DrugNav: A Benchmark Dataset of Expert Trajectories for Developing and Evaluating LLM Agents in Multi-Step Drug Discovery

Anonymous Author(s)
Affiliation
Address
email

Abstract

The potential of Large Language Models (LLMs) in drug discovery is constrained by inadequate benchmarks. Current benchmarks, focused on single-tool calls in general scientific domains, fail to capture the complex, multi-step reasoning and execution required for pharmaceutical R&D. To address this critical gap, we introduce DrugNav, a new, open dataset of expert tool-calling trajectories tailored for drug discovery. DrugNav consists of high-fidelity, sequential tool interactions that solve complex queries, from target identification to lead optimization. Each trajectory documents the complete workflow of tool calls, intermediate reasoning, and outcomes, providing the necessary data to train agentic models on complex, multi-tool tasks. By providing a curated set of successful solution pathways, DrugNav is specifically designed to facilitate end-to-end, tool-integrated Reinforcement Learning for LLM agents. Our work will accelerate the development of capable autonomous systems, significantly reducing the time and cost of drug discovery and advancing AI-driven science.

1 AI Task Definition

Developing autonomous agents for drug discovery requires moving beyond an LLM's declarative knowledge [32, 7] to procedural, tool-driven execution [10, 24, 29, 16, 4, 21]. Progress is currently bottlenecked by benchmarks [21, 20, 4, 26, 27] that lack the complexity of real-world pharmaceutical research. They typically evaluate isolated, single-tool actions, not the long-horizon, multi-stage workflows central to the field—such as chaining bioinformatics queries with molecular simulation tools. A meaningful AI task must therefore move beyond single-step evaluations and instead benchmark the strategic orchestration of a diverse toolset to solve complex scientific problems.

To rigorously evaluate agents in this context, we formalize the AI task as learning a research policy $\pi(a_t|s_t)$. The state s_t represents the agent's complete experimental notebook at timestep t. It includes the initial user query (Q), such as "Find potential inhibitors for the EGFR T790M mutant," and the history of all preceding thoughts, actions, and observations $\{(th_i, a_i, o_i)\}_{i=0}^{t-1}$. The action a_t is a computational experiment, chosen from a high-dimensional action space $\mathcal{A}$ of tool calls. The available toolset $\mathcal{T}$ is tailored for drug discovery, containing essential bioinformatics tools (e.g., BLAST [3] for sequence search, AlphaFold [15, 2] for structure prediction), cheminformatics libraries (e.g., RDKit [19] for molecule manipulation), and simulation engines (e.g., AutoDock Vina [9, 33] for molecular docking). A full trajectory $\tau = \{(s_t, a_t, o_t)\}_{t=0}^{N}$ represents an stage-wise end-to-end research workflow. The agent's objective is to generate a trajectory that achieves a successful outcome, defined by criteria like identifying a molecule with a predicted binding affinity below a certain threshold or proposing a valid synthesis route. This framework allows us to benchmark the core capabilities essential for a computational chemist: strategic planning, correct tool selection and parameterization, error handling from failed simulations, and the synthesis of chemical and biological data.

Submitted to AI for Science workshop (NeurIPS 2025). Do not distribute.

2 Dataset Rational and Design

Existing agent datasets are often knowledge-centric and lack the procedural trajectories required for complex scientific workflows. DrugNav is architected to fill this "workflow gap," providing complete action sequences for imitation and reinforcement learning. To ensure real-world relevance, its tasks are structured around the canonical stages of drug discovery—from target identification to lead optimization—covering a spectrum of causally-linked and increasingly complex challenges.

Dataset Design Principles Our design philosophy is centered on creating a dataset that mirrors the complexity and structure of real-world computational drug discovery.

- **Stage-Wise Coverage of Drug Discovery:** Trajectories will be stratified across four canonical stages (Target Identification & Validation, Hit/Lead Discovery, Lead Optimization, and Preclinical Research), ensuring coverage of causally linked and progressively complex tasks.
- **Hierarchical and Compositional Toolset:** The tool suite will include:
  - *Deep Learning Models:* Pre-trained models for tasks like protein structure prediction [22, 18, 35], de novo molecular design [28, 11, 14, 30, 25] and property prediction [37, 36].
  - *Domain-Specific Utilities:* Deterministic computational tools for tasks like molecular fingerprinting (RDKit), sequence alignment (BLAST), and virtual high-throughput screening [33]
  - *Simulators:* Physics-based engines for molecular dynamics [1, 5, 23, 31, 13] and docking [12, 33] that produce complex, structured outputs.
  - *Information Retrieval APIs:* Interfaces to structured databases (e.g., PDB [6], UniProt [8], DrugBank [34], PubChem [17]) and knowledge sources (e.g., ArXiv, Google search).
- **Trajectory Schema:** Each data sample will be a rich JSON object capturing the full reasoning process, inspired by the ReAct framework to expose the model's chain-of-thought for interpretability and targeted learning. We provide an example shown in Appendix 4.2.
- **Data Generation Pipeline and Scale:** We project an initial dataset size of 2,000 trajectories. Our generation pipeline is a rigorous, expert-driven process:
  - *Tool Curation:* We will structure our curation around the drug discovery pipeline, breaking it into tasks and subtasks. For each subtask, we will collect and standardize tools, creating a diverse suite of over 50. Each tool will receive a formal, OpenAPI-like specification detailing its interface, description, and operational semantics to ensure variety and clarity.
  - *Task Authoring:* Domain experts will design tasks with varying complexity, each with explicit success criteria and evaluation checkpoints.
  - *Trajectory Generation:* We will employ an expert-in-the-loop framework where a human expert guides a baseline LLM agent with a ReAct-like prompt (thought - tool calling - observation) to produce a "gold" trajectory. The expert can correct suboptimal actions, refine reasoning, and ensure the trajectory follows a scientifically valid path.
  - *Automated and Manual Quality Control:* Trajectories will be validated against a rubric assessing (1) tool call validity, (2) argument correctness, (3) scientific plausibility of the reasoning steps, and (4) task success. A separate LLM-based "judge" will provide an initial quality score, with final verification performed by human experts.
- **Estimated Cost:** The initial construction (2,000 trajectories) is estimated to require approximately 2,000 expert-hours and 500 GPU-hours for agent interaction and validation, reflecting the high-fidelity nature of the dataset.

3 Impact and Potential

DrugNav's primary impact is establishing a rigorous, reproducible benchmark for long-horizon reasoning in the high-stakes domain of drug discovery. As an open-source resource, it will spur the development of advanced agent architectures and fuel progress in offline reinforcement and imitation learning from expert data. Ultimately, by accelerating the development of autonomous systems for the pharmaceutical pipeline, DrugNav will help reduce the significant time and cost of R&D and unlock a new era of AI-accelerated science.

References

[1] Mark James Abraham, Teemu Murtola, Roland Schulz, Szil'ard P'all, Jeff C. Smith, Berk Hess, and Erik Lindahl. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1-2:19–25, sep 2015. doi: 10.1016/j.softx.2015.06.001. URL https://doi.org/10.1016%2Fj.softx.2015.06.001.

[2] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, Sebastian Bodenstein, David A Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander Imani Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M R Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Vic-613 tor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630:493 – 500, 2024. URL https://api.semanticscholar.org/CorpusID:269633210.

[3] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.

[4] Reza Averly, Frazier N. Baker, Ian A. Watson, and Xia Ning. Liddia: Language-based intelligent drug discovery agent, 2025.

[5] H.J.C. Berendsen, D. van der Spoel, and R. van Drunen. GROMACS: A message-passing parallel molecular dynamics implementation. *Computer Physics Communications*, 91(1-3): 43–56, oct 1995. doi: 10.1016/0010-4655(95)00042-e. URL https://doi.org/10.1016%2F0010-4655%2895%2900042-e.

[6] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, 2000. doi: 10.1093/nar/28.1.235.

[7] He Cao, Zijing Liu, Xingyu Lu, Yuan Yao, and Yu Li. Instructmol: Multi-modal integration for building a versatile and reliable molecular assistant in drug discovery, 2024. URL https://arxiv.org/abs/2311.16208.

[8] The UniProt Consortium. UniProt: the universal protein knowledgebase in 2023. *Nucleic Acids Research*, 51(D1):D523–D531, 2023. doi: 10.1093/nar/gkac1052.

[9] Jérôme Eberhardt, Diogo Santos-Martins, Andreas F. Tillack, and Stefano Forli. Autodock vina 1.2.0: New docking methods, expanded force field, and python bindings. *Journal of chemical information and modeling*, 2021. URL https://api.semanticscholar.org/CorpusID:236092162.

[10] Bowen Gao, Yanwen Huang, Yiqiao Liu, Wenxuan Xie, Wei-Ying Ma, Ya-Qin Zhang, and Yanyan Lan. Pharmagents: Building a virtual pharma with large language model agents, 2025.

[11] Jiaqi Guan, Wesley Wei Qian, Xingang Peng, Yufeng Su, Jian Peng, and Jianzhu Ma. 3d equivariant diffusion for target-aware molecule generation and affinity prediction, 2023. URL https://arxiv.org/abs/2303.03543.

[12] Thomas A. Halgren, Robert B. Murphy, Richard A. Friesner, Hege S. Beard, Leah L. Frye, W. Thomas Pollard, and Jay L. Banks. Glide: a new approach for rapid, accurate docking and scoring. 2. enrichment factors in database screening. *Journal of medicinal chemistry*, 47 7: 1750–9, 2004. URL https://api.semanticscholar.org/CorpusID:1095757.

[13] Berk Hess, Carsten Kutzner, David van der Spoel, and Erik Lindahl. GROMACS 4: Algorithms for highly efficient, load-balanced, and scalable molecular simulation. *Journal of Chemical Theory and Computation*, 4(3):435–447, jan 2008. doi: 10.1021/ct700301q. URL https://doi.org/10.1021%2Fct700301q.

[14] Yuanyuan Jiang, Guo Zhang, Jing You, Hailin Zhang, Rui Yao, Huanzhang Xie, Liyun Zhang, Ziyi Xia, Mengzhe Dai, Yunjie Wu, Linli Li, and Shengyong Yang. Pocketflow is a data-and-knowledge-driven structure-based molecular generative model. *Nat. Mac. Intell.*, 6:326–337, 2024. URL https://api.semanticscholar.org/CorpusID:268454418.

[15] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, August 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2. URL https://doi.org/10.1038/s41586-021-03819-2.

[16] Hyomin Kim, Yunhui Jang, and Sungsoo Ahn. Mt-mol:multi agent system with tool-based reasoning for molecular optimization, 2025.

[17] S. Kim, J. Chen, T. Cheng, A. Gindulyte, J. He, S. He, Q. Li, B. A. Shoemaker, P. A. Thiessen, B. Yu, L. Zaslavsky, J. Zhang, and E. E. Bolton. Pubchem in 2021: new data content and improved web interfaces. *Nucleic Acids Research*, 49(D1):D1388–D1395, 2021. doi: 10.1093/nar/gkaa971.

[18] Rohith Krishna, Jue Wang, Woody Ahern, Pascal Sturmfels, Preetham Venkatesh, Indrek Kalvet, Gyu Rie Lee, Felix S Morey-Burrows, Ivan V. Anishchenko, Ian R. Humphreys, Ryan McHugh, Dionne K. Vafeados, Xinting Li, George A. Sutherland, Andrew Hitchcock, C. Neil Hunter, Alex Kang, Evans Brackenbrough, Asim K. Bera, Minkyung Baek, Frank DiMaio, and David Baker. Generalized biomolecular modeling and design with rosettafold all-atom. *bioRxiv*, 2023. URL https://api.semanticscholar.org/CorpusID:264039660.

[19] Greg Landrum et al. Rdkit: Open-source cheminformatics. https://www.rdkit.org, 2006–. Version 2025.

[20] Namkyeong Lee, Edward De Brouwer, Ehsan Hajiramezanali, Tommaso Biancalani, Chanyoung Park, and Gabriele Scalia. Rag-enhanced collaborative llm agents for drug discovery, 2025. URL https://arxiv.org/abs/2502.17506.

[21] Kun Li, Zhennan Wu, Shoupeng Wang, Jia Wu, Shirui Pan, and Wenbin Hu. Drugpilot: Llm-based parameterized reasoning agent for drug discovery, 2025.

[22] Zeming Lin, Halil Akin, Roshan Rao, Brian L. Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic level protein structure with a language model. *bioRxiv*, 2022. URL https://api.semanticscholar.org/CorpusID:253259177.

[23] Erik Lindahl, Berk Hess, and David van der Spoel. GROMACS 3.0: a package for molecular simulation and trajectory analysis. *Journal of Molecular Modeling*, 7(8):306–317, jan 2001. doi: 10.1007/s008940100045. URL https://doi.org/10.1007%2Fs008940100045.

[24] Sizhe Liu, Yizhou Lu, Siyu Chen, Xiyang Hu, Jieyu Zhao, Tianfan Fu, and Yue Zhao. Drugagent: Automating ai-aided drug discovery programming through llm multi-agent collaboration, 2024.

[25] Hannes H. Loeffler, Jiazhen He, Alessandro Tibo, Jon Paul Janet, Alexey Voronov, Lewis H. Mervin, and Ola Engkvist. Reinvent 4: Modern ai–driven generative molecule design. *Journal of Cheminformatics*, 16, 2024. URL https://api.semanticscholar.org/CorpusID:267778326.

[26] Janghoon Ock, Radheesh Sharma Meda, Srivathsan Badrinarayanan, Neha S. Aluru, Achuth Chandrasekhar, and Amir Barati Farimani. Large language model agent for modular task execution in drug discovery, 2025. URL https://arxiv.org/abs/2507.02925.

[27] Qihua Pan, Dong Xu, Jenna Xinyi Yao, Lijia Ma, Zexuan Zhu, and Junkai Ji. Frogent: An end-to-end full-process drug design agent, 2025. URL https://arxiv.org/abs/2508.10760.

[28] Xingang Peng, Shitong Luo, Jiaqi Guan, Qi Xie, Jian Peng, and Jianzhu Ma. Pocket2mol: Efficient molecular sampling based on 3d protein pockets, 2025. URL https://arxiv.org/abs/2205.07249.

[29] Michael Retchin, Yuanqing Wang, K. Takaba, and J. Chodera. Druggym: A testbed for the economics of autonomous drug discovery, 2024.

[30] Arne Schneuing, Charles Harris, Yuanqi Du, Kieran Didi, Arian Jamasb, Ilia Igashov, Weitao Du, Carla Gomes, Tom Blundell, Pietro Lio, Max Welling, Michael Bronstein, and Bruno Correia. Structure-based drug design with equivariant diffusion models, 2024. URL https://arxiv.org/abs/2210.13695.

[31] David Van Der Spoel, Erik Lindahl, Berk Hess, Gerrit Groenhof, Alan E. Mark, and Herman J. C. Berendsen. GROMACS: Fast, flexible, and free. *Journal of Computational Chemistry*, 26(16): 1701–1718, 2005. doi: 10.1002/jcc.20291. URL https://doi.org/10.1002%2Fjcc.20291.

[32] Jinyuan Sun, Auston Li, Yifan Deng, and Jiabo Li. ChatMol copilot: An agent for molecular modeling and computation powered by LLMs. In Carl Edwards, Qingyun Wang, Manling Li, Lawrence Zhao, Tom Hope, and Heng Ji (eds.), *Proceedings of the 1st Workshop on Language + Molecules (L+M 2024)*, pp. 55–65, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.langmol-1.7. URL https://aclanthology.org/2024.langmol-1.7/.

[33] Oleg Trott and Arthur J. Olson. Autodock vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of Computational Chemistry*, 31, 2009. URL https://api.semanticscholar.org/CorpusID:30245244.

[34] D. S. Wishart, Y. D. Feunang, A. C. Guo, E. J. Lo, A. Marcu, J. R. Grant, T. Sajed, D. Johnson, C. Li, Z. Saye-M, N. Assempour, I. Iynkkaran, Y. Liu, A. Maciejewski, N. Gale, A. Wilson, L. Chin, R. Cummings, D. Le, A. Pon, C. Knox, and M. Wilson. Drugbank 5.0: a major update to the drugbank database for 2018. *Nucleic Acids Research*, 46(D1):D1074–D1082, 2018. doi: 10.1093/nar/gkx1037.

[35] Ruidong Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, Bonnie Berger, et al. High-resolution de novo structure prediction from primary sequence. *BioRxiv*, pp. 2022–07, 2022.

[36] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59 (8):3370–3388, 2019.

[37] Gengmo Zhou, Zhifeng Gao, Qiankun Ding, Hang Zheng, Hongteng Xu, Zhewei Wei, Linfeng Zhang, and Guolin Ke. Uni-mol: A universal 3d molecular representation learning framework. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6K2RM6wVqKu.

4 Appendix

4.1 Unified Tool Specification

To ensure that the LLM agent can reliably and consistently understand and use the diverse set of tools, we adopt a unified tool specification format that fits with MCP. Each tool is described by a machine-readable schema that details its functionality, parameters, and expected inputs.

The core of this specification is a JSON Schema definition for the tool's input parameters. This provides strong typing, constraints (such as required fields or enumerated values), and rich natural language descriptions that are crucial for the LLM's reasoning process.

For instance, the specification for the `pdb_search` tool, which queries the Protein Data Bank, is defined as follows:

```
{
  "name": "pdb_search",
  "description": "Searches the Protein Data Bank (PDB) for protein structures.
  ↪  Can be queried using keywords like protein name, gene name, or organism.
  ↪  Returns a list of matching PDB entries with their metadata.",
  "parameters": {
    "type": "object",
    "required": ["query"],
    "properties": {
      "query": {
        "type": "string",
        "description": "The search keyword, such as a protein name ('EGFR kinase
        ↪  domain'), gene name ('BRAF'), or organism ('human')."
      },
      "limit": {
        "type": "integer",
        "default": 5,
        "description": "The maximum number of search results to return. Defaults
        ↪  to 5."
      },
      "method": {
        "type": "string",
        "enum": [
          "X-RAY DIFFRACTION",
          "NMR",
          "ELECTRON MICROSCOPY"
        ],
        "description": "Filter results by the experimental method used to
        ↪  determine the structure."
      }
    }
  }
}
```

This structured format provides several key advantages: (1) it reduces hallucinated or malformed tool calls by enforcing strict parameter validation; (2) it enables the agent to self-correct by inspecting the schema if a call fails; and (3) it makes the entire tool suite more transparent and easier to extend.

4.2 Trajectory Schema

The Trajectory Schema is designed to capture complex scenarios, including error handling, which is crucial for training robust agents. The observation field can contain either successful results or structured error messages from the tools. The agent's subsequent thought process must then demonstrate its ability to diagnose and recover from the error. Below is an example of a two-turn sequence where the agent makes a typo, receives an error, and corrects it.

```
{
  "task_id": "TD-015",
  "query": "Find the PDB structure for the human protein BRAF.",
  "trajectory": [
    {
      "turn": 1,
```

```
      "thought": "I need to find the structure of the human BRAF protein. I will
      ↪  use the pdb_search tool to query the Protein Data Bank.",
      "action": "pdb_search(query='human B-RAF')",
      // Note: The correct identifier is typically 'BRAF', not 'B-RAF'. This may
      ↪  cause an error.
      "observation": "{'status': 'error', 'message': 'Query \"human B-RAF\"
      ↪  returned no results. No exact matches found.'}"
    },
    {
      "turn": 2,
      "thought": "The previous search for 'human B-RAF' failed. It's possible I
      ↪  used an incorrect alias or formatting. The standard gene name is BRAF.
      ↪  I will try searching again with the correct identifier 'BRAF' to see if
      ↪  I can get a result.",
      // The agent correctly diagnoses the likely error (incorrect name) and
      ↪  plans a corrective action.
      "action": "pdb_search(query='human BRAF kinase domain')",
      "observation": "{'status': 'success', 'results': [{'pdb_id': '1UWJ',
      ↪  'resolution': '2.65 A', 'method': 'X-RAY DIFFRACTION'}]}"
    }
    // ... subsequent turns of thought, action, observation
  ]
}
```

4.3 Evaluation Framework

To ensure rigorous and multi-faceted evaluation of LLM agents, we propose a comprehensive evaluation framework that will be released alongside the DrugNav dataset. This framework includes standardized metrics and a set of baseline models.

Agent performance will be assessed using a suite of primary and secondary metrics:

Primary Metric

- **Task Success Rate:** A binary metric indicating whether the agent-generated trajectory achieves the final objective defined in the task's `success_metrics` (e.g., `found_lead: true`, `pIC50_met: true`). This is the ultimate measure of an agent's capability.

Secondary Metrics

Efficiency and Cost

- **Path Length:** The total number of turns (thought-action-observation steps) required to complete the task. Shorter paths are generally more efficient.
- **Tool Call Count:** The total number of times the agent invokes any tool. This serves as a proxy for both computational cost and API usage.
- **Computational Cost:** For tasks involving simulators (e.g., molecular docking, MD simulations), we will record the estimated GPU/CPU hours, providing a real-world cost metric.

Quality and Robustness

- **Trajectory Quality Score:** A score from 1-5 provided by the final LLM-based "judge" and verified by a human expert, assessing the scientific plausibility and elegance of the agent's reasoning path.
- **Error Handling Rate:** The percentage of instances where an agent successfully recovers from a tool-generated error (e.g., failed simulation, invalid input, API timeout) and proceeds towards a valid solution.