Machine Learning-Enhanced Computational Modeling of Metal-Protein Interactions

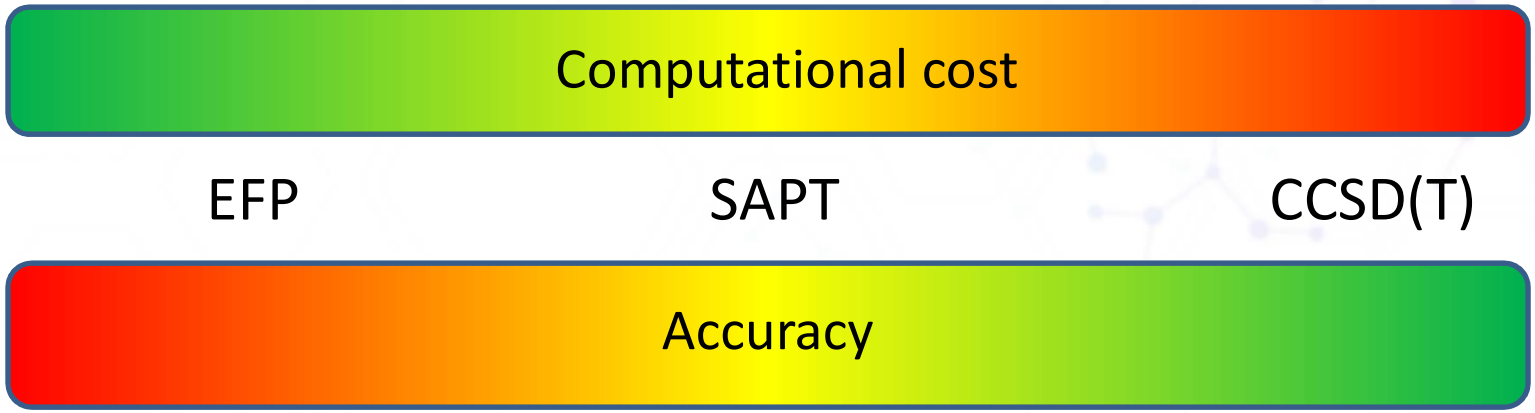
Presented by Alex Haskel

Background

- Interaction energy: contribution to the total energy in a system caused by interactions between objects being considered.

- Calculating interaction energies is key to understanding the structures of large molecules, the nature of gas–liquid and liquid–solid phenomena, and more.

- Different computation methods offer different tradeoffs:



EFP Theory

- EFP is computationally inexpensive, and can handle larger systems than other more accurate, but more costly methods.

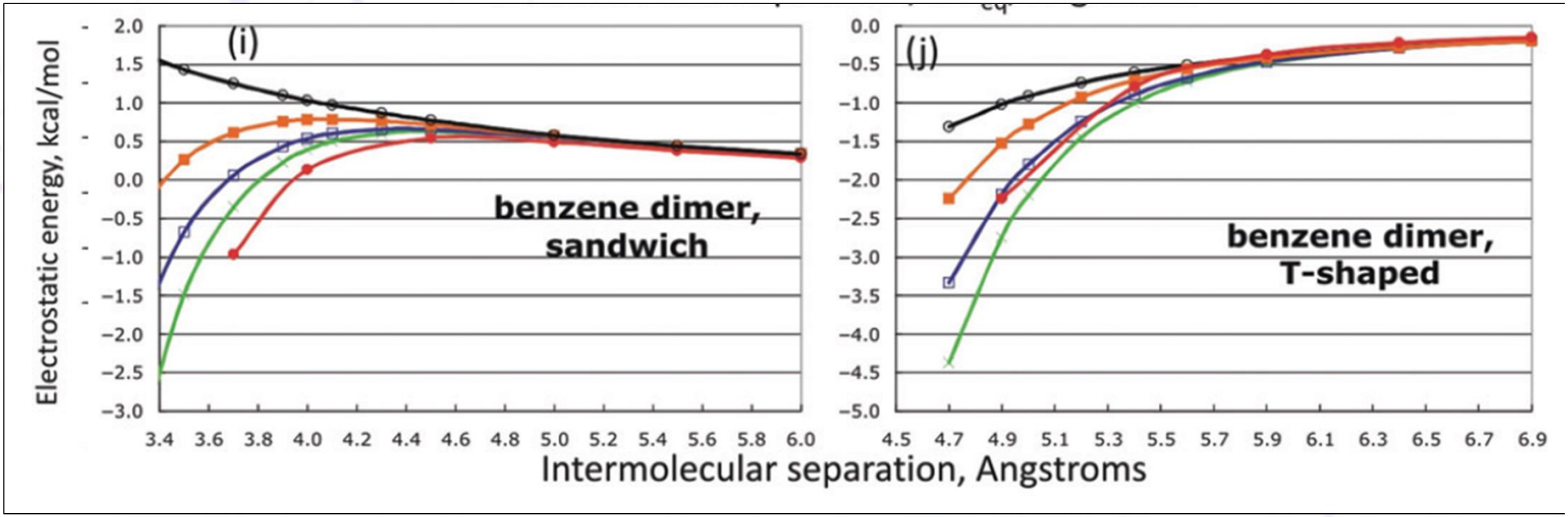
- $E_{int} = E_{Coul} + E_{ind} + E_{exrep} + E_{disp} + E_{ct}$

- Some terms involve overlap between electron clouds, a phenomenon that's hard to model accurately with simple classical equations.

- In these cases, EFP uses "damping functions," corrective factors that turn down (dampen) unrealistic energy contributions at short distances.

On Damping Functions

- Damping functions improve the predictions of EFP theory, but are handcrafted formulas, guided by theory and empirical observation of what seems to fit data from more precise methods, like SAPT. There's no single "true" damping scheme.



RED line: Electrostatic energy prediction from SAPT (target)

BLACK line : Undamped EFP Electrostatic energy

GREEN/PURPLE/ORANGE: EFP electrostatic damping schemes

Credit: Slipchenko, L. V., & Gordon, M. S. (2009). Damping functions in the effective fragment potential method. *Molecular Physics*, *107*(8–12).

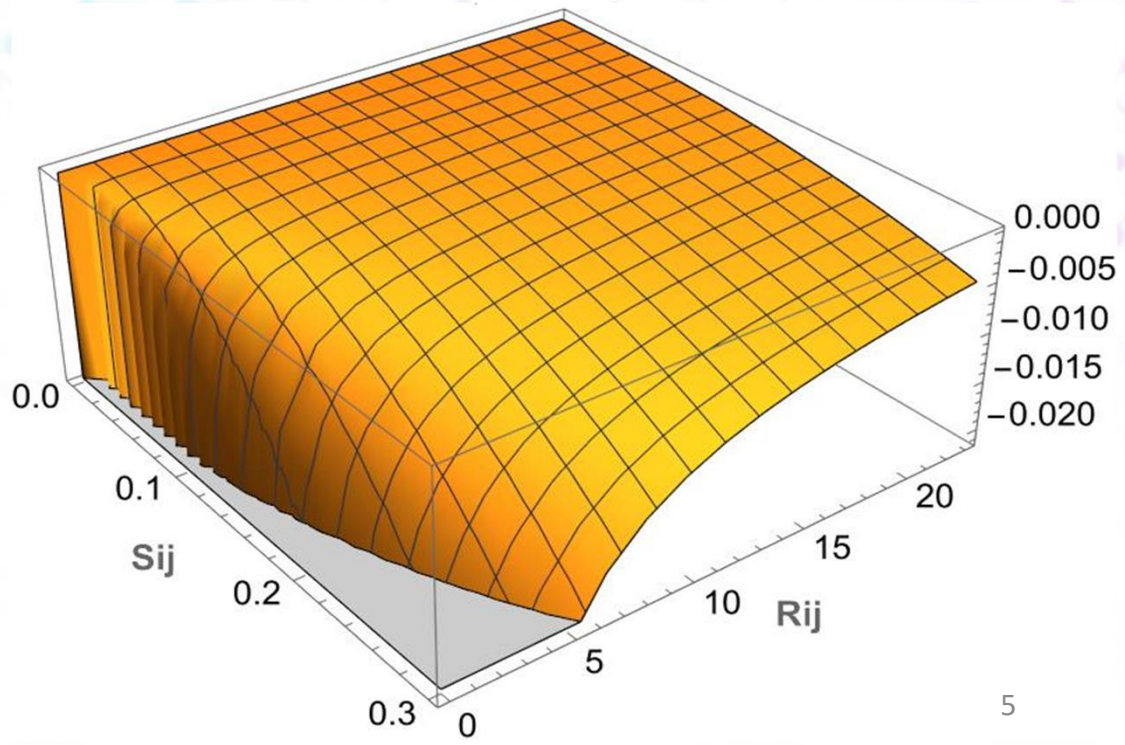
Current Focus: Coulomb Damping

- We'd like to find out if there are better forms for these damping functions that EFP theorists could use.

- For now, we are focused on the simplest damping term, the *Overlap Penetration Energy* formula for electrostatic damping:

$$E^{Pen} = -2\left(\frac{1}{-2\ln(Sij)}\right)^{\frac{1}{2}} \frac{Sij^2}{Rij}$$

$$= f(Sij, Rij)$$

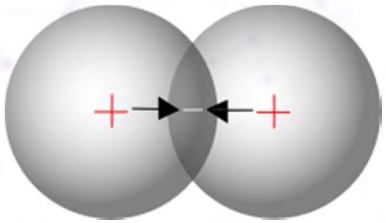


What are Sij and RIJ? + Our Research

- Sij measures ***overlap*** between the i^{th} molecular orbital of one fragment and the j^{th} molecular orbital of another fragment. Rij measures the ***distance*** between them.

- Each system in our database has many (Sij, Rij) pairs, depending on how many fragments and orbitals there are.

- For an entire system with many fragments, the final damping correction to the electrostatic energy is given by the sum of corrections on all pairs:

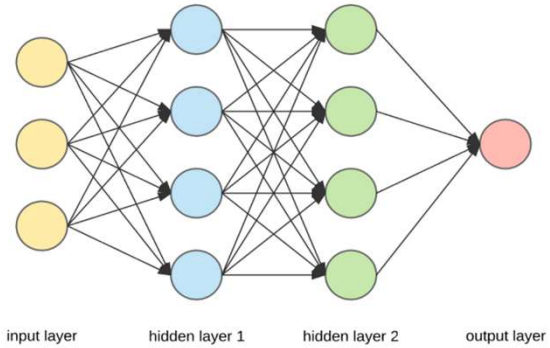


$$\sum_{j \in B} \sum_{i \in A} f(Sij, Rij)$$

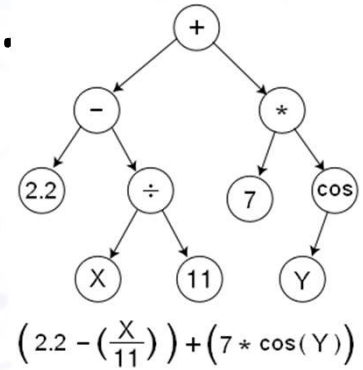
- We want to find a Coulomb damping formula, $f^*(Sij, Rij)$ that ideally is:
  - A function of (Sij, Rij) (physically meaningful variables)
  - Interpretable
  - More accurate than the existing one

Possible ML Approaches

- Approach 1: Use a neural network.



- Approach 2: Use symbolic regression.



$$\left(2.2-\left(\frac{X}{11}\right)\right)+\left(7*\cos(Y)\right)$$

- Approach 3: Hybrid approach (most promising???)

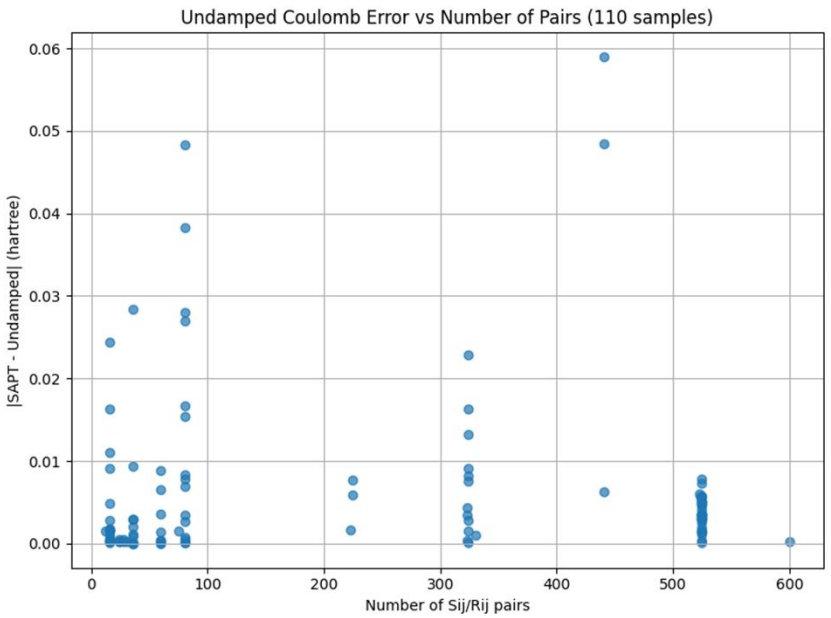
Approach 1: Neural Network

- One can use a NN to learn a better damping function directly from the SAPT data.
- Pros & Cons:
  - "Black box;" not easily interpretable
  - We have ground truth values at the system level from SAPT data, but no truth values at the (Sij, Rij) pair level within each system
  - We'd have to make a prediction for every pair in each system, but we can only compare the total sum vs SAPT. So, we could have a ratio:

$$\frac{200\ predictions}{1\ optimization\ step}\ (not\ ideal)$$

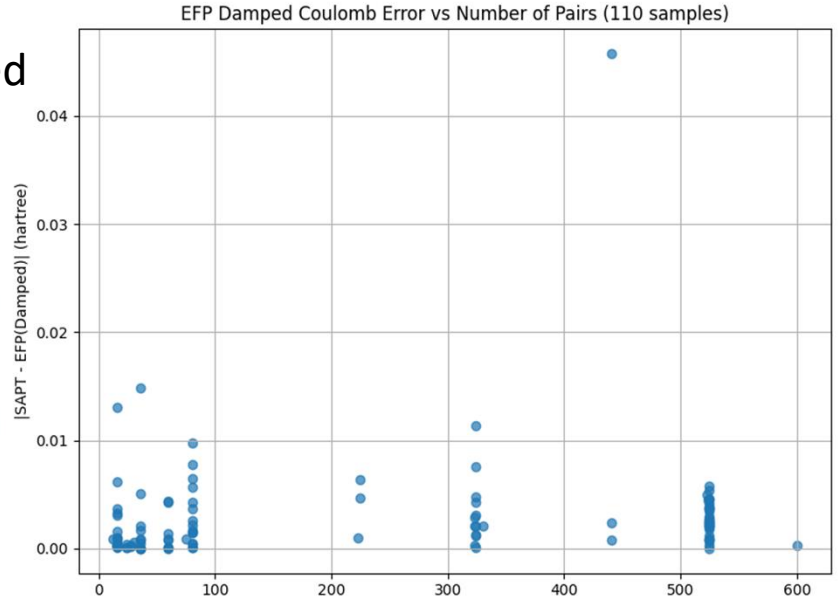
  - NN are "data hungry," and there aren't many systems in our database, if we want to do system-level learning only.
  - Bonus: Neural networks are cool

Some NN Experiments & Baselines

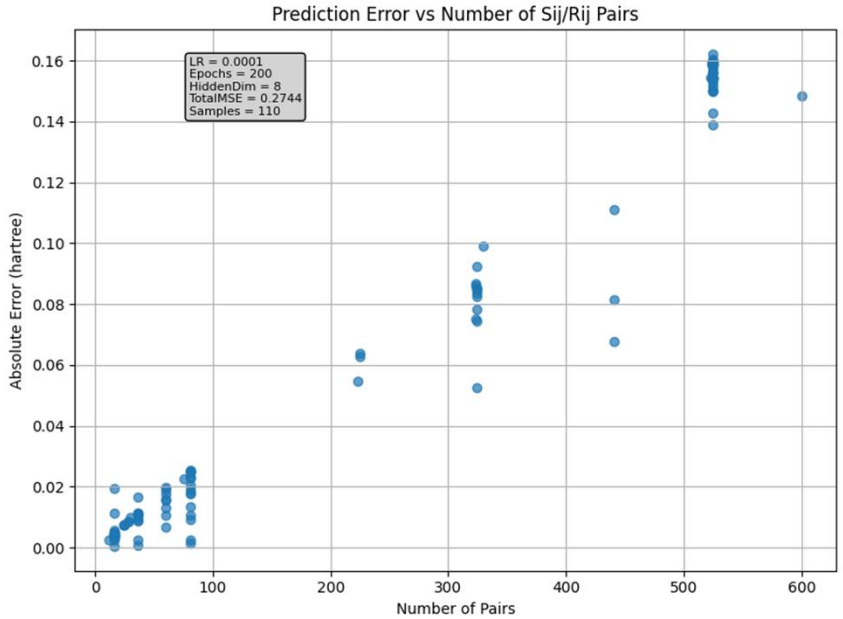


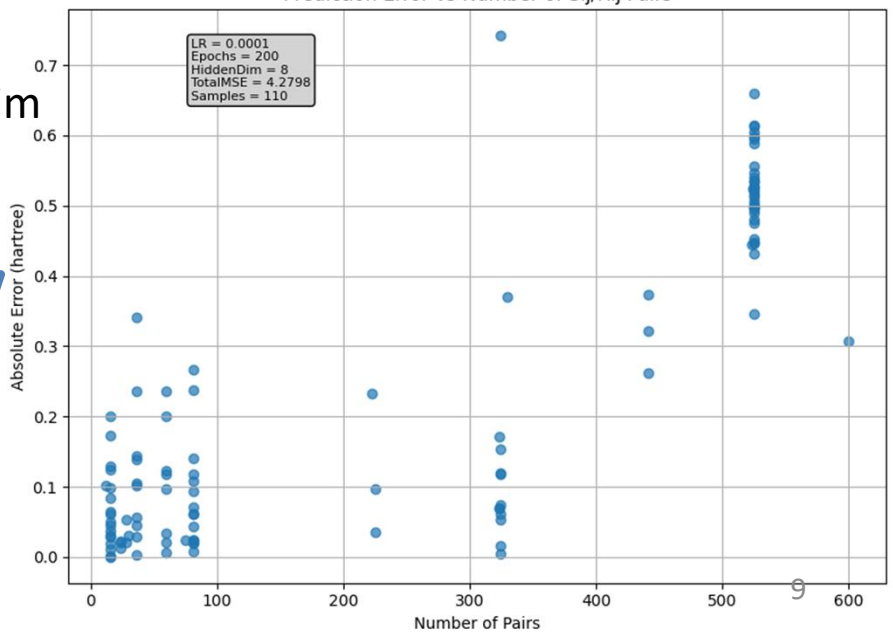
Undamped Baseline

Damped w/f(sij,rij)



NN Experiments





Approach 2: Symbolic Regression

- SR is a regression method that searches for expressions that best fits a given dataset, both in terms of accuracy and simplicity.
- Pros & Cons:
  - More niche method; less documentation out there
  - Designed to find **interpretable** symbolic formulas in scientific settings, several papers about this
  - We'd still have the same issue of optimizing at the system level, but having to guess many targets at the pair level
  - Perhaps doesn't need as much data
  - Could be more expensive computationally
  - Bonus: Also cool

Approach 2: Symbolic Regression

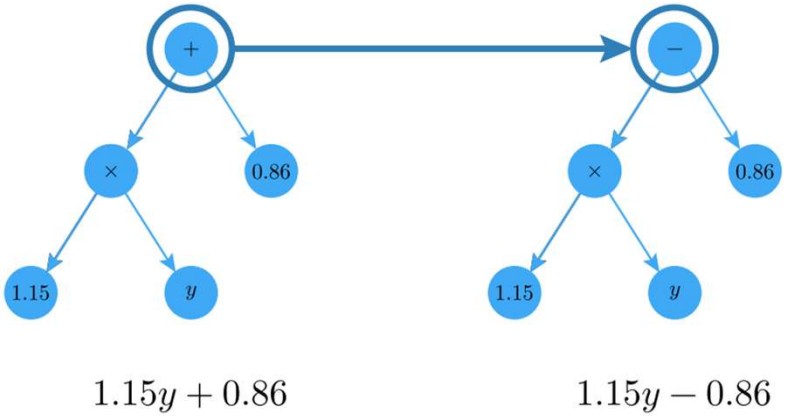
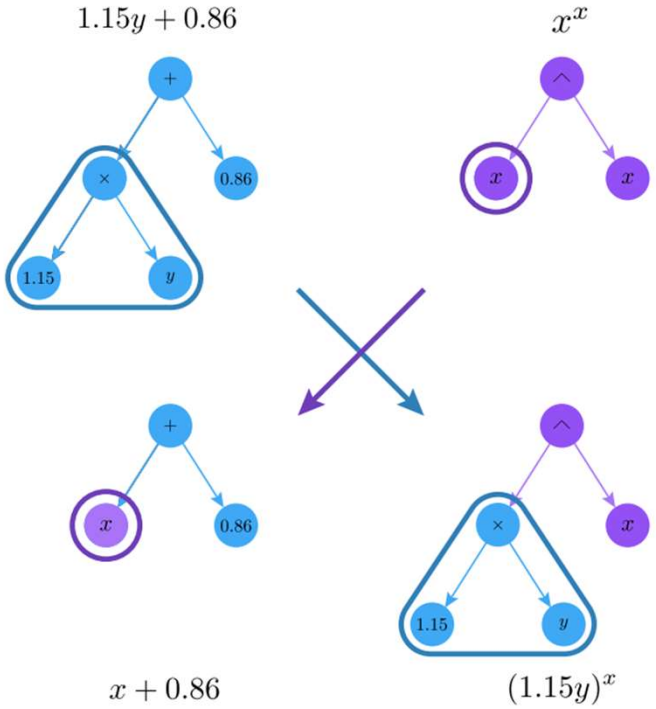


Fig.1 Mutation ↑ & crossover↓



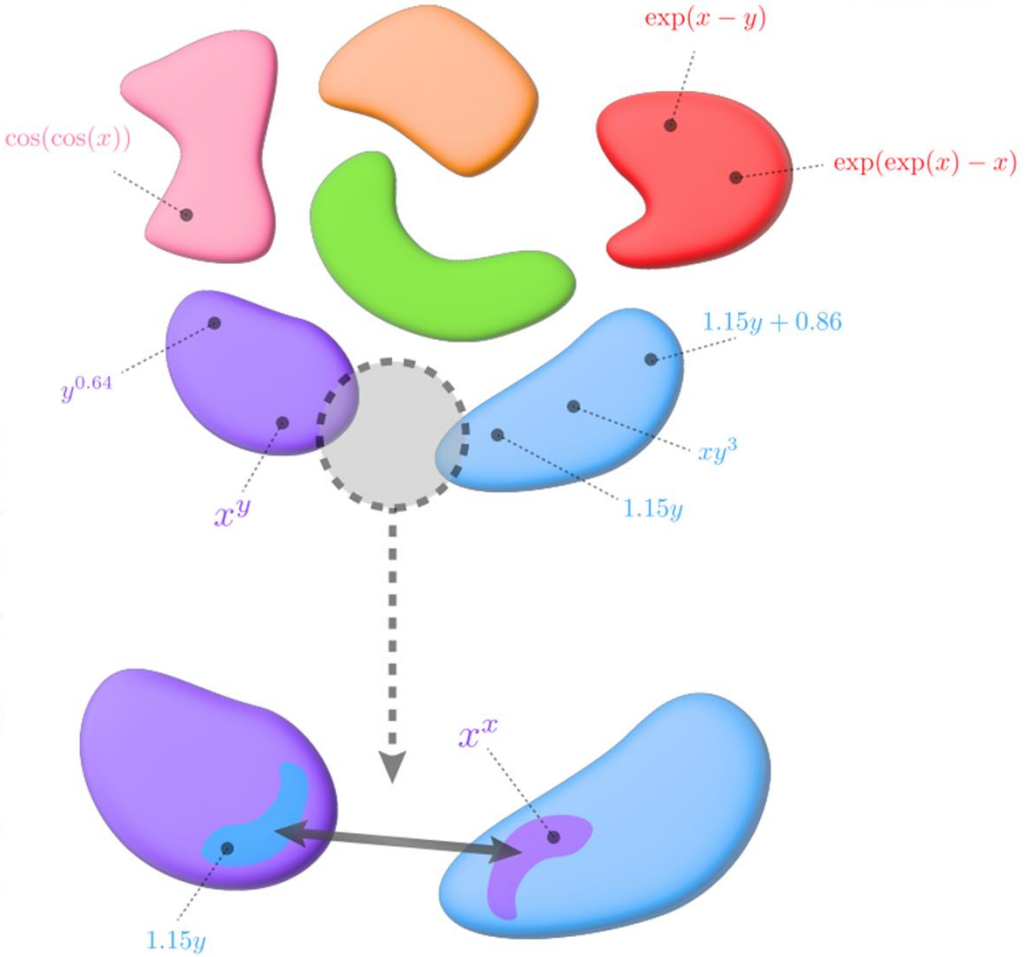


Fig. 2: Independent “islands of expressions” undergoing evolution, with island migration operation

Credit: Miles Cranmer. (2023). Interpretable Machine Learning for Science with PySR and SymbolicRegression.jl.

Symbolic Regression Example

- I wanted to see if PySR could recover the existing damping formula just by looking at data, so I gave it 10,000 (Sij,Rij) pairs, each with corresponding values from the existing f(Sij, Rij) formula

Equation	Complexity	Loss	Score
$y = Sij^3 \left(-2.15\right)$	6	$1.17 \cdot 10^{-10}$	1.41
$y = Sij \left(0.000146 + \frac{Sij \left(-Sij - Sij - 0.733\right)}{Rij}\right)$	18	$7.35 \cdot 10^{-14}$	0.913
$y = \frac{Sij^2 \left(-2Sij + 0.0486 \left(0.530 \log\left(Sij\right) + 1\right)^2 - 0.733\right)}{Rij}$	25	$1.53 \cdot 10^{-14}$	1.26

- The full list of candidates is much longer. Very small loss, but more complicated expressions. I'd like to run more tests, play with hyper-parameters, maybe test on our full dataset (26,000 pairs as of now; more to come)

Approach 3: Hybrid (my current favored approach)

- Both of the previous approaches suffer from the same problem: lack of data, and, in particular, a lack of per-pair truth values.

- But we can do this:

  - **Phase 1:** Train a neural network to learn the existing damping formula. Just try to match the current f(sij, rij).

    - This gives us a per-pair target; no more infrequent optimization. 1 guess → 1 correction step.

    - A lot more data, we have 26,000 pairs, but only ~110 systems

    - The current approximation is already very good, so there is no reason to start from scratch. Better to start from there.

Approach 3: Hybrid (my current favored approach)

- If we can get the neural network to replicate the performance of the current damping formula (big if), we move to phase 2.
  - **Phase 2:** Take the same neural network, maybe freeze some of the weights so it doesn't "unlearn" too much (aka fine-tuning) and train it again, but minimizing loss only at the level of total sum of corrections per system vs SAPT data, not at the pair level:
    - This would allow the NN to start from the current damping performance, not from scratch
    - The system-level data seems scarce (only a few 1000s SAPT data points at most, which is very little by NN standards). However, we'd only have to make small adjustments, so, maybe it could work.

Approach 3: Hybrid (my current favored approach)

- Assuming we can surpass the performance of the current damping function, we still have the problem that the NN is a black box. So:
  - **Phase 3:**
    - If 1 & 2 worked, take the NN and create a full set of improved ground truth values at the pair level for all pairs in our database (we can easily get 10s or 100s of thousands of these).
    - Feed this new pairwise data to the symbolic regressor; have it find expressions that match this improved data.
    - If all this works, we'd now have a set of expressions that are:
      - Functions of (Sij, Rij) (physically meaningful variables)
      - Interpretable
      - More accurate than the existing one

… but I’m still working on it.

Acknowledgements

- **NSF, Simcodes Program, Iowa State University, and Ames National Lab, for funding and running this REU opportunity.**