DynAgent: A Modular Multi-Agent Framework for Autonomous Protein-Ligand Molecular Dynamics Simulations

Salomé Guilbert[1,†], Cassandra Masschelein[1,†], Jeremy Goumaz [1], Bohdan Naida[1], Philippe Schwaller[1,2]
[1]École Polytechnique Fédérale de Lausanne (EPFL)
[2]National Centre of Competence in Research (NCCR) Catalysis
{salome.guilbert,cassandra.masschelein,philippe.schwaller}@epfl.ch

Abstract

Molecular dynamics (MD) simulations are indispensable for probing the structure, dynamics, and functions of biomolecular systems, including proteins and protein–ligand complexes. Despite their broad utility in drug discovery and protein engineering, the technical complexity of MD setup—encompassing parameterization, input preparation, and software configuration—remains a major barrier for widespread and efficient usage. Agentic LLMs have demonstrated their capacity to autonomously execute multi-step scientific processes, and to date, they have not successfully been used to automate protein-ligand MD workflows. Here, we present **DynAgent**, a modular multi-agent framework that autonomously designs and executes complete MD workflows for both protein and protein–ligand systems, and integrates free energy binding affinity calculations with the MM-PB(GB)SA method. The framework integrates dynamic tool use, web search, PaperQA, and a self-correcting behavior. **DynAgent** comprises three specialized agents, interacting to plan the experiment, perform the simulation, and analyze the results. We evaluated its performance across eight benchmark systems of varying complexity, assessing success rate, efficiency, and adaptability. **DynAgent** reliably performed full MD simulations, corrected runtime errors through iterative reasoning, and produced meaningful analyses of protein–ligand interactions. This automated framework paves the way toward standardized, scalable, and time-efficient molecular modeling pipelines for future biomolecular and drug design applications.

1 Introduction

Molecular dynamics (MD) simulations are a cornerstone of computational chemistry and biophysics, enabling atomistic modeling of molecular interactions and conformational changes over time (1; 2; 3; 4). They provide crucial insights into the structure, stability, and function of biomolecules such as proteins, nucleic acids, and membranes, and have become indispensable tools in areas such as drug design and protein engineering.

In practice, MD simulations are typically performed using specialized engines such as GROMACS (5), OpenMM (6), or AmberTools (7). However, preparing a molecular system for simulation remains one of the most error-prone and labor-intensive steps in the workflow (8). Several frameworks have sought to automate MD simulations, including CHAPERONg (9) and PyAutoFEP (10), which facilitate simulation setup and analysis or integrate free-energy perturbation (FEP) workflows (11; 12). However, these pipelines remain rigid and domain-specific: extending them to new systems or simulation engines often requires significant manual modification and expert knowledge.

[†]These authors contributed equally.

European Conference on Neural Information Processing Systems (EurIPS) Workshop: SimBioChem 2025.

Recent advances in large language models (LLMs) (13; 14; 15) have introduced a new paradigm for scientific automation. LLMs can interpret natural language instructions, reason about domain-specific constraints, and orchestrate external tools, offering a flexible control layer that can adapt to new workflows and dynamically recover from errors. The emergence of **agentic LLMs** (16; 17; 18) has demonstrated their capacity to autonomously execute multi-step scientific processes, including early efforts in MD workflow automation through MDCrow (19) and NAMD-Agent (20). Both systems successfully automate protein-only simulations using OpenMM or CHARMM-GUI, respectively, and can handle basic preprocessing and simulation stages. Yet, they remain limited in scope—unable to process protein–ligand systems, perform adaptive recovery from simulation errors, or generalize across simulation platforms.

In this work, we introduce **DynAgent**, a modular multi-agent framework that autonomously designs and executes complete MD workflows for both protein and protein–ligand systems, with the option to perform free energy binding affinity calculations with the MM-PB(GB)SA method. Unlike prior systems, **DynAgent** separates high-level reasoning (e.g., parameter planning, structure retrieval) from low-level execution (e.g., simulation setup, file handling). This design enables dynamic tool use, retrieval-augmented parameter selection, and self-correcting behavior during simulation runs. By combining domain-aware reasoning with multi-agent coordination (21; 18), **DynAgent** bridges natural-language goal specification and robust computational execution in molecular simulation.

Table S1 summarizes key differences between existing agentic MD frameworks and **DynAgent**. Our system uniquely supports protein–ligand systems, integrates retrieval from external databases (web search as well as literature search with PaperQA (22; 17; 23)), and implements adaptive tool selection and error correction, thus extending the capabilities of current MD automation pipelines.

2 Methods

MD Workflow Our system, **DynAgent**, is designed as a multi-agent framework that autonomously prepares, executes, and analyzes MD simulations. The architecture extends previous agentic approaches such as MDCrow (19) and NAMD-Agent (20) by introducing a dedicated planner agent, dynamic feedback with error correction, and by enabling simulations of protein–ligand complexes.

The workflow (Figure 1) is organized into three main components:

1. **Planner Agent (`PrepAgent`)**: Extracts user intent and scientific requirements from natural language input, retrieves structural information, and constructs a context-aware simulation plan. The planner leverages retrieval-augmented reasoning (24) and tool invocation to retrieve the PDB file and select appropriate parameters for the MD simulation such as temperature and duration.
2. **Molecular Dynamics Agent (`DynAgent`)**: Interprets the plan produced by the `PrepAgent` and autonomously executes it through a loop of tool invocation $\rightarrow$ validation $\rightarrow$ reflection (18). Each subtask is validated against the filesystem and output logs, allowing the agent to detect and repair common runtime errors by regenerating or modifying input files (e.g., correcting missing atom names or updating topology directives).
3. **Analyzer Agent**: Post-processes MD trajectories and outputs descriptive analyses. Standard GROMACS modules are used to compute RMSD, radius of gyration, RMSF, and hydrogen-bond statistics. The agent further provides qualitative summaries of stability, convergence, and ligand–protein interactions in natural language form.
4. **Execution environment**: Both agents operate within a sandboxed directory structure, ensuring safe read/write operations and reproducibility. Interactions with external LLMs are routed through LiteLLM, which acts as a unified interface for model invocation, request routing, and provider abstraction. The framework supports both AmberTools and GROMACS, providing flexibility beyond prior works restricted to OpenMM or CHARMM-GUI.

A representative workflow is shown in Figure S1, illustrating the difference between a protein-only simulation and a protein–ligand complex. The planner dynamically selects additional steps for ligand parameterization and topology construction, automatically adjusting the downstream MD setup.

Agent Toolset Each agent interacts with a structured set of domain-specific tools, defined through a formal schema that includes tool names, descriptions, and JSON input specifications. The `PrepAgent` loads these tool schemes at runtime and delegates tasks to corresponding Python functions. The major

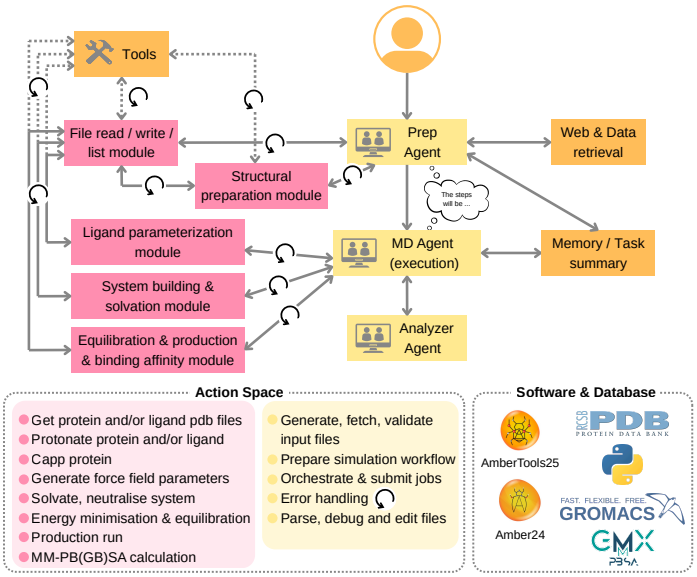


Figure 1: Overview of the framework. The `PrepAgent` constructs a context-aware simulation plan, the `DynAgent` executes it with error-corrective reasoning, and the Analyzer interprets the resulting trajectories.

tool families include structural preparation, ligand parameterization, system building and solvation, equilibration and production, and web and data retrieval (Figure 1).

- **Structural preparation**: cleaning and protonation of PDB files, capping termini, and correcting atom naming inconsistencies.
- **Ligand parameterization**: generation of topology and force field parameters using `acpype` or `antechamber`, automatically detected from the ligand identifier.
- **System building and solvation**: generation of topology and coordinate files with `tleap`, using the AMBER force fields 14ffsb (25) for proteins and GAFF2 (26) for ligands. The system is solvated with the TIP3P model (27) and neutralized by ion placement.
- **Equilibration and production**: energy minimization, NVT/NPT equilibration, and production MD runs executed through `GROMACS`.
- **Web and data retrieval**: web search or PaperQA (22; 17; 23) queries to determine experimental conditions such as recommended temperature or ligand information.

This modular design allows for tool reuse and extension across multiple agents and systems, and facilitates transparent LLM–tool interactions.

Error Correction and Feedback Memory A key feature of the framework is its iterative feedback mechanism, which is similar to the mechanism outlined in (28; 18), which enables self-correction for failed or incomplete subtasks. Each tool invocation is sandboxed, logged, and summarized into a compressed message context to maintain continuity between attempts. If a subtask fails (ie. due to missing files, naming errors, or incompatible parameters) the agent analyzes the log output and proposes a corrected re-execution plan.

Retrieval-Augmented Generation for Parameter Selection The `PrepAgent` integrates retrieval-augmented generation (RAG) to enable domain-aware parameterization. During the planning phase, the agent can use the its built-in `web_search` tool to query the internet and obtain up-to-date knowledge and parameter recommendations, while PaperQA (22; 17; 23) can be used to search uploaded literature for system specific references regarding the protein and ligands, as well as software details. The LLM integrates retrieved information into its reasoning to select MD parameters (as well as to identify possible errors). This allows the LLM to justify and select simulation parameters (e.g., 310 K for human proteins, 298 K for mesophilic enzymes). By grounding parameter decisions

in primary data sources, the agent improves the scientific interpretability and reproducibility of molecular dynamics setups.

Analysis and Visualization Upon successful completion of the MD run, the Analyzer agent performs automated trajectory analysis. It generates RMSD, radius of gyration, RMSF, and hydrogen-bond statistics, using GROMACS analysis tools. Each plot is accompanied by an automatically generated textual interpretation describing structural stability, folding dynamics, and binding behavior.

3 Results

The performance and generalizability of **DynAgent** was evaluated across eight distinct simulation setups, five of which were well-established protein-ligand complexes that are widely used in the literature for benchmarking MD and free energy calculation methodologies (eight systems detailed in Table S2). The other three systems didn't contain a ligand so that we could test the agent's ability to adapt its workflow and use of tools based on the system to simulate. The inputs directly originated from the PDB (29), as requested either by the user prompt or the user upload. The temperature and simulation duration was chosen by the agent. The agent's success was measured by its ability to correctly execute the tools required for the simulation to succeed, the creation of the necessary (non-empty) files, and the stability of the simulation.

The selected systems span diverse protein families and ligand chemotypes, providing a representative testbed for assessing the agent's ability to set up, parameterize, and execute MD workflows across different biochemical contexts. The chosen systems and PDB IDs are detailed in the Supplementary Information.

DynAgent successfully performed a production MD run for all five systems, as evidenced by the output RMSD, RMSF, and radius of gyration plots, consistent with equilibrated systems (outputs for PDB:5UEZ system shown in Figure 2). Human inspection of the generated files confirmed that each system was correctly parameterized by the agent.

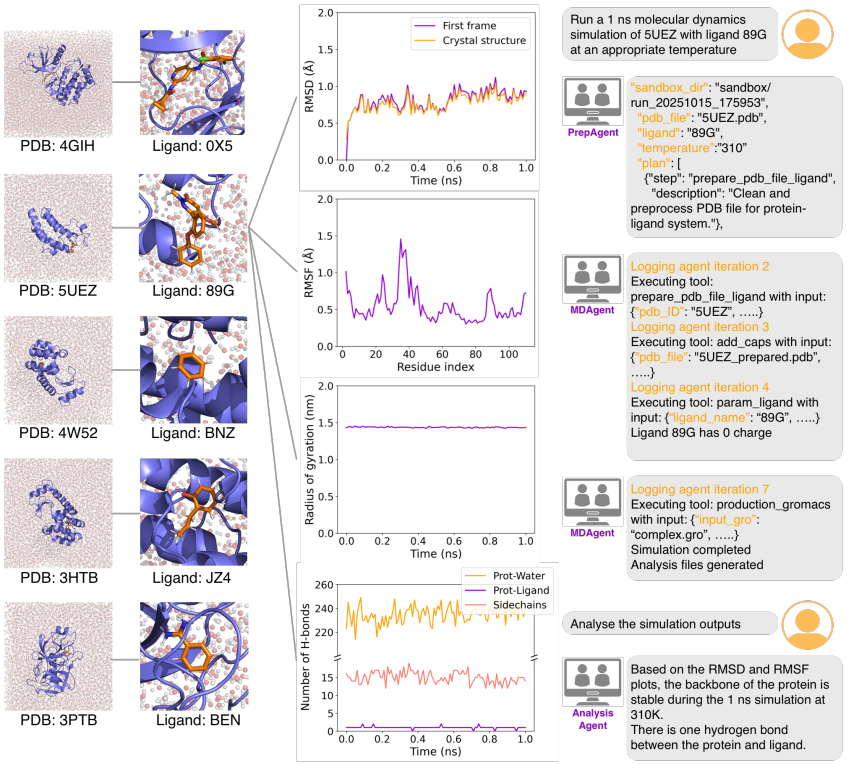
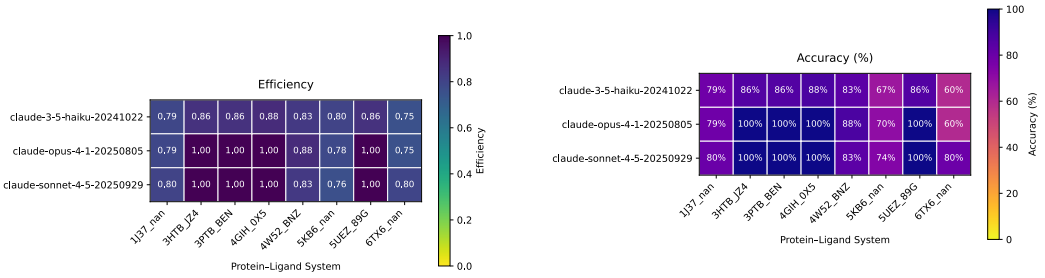


Figure 2: Solvated protein-ligand complex systems. Representative output plots automatically generated by the Analyzer for the 5UEZ system: RMSD, RMSF, radius of gyration, and hydrogen-bond analysis. Example of a dynamic workflow with the agent.

We evaluated the performance of **DynAgent** with three different Claude agents: Haiku 3.5, Opus 4.1, and Sonnet 4.5. **DynAgent's** error correction efficiency was calculated from the number of iterations performed per experiment, divided by the minimum number of iterations required for completion (Figure 3a). Claude Opus and Sonnet were as efficient, while Haiku underperformed for all experiments. Efficiency losses most often occur when the agent uses tools in the wrong order, or with incorrect inputs, which increases the number of iterations and setup time required. Next, accuracy (defined as the proportion of successfully completed tasks out of the human defined required tasks) was evaluated (Figure 3b). Required subtasks are detailed in Figure S1. Claude Opus and Sonnet outperformed Haiku again. 100 % accuracy was obtained for 4 protein-ligand systems. The fifth system, 4W52_BNZ, completed the MD simulation, and the 80 % accuracy demonstrated the agent's ability to correct errors. Indeed, the tools were not all used successfully in the first attempt, but with more iterations, the simulation reached completion. Finally, three systems did not reach full completion: 1J37, 5KB6, and 6TX6. The structures with missing residues and PDB input files with unknown atom names were the cause of the Amber and GROMACS tools failing in these cases.



(a) Efficiency of the different LLMs, defined as minimum required iterations divided by total iterations.

(b) Accuracy of the different LLMs, defined as the proportion of successfully completed tasks out of the human defined required tasks.

Figure 3: Comparison of efficiency and accuracy across different LLM systems, shown side by side.

Once a protein-ligand simulation is complete, **DynAgent** offers the option to perform MM-PG(GB)SA calculations to obtain ligand binding free energies, with the gmx_MMPBSA tool. (12; 30) The integration of this binding affinity tool was tested against a set of inhibitor molecules for the bromodomain 1 of Bromodomain-containing protein 4 (BRD4 BD1). The binding affinity values obtained with **DynAgent** were compared with experimental IC_{50} values. (31) Two inhibitors (compounds **17** and **25** from Jiang et al. (31)) were first docked into BRD4 BD1 (PDB: 6JJ3 (31)) with the software GNINA (32). **DynAgent** then autonomously performed 10 ns MD simulations of the complex structures and determined a binding free energy value with the PB solvation model in each case. As seen in Table 1, the $\Delta\Delta G$ values given by the agent allow to differentiate a very potent inhibitor (compound **25**) from a less potent one (compound **17**). The MM-PBSA method further provides more insights into the nature of the binding interactions. For instance, while Van der Waals interactions stabilize better compound **25** with respect to compound **17**, stronger electrostatic interactions interactions seem to be the cause of the major difference in binding between the two inhibitors.

Table 1: Comparison of the MM-PBSA binding affinity values obtained with **DynAgent** and experimental IC_{50} values (31), on a set of two inhibitor compounds binding to BRD4 BD1.

Compounds	**Experimental** IC_{50} (nM)	**DynAgent MM-PBSA** $\Delta\Delta G$ (kcal/mol)	**DynAgent** Δ**VdVaals** (kcal/mol)	**DynAgent** Δ**EEL** (kcal/mol)
Compound **17**	2436±290	-23.86	-33.22	-10.75
Compound **25**	49.5±10.3	-26.06	-34.09	-16.57

4 Discussion

Agentic systems like **DynAgent** illustrate the feasibility of "AI scientists" (17; 33) capable of performing full experimental workflows in computational chemistry. Unlike rigid automation scripts, our multi-agent architecture generalizes across simulation goals and tools by reasoning about both

physical constraints and software syntax. The streamlined usability of **DynAgent** enables more efficient incorporation of computational binding-affinity calculations into drug-optimization campaigns. However, current limitations include dependency on external LLM APIs, lack of long-term memory across sessions, and limited quantitative evaluation of trajectory quality. Future work will integrate reinforcement feedback from simulation outcomes and benchmarking datasets (e.g., MD17 (34), PDBbind (35)) to enhance self-improvement.

The modularity of the agentic architecture will allow the facile extension of possible experiments. The context-aware nature of the `PrepAegent` makes it possible to suggest the most appropriate workflows for a series of new tasks. The aim is to introduce diverse systems into the pipeline, such as DNA, membrane proteins, and multimers containing several ligands. The goal is also to explore the analytic capabilities of the agent, to compare different simulations, and to analyze the outputs with the knowledge of the specific system.

Acknowledgment

CM acknowledges Valence Labs for financial support. SG acknowleges support from Intel and Merck KGaA via the AWASES programme, and the Swiss National Science Foundation (SNSF).

References

[1] S. A. Hollingsworth, R. O. Dror, Molecular Dynamics Simulation for All , Neuron 99 (6) (2018) 1129–1143. doi:https://doi.org/10.1016/j.neuron.2018.08.011.

[2] M. Karplus, J. A. McCammon, Molecular dynamics simulations of biomolecules, Nature structural biology 9 (9) (2002) 646–652.

[3] T. Schlick, S. Portillo-Ledesma, Biomolecular modeling thrives in the age of technology, Nature computational science 1 (5) (2021) 321–331.

[4] T. Schlick, R. Collepardo-Guevara, L. A. Halvorsen, S. Jung, X. Xiao, Biomolecular modeling and simulation: a field coming of age, Quarterly reviews of biophysics 44 (2) (2011) 191–228.

[5] M. J. Abraham, T. Murtola, R. Schulz, S. Páll, J. C. Smith, B. Hess, E. Lindahl, Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers, SoftwareX 1 (2015) 19–25.

[6] P. Eastman, J. Swails, J. D. Chodera, R. T. McGibbon, Y. Zhao, K. A. Beauchamp, L.-P. Wang, A. C. Simmonett, M. P. Harrigan, C. D. Stern, et al., Openmm 7: Rapid development of high performance algorithms for molecular dynamics, PLoS computational biology 13 (7) (2017) e1005659.

[7] D. A. Case, H. M. Aktulga, K. Belfon, D. S. Cerutti, G. A. Cisneros, V. W. D. Cruzeiro, N. Forouzesh, T. J. Giese, A. W. Gotz, H. Gohlke, et al., Ambertools, Journal of chemical information and modeling 63 (20) (2023) 6183–6191.

[8] J. A. Lemkul, Introductory tutorials for simulating protein dynamics with gromacs, The Journal of Physical Chemistry B 128 (39) (2024) 9418–9435.

[9] A. A. Yekeen, O. A. Durojaye, M. O. Idris, H. F. Muritala, R. O. Arise, Chaperong: A tool for automated gromacs-based molecular dynamics simulations and trajectory analyses, Computational and structural biotechnology journal 21 (2023) 4849–4858.

[10] L. Carvalho Martins, E. A. Cino, R. S. Ferreira, Pyautofep: An automated free energy perturbation workflow for gromacs integrating enhanced sampling methods, Journal of Chemical Theory and Computation 17 (7) (2021) 4262–4273.

[11] E. B. Lenselink, J. Louvel, A. F. Forti, J. P. van Veldhoven, H. de Vries, T. Mulder-Krieger, F. M. McRobb, A. Negri, J. Goose, R. Abel, et al., Predicting binding affinities for gpcr ligands using free-energy perturbation, ACS omega 1 (2) (2016) 293–304.

[12] M. S. Valdés-Tresanco, M. E. Valdés-Tresanco, P. A. Valiente, E. Moreno, gmx_mmpbsa: a new tool to perform end-state free energy calculations with gromacs, Journal of chemical theory and computation 17 (10) (2021) 6281–6291.

[13] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al., Improving language understanding by generative pre-training (2018).

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[15] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[16] A. M. Bran, S. Cox, O. Schilter, C. Baldassari, A. D. White, P. Schwaller, Augmenting large language models with chemistry tools, Nature Machine Intelligence 6 (5) (2024) 525–535.

[17] M. D. Skarlinski, S. Cox, J. M. Laurent, J. D. Braza, M. Hinks, M. J. Hammerling, M. Ponnapati, S. G. Rodriques, A. D. White, Language agents achieve superhuman synthesis of scientific knowledge, arXiv preprint arXiv:2409.13740 (2024).

[18] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, S. Yao, Reflexion: Language agents with verbal reinforcement learning, Advances in Neural Information Processing Systems 36 (2023) 8634–8652.

[19] Q. Campbell, S. Cox, J. Medina, B. Watterson, A. D. White, Mdcrow: Automating molecular dynamics workflows with large language models, arXiv preprint arXiv:2502.09565 (2025).

[20] A. Chandrasekhar, A. B. Farimani, Automating md simulations for proteins using large language models: Namd-agent, arXiv preprint arXiv:2507.07887 (2025).

[21] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, A. Anandkumar, Voyager: An open-ended embodied agent with large language models, arXiv preprint arXiv:2305.16291 (2023).

[22] S. Narayanan, J. D. Braza, R.-R. Griffiths, M. Ponnapati, A. Bou, J. Laurent, O. Kabeli, G. Wellawatte, S. Cox, S. G. Rodriques, A. D. White, Aviary: training language agents on challenging scientific tasks, arXiv preprent arXiv:2412.21154 (2024).
URL https://doi.org/10.48550/arXiv.2412.21154

[23] J. Lála, O. O'Donoghue, A. Shtedritski, S. Cox, S. G. Rodriques, A. D. White, Paperqa: Retrieval-augmented generative agent for scientific research, arXiv preprint arXiv:2312.07559 (2023).
URL https://doi.org/10.48550/arXiv.2312.07559

[24] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, Advances in neural information processing systems 33 (2020) 9459–9474.

[25] J. A. Maier, C. Martinez, K. Kasavajhala, L. Wickstrom, K. E. Hauser, C. Simmerling, ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from ff99SB, J. Chem. Theory Comput. 11 (8) (2015) 3696–3713. doi:https://doi.org/10.1021/acs.jctc.5b00255.

[26] X. He, V. H. Man, W. Yang, T.-S. Lee, J. Wang, A fast and high-quality charge model for the next generation general amber force field, The Journal of chemical physics 153 (11) (2020).

[27] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, M. L. Klein, Comparison of simple potential functions for simulating liquid water, J. Chem. Phys. 79 (2) (1983) 926–935. doi:https://doi.org/10.1063/1.445869.

[28] Y. Zou, A. H. Cheng, A. Aldossary, J. Bai, S. X. Leong, J. A. Campos-Gonzalez-Angulo, C. Choi, C. T. Ser, G. Tom, A. Wang, et al., El agente: An autonomous agent for quantum chemistry, Matter 8 (7) (2025).

[29] S. K. Burley, R. Bhatt, C. Bhikadiya, C. Bi, A. Biester, P. Biswas, S. Bittrich, S. Blaumann, R. Brown, H. Chao, et al., Updated resources for exploring experimentally-determined pdb structures and computed structure models at the rcsb protein data bank, Nucleic acids research 53 (D1) (2025) D564–D574.

[30] B. R. Miller III, T. D. McGee Jr, J. M. Swails, N. Homeyer, H. Gohlke, A. E. Roitberg, Mmpbsa. py: an efficient program for end-state free energy calculations, Journal of chemical theory and computation 8 (9) (2012) 3314–3321.

[31] F. Jiang, Q. Hu, Z. Zhang, H. Li, H. Li, D. Zhang, H. Li, Y. Ma, J. Xu, H. Chen, et al., Discovery of benzo [cd] indol-2 (1 h)-ones and pyrrolo [4, 3, 2-de] quinolin-2 (1 h)-ones as bromodomain and extra-terminal domain (bet) inhibitors with selectivity for the first bromodomain with potential high efficiency against acute gouty arthritis, Journal of medicinal chemistry 62 (24) (2019) 11080–11107.

[32] A. T. McNutt, Y. Li, R. Meli, R. Aggarwal, D. R. Koes, Gnina 1.3: the next increment in molecular docking with deep learning, Journal of Cheminformatics 17 (1) (2025) 28.

[33] H. Wang, T. Fu, Y. Du, W. Gao, K. Huang, Z. Liu, P. Chandak, S. Liu, P. Van Katwyk, A. Deac, et al., Scientific discovery in the age of artificial intelligence, Nature 620 (7972) (2023) 47–60.

[34] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, K.-R. Müller, Machine learning of accurate energy-conserving molecular force fields, Science advances 3 (5) (2017) e1603015.

[35] R. Wang, X. Fang, Y. Lu, S. Wang, The pdbbind database: Collection of binding affinities for protein- ligand complexes with known three-dimensional structures, Journal of medicinal chemistry 47 (12) (2004) 2977–2980.

[36] N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, G. R. Hutchison, Open babel: An open chemical toolbox, Journal of cheminformatics 3 (1) (2011) 33.

[37] M. Marquart, J. Walter, J. Deisenhofer, W. Bode, R. Huber, The geometry of the reactive site and of the peptide groups in trypsin, trypsinogen and its complexes with inhibitors, Structural Science 39 (4) (1983) 480–490.

[38] L. Wang, J. K. Pratt, T. Soltwedel, G. S. Sheppard, S. D. Fidanze, D. Liu, L. A. Hasvold, R. A. Mantei, J. H. Holms, W. J. McClellan, et al., Fragment-based, structure-enabled discovery of novel pyridones and pyridone macrocycles as potent bromodomain and extra-terminal domain (bet) family bromodomain inhibitors, Journal of medicinal chemistry 60 (9) (2017) 3828–3850.

[39] J. Liang, V. Tsui, A. Van Abbema, L. Bao, K. Barrett, M. Beresini, L. Berezhkovskiy, W. S. Blair, C. Chang, J. Driscoll, et al., Lead identification of novel and selective tyk2 inhibitors, European journal of medicinal chemistry 67 (2013) 175–187.

[40] M. Merski, M. Fischer, T. E. Balius, O. Eidam, B. K. Shoichet, Homologous ligands accommodated by discrete conformations of a buried cavity, Proceedings of the National Academy of Sciences 112 (16) (2015) 5039–5044.

[41] S. E. Boyce, D. L. Mobley, G. J. Rocklin, A. P. Graves, K. A. Dill, B. K. Shoichet, Predicting ligand binding affinity with alchemical free energy methods in a polar model binding site, Journal of molecular biology 394 (4) (2009) 747–763.

[42] H. M. Kim, D. R. Shin, O. J. Yoo, H. Lee, J.-O. Lee, Crystal structure of drosophila angiotensin i-converting enzyme bound to captopril and lisinopril, FEBS letters 538 (1-3) (2003) 65–70.

[43] R. Oliveira, R. Neto, C. Polo, C. Tonoli, M. Murakami, K. Franchini, High-resolution structure of the adenosine kinase from mus musculus in complex with adenosine, To be published (2017).

[44] S. W. Draxler, M. Bauer, C. Eickmeier, S. Nadal, H. Nar, D. Rangel Rojas, D. Seeliger, M. Zeeb, D. Fiegen, Hybrid screening approach for very small fragments: X-ray and computational screening on fkbp51, Journal of medicinal chemistry 63 (11) (2020) 5856–5864.

Supplementary Information

A DynAgent Capabilities

Here, we outline the current state of both existing agentic frameworks that run MD simulations, and our framework. We specify both the types of systems it can accept, and the range of tasks it can perform.

Table S1: Comparison of capabilities across existing agentic MD frameworks. **DynAgent** extends prior systems by enabling ligand handling, adaptive error correction, and retrieval-augmented parameter selection.

Capability	**MDCrow**	**NAMD-Agent**	**DynAgent (ours)**
Protein-only simulations	Yes	Yes	Yes
Protein–ligand simulations	No	No	Yes
Automated structure cleaning	Partial	Yes	Yes
Force field parameterization	Yes (OpenMM)	Yes (CHARMM-GUI)	Yes (AmberTools/GROMACS)
Retrieval from literature/databases	Yes (limited)	No	Yes (RAG via RCSB/PubChem/Google)
Adaptive tool selection	No	Partial	Yes
Error correction and recovery	No	No	Yes
Natural-language task specification	Yes	Yes	Yes
Modular multi-agent architecture	No	Partial	Yes
Cross-platform compatibility	OpenMM only	CHARMM only	GROMACS / AmberTools

B Human defined workflow

There is a series of tools provided to the agent, that it should use correctly for simulation success. The tools provided include:

- Fetch and save the pdb file.
- Preparation of pdb files. This comprises cleaning the protein file, capping it (C- and N-termini with ACE and NME), extracting the ligand file, and protonating it at pH = 7 using Open Babel (36).
- Preparation of ligand parameters with AmberTools.
- Generation of the protein-ligand complex file if a ligand is present.
- Web and data retrieval: query of external sources (RCSB, PubChem, Google) to determine experimental conditions such as recommended temperature or ligand information.
- Generation parameters for the protein or protein-ligand systems using tLEaP. The protein is parameterized with the ff14sb force field and the ligand with GAFF2, with AM1-BCC charges.
- Preparation the simulation box with AmberTools, by solvating with the TIP3P model and neutralising with ions.
- Equilibration the system with GROMACS: it includes energy minimisation until the maximum force on all atoms is inferior to 10.0 kJ/mol, 100 ps NVT and 100 ps NPT simulations to equilibrate the temperature and pressure of the system. Both properties are plotted and analyzed for convergence.
- Generation of the NPT production run with GROMACS, with the temperature either specified by the user, or determined by the agent with its web-search tool.
- Analysis of the RMSD, RMSF, radius of gyration, number of hydrogen-bonds (between protein side-chains, protein-ligand and protein-water) plots generated by GROMACS.

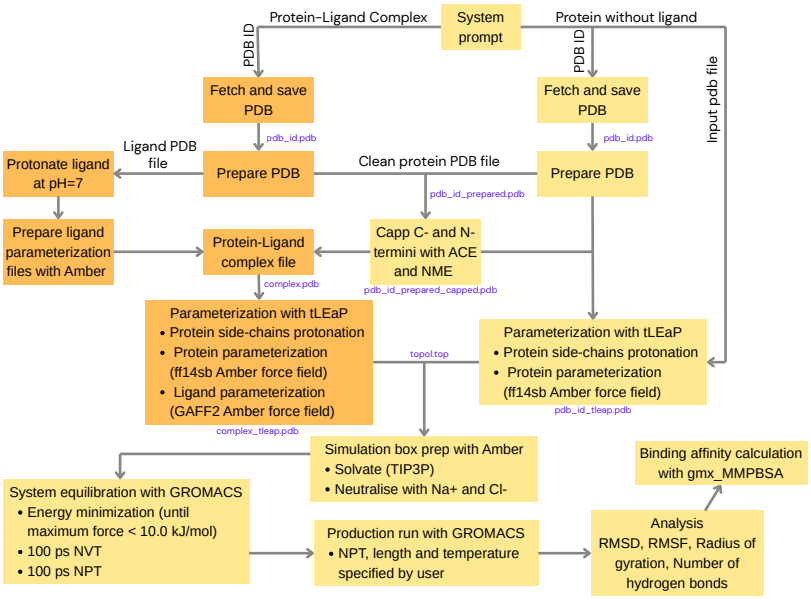


Figure S1: Agentic workflow for a protein-only system (yellow) and a protein-ligand system (orange+yellow). Steps include structure retrieval, preprocessing, generation of input force field parameters, solvation, equilibration, and production.

C Reference Systems for Evaluation

First, β-Trypsin–Benzamidine in complex with benzamidine (PDB ID: 3PTB(37)) is one of the most popular and historically important benchmarks for protein–ligand simulations, owing to its well-characterized binding pocket and simple ligand structure. Second, Bromodomain-containing Protein 4 bound to 5-methoxy-2-methyl-6-(2-phenoxyphenyl)pyridazin-3(2H)-one (PDB ID: 5UEZ(38)) represents a pharmaceutically relevant target. This system introduces a more chemically complex ligand with multiple aromatic and heterocyclic moieties, allowing evaluation of the agent's handling, protonation, and parameterization of flexible ligands. Third, the Tyrosine Kinase 2 (JH1 domain) domain complexed with 2,6-dichloro-N-2-[(cyclopropylcarbonyl)amino]pyridin-4-ylbenzamide (PDB ID: 4GIH(39)) is a well-known benchmark system for free energy methods. The presence of halogen atoms and multiple rotatable bonds makes it an ideal test case for assessing the robustness of ligand parameterization and sampling accuracy. Then, the T4 Lysozyme L99A–Benzene mutant complexed with benzene (PDB ID: 4W52(40)) is frequently used as a model system for studying hydrophobic binding, validating force fields, and comparing computational and experimental binding thermodynamics. Finally, the T4 Lysozyme L99A/M102Q–2-Propylphenol mutant bound to 2-propylphenol (PDB ID: 3HTB(41)) extends the previous system to include a polar binding environment. This system is a common model for assessing protein-ligand MD simulations and binding affinities. Together, these five systems encompass a range of protein classes, ligand complexities, and interaction types, thereby providing a comprehensive evaluation of the LLM agent's capability to autonomously set up and conduct MD simulations across chemically diverse systems.

Table S2: Systems used to evaluate the LLM agent for molecular dynamics simulations (Protein-ligand complexes and Protein systems).

PDB ID	Protein	Ligand	Rationale for Selection
1J37 (42)	Drosophila AnCE	/	High resolution crystal structure of a protein monomer.
3HTB (41)	T4 Lysozyme L99A/M102Q	JZ4	Common model for studying polar cavity binding; probes hydrogen bonding and solvation effects.
3PTB (37)	β-Trypsin	BEN	Canonical benchmark system for protein–ligand MD; simple and well-characterized binding site.
4GIH (39)	Tyk2 (JH1 domain)	0X5	Popular benchmark for free energy methods; halogenated, conformationally complex inhibitor.
4W52 (40)	T4 Lysozyme L99A	BNZ	Minimal hydrophobic binding model; widely used to test force fields and thermodynamics.
5KB6 (43)	adenosine kinase	/	High resolution crystal structure of a well-studied protein.
5UEZ (38)	BRD4 (Bromodomain-containing protein 4)	89G	Pharmacologically relevant target; tests handling of flexible, aromatic ligands.
6TX6 (44)	Human FKBP51 FK1 Domain A19T mutant	/	High resolution crystal structure of a well-studied protein.