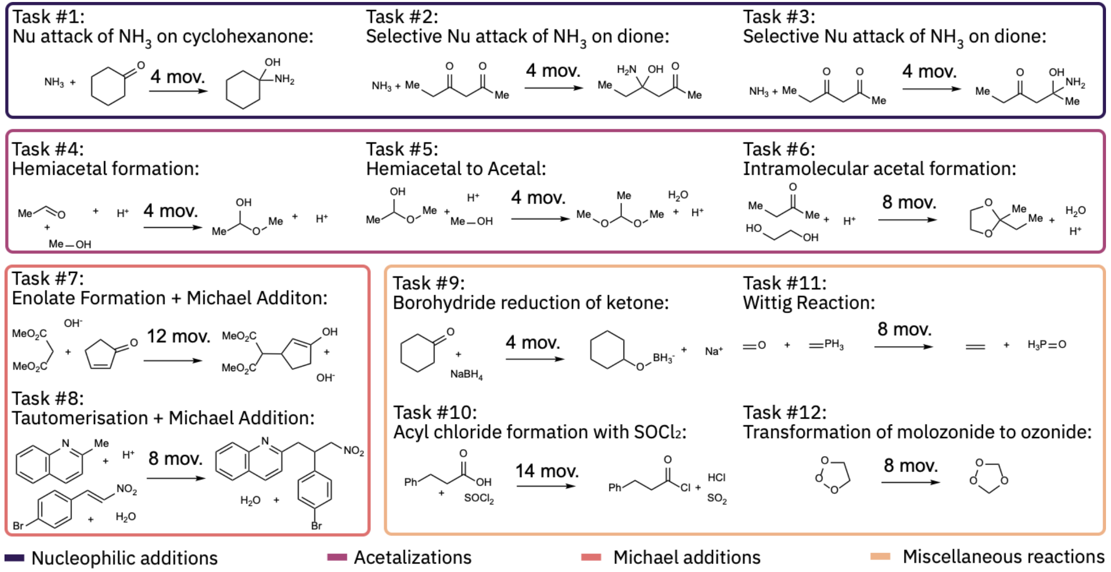

Reaction mechanism elucidation

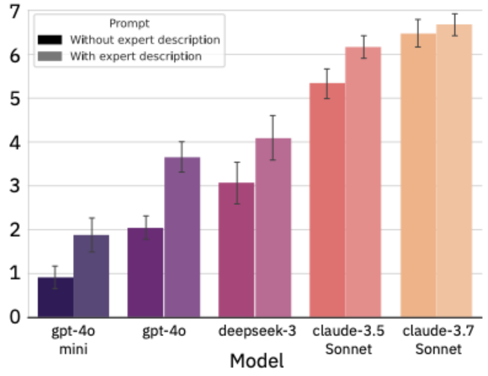
Actions: elementary steps

- Ionization moves
- Attack moves









Expert reaction description in prompt helps weaker models.

LLMs enable chemists to talk to machine learning tools in their language.

Code and report submission

Updated deadline: Sunday, June 1st (end of day, CET) following a request by one of your fellow students.