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

Anonymous Author(s)
Affiliation
Address
email

Abstract

We present the Darwin–Gödel Discovery Machine (DGDM), a dual-loop framework for bounded-risk self-improving AI4Science. The inner Darwinian loop evolves candidate solutions—demonstrated here with molecular ligands—through reinforcement learning (RL) guided variation, fitness evaluation, and constraint-based retention, ensuring validity and incremental improvement. The outer Gödelian loop adapts the discovery pipeline itself, governed by statistical safeguards (PAC–style) that limit harmful modifications. In a proof-of-concept docking study on four seed ligands, DGDM improved median binding affinity from -4.457 to -5.422 kcal/mol while preserving 100% chemical validity. These results illustrate how bounded-risk inner-loop evolution can yield scientifically meaningful advances, while motivating future extensions of the outer loop for trustworthy pipeline optimization. Although preliminary in scope, this work highlights the potential of dual-loop architectures to push the boundaries of AI in scientific discovery while explicitly accounting for risk. Looking ahead, RL strategies and large language models with domain-grounded retrieval offer natural mechanisms to enrich inner-loop adaptation and outer-loop self-improvement, advancing the vision of trustworthy, self-improving AI4Science. An anonymized reproducibility package will be released to facilitate community feedback.

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

*Submitted to AI for Science Workshop (NeurIPS 2025)

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 two-level dual-loop framework for bounded-risk self-improvement in AI4Science. The DGDM integrates two complementary principles: a Darwinian inner loop, which evolves candidate solutions—demonstrated here with molecular ligands—via reinforcement-learning–guided variation and selection, ensuring validity and incremental improvement; and a Gödelian outer loop, which adapts the discovery pipeline itself, but only under statistical safeguards (PAC-style) that limit harmful modifications. Together, these loops couple the creativity of evolutionary search with the rigor of principled safeguards. While our proof-of-concept experiments focus on drug discovery, the framework is designed for general application across scientific domains.

Contributions. Our contributions are threefold:

1. We introduce a **bounded-risk dual-loop self-improving AI framework** designed for AI4Science, with initial experiments in drug discovery to demonstrate feasibility.
2. We provide a **proof-of-concept docking study** showing that DGDM improved median binding affinity across four seed ligands while preserving 100% chemical validity.
3. We outline a **roadmap** for extending the Gödelian outer loop with PAC-style acceptance rules, reinforcement learning, and language model–driven hypothesis generation.

Taken together, these contributions suggest that bounded-risk dual-loop architectures can produce scientifically meaningful improvements while advancing more trustworthy forms of self-improving AI. This paper presents a *work-in-progress* framework: the Darwinian inner loop is validated in a molecular docking setting, while the Gödelian outer loop is developed at the level of mathematical formulation and PAC-style derivation, with experimental validation left for future work.

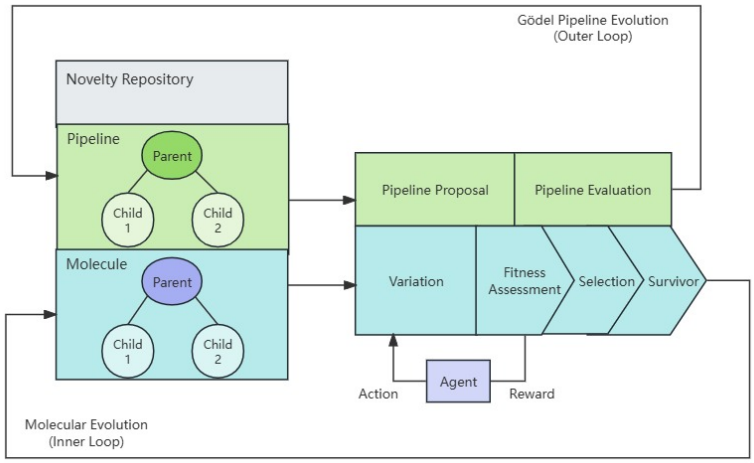

Figure 1: Conceptual schematic of the Darwin–Gödel Discovery Machine (DGDM). The dual-loop design couples inner-loop solution evolution with outer-loop pipeline adaptation. Molecules are generated, modified, and optimized under reinforcement-learning–based fitness assessment, while the pipeline itself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

2 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 PAC–style statistical safeguards.

2.1 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.

2.2 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 PAC–style 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.

PAC 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), Hoeffding's inequality gives

$$\Pr\left(\mu < \hat{\mu} - (b-a)\sqrt{\tfrac{1}{2n} \ln \tfrac{2}{\delta}} \right) \leq \delta,$$

where $\mu = \mathbb{E}[Y_i]$ is the true mean gain. We therefore accept the pipeline modification only if

$$\hat{\mu} - (b-a)\sqrt{\tfrac{1}{2n} \ln \tfrac{2}{\delta}} \;\geq\; 0,$$

which guarantees (with probability at least $1 - \delta$) that the modification does not reduce performance on average.

In practice, we use paired t-tests as an exploratory check of statistical significance, while the PAC-style acceptance rule serves as the formal safeguard that bounds the risk of harmful changes. *A full derivation is provided in Appendix 1.*

2.3 Loop Interaction

The inner loop evolves ligands under a fixed pipeline until convergence or stagnation:

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

or until a maximum budget $T_{\max}$ is reached. Its outcomes are aggregated and passed to the outer loop, which then decides whether to accept a proposed pipeline modification. A single outer-loop update typically relies on many inner-loop cycles, grounding meta-level decisions in stable evidence. Note that here Δ refers to the median docking score difference per ligand, whereas $\hat{\mu}$ in the outer loop analysis refers to the mean improvement across paired runs.

2.4 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 = improvement).
- **Pass rate:** proportion of ligands satisfying chemical validity and drug-likeness constraints.
- **Trajectory analysis:** qualitative tracing of modifications leading to observed improvements.

Note: we report $\Delta = \text{R1} - \text{R0}$ (negative Δ indicates improvement), while the PAC test uses $Y := \text{R0} - \text{R1} = -\Delta$ so that larger is better.

All runs used fixed seeds and parameter settings; full environment manifests are provided in the supplementary repository. While evaluated here in molecular docking, the DGDM framework is domain-agnostic and in principle applicable across AI4Science tasks requiring bounded-risk self-improvement.

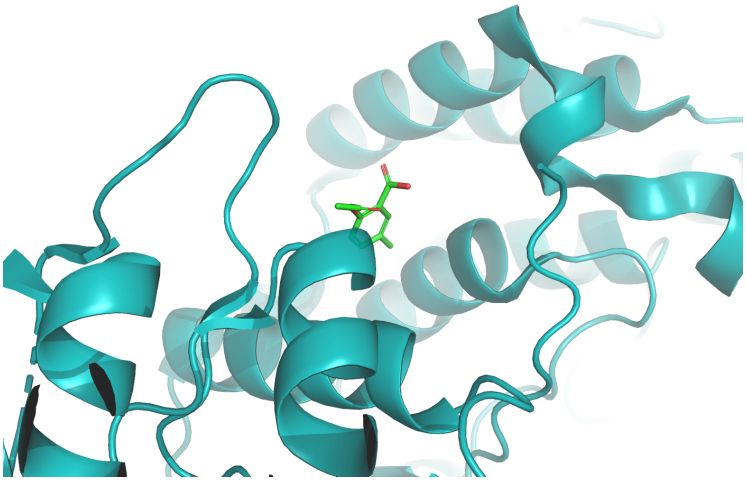

Figure 2: Docked pose of DGDM-optimized Aspirin_mut2 (green) in the target pocket (teal).

3 Experiments

We conducted a proof-of-concept (PoC) study to test whether DGDM can improve docking-based binding affinity predictions under bounded-risk constraints. Four seed ligands (Aspirin, LIG3, LIG4,

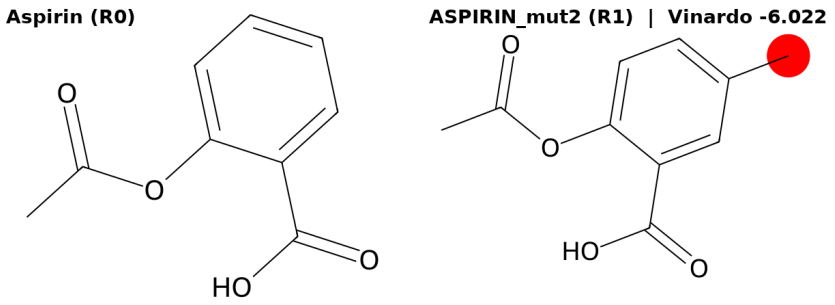

Figure 3: Structural comparison of baseline Aspirin (R0) and optimized variant Aspirin_mut2 (R1). Red highlights the DGDM-induced modification.

and Pyridine) were chosen to span diverse scaffolds and pharmacophores. Each ligand was docked in two stages: a baseline run (R0) and an optimized run (R1) where DGDM-generated variants were filtered and re-docked under identical conditions.

3.1 Setup

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

3.2 Metrics

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

3.3 Results

DGDM consistently improved median binding affinity across all ligands while preserving 100% chemical validity (Table 1). Improvements ranged from -0.8 to -1.5 kcal/mol, with Aspirin showing the strongest enhancement ($\Delta = -1.27$ kcal/mol).

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**

3.4 Limitations of Proof-of-Concept

This proof-of-concept study has several limitations. Evaluation was restricted to a small ligand panel and a single protein target, with docking scores serving only as approximate surrogates for binding affinity. Furthermore, the present experiments validate only the *inner loop* of DGDM; extending to the *outer loop*—where pipeline modifications are statistically tested under PAC-style safeguards—remains future work. These caveats underscore that the reported results should be interpreted as preliminary, motivating larger-scale benchmarking and experimental validation.

3.5 Qualitative Insights

Trajectory analysis revealed chemically interpretable improvements: variants typically introduced or repositioned hydrogen-bond donors/acceptors while reducing steric clashes. The top candidate, Aspirin_mut2 (Figure 2, 3), achieved a Vinardo score of -6.022 kcal/mol by adding a polar substitution that improved complementarity within the binding pocket. Notably, no ligand regressed relative to its baseline, underscoring the role of constraint-based survivor retention and PAC-style safeguards in preventing detrimental modifications.

4 Conclusion

We presented DGDM, a Darwin–Gödel–inspired dual-loop framework for molecular design that evolves both candidate structures and the optimization process itself. In a proof-of-concept study, DGDM consistently improved docking-based binding affinity while preserving chemical validity, highlighting the feasibility of bounded-risk generative modification.

Looking ahead, scaling DGDM will require integration with rescoring, molecular dynamics, and ultimately wet-lab assays to provide empirical feedback. Responsible deployment will also demand transparent benchmarking, auditable operators, and strong governance, given the higher standards of safety and reproducibility in drug discovery. While our experiments begin in molecular design, the underlying bounded-risk self-improvement framework is general and holds promise across AI for Science domains.

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.

Appendix 1 Error-Bounded Acceptance in the Gödel Loop

We formalize the statistical safeguard used in the Gödelian outer loop. Recall: in the main text we define

$$\Delta = R_1 - R_0 \quad \text{(negative } \Delta \text{ indicates improvement)},$$

while for analysis we use

$$Y := R_0 - R_1 = -\Delta,$$

so that larger Y indicates better performance of the modified pipeline.

In particular, we adopt a *Probably Approximately Correct (PAC)–style bound* via Hoeffding's inequality, which provides a distribution-free guarantee that harmful modifications are accepted with probability at most δ.

Lemma 1 (Hoeffding Confidence Bound)**.** *Let $Y_1, \ldots, Y_n \in [a, b]$ be i.i.d. paired improvements with true mean $\mu = \mathbb{E}[Y_i]$ and empirical mean*

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

Then for any $\delta \in (0, 1)$,

$$\Pr\left(\mu < \hat{\mu} - (b - a) \sqrt{\tfrac{1}{2n} \ln \tfrac{2}{\delta}} \right) \leq \delta.$$

Proof. Direct application of Hoeffding's inequality for bounded random variables. □

Corollary 1 (PAC Acceptance Rule)**.** *With probability at least $1 - \delta$, the true mean improvement μ is nonnegative provided that:*

$$\hat{\mu} - (b - a)\sqrt{\tfrac{1}{2n} \ln \tfrac{2}{\delta}} \;\geq\; 0.$$

Thus, adopting a pipeline modification under this criterion ensures that the risk of accepting a harmful modification is bounded by δ.

Sample Size Requirement. To guarantee error tolerance $\epsilon > 0$, it suffices to use

$$n \;\geq\; \frac{(b-a)^2}{2\epsilon^2} \ln \tfrac{2}{\delta}.$$

Practical Note. In docking applications, Y is bounded by physical score limits. We clip to $[a, b] = [-B, B]$ with $B = 5$ kcal/mol, consistent with typical Vinardo ranges. This bound is conservative, since docking scores rarely approach ± 5 in practice; tighter thresholds could be obtained with variance-adaptive inequalities such as the empirical Bernstein bound. In practice, paired t-tests are also applied as exploratory diagnostics; however, only the Hoeffding-based PAC acceptance rule provides a distribution-free guarantee.

Appendix 2 Algorithmic Details

Algorithm 1 Outer Loop: Pipeline Self-Adaptation

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

$\theta \leftarrow \theta_0,\ \mathcal{R} \leftarrow \{(\theta_0, \texttt{baseline})\}$
for $t = 1$ to T **do**
  $(\mathcal{C}, \mathbf{m}) \leftarrow$ RUNINNERLOOP$(\theta, \mathcal{H})$
  if STAGNANT$(\mathbf{m})$ **then**
    $\theta' \leftarrow$ PROPOSEEDIT$(G, \theta, \mathcal{R})$
    $\Delta \leftarrow$ EVALUATEPAIR$(\mathcal{H}, \theta, \theta', \mathcal{C})$
    if SIGNIFICANTIMPROVEMENT(Δ) **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 (w/o RL)

Input: Current configuration θ, harness $\mathcal{H}$, population size M
Output: 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 $(\mathcal{C}, \mathcal{S})$