The Darwin–Gödel Discovery Machine: Toward Bounded-Risk Self-Improving AI4Science

Xuening Wu[1], Yanlan Kang[1], Xinhang Zhang[1], Qianya Xu[2], Honggang Wang[3], Zeping Chen[3]

[1]Fudan University, Shanghai, China
[2]University of California San Diego, La Jolla, USA
[3]Tongji University, Shanghai, China

Abstract

We present the Darwin–Gödel Discovery Machine (DGDM), a dual-loop system for risk-aware self-improving AI4Science. The inner Darwinian loop evolves candidate solutions—demonstrated here with molecular ligands—via reinforcement learning–guided variation, fitness evaluation, and constraint-based retention, ensuring chemical validity and incremental improvement. Surrounding this process, an outer Gödelian loop adapts elements of the discovery pipeline itself, using confidence-based acceptance to regulate potentially harmful modifications. In a proof-of-concept molecular docking study on four seed ligands, DGDM improves median binding affinity from -4.457 to -5.422 kcal/mol while maintaining 100% chemical validity. These results illustrate how bounded-risk inner-loop evolution can yield scientifically meaningful gains, while highlighting the role of risk-aware acceptance in stabilizing self-directed discovery pipelines. Although preliminary in scope, this work demonstrates the feasibility of dual-loop architectures for AI-driven scientific discovery and motivates future extensions toward more robust and trustworthy self-improving AI4Science systems. A reproducibility package will be released upon publication.

1 Introduction

Artificial intelligence (AI) has begun to transform scientific discovery, from protein folding [1] to climate modeling [2]. Yet a fundamental challenge remains: how can we design AI systems that not only advance individual tasks but also continuously improve the pipelines that integrate them into end-to-end scientific discovery? The notion of a Gödel Machine [3] provides a theoretical "yes," as it guarantees improvement whenever a provably better modification is found—but such proofs are rarely feasible in practice. By contrast, most existing systems function as fixed pipelines: once trained, they are applied in a static manner without the capacity for self-improvement. At the other extreme, unconstrained self-modification can lead to unreliable trajectories of improvement, including performance degradation, systematic errors, and invalid outputs. The central challenge, therefore, is to develop AI frameworks capable of self-improvement at both the task and pipeline levels, while ensuring progress under bounded risk.

Recent advances illustrate both the promise and the limits of current approaches. In molecular discovery, generative models for de novo design [4, 5], reinforcement learning for synthesis and property optimization [6, 7], and autonomous laboratory platforms for iterative experimentation [8–11] have shown the potential of self-directed AI. These "self-driving labs" close the loop between hypothesis generation and wet-lab validation, offering a physical realization of self-improving pipelines. At the protein level, breakthroughs in structure prediction [1, 12] and structure-conditioned generators such as ProteinMPNN and Chroma [13, 14], together with language-model–based predictors (e.g., ESM-2) and diffusion-based docking approaches such as DiffDock [15, 16], have expanded the design

space and improved accuracy. Meanwhile, classical docking workflows—including AutoDock Vina, Vinardo, RDKit, and OpenBabel—remain widely adopted [17–20]. Beyond chemistry, coding-agent frameworks such as the Darwin Gödel Machine [21] illustrate the potential of self-improving agents, but direct transfer to molecular discovery is challenging due to noisy, continuous chemical spaces with strict validity and safety requirements. Overall, current systems demonstrate creativity but still operate within largely fixed pipelines and often lack explicit safeguards against invalid or risky outcomes.

To address this gap, we propose the **Darwin–Gödel Discovery Machine (DGDM)**, a dual-loop system for risk-aware self-improvement in AI4Science. DGDM combines two complementary design principles: a Darwinian inner loop, which evolves candidate solutions—demonstrated here with molecular ligands—via reinforcement-learning–guided variation and selection to ensure validity and incremental improvement; and a Gödelian outer loop, which governs adaptations to the discovery pipeline itself through confidence-based acceptance of proposed modifications. Together, these loops couple creative exploration with explicit risk awareness. While our proof-of-concept focuses on drug discovery, the design is intended to be illustrative of broader AI-driven scientific workflows.

Contributions. This work makes the following contributions:

1. We introduce a **dual-loop system design** for risk-aware self-improvement in AI4Science, illustrating how task-level optimization and pipeline-level adaptation can be coupled.
2. We present a **proof-of-concept molecular docking study** showing that DGDM improves median binding affinity across four seed ligands while preserving 100% chemical validity.
3. We discuss a **forward-looking design direction** for incorporating principled risk-aware acceptance mechanisms into self-improving scientific pipelines.

Taken together, these contributions position DGDM as a *work-in-progress* system. In this paper, we validate the Darwinian inner loop in a molecular docking setting, while the Gödelian outer loop is presented at the level of system design and risk-aware acceptance principles. A full formal treatment and extensive empirical validation of outer-loop self-modification are left for future work.

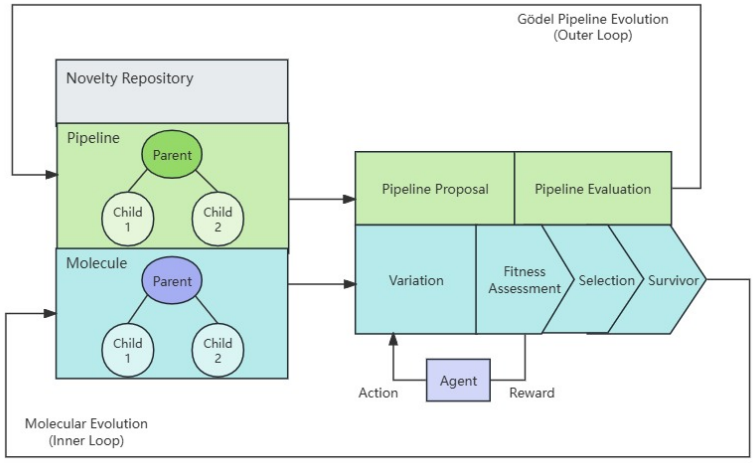

Figure 1: Conceptual schematic of the Darwin–Gödel Discovery Machine (DGDM). The dual-loop design couples inner-loop solution evolution with outer-loop pipeline adaptation. Molecules are generated, modified, and optimized under reinforcement-learning–based fitness assessment, while the pipeline itself can adapt through proposed modifications to models, scoring functions, or search strategies.

2 Related Work

Protein structure prediction. Breakthroughs in protein structure prediction have significantly advanced drug discovery. AlphaFold achieves near-atomic accuracy across diverse proteins, while single-sequence models such as ESMFold leverage protein language models for high-throughput

prediction [1, 12]. These predictors substantially expand structural availability, providing upstream inputs that downstream design systems—including ours—can readily integrate.

Generative modeling and docking workflows. Generative AI has broadened both molecular and protein design spaces. For docking, diffusion-based models such as DiffDock frame pose prediction as a generative sampling problem, improving accuracy and enrichment [15, 16]. At the protein level, transformer-based language models (e.g., ESM-2) and structure-conditioned generators such as ProteinMPNN and Chroma enable sequence and structure design [13, 14]. In small-molecule discovery, reinforcement learning methods such as REINVENT [6] and graph-based approaches including the Graph Convolutional Policy Network (GCPN) [22] have been used to optimize chemical properties and synthesizability. Meanwhile, classical docking components—AutoDock Vina, Vinardo scoring, RDKit, and OpenBabel—remain widely adopted in practice [17–20]. Rather than proposing a new generator, our approach integrates with existing generative and docking pipelines, introducing a meta-level control mechanism to regulate and enhance their behavior.

Self-improvement paradigms and cross-domain inspirations. Beyond individual generators, prior work has explored systems that adapt and optimize the discovery process as a whole. Gödel machines formalize agents that rewrite themselves once a proof guarantees higher expected utility [3]. While such proof-based guarantees are infeasible in scientific discovery pipelines, the underlying idea motivates practical relaxations: proposing modifications, evaluating them empirically, and retaining only validated improvements. Zhang et al. [21] introduced a Darwinian Gödel Machine for software agents, illustrating how generative backbones and constraint-based filtering can support adaptive self-evolution. However, direct transfer to molecular discovery is challenging, as chemical design operates in noisy and continuous spaces with delayed feedback and strict validity requirements. Our work draws inspiration from these ideas while adapting them to the constraints and uncertainties of AI-driven scientific discovery.

3 Method

The **Darwin–Gödel Discovery Machine (DGDM)** is organized into two nested optimization loops (Figure 1), enabling self-improvement at both the molecular and pipeline levels. The inner Darwinian loop refines molecules via reinforcement learning–guided evolution, while the outer Gödelian loop adaptively reconfigures the discovery pipeline under risk-aware statistical safeguards.

3.1 Inner Loop: Reinforcement-Learning–Guided Molecular Evolution

The inner loop follows a Darwinian cycle with four stages: (1) *variation*, (2) *fitness assessment*, (3) *selection*, and (4) *constraint-based retention*. In our conceptual design, reinforcement learning (RL) biases this process: docking scores and constraint outcomes provide reward signals that guide exploration.

Variation. Molecular diversity is introduced via perturbations generated by diffusion models, graph-based generators, or language-model–based chemistry models. RL agents parameterize these operators, learning which transformations are most productive.

Fitness assessment. Modified ligands are docked against the target receptor. Scoring functions (e.g., Vinardo in AutoDock Vina) provide approximate binding free energies, while AlphaFold [1] or ESMFold [12] can supply receptor structures when needed. Docking energies serve as quantitative rewards for RL.

Selection. High-affinity candidates are preferentially retained, maintaining evolutionary pressure toward stronger binding while preserving structural diversity.

Constraint filtering. Survivors must satisfy chemical validity and drug-likeness checks (e.g., Lipinski's rules, synthetic accessibility, toxicity alerts). Failures provide negative reinforcement, discouraging unproductive modification strategies.

This RL-augmented Darwinian cycle balances stochastic exploration (via generative perturbations) with directed exploitation (via docking and constraints), producing progressively higher-quality ligands across generations.

3.2 Outer Loop: Gödelian Pipeline Self-Adaptation

The outer loop adapts the *pipeline configuration*—the sequence and parameters of operators controlling molecular search. Inspired by the Gödel Machine [3], it introduces meta-level self-modification, but replaces infeasible proof-based guarantees with tractable risk-aware statistical safeguards.

Proposal generation. Candidate modifications are generated, e.g., by large language models (LLMs) augmented with retrieval-augmented generation (RAG). Examples include inserting refinement steps or altering filtering thresholds.

Risk-aware acceptance test. Define the paired improvement (gain) for replicate i as

$$Y_i := R_{0,i} - R_{1,i} \quad \text{(larger is better)},$$

so that negative values correspond to degradation. Let

$$\hat{\mu} = \tfrac{1}{n} \sum_{i=1}^{n} Y_i$$

denote the empirical mean improvement across n paired runs. Assuming each $Y_i \in [a, b]$ (enforced by clipping), we evaluate proposed pipeline modifications using paired experimental runs and compute $\hat{\mu}$.

A modification is accepted only if a conservative lower-confidence estimate remains non-negative:

$$\hat{\mu} - \mathrm{margin}(n, \delta) \geq 0,$$

where $\mathrm{margin}(n, \delta)$ increases as the number of paired runs decreases and as the risk tolerance δ becomes smaller. This criterion provides a practical safeguard against adopting pipeline changes that appear beneficial due to noise or limited sampling.

In practice, we use paired t-tests as an exploratory check of statistical significance, while the risk-aware acceptance rule offers a simple and interpretable mechanism for regulating self-modification.

3.3 Loop Interaction

The inner loop evolves ligands under a fixed pipeline until convergence or stagnation is observed,

$$\Delta_t < \epsilon \quad \text{for } K \text{ generations},$$

or until a maximum computational budget $T_{\max}$ is reached. The resulting outcomes are aggregated and passed to the outer loop, which then determines whether to accept a proposed pipeline modification. A single outer-loop update typically relies on multiple inner-loop cycles, grounding meta-level decisions in empirically stable evidence.

Note that Δ denotes the median docking score difference per ligand within an inner-loop cycle, whereas $\hat{\mu}$ in the outer loop refers to the mean improvement computed across paired pipeline runs.

3.4 Evaluation Setup

To ensure reproducibility, each ligand's binding score was reported as the median across three docking poses. Baseline (R0) and modified (R1) pipelines were run under identical frozen settings. Metrics include:

- **Binding affinity:** docking energies (kcal/mol) from Vinardo.
- **Score improvement:** $\Delta = \text{R1 median} - \text{R0 median}$ (negative Δ indicates improvement).
- **Pass rate:** proportion of ligands satisfying chemical validity and drug-likeness constraints.
- **Trajectory analysis:** qualitative tracing of molecular modifications leading to observed improvements.

Note: We report $\Delta = \text{R1} - \text{R0}$ (negative Δ indicates improvement), while the risk-aware acceptance rule operates on the paired gain $Y := \text{R0} - \text{R1} = -\Delta$ so that larger values correspond to improvement.

All runs used fixed seeds and parameter settings; full environment manifests and scripts are provided as part of the reproducibility package. While evaluated here in a molecular docking setting, DGDM is presented as an illustrative example of how risk-aware self-improvement can be integrated into AI-driven scientific workflows.

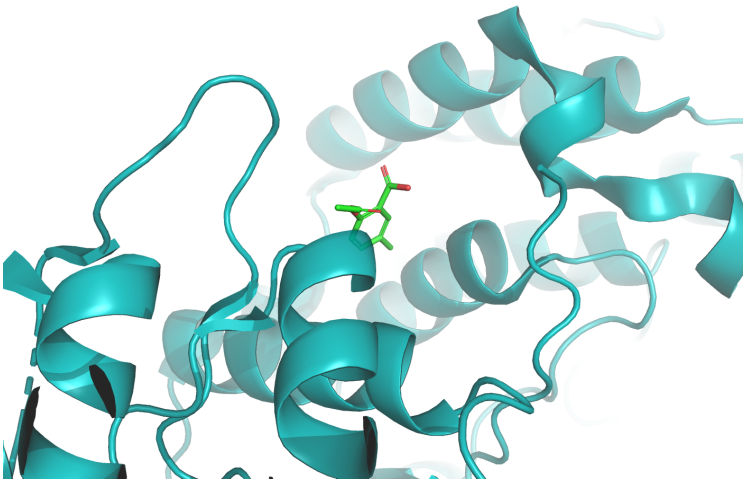

Figure 2: Docked pose of DGDM-optimized Aspirin_mut2 (green) in the target pocket (teal).

4 Experiments

We conducted a proof-of-concept (PoC) study to examine whether DGDM can improve docking-based binding affinity predictions under conservative constraints. Four seed ligands (Aspirin, LIG3, LIG4, and Pyridine) were selected to span diverse scaffolds and pharmacophores. Each ligand was evaluated in two stages: a baseline run (R0) and an optimized run (R1), in which DGDM-generated variants were filtered and re-docked under identical conditions.

4.1 Setup

Docking was performed using `AutoDock Vina` with the Vinardo scoring function, with exhaustiveness fixed at 12. For each ligand, three poses were sampled and the median score reported. Constraint filters (Lipinski rules, synthetic accessibility, and toxicity/reactivity alerts) enforced chemical validity; invalid candidates were discarded and penalized. Docking scores are interpreted as *relative indicators* of binding propensity, consistent with prior work.

4.2 Metrics

We report four metrics: (1) median docking affinity (kcal/mol); (2) improvement Δ (R1–R0, with negative values indicating stronger binding); (3) chemical validity pass rate; and (4) qualitative trajectory analysis of modifications contributing to observed changes.

4.3 Results

Across all evaluated ligands, DGDM improved median docking affinity while preserving 100% chemical validity (Table 1). Observed improvements ranged from -0.8 to -1.5 kcal/mol, with Aspirin showing the largest change ($\Delta = -1.27$ kcal/mol). These results indicate that the inner-loop optimization process can identify chemically valid modifications associated with stronger docking scores under the tested conditions.

4.4 Limitations of the Proof-of-Concept

This study is limited in scope. Evaluation was restricted to a small ligand panel and a single protein target, and docking scores serve only as approximate surrogates for binding affinity. Moreover, the present experiments validate only the *inner loop* of DGDM; empirical evaluation of the *outer loop*, which governs pipeline-level adaptation, is left for future work. Accordingly, the reported

Table 1: Proof-of-concept docking outcomes. Negative Δ indicates improvement.

Ligand	R0 Median	R1 Median	Δ (R1–R0)	Pass Rate (%)
Aspirin	-4.752	-6.022	-1.270	100
LIG3	-4.374	-5.181	-0.807	100
LIG4	-4.541	-6.020	-1.479	100
Pyridine	-3.428	-4.260	-0.832	100
Median	-4.457	-5.422	-0.965	**100**

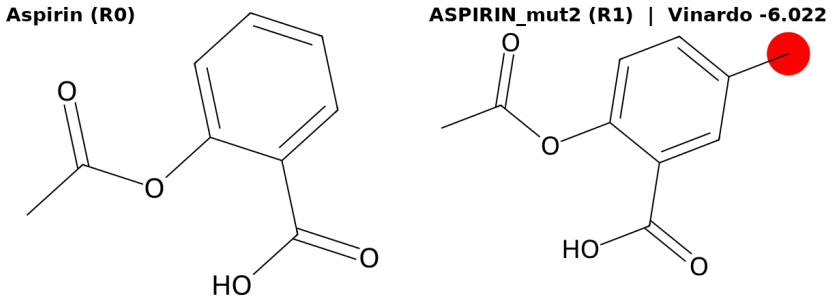

Figure 3: Structural comparison of baseline Aspirin (R0) and optimized variant Aspirin_mut2 (R1). Red highlights the DGDM-induced modification.

results should be interpreted as preliminary and intended to motivate larger-scale benchmarking and experimental validation.

4.5 Qualitative Insights

Qualitative trajectory analysis revealed chemically interpretable patterns. Optimized variants frequently introduced or repositioned hydrogen-bond donors or acceptors while reducing steric clashes. The top candidate, Aspirin_mut2 (Figures 2 and 3), achieved a Vinardo score of -6.022 kcal/mol through a polar substitution that improved complementarity within the binding pocket. Under the evaluated conditions, no ligand exhibited a degradation relative to its baseline, reflecting the effect of constraint-based survivor filtering in this proof-of-concept setting.

5 Conclusion

We presented DGDM, a Darwin–Gödel–inspired dual-loop system for molecular design that evolves candidate structures while enabling adaptive refinement of the optimization process. In a proof-of-concept study, DGDM improved docking-based binding affinity for the evaluated ligands while preserving chemical validity, demonstrating the feasibility of bounded-risk generative modification in this setting.

Looking ahead, scaling DGDM will require integration with more accurate evaluation pipelines, including rescoring, molecular dynamics, and ultimately wet-lab assays to provide richer empirical feedback. Responsible deployment in drug discovery will also necessitate transparent benchmarking, auditable operators, and appropriate governance, reflecting the high standards of safety and reproducibility in this domain. While our experiments focus on molecular design, the ideas explored here illustrate how risk-aware self-improvement mechanisms may be incorporated into broader AI-for-Science workflows.

References

[1] John Jumper, Richard Evans, Alexander Pritzel, and et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596:583–589, 2021.

[2] E. Bevacqua, Laura Suarez-Gutierrez, Aglaé Jézéquel, F. Lehner, M. Vrac, P. Yiou, and J. Zscheischler. Advancing research on compound weather and climate events via large ensemble model simulations. *Nature Communications*, 14, 2023. doi: 10.1038/s41467-023-37847-5.

[3] Jürgen Schmidhuber. Gödel machines: Fully self-referential optimal universal self-improvers. In *Artificial general intelligence*, pages 199–226. Springer, 2007.

[4] Marwin HS Segler, Mike Preuss, and Mark P Waller. Planning chemical syntheses with deep neural networks and symbolic ai. *Nature*, 555(7698):604–610, 2018.

[5] Benjamin Sánchez-Lengeling, Emily Reif, Andrew Pearce, et al. A gentle introduction to inverse molecular design: Machine learning for molecular discovery. *ACS Central Science*, 7 (8):1159–1173, 2021.

[6] Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. *Science Advances*, 4(7):eaap7885, 2018.

[7] Zhenpeng Zhou, Steven Kearnes, Li Li, Richard N Zare, and Patrick Riley. Optimization of molecules via deep reinforcement learning. *Scientific Reports*, 9:10752, 2019.

[8] Aldo Enrique Gongora, Bin Xu, William S Perry, Chukwudi Okoye, and Rigoberto Hernandez. Automatic discovery of materials and catalysts by quantum mechanics and machine learning. *Nature Reviews Materials*, 5(7):531–550, 2020.

[9] Ben Burger, Phillip M Maffettone, Vladimir V Gusev, Christopher M Aitchison, Yu Bai, Xiao Wang, Xiaobo Li, Ben M Alston, Bingqing Li, Robert Clowes, Neil Rankin, Jonathan B Harris, R James Sprick, and Andrew I Cooper. A mobile robotic chemist. *Nature*, 583(7815):237–241, 2020.

[10] Connor W Coley, Nathaniel S Eyke, and Klavs F Jensen. Autonomous discovery in the chemical sciences part ii: Outlook. *Angewandte Chemie International Edition*, 59(52):23414–23436, 2020.

[11] Benjamin P MacLeod, Fiona G L Parlane, T R Morrissey, Frédéric Hase, Luke M Roch, Kevin E Dettelbach, Rafael Moreira, Luke P E Yunker, Michael B Rooney, Joel R Deeth, Vivian Lai, Grace J Ng, Heather Situ, Rong Zhang, Mark S Elliott, Brandon Haley, Leroy Cronin, Jason E Hein, Alán Aspuru-Guzik, and Curtis P Berlinguette. Self-driving laboratory for accelerated discovery of thin-film materials. *Science Advances*, 6(20):eaaz8867, 2020.

[12] Zeming Lin, Heewook Akin, Roshan Rao, and et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2023. doi: 10.1101/2022.07.20.500902.

[13] Joël Dauparas, Ivan Anishchenko, and David Baker. Robust deep learning–based protein design using structure-informed graph representations. *Science*, 378(6615):49–56, 2022.

[14] John Ingraham et al. Chroma: Generative modeling of protein–ligand complexes. *Nature*, 2023. doi: 10.1038/s41586-023-06728-8.

[15] Giacomo Corso, Hendrik Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. *International Conference on Learning Representations (ICLR)*, 2023.

[16] Jack Qiu, Boyuan Pang, et al. Pocket-conditioned diffusion model for protein-ligand complex generation. *bioRxiv*, 2023. doi: 10.1101/2023.11.22.568238.

[17] Jerome 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*, 61(8):3891–3898, 2021.

[18] Rodrigo Quiroga and Marcos A Villarreal. Vinardo: A scoring function based on autodock vina improves scoring, docking, and virtual screening. *PloS one*, 11(5):e0155183, 2016.

[19] RDKit: Open-source cheminformatics. `https://www.rdkit.org`. accessed 2025/08/14.

[20] Noel M O'Boyle, Michael Banck, Craig A James, Chris Morley, Tim Vandermeersch, and Geoffrey R Hutchison. Open babel: An open chemical toolbox. *Journal of cheminformatics*, 3 (1):33, 2011.

[21] Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin godel machine: Open-ended evolution of self-improving agents. *arXiv preprint arXiv:2505.22954*, 2025.

[22] Jiaxuan You, Bowen Liu, Rex Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In *Advances in Neural Information Processing Systems*, 2018. URL `https://arxiv.org/abs/1806.02473`.

Appendix 1 Algorithmic Details

Algorithm 1 Outer Loop: Pipeline Self-Adaptation (Conceptual)

Require: Initial configuration θ_0, generator G, harness $\mathcal{H}$, max rounds T
Ensure: Final configuration $\theta^\star$, registry $\mathcal{R}$

$\theta \leftarrow \theta_0,\ \mathcal{R} \leftarrow \{(\theta_0, \texttt{baseline})\}$
for $t = 1$ to T **do**
  $(\mathcal{C}, \mathbf{m}) \leftarrow \textsc{RunInnerLoop}(\theta, \mathcal{H})$
  if $\textsc{Stagnant}(\mathbf{m})$ **then**
    $\theta' \leftarrow \textsc{ProposeEdit}(G, \theta, \mathcal{R})$
    $\Delta \leftarrow \textsc{EvaluatePair}(\mathcal{H}, \theta, \theta', \mathcal{C})$
    if $\textsc{SufficientImprovement}(\Delta)$ **then**
      $\theta \leftarrow \theta',\ \mathcal{R} \leftarrow \mathcal{R} \cup \{(\theta', \texttt{accepted})\}$
    else
      $\mathcal{R} \leftarrow \mathcal{R} \cup \{(\theta', \texttt{rejected})\}$
    end if
  end if
end for
return $\theta^\star \leftarrow \theta,\ \mathcal{R}$

Algorithm 2 Inner Loop: Ligand Evolution (Illustrative)

Require: Current configuration θ, harness $\mathcal{H}$, population size M
Ensure: Candidate batch $\mathcal{C}$, survivors $\mathcal{S}$

$\mathcal{C} \leftarrow \emptyset,\ \mathcal{S} \leftarrow \emptyset$
for $i = 1$ to M **do**
    $x \leftarrow \textsc{SampleLigand}(\theta)$
    $x' \leftarrow \textsc{ModifyLigand}(x)$
    $s \leftarrow \textsc{Evaluate}(x', \mathcal{H})$
    $\mathcal{C} \leftarrow \mathcal{C} \cup (x', s)$
    if $\textsc{Survives}(s)$ **then**
        $\mathcal{S} \leftarrow \mathcal{S} \cup x'$
    end if
end for
return $(\mathcal{C}, \mathcal{S})$