PolyComp: open-source field theory for polymers

Emmit Pert[1] **and Grant Rotskoff**[1¶]

1 Stanford University ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 05 November 2025
Published: unpublished

Summary

Field theoretic methods provide insight into the phase behavior of complex polymer assemblies, but sophisticated numerical methods are required to efficiently simulate nontrivial systems. By representing the constituent polymers through their density and chemical potential fields, the mesoscale organization can be determined without explicitly modeling the microscopic degrees of freedom, which leads to superior scalability compared with particle-based simulations. Polymer field theories accurately model a wide variety of systems, but currently open-source codes that implement state-of-the-art methods for field theoretic simulation are not broadly available. This software provides a flexible platform for numerical simulations of polymer field theories, focused on applications to charged polymers.

Statement of Need

`PolyComp` is a Python-based implementation of Field Theoretic Simulation for an Auxiliary Field Theory with Complex Langevin integration. It is optimized for and runs on GPU hardware using CuPy to handle lower-level operations while still being readable to any user familiar with NumPy. The API is designed to allow for straightforward simulation of linear polymer melts and can also accommodate nanoparticles and affixed polymer brushes. The development of the underlying methods has been ongoing for many years and our implementation provides small modifications to the common polymer field theory methods (G. Fredrickson, 2006) (G. H. Fredrickson & Delaney, 2023), mainly around the handling of the system's incompressibility constraint. Although the field is well-developed, the lack of open-source code was a significant impediment to our work, and this release of a simpler, Python-based package will save significant time for those entering the field and looking to apply these methods. The source code for PolyComp has been archived to Zenodo with the linked DOI: (E. Pert & Rotskoff, 2025). Detailed derivations of methods are available in a previous publication (E. K. Pert, Hurst, et al., 2025).

State of the Field

The dominant group producing work in this space is the Fredrickson Group, but their codebase is currently closed-source, so this project was initiated as a way to construct an open-source polymer field theory simulation package. Another codebase, `langevin-fts` (Yong & Kim, 2025), was developed independently and concurrently with our work, which was highly developed by the time of the first release. Today, these two packages have diverged into different specializations: whereas the Kim group focuses on efficient algorithms for branched polymers and machine-learning integration, our work focuses on biological systems involving linear polymers and the necessity for precise phase determination. There is sophisticated, open-source work for computing self-consistent field theories (Arora et al., 2016). This code is not suitable for the coacervation-type problems we are interested in studying, as it cannot run Complex Langevin sampling. We built this project in response to the lack of an open-source codebase in the field. It will allow easier implementation by future researchers of the techniques described by existing field theory methods (G. Fredrickson, 2006), particularly those relevant for biological systems.

Software Design

PolyComp's main differentiating feature upon development was its open-source nature. The design principles for the codebase were transparency, functionality, and efficiency. Because the math motivating field theory simulations is quite dense, the code is written to be as transparent as possible, allowing for easy comprehension, modification, and verification. In particular, the usage of CuPy for the bulk of the operations means that the code is parseable by any programmer who is familiar with the common NumPy library. Secondly, the code is functional, with a particular emphasis on interfacing with complex biological systems and machine learning methods for optimization. It has been extensively tested for correctness, and the native Python makes interfaces with modern machine learning methods for optimization easy. Finally, the code is efficient, having been optimized for GPU performance with both CuPy's FFT library and our own kernels to speed up the expensive operation of solving the modified diffusion equation.

Research impact statement

PolyComp has been used to simulate biological systems examining mRNA encapsulation for drug delivery (E. K. Pert, Hurst, et al., 2025), formation of nanoparticle superlattices (Ye et al., 2025) and as a base for the development of machine-learned acceleration to general polymer field theories (E. K. Pert, Batton, et al., 2025). The code is well-adapted for biological simulation and is currently being used to simulate intrinsically disordered proteins in our group. While the usage of the code thus far has been limited to our group, it is becoming a workhorse program internally and its simple usage, documentation, and proven usefulness means that we hope it will be adopted more broadly for those looking to start learning about polymer field theory simulation or running experiments with polymeric systems.

AI usage disclosure

The core functionality of the code was primarily developed from 2020-2022, without the use of any AI assistance. Later Google Gemini was used to help develop unit tests, proofread code, and build documentation to prepare for public release.

Acknowledgements

We acknowledge contributions from the many members of the Rotskoff Group who have helped with conceptualization, debugging, and optimization of this project, in particular Sherry Li, Clay Batton, Andy Mitchell, and Nicholas Juntunen.

References

Arora, A., Qin, J., Morse, D. C., Delaney, K. T., Fredrickson, G. H., Bates, F. S., & Dorfman, K. D. (2016). Broadly Accessible Self-Consistent Field Theory for Block Polymer Materials Discovery. *Macromolecules*, *49*(13), 4675–4690. https://doi.org/10.1021/acs.macromol.6b00107

Fredrickson, G. (2006). *The Equilibrium Theory of Inhomogeneous Polymers*. Oxford University Press.

Fredrickson, G. H., & Delaney, K. T. (2023). *Field-Theoretic Simulations in Soft Matter and Quantum Fluids* (1st ed.). Oxford University PressOxford. https://doi.org/10.1093/oso/9780192847485.001.0001

Pert, E. K., Batton, C. H., Li, X., Dunne, S., & Rotskoff, G. M. (2025). Scaling Field-Theoretic Simulation for Multicomponent Mixtures with Neural Operators. *Journal of Chemical Theory and Computation*. https://doi.org/10.1021/acs.jctc.5c00102

Pert, E. K., Hurst, P. J., Waymouth, R. M., & Rotskoff, G. M. (2025). Coacervation drives morphological diversity of mRNA encapsulating nanoparticles. *The Journal of Chemical Physics*, *162*(7), 074902. https://doi.org/10.1063/5.0235799

Pert, E., & Rotskoff, G. (2025). *PolyComp: v1.0.0*. https://doi.org/10.5281/zenodo.17497822

Ye, M., Pert, E. K., Lee, M. S., Li, Y., Nishimura, A., Li, R. L., Ngo, S. H., Huang, W. Y., Rotskoff, G. M., & Macfarlane, R. J. (2025). Nanoparticle Superlattices Assembled via Rapid Solvent Destabilization of Macromolecular Ligands. *ACS Nano*, *19*(39), 34847–34857. https://doi.org/10.1021/acsnano.5c10677

Yong, D., & Kim, J. U. (2025). Dynamic Programming for Chain Propagator Computation of Branched Block Copolymers in Polymer Field Theory Simulations. *Journal of Chemical Theory and Computation*, *21*(7), 3676–3690. https://doi.org/10.1021/acs.jctc.5c00103