Force Field Models in Polymer Informatics

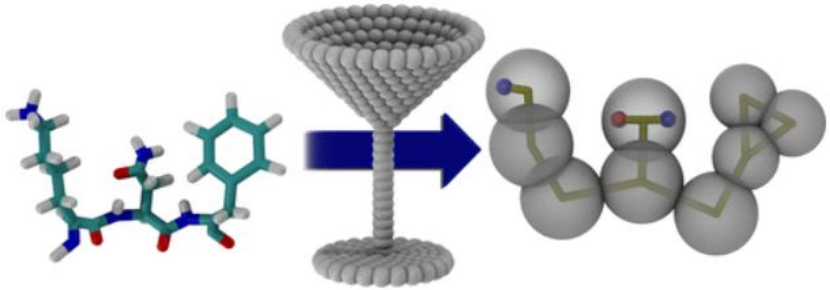
Luis Martinez

Motivation

- **Goal:** Predict the properties of a polymer given its conformation.
- Simulations are absolutely necessary.
  - Already existing options (e.g. all-atom simulations) are **slow**.
- Crystallization simulations are also slow, while involving mathematical assumptions.
- **We need a model for feature representation that allows for computational speed.**

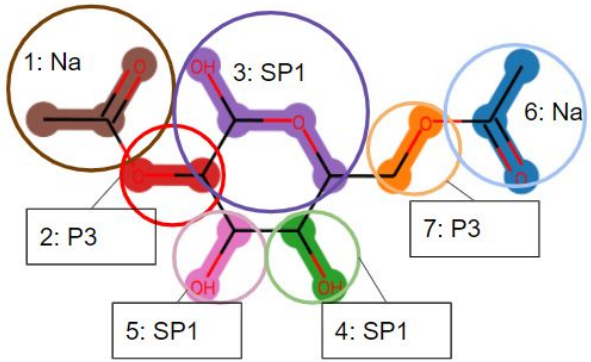
Objective: MARTINI Model

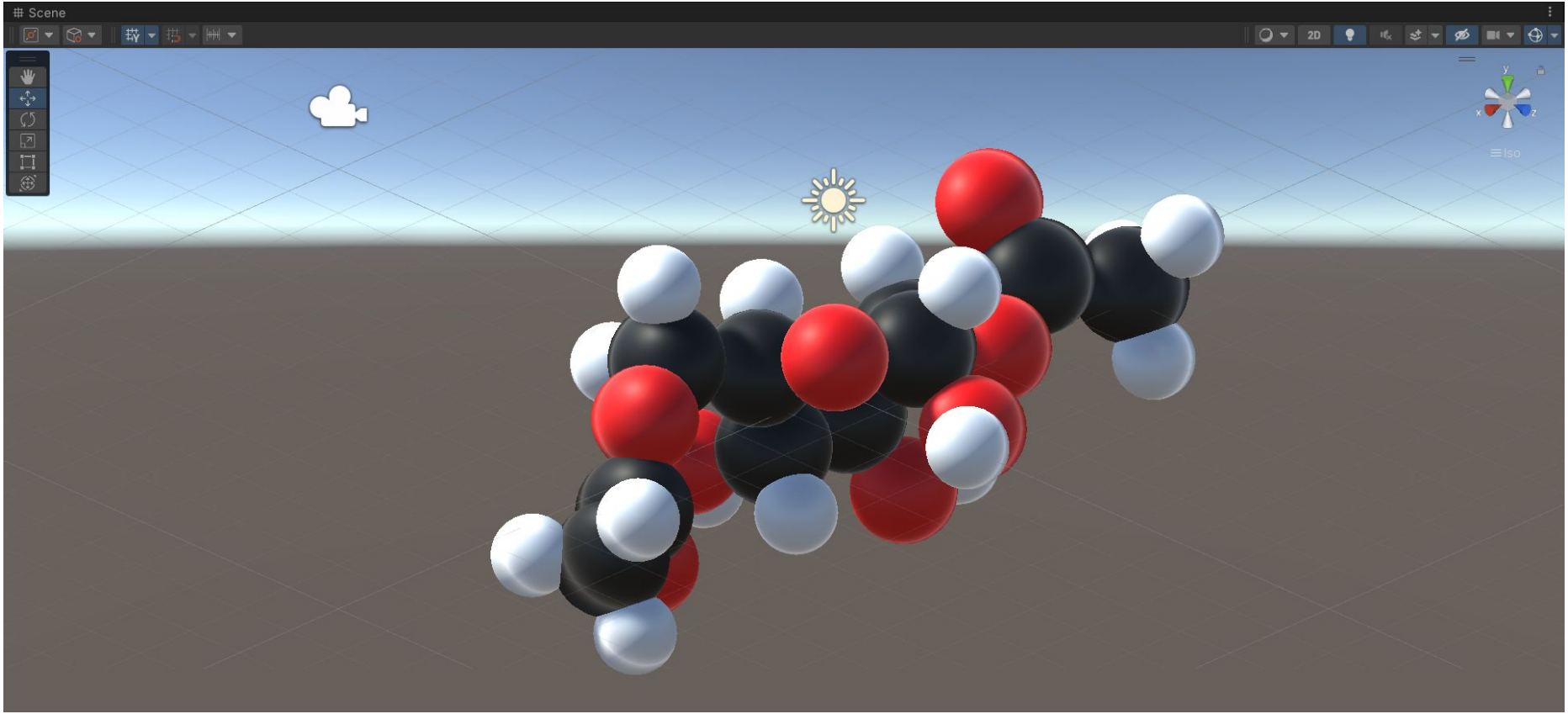
- **Main Objective:** Construct a generalize architecture to predict polymer predictions more efficiently.
- **Secondary Objective:** Decrease the resolution of our polymer chain and model molecular dynamics.
    - Use the MARTINI model; which categorizes groups of atoms (4-to-1) into certain types and subtypes. These categories are used to parametrize a set of bonded and non-bonded potential equations.
    - Model the molecular dynamics of the polymer through the use of MARTINI.



Approach

- **Construction of a pipeline:** GNN for the production of embeddings, and the use of a reinforcement learning architecture.
- **Tool:** Unity.
  - Representation of the whole molecule as a series of agents (beads) with scripts that describe the functionality of the MARTINI model.
  - Scripting code made use of C#.
  - Toy model: Cellulose Acetate




Scene
2D
Iso

Approach

- **Embeddings:** Defined embeddings for each bead including the information necessary (i.e. type and subtype).
- **Non-bonded Interactions:**
  - Inclusion of interaction matrix the uses the embedded information to determine the parameters of Lennard-Jones equations for each pair of non-bonded beads.
- **Bonded Interactions:**
  - Inclusion of previous simulation data in order to parametrize the bonded interaction equations.
  - Use of graph-search algorithm in order to iterate through the bonded beads of the polymer chain to define both bond angle and dihedral angle conformations.

$$V_{LJ}(r_1 \ldots r_n) + V_{Coul}(r_1 \ldots r_n) = \sum_{i,j} 4\varepsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] + \sum_{i,j}\frac{q_i q_j}{4\pi\epsilon_0\epsilon_r r_{ij}}$$

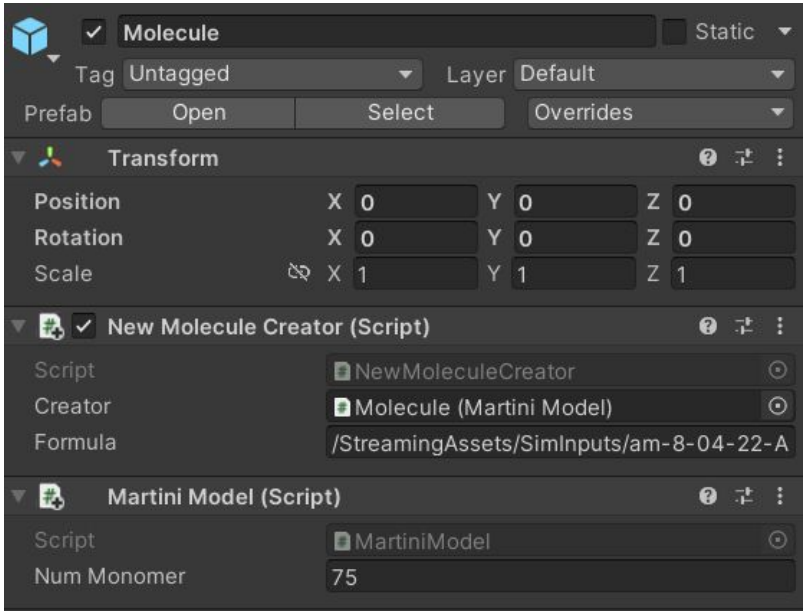
$$V_{\mathrm{b}} = 1/2 K_{\mathrm{b}}(d_{ij} - d_{\mathrm{b}})^2,$$

$$V_{\mathrm{a}} = 1/2 K_{\mathrm{a}}[\cos(\varphi_{ijk}) - \cos(\varphi_{\mathrm{a}})]^2,$$

$$V_{\mathrm{d}} = K_{\mathrm{d}}[1 + \cos(\theta_{ijkl} - \theta_{\mathrm{d}})],$$

```
public static IReadOnlyDictionary<(string, string), int> typeMappings = new Dictionary<(string,string), int>() {
        {("Q", "da"), 0}, {("Q", "d"), 1}, {("Q", "a"), 2}, {("Q", "0"), 3},
        {("P", "5"), 4}, {("P", "4"), 5}, {("P", "3"), 6}, {("P", "2"), 7}, {("P", "1"), 8},
        {("N", "da"), 9}, {("N", "d"), 10}, {("N", "a"), 11}, {("N", "0"), 12},
        {("C", "5"), 13}, {("C", "4"), 14}, {("C", "3"), 15}, {("C", "2"), 16}, {("C", "1"), 17},
        {("BP", "4"), 18}
    };

public static readonly string[,] levels = new string[,]
{
    {"O", "O", "O", "II", "O", "O", "O", "I", "I", "I", "I", "I", "IV", "V", "VI", "VII", "IX", "IX", "O"},
    {"O", "I", "O", "II", "O", "O", "O", "I", "I", "I", "III", "I", "IV", "V", "VI", "VII", "IX", "IX", "O"},
    {"O", "O", "I", "II", "O", "O", "O", "I", "I", "I", "III", "IV", "V", "VI", "VII", "IX", "IX", "O"},
    {"II", "II", "II", "IV", "I", "O", "I", "II", "III", "III", "III", "III", "IV", "V", "VI", "VII", "IX", "IX", "O"},
    {"O", "O", "O", "I", "O", "O", "O", "O", "O", "I", "I", "I", "IV", "V", "VI", "VI", "VII", "VIII", "O"},
    {"O", "O", "O", "O", "O", "I", "I", "II", "II", "III", "III", "III", "IV", "V", "VI", "VI", "VII", "VIII", "O"},
    {"O", "O", "O", "I", "O", "I", "I", "II", "II", "II", "II", "II", "IV", "IV", "V", "V", "VI", "VII", "O"},
    {"I", "I", "I", "II", "O", "II", "II", "II", "II", "II", "II", "II", "III", "IV", "IV", "V", "VI", "VII", "O"},
    {"I", "I", "I", "III", "O", "II", "II", "II", "II", "II", "II", "II", "III", "IV", "IV", "IV", "V", "VI", "O"},
    {"I", "I", "I", "III", "I", "III", "II", "II", "II", "II", "II", "II", "IV", "IV", "V", "VI", "VI", "VI", "O"},
    {"I", "III", "I", "III", "I", "III", "II", "II", "II", "II", "III", "II", "IV", "IV", "V", "VI", "VI", "VI", "O"},
    {"I", "I", "III", "III", "I", "III", "II", "II", "II", "II", "II", "III", "IV", "IV", "V", "VI", "VI", "VI", "O"},
    {"IV", "IV", "IV", "IV", "IV", "IV", "IV", "III", "III", "IV", "IV", "IV", "IV", "IV", "IV", "IV", "V", "VI", "O"},
    {"V", "V", "V", "V", "V", "V", "IV", "IV", "IV", "IV", "IV", "IV", "IV", "IV", "IV", "V", "V", "O"},
    {"VI", "VI", "VI", "VI", "VI", "VI", "V", "IV", "IV", "V", "V", "V", "IV", "IV", "IV", "IV", "V", "V", "O"},
    {"VI", "VI", "VI", "VI", "VI", "VI", "V", "IV", "IV", "V", "V", "V", "IV", "IV", "IV", "IV", "V", "V", "O"},
    {"VII", "VII", "VII", "VII", "VI", "VI", "V", "V", "IV", "VI", "VI", "VI", "IV", "IV", "IV", "IV", "IV", "IV", "O"},
    {"IX", "IX", "IX", "IX", "VII", "VII", "VI", "VI", "V", "VI", "VI", "VI", "V", "V", "V", "IV", "IV", "IV", "O"},
    {"IX", "IX", "IX", "IX", "VIII", "VIII", "VII", "VII", "VI", "VI", "VI", "VI", "VI", "V", "V", "IV", "IV", "IV", "O"},
    {"O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O"}
};
```




```
public static void findAngles (Dictionary< GameObject, List<GameObject> > graph, GameObject bead, int length, List< List<GameObject> > paths, List<GameObject> path = null) {
    path = path ?? new List<GameObject>();
    if (path != null) {
        path.Add(bead);
        List<GameObject> pathr = new List<GameObject>(path);
        pathr.Reverse();
        if (path.Count == length) {
            if (!paths.Contains(path) && !paths.Contains(pathr)) {
                paths.Add(path);
            }
        }
        else {
            foreach (GameObject neighbour in graph[bead]) {
                findAngles(graph, neighbour, length, paths, path);
            }
        }
    }
}
```

Results

- Expanded my knowledge regarding molecular simulations.
  - In particular, obtained an improved understanding of the mathematical assumptions made in order to model simulations; both in atomistic and coarse-grained models.
- Improved my understanding of molecular property prediction architectures.
  - Reviewed the existing literature and gained an understanding of the advantages and disadvantages of existing approaches.
- Gained expertise in the use of Unity3D as a tool.
  - Learned how to develop C# scripts oriented towards the modelling of force fields and the implementation of reinforcement learning simulations.

Next Steps

- Finish the construction of the reinforcement learning implementation.
  - Construct a RL simulation through the use of the Unity ML-Agents toolkit by assigning C# scripts to our modeled polymer chain.
- Start the training stage of our architecture.
  - Introduce data to our model.
- Include the entanglement of polymer chains through reinforcement learning.
  - Model the dynamics involved between polymer chains in Unity.
- Compare results with the existing literature.