Student Project Proposal

PolySTRIPES: Symbolic AI Representation of Molecular Dynamics Trajectories

Master area	AI for Drug Discovery; molecular simulation; cheminformatics; biomaterials
Project type	Computational project with Python implementation and data analysis
Duration	2-3 months
Supervision	One tutor or co-tutor; weekly or bi-weekly progress meetings
Recommended level	Basic Python and machine learning; basic knowledge of molecular modelling is useful
Main output	A documented notebook/workflow, a technical report, and a short final presentation
Possible datasets	Small polymer, lipid membrane, or membrane-drug/peptide MD trajectories; public or tutor-provided data

Short description. The project asks the student to prototype a simplified PolySTRIPES workflow that converts molecular dynamics trajectories into symbolic sequences of local interactions. The final representation should be compact, interpretable, and suitable for downstream AI models such as sequence models, graph embeddings, or graph neural networks.

1. Background and rationale

Molecular dynamics simulations describe the time evolution of molecular systems at atomistic or coarse-grained resolution. They are highly informative but difficult to analyse systematically because they generate large, redundant and dynamically complex trajectories. This limitation is especially relevant for polymers, lipid membranes, drug-delivery systems and membrane-active compounds, where repeated structural units and slow collective motions produce rich interaction patterns that are not easily summarized by a few geometric descriptors.

PolySTRIPES addresses this problem by replacing raw coordinates with a symbolic language of local molecular interactions. Chemically meaningful sites are identified in the simulated system, local contacts are detected frame by frame, and each interaction state is encoded as a discrete token. The resulting token sequences can then be analysed with methods derived from sequence analysis, graph representation learning and machine learning.

In the context of AI for drug discovery, this approach is relevant because many pharmacological and formulation problems depend on dynamic molecular behaviour: membrane binding, polymer-drug interactions, stability of delivery systems, local flexibility, hydration, and persistent interaction motifs. A compact symbolic representation may help compare simulations, identify recurrent patterns, and build machine learning-ready descriptors for future predictive or generative models.

2. Project objective

The objective of the project is to implement and test a simplified PolySTRIPES pipeline for transforming molecular dynamics trajectory data into symbolic interaction sequences and graph-based representations suitable for AI analysis.

The project should remain realistic for a 2-3 month student activity. It is not necessary to develop a complete production software package or to run long new molecular dynamics simulations. The student should focus on a well-defined prototype applied to one or a few small trajectory examples.

3. Specific aims

- Define a simple site-based representation for a polymer, lipid membrane, or membrane-related molecular system.
- Detect local interactions between sites over time using clear geometric and chemical rules.
- Convert interaction states into symbolic tokens containing interaction type, partner class and local structural context.
- Introduce temporal windowing and calculate simple entropy-based descriptors of local dynamical disorder.
- Build a compact machine learning-ready output, such as a token table, sequence dataset, or graph with node-level token sequences.
- Evaluate whether the symbolic representation captures interpretable recurrent motifs or differences between systems.

4. Proposed methodology

4.1 Input data

The student will start from standard molecular dynamics files: a topology file, typically PDB or equivalent, and a trajectory file such as XTC, DCD, TRR or a simplified exported coordinate series. A small public trajectory or a tutor-provided dataset should be used to keep the project computationally manageable.

4.2 Site definition

The molecular system will be segmented into chemically meaningful analytical sites. Depending on the dataset, these sites may correspond to polymer monomers, backbone regions, carbonyl groups, lipid headgroups, tail segments, solvent groups, ions, ligands or peptide residues. The goal is to reduce dimensionality while preserving chemical interpretability.

4.3 Interaction detection and tokenization

For each frame, neighbouring sites will be identified using distance-based criteria. The student will then classify simple interaction types, for example hydrogen bonds, polar contacts, hydrophobic contacts, ionic contacts or solvent-mediated contacts. Each interaction state will be encoded as a symbolic token. A minimal token could include: site identifier, interaction class, partner class, structural region and time window.

4.4 Temporal compression and entropy descriptors

To reduce trajectory redundancy, frames will be grouped into time windows. For each site and each window, the student will extract dominant tokens, token frequencies and a local entropy value. Low entropy should indicate a stable interaction state, whereas high entropy should indicate a site that fluctuates among multiple interaction modes.

4.5 AI-ready representation

The final output should combine local token sequences with structural adjacency. In its simplest form, this can be a table of token sequences and a graph in which nodes are analytical sites and edges represent molecular connectivity or persistent contacts. This representation may then be used for clustering, embedding, visualization, motif analysis or, in an optional extension, a simple supervised or self-supervised machine learning experiment.

5. Indicative work plan

Period	Main task	Expected output
Weeks 1-2	Problem definition and dataset selection	Short technical note describing the selected system, files, software tools and analytical sites.
Weeks 3-4	Trajectory parsing and site mapping	Python functions/notebook for reading coordinates, identifying species and assigning sites.
Weeks 5-6	Interaction detection and tokenization	First token table generated from one trajectory; validation on selected frames.
Weeks 7-8	Temporal compression and entropy	Window-based token summaries and entropy descriptors for each analytical site.
Weeks 9-10	Pattern analysis and visualization	Plots, motif counts, token heatmaps or simple embeddings showing recurrent dynamical signatures.
Weeks 11-12	Final integration and reporting	Final notebook, short report and presentation summarizing method, results and limitations.

6. Expected deliverables

- A documented Python notebook or small code repository implementing the prototype workflow.
- A processed example dataset or output tables containing symbolic interaction tokens.
- Entropy descriptors and at least one visualization of local dynamical behaviour.
- A graph or sequence-based representation that can be used as input for machine learning.
- A final technical report of approximately 3-5 pages, including aims, methods, results, limitations and possible extensions.
- A final presentation of approximately 10-15 minutes.

7. Skills acquired by the student

- Practical handling of molecular simulation data and trajectory-derived features.
- Design of chemically interpretable descriptors for complex molecular systems.
- Implementation of symbolic representations and tokenization strategies.
- Use of entropy or frequency-based descriptors to summarize molecular dynamics.
- Preparation of graph- or sequence-based inputs for AI models.
- Critical evaluation of methodological limits in AI-assisted molecular modelling.

8. Suggested software and resources

The project can be implemented in Python using standard scientific libraries. Depending on the dataset and the student background, useful packages may include MDAnalysis or MDTraj for trajectory processing, RDKit for chemical handling where relevant, NumPy and pandas for data processing, NetworkX for graph construction, scikit-learn for baseline analysis, matplotlib for visualization, and optionally PyTorch or PyTorch Geometric for advanced machine learning extensions.

The tutor should provide, or help the student identify, a small trajectory dataset suitable for testing. The project should initially avoid very large systems or long production trajectories unless preprocessing has already been completed.

9. Evaluation criteria

Criterion	What should be assessed
Scientific clarity	The project question and molecular interpretation are clearly defined.
Technical implementation	The code is readable, documented and reproducible on the selected dataset.
AI relevance	The final representation is suitable for machine learning, embedding, clustering or graph analysis.
Interpretability	The tokens and descriptors retain meaningful chemical or dynamical information.
Critical discussion	The student discusses limitations, assumptions and possible future developments.

10. Optional extensions

For students with stronger computational skills, the project may be extended by comparing two different systems, implementing unsupervised clustering of token sequences, training a simple graph neural network on synthetic labels, or testing whether the symbolic representation can distinguish different polymer compositions, membrane phases or ligand-binding conditions.

Note for tutors. The project should be adapted to the available dataset and to the student background. The minimal successful outcome is a clear, reproducible prototype; advanced machine learning components should be treated as optional extensions rather than mandatory requirements.