Accelerating Molecular Generation through Representation Alignment

An MPhil project proposal

Shreyas Ravishankar (sr2173), Hughes Hall

Project Supervisor: Professor Pietro Lio

Abstract

Training generative models for molecular design is computationally expensive because models must simultaneously perform two difficult tasks: (1) representation learning, i.e. encoding data into meaningful representations; and (2) generation, i.e. decoding learned representations into data. This project investigates whether Representation Alignment (REPA), a novel technique that aligns generative model representations with pre-trained encoders, can alleviate this bottleneck. We will first implement REPA for small molecule generation tasks, and assess whether aligning representations (1) makes training more efficient; and (2) produces more chemically valid molecules. If successful, we will then investigate the utility of this approach for protein backbone generation.

1 Introduction, approach and outcomes

1.1 Motivation

Deep generative models for molecular design, such as diffusion and flow-based models, hold promise for various scientific endeavours. However, training these models is both data intensive and computationally expensive. Recent work in image generation [1] argues that this bottleneck arises because generative models must simultaneously: (1) learn high-quality internal representations of complex data; and (2) learn to construct data from these representations. The standard denoising or flow-matching objective may not provide sufficient signal for representation learning, leading to slow convergence.

Representation Alignment (REPA) [1] addresses this by forcing internal representations to match those from known-good pre-trained encoders, via a regularisation term in the training objective. In image generation, this simple intervention accelerates training by 17.5× while improving output quality.

1.2 Research Goals and Outcomes

The goal is to contribute methodological insights about representation learning in molecular generative models. The core question is: *can these models benefit, in terms of training efficiency and generation quality, from representation alignment with pre-trained models?* We aim to characterise when and why this works, including failure modes and limitations. The project will produce a written report documenting methodology, results, and insights, along with documented code and experimental scripts.

1.3 Non Goals

1. We will not train new encoder models.
2. We will not invent new metrics for efficiency or quality.
3. We do not aim to produce a hyper-optimised model; rather, we intend to investigate if REPA works or not.

1.4 Approach

This project takes a staged approach, starting with small molecules before extending to proteins.

Phase 1: Small Molecule Generation

We will investigate if we can improve Tabasco [2], a flow-based small molecule generator, by aligning with representations from different molecular encoders:

- *MACE* [6], a machine learned interatomic potential (MLIP) that encodes atomic geometry.
- *Chemprop* [3] or similar graph neural network trained on molecular property prediction.

We will measure the following metrics, ablating with and without REPA:

1. Training efficiency
   (a) Frechet ChemNet Distance vs training iteration
2. Generation quality
   (a) Validity
   (b) Novelty
   (c) Diversity
   (d) Posebusters
   (e) JS divergence between distributions of bond length, bond angles, bond types, and atom types

Phase 2: Extension to Proteins

If Phase 1 yields promising results, we will investigate protein backbone generation, testing alignment with different protein encoders:

- *ESM2*, a protein language model encoding structural knowledge from single sequences at scale.
- *ProteinMPNN*, an inverse folding model encoding priors on designability.
- *MACE* [6], encoding atomic geometry.

We will measure the following self-consistency metrics as proxy for quality, ablating with and without REPA:

1. scRMSD
2. pdbTM

3. Diversity
4. Frechet protein distance
5. pLDDT

2 Workplan

Weeks 1–2 (8 Dec – 21 Dec): Background and Setup

Study flow-matching fundamentals, REPA paper [1], and MACE paper [6]. Set up Python environment and identify molecular datasets of interest. **Target:** Working environment; understanding of REPA and flow-matching.

Weeks 3–4 (22 Dec – 4 Jan): Codebase Familiarization

Study Tabasco [2] paper and explore baseline codebase. Identify alignment points in model architecture. **Target:** Understanding of baseline model structure.

Weeks 5–6 (5 Jan – 18 Jan): Baseline Implementation

Set up baseline flow model, pre-process datasets, and train on small subset. Implement evaluation metrics (validity, uniqueness, diversity). **Target:** Reproducible baseline with documented metrics.

Weeks 7–8 (19 Jan – 1 Feb): Encoder Integration

Select and integrate pre-trained molecular encoders (MACE, Chemprop). Extract embeddings for dataset molecules and verify outputs. **Target:** Working encoder inference pipeline.

Weeks 9–10 (2 Feb – 15 Feb): REPA Implementation

Implement alignment loss in training loop with tunable weight parameter. Debug gradient flow and numerical stability. Run initial alignment experiments. **Target:** Functional REPA implementation with preliminary results.

Weeks 11–12 (16 Feb – 1 Mar): Initial Experiments

Systematically vary alignment strength. Evaluate training efficiency and sample quality. Document which configurations work best. **Target:** Evidence for/against REPA effectiveness.

Weeks 13–14 (2 Mar – 15 Mar): Ablation Studies

Compare alignment layers (early/middle/late), loss formulations (L2/cosine), and encoder architectures (2D/3D). Generate comparison tables and plots. **Target:** Comprehensive ablation results.

Week 15 (16 Mar – 22 Mar): Progress Review with Supervisor

Prepare presentation of results to date. Refine experimental plan based on feedback. **Target:** Progress review completed; clear priorities for Easter Term.

Week 16 (23 Mar – 29 Mar): Planning and Transition

Finalize experimental plan for Easter Term. Clean and document codebase. Begin outlining dissertation structure. **Target:** Ready for intensive experimental phase.

Weeks 17–18 (30 Mar – 12 Apr): Intensive Experiments

Conduct comprehensive hyperparameter sweeps and final experiments. If results are strong, explore protein extension with Proteina [7]. Otherwise, perform deep analysis of molecular results. **Target:** Comprehensive experimental results OR protein proof-of-concept.

Weeks 19–20 (13 Apr – 26 Apr): Final Experiments

Complete all remaining experiments. Generate final figures, tables, and visualizations. Organize all results into clear narrative. **Target:** All experimental work complete.

Weeks 21–22 (27 Apr – 10 May): Core Writing

Write Introduction, Background, Related Work, and Methodology chapters. Create high-quality figures. **Target:** Core chapters drafted.

Weeks 23–24 (11 May – 24 May): Results and Discussion

Write Results and Discussion chapters with all figures and tables. Draft Conclusion. Share complete draft with supervisor for feedback. **Target:** Complete first draft. *Note: Title change deadline 26 May.*

Week 25 (25 May – 31 May): Revision and Contingency

Incorporate supervisor feedback. Polish writing and verify all citations. Format according to guidelines. Prepare abstract. **Target:** Polished dissertation ready for final checks.

Week 26 (1 Jun – 9 Jun): Final Checks and Submission

Final proofreading and formatting checks. **Submit by 11:00 AM Tuesday 9 June 2026. Target:** Dissertation submitted with time to spare.

Post-Submission (10 Jun – 18 Jun): Presentation

Prepare and practice presentation for 16–18 June mini-conference.

Risk Mitigation

Protein extension (Weeks 17–18) is optional; project succeeds with strong small molecule results alone. If behind at progress review, focus exclusively on molecules. Five weeks allocated for writing with Week 26 as buffer.

References

[1] S. Yu et al. Representation Alignment for Generation: Training Diffusion Transformers Is Easier Than You Think. 2025. arXiv: 2410.06940 [cs.CV]. url: https://arxiv.org/abs/2410.06940.

[2] C. Vonessen et al. TABASCO: A Fast, Simplified Model for Molecular Generation with Improved Physical Quality. 2025. arXiv: 2507.00899 [cs.LG]. url: https://arxiv.org/abs/2507.00899.

[3] K. Yang et al. "Analyzing Learned Molecular Representations for Property Prediction". In: Journal of Chemical Information and Modeling 59.8 (2019), pp. 3370–3388. doi: 10.1021/acs.jcim.9b00237.

[4] K. T. Schütt et al. "SchNet: A continuous-filter convolutional neural network for modeling quantum interactions". In: Advances in Neural Information Processing Systems 30 (2017).

[5] J. Klicpera et al. "Directional Message Passing for Molecular Graphs". In: International Conference on Learning Representations (2020).

[6] I. Batatia et al. "A foundation model for atomistic materials chemistry". In: The Journal of Chemical Physics 163.18 (2024), 184110. doi: 10.1063/5.0155322.

[7] T. Geffner et al. Proteina: Scaling Flow-based Protein Structure Generative Models. 2025. arXiv: 2503.00710 [cs.LG]. url: https://arxiv.org/abs/2503.00710.

[8] Z. Lin et al. "Evolutionary-scale prediction of atomic-level protein structure with a language model". In: Science 379.6637 (2023), pp. 1123–1130. doi: 10.1126/science.ade2574.

[9] J. Dauparas et al. "Robust deep learning–based protein sequence design using ProteinMPNN". In: Science 378.6615 (2022), pp. 49–56. doi: 10.1126/science.add2187.