Retrosynthesis Planning via Worst-path Policy Optimisation in Tree-structured MDPs

Mianchu Wang* **Giovanni Montana*†**
*University of Warwick
†The Alan Turing Institute
{mianchu.wang, g.montana}@warwick.ac.uk

Abstract

Retrosynthesis planning aims to decompose target molecules into available building blocks, forming a synthetic tree where each internal node represents an intermediate compound and each leaf ideally corresponds to a purchasable reactant. However, this tree becomes invalid if any leaf node is not a valid building block, making the planning process vulnerable to the "weakest link" in the synthetic route. Existing methods often optimise for average performance across branches, failing to account for this worst-case sensitivity. In this paper, we reframe retrosynthesis as a worst-path optimisation problem within tree-structured Markov Decision Processes (MDPs). We prove that this formulation admits a unique optimal solution and provides monotonic improvement guarantees. Building on this insight, we introduce Interactive Retrosynthesis Planning (InterRetro), a method that interacts with the tree MDP, learns a value function for worst-path outcomes, and improves its policy through self-imitation, preferentially reinforcing past decisions with high estimated advantage. Empirically, InterRetro achieves state-of-the-art results – solving 100% of targets on the Retro*-190 benchmark, shortening synthetic routes by 4.9%, and achieving promising performance using only 10% of the training data.

1 Introduction

Retrosynthesis aims to identify the reactants needed to synthesise a target molecule with desired properties. As a fundamental task in computer-aided molecular design, retrosynthesis underpins progress in drug discovery and materials science [4, 39]. A key challenge in retrosynthesis is single-step prediction, which involves predicting the reactants for a given product (illustrated in Figure 1). While recent approaches using supervised learning have achieved human-level accuracy in this task [2, 38], their practical applicability is limited, as the suggested reactants are often commercially unavailable. Unlike single-step prediction, multi-step planning recursively decomposes the target into simpler intermediates, aiming to construct a synthetic route whose leaf nodes are purchasable compounds. This process naturally forms a sequential decision-making problem, where early decisions influence future steps and the overall outcome [7, 21].

Most methods tackle this decision-making problem using heuristic search, with Monte Carlo Tree Search (MCTS) being widely adopted for its ability to balance exploration and exploitation [17, 23]. In this framework, a single-step retrosynthesis model suggests candidate reactions and predicts the most likely reactants for a given product. During simulations, MCTS selects reactions by weighing their estimated value against their exploration potential. After simulating a complete synthetic route, the values of the decisions made along the path are updated to guide future searches more effectively. Building on this foundation, recent studies have proposed various enhancements to improve the exploration–exploitation trade-off or to stabilise value estimation [1, 7, 31, 37]. However, these

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

methods heavily rely on time-consuming real-time search, requiring hundreds of model calls for each molecule, limiting their practical utility in large-scale molecular design scenarios.

To improve search efficiency, recent work fine-tunes pre-trained single-step models by imitating decisions made during successful heuristic search trajectories [9, 10, 34]. The key insight is that reaction choices selected after look-ahead planning are often more reliable than those proposed by the original model [22, 28]. By training on these improved decisions, the resulting policies can propose more plausible reactions and accelerate the search for viable synthetic routes. However, this fine-tuning process adapts the model to the distribution of molecules encountered during search, rather than those seen in direct inference. As a result, the model may perform well when used within the search loop but struggle to construct full synthetic routes independently, limiting its utility in settings where fast, search-free planning is desired.

In this paper, we propose a novel perspective: reframing retrosynthesis planning as a worst-path optimisation problem in tree-structured Markov Decision Processes (tree MDPs). We observe that existing approaches typically optimise for average or cumulative rewards across all root-to-leaf paths in the synthetic tree [10, 21], overlooking a critical insight: a synthetic route is only valid if every root-to-leaf path terminates at a purchasable compound. Even a single unsuccessful path invalidates the entire synthetic route, making the worst-performing path the limiting factor in overall performance. This observation leads us to introduce a new "worst-path" objective that focuses explicitly on improving the most challenging path in the synthetic route.

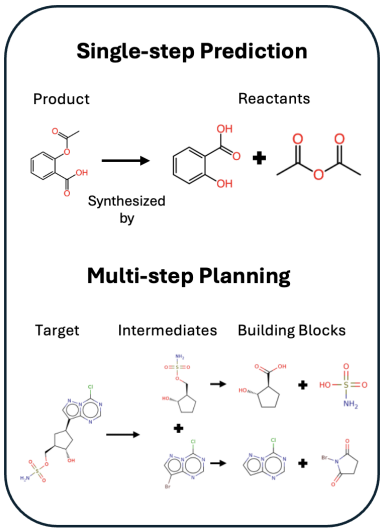


Figure 1: Single-step prediction decomposes a molecule into reactants, whereas multi-step planning searches for a synthetic route, aiming to reach purchasable building blocks.

Building on this novel formulation, we develop Interactive Retrosynthesis Planning (InterRetro) — a framework for multi-step retrosynthesis that learns to generate complete synthetic routes without search at inference time. InterRetro treats a single-step model as an agent operating within the tree MDP, recursively decomposing molecules into reactants through environment interactions. The agent is improved via weighted self-imitation of its past successful decisions to encourage shallow synthetic trees, effectively bootstrapping its performance. In particular, InterRetro operates in three key steps. Firstly, the agent interacts with the tree MDP to construct complete synthetic routes. Secondly, it identifies successful subtrees whose leaf nodes correspond to commercially available compounds. Finally, it fine-tunes the policy to imitate decisions from these subtrees, with theoretical guarantees of monotonic improvement and convergence to a unique optimal value function. This iterative self-improvement allows InterRetro to eliminate the need for real-time search while maintaining high planning quality.

This paper makes four key contributions: (1) we introduce the novel worst-path optimisation framework for tree-structured MDPs, specifically designed for problems where the weakest component determines the overall success; (2) we develop a weighted self-imitation learning algorithm with monotonic improvement guarantees for optimising this worst-path objective, proving the existence of a unique optimal solution through Bellman optimality analysis; (3) we apply this framework to retrosynthesis through InterRetro, a search-free approach to multi-step planning; and (4) we empirically demonstrate that InterRetro outperforms state-of-the-art (SOTA) methods in terms of success rate (achieving up to 100% on benchmark datasets), route length (reducing steps by 4.9%), and sample efficiency (reaching 92% of full performance with only 10% of training data).

2 Related Work

Single-step Prediction. The fundamental task in retrosynthesis is single-step prediction, which identifies the reactants that produce a given target molecule. Early approaches are template-based,

where reaction templates are extracted from patents and literature data [20], and a model selects the most appropriate template to decompose a molecule [2, 3]. Later methods propose template-free strategies, directly mapping the SMILES string of the target molecule to that of the reactants without relying on predefined transformation patterns [8, 26]. Recent studies introduce semi-template-based approaches, which combine both paradigms by first identifying intermediate structures and then completing them into full reactants — either by generating leaving groups [25], SMILES tokens [5, 32], or graph edits [30, 38]. While these single-step methods achieve high accuracy in predicting known reactions, they typically optimise for likelihood of historical reactions rather than synthesisability from commercially available compounds. In contrast, our approach explicitly favours reactions that lead to purchasable building blocks, enhancing downstream planning efficiency and practical applicability.

Multi-step Planning. Multi-step planning aims to construct a synthetic tree rooted at the target molecule, expanded by a single-step model, and terminating at commercially available compounds. Existing methods rely on heuristic search to find viable synthetic routes. For example, [6] formulates the retrosynthesis problem as an AND/OR tree search and adopts proof-number search to find the optimal solution; Retro* [1] explores the AND/OR tree using A* search and proposes value estimation methods in this setting; To improve exploration, [23] and [37] adopt MCTS, which better balances exploitation and exploration compared to A*. These search-based approaches, however, require extensive computation at inference time, often necessitating hundreds of model calls per molecule. Recent efforts such as [9] and [10] address this limitation by fine-tuning the single-step model using successful trajectories collected during the planning phase, reducing search iterations and improving synthesis success rates. Yet, these methods still ultimately rely on search during inference, merely reducing rather than eliminating this computational burden.

Self-imitation Learning. Self-imitation learning improves policy performance by encouraging the agent to replicate its own high-return past behaviours [15, 16]. A central aspect of self-imitation is the use of support constraints — the learned policy is regularised to remain close to the data-generating policy, which stabilises training and mitigates distributional shift [16, 29]. This principle has been effectively applied in offline and offline-to-online reinforcement learning [14, 24, 27], where self-imitation facilitates safe policy improvement without extensive exploration. In our work, we apply self-imitation to retrosynthesis planning by initialising the single-step model to generate reactions from the dataset and refining it through its past successful decisions, ensuring that its proposed reactions remain chemically plausible throughout training while progressively favouring those that lead to commercially available building blocks.

3 Retrosynthesis via Worst-path Optimisation in Tree MDPs

Prior work in multi-step retrosynthesis typically minimises the total cost of a synthetic route by optimising cumulative rewards across all root-to-leaf paths in the synthetic tree. In contrast, our worst-path objective targets the weakest path, ensuring viability by requiring all paths to terminate at purchasable compounds. To efficiently optimise this criterion, we propose a weighted self-imitation algorithm that leverages successful trajectories while prioritising improvement on failure-prone paths.

3.1 Tree-structured MDPs

We formalise the retrosynthesis problem as a tree-structured Markov Decision Process (tree MDP), denoted by $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, r, \mathcal{S}_{bb} \rangle$. Each state $s \in \mathcal{S}$ represents a molecule, and each action $a \in \mathcal{A}$ corresponds to a chemical reaction. Unlike standard MDPs, where each transition leads to a single successor state, a chemical reaction may decompose a molecule into multiple reactants. To capture this branching structure, we define the power set of all possible molecules as $2^{\mathcal{S}}$ and introduce a branching transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to 2^{\mathcal{S}}$, which maps a product molecule s and a reaction a to a set of reactants. These reactants become the children of s in the synthetic tree. The reward function $r : \mathcal{S} \to \mathbb{R}$ assigns a numerical value to each molecule, and $\mathcal{S}_{bb} \subset \mathcal{S}$ denotes the set of commercially available building blocks. A synthetic tree is considered successful only if all of its leaf nodes belong to $\mathcal{S}_{bb}$. A policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ defines a probability distribution over feasible reactions for a molecule s, thereby determining how the synthetic tree expands. Throughout this paper, we denote a complete synthetic tree by τ, and let $P(\tau)$ represent the set of all root-to-leaf paths within τ. The tree MDP formulation is illustrated in Figure 3.

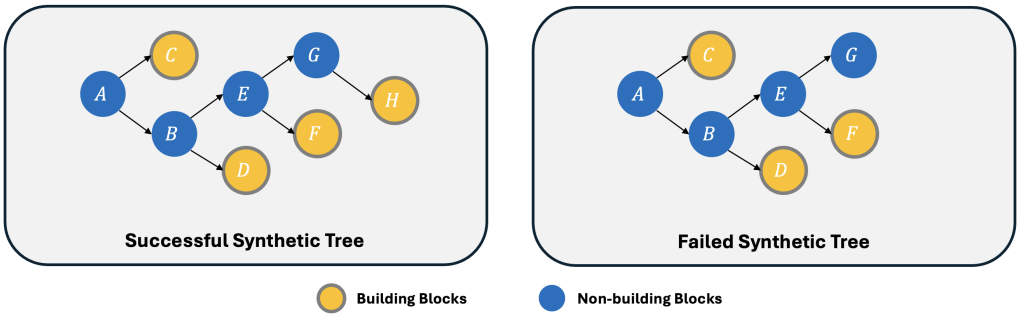


Figure 3: Examples illustrating the tree MDP formulation. Each non-leaf node represents a molecule that is decomposed into one or more reactants. *Left tree*: A successful synthetic route for target molecule A. It contains 4 root-to-leaf paths: $P(\tau) = \{\text{ABD}, \text{ABEF}, \text{ABEGH}, \text{AC}\}$. Since all leaf nodes are building blocks, each path receives a value of γ^T, where T is the path length. The tree's overall value is $\min_{p \in P(\tau)}\{\gamma^2, \gamma^3, \gamma^4, \gamma\} = \gamma^4$, determined by the longest path. *Right tree*: A failed synthesis attempt for molecule A. One of its paths, ABEG, terminates at G, which is not a building block. This gives path ABEG a value of 0, making the tree's overall value $\min_{p \in P(\tau)}\{\gamma^2, \gamma^3, 0, \gamma\} = 0$, illustrating why a single failing path invalidates the entire route.

3.2 Worst-path Objective

The width of synthetic trees is naturally bounded, as most molecules can be synthesised from only a few reactants. As shown in Figure 2, 98.6% of reactions in the USPTO-50k dataset involve three or fewer reactants. This suggests that the quality of a synthetic route is primarily determined by its depth rather than its branching factor. Accordingly, we evaluate each synthetic tree by the length of its longest root-to-leaf path. Figure 3 illustrates this concept: the left tree successfully synthesises molecule A, with its value determined by the longest path, whereas the right tree fails due to an unsynthesisable intermediate G, yielding a return of 0.

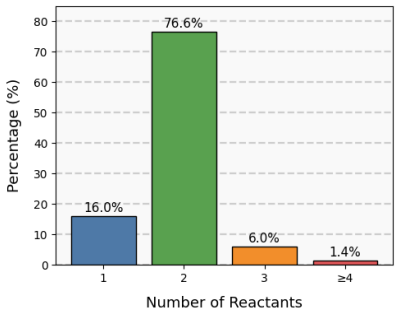


Figure 2: Distribution of reactant counts in the USPTO-50k dataset.

Formally, we define the reward function as

$$r(s) = \begin{cases} 1, & \text{if } s \in \mathcal{S}_{bb}, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

This assigns a reward of 1 to commercially available building blocks and 0 otherwise. Since the reward is non-zero only at terminal states (i.e., leaf nodes), the return of any root-to-leaf path $p = (s_0, a_0, s_1, \ldots, s_T)$ is

$$\sum_{t=0}^{T} \gamma^t r(s_t) = \gamma^T r(s_T), \tag{2}$$

where $\gamma \in (0, 1)$ is the discount factor. Hence, a successful path with $r(s_T) = 1$ receives value γ^T, penalising deeper decompositions, while dead-end paths with $r(s_T) = 0$ yield zero return. We then define the worst-path objective as

$$J(\pi) = \mathbb{E}_{\tau \sim \pi}\left[\min_{p \in P(\tau)} \sum_{t=0}^{T} \gamma^t r(s_t)\right]. \tag{3}$$

This objective captures the intuition that a synthetic route is only as viable as its weakest link: if any path leads to a dead-end, the entire tree's value becomes zero. Maximising the worst-path return therefore encourages the policy to ensure that every branch terminates at building blocks through the shortest possible routes.

3.3 Advantage Estimation

To optimise the worst-path objective, the agent interacts with the tree-structured MDP and learns to reproduce its past advantageous decisions. In this section, we formally define the advantage to quantify how beneficial a reaction is relative to the policy's average behaviour. We then propose to estimate this advantage using a value function learned from interaction experiences. Finally, we analyse its theoretical properties and establish the existence of an optimal policy that maximises the worst-path objective.

Given a molecule s as the root node, a Q-function estimates the expected worst-path return when first expanding s with reaction a and subsequently following policy π:

$$Q^\pi(s,a) = \mathbb{E}_{\tau\sim\pi}\left[\min_{p\in P(\tau)} \sum_{t=0}^{T} \gamma^t r(s_t) \;\middle|\; s_0 = s, a_0 = a\right]. \tag{4}$$

Here, s_0 and a_0 denote the root molecule and the initial reaction, respectively. Similarly, the value function $V^\pi(s)$ represents the expected worst-path return when all subsequent reactions follow policy π:

$$V^\pi(s) = \mathbb{E}_{\tau\sim\pi}\left[\min_{p\in P(\tau)} \sum_{t=0}^{T} \gamma^t r(s_t) \;\middle|\; s_0 = s\right]. \tag{5}$$

With these definitions in place, we can express the advantage function, which quantifies the relative benefit of applying reaction a to molecule s compared to following the policy:

$$A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s). \tag{6}$$

A positive advantage indicates that reaction a leads to better outcomes than the policy's average behaviour. Proposition 1 establishes a relationship between $Q^\pi(s,a)$ and $V^\pi(s)$:

Proposition 1. *The Q-function $Q^\pi(s,a)$ equals its immediate reward plus the discounted value of its next states:*

$$Q^\pi(s,a) = r(s) + \gamma(1-r(s)) \min_{s'\in\mathcal{T}(s,a)} V^\pi(s'). \tag{7}$$

The proof is provided in Appendix B.1.

Thus, we can estimate the advantage as:

$$A^\pi(s,a) = r(s) + \gamma(1-r(s)) \min_{s'\in\mathcal{T}(s,a)} V^\pi(s') - V^\pi(s). \tag{8}$$

Next, we derive a recursive form of the value function, which forms the foundation for learning a parameterised value function.

Proposition 2. *The value function $V^\pi(s)$ satisfies the recursion:*

$$V^\pi(s) = r(s) + \gamma(1-r(s)) \sum_{a\in\mathcal{A}} \pi(a \mid s) \min_{s'\in\mathcal{T}(s,a)} V^\pi(s'). \tag{9}$$

The proof is provided in Appendix B.2.

Beyond the value functions for a given policy, we can define the optimal worst-path value function, V^*, which represents the maximum possible worst-path return achievable from any state.

Proposition 3. *This optimal value function V^* uniquely satisfies the Bellman optimality equation for the worst-path objective:*

$$V^*(s) = r(s) + \gamma(1-r(s)) \max_{a\in\mathcal{A}} \left[\min_{s'\in\mathcal{T}(s,a)} V^*(s')\right], \tag{10}$$

The proof is provided in Appendix B.3.

In the proof, we show that the corresponding Bellman optimality operator is a contraction mapping, which guarantees the existence and uniqueness of V^* and the convergence of value iteration to V^*. Furthermore, there exists at least one deterministic stationary policy π^* that is greedy with respect to V^* and is therefore an optimal policy.

3.4 Self-imitation Learning for Retrosynthesis

A key challenge in retrosynthesis is ensuring that the learned policy proposes chemically valid reactions at each step. To address this, we constrain policy learning within the support of a pre-trained single-step model, which can empirically capture high-fidelity chemical transformations. Specifically, we leverage a pre-trained single-step model π^0, trained to reflect expert or empirical reaction knowledge from the dataset [38], and define the constrained policy set:

$$\Pi = \{\pi \mid \pi(a \mid s) = 0 \text{ whenever } \pi^0(a \mid s) = 0\}. \tag{11}$$

By restricting π to actions that π^0 assigns non-zero probability, we eliminate the risk of generating unrealistic reactions while allowing flexibility to re-weight feasible actions based on their effectiveness for multi-step planning.

Our objective is to find a policy $\pi \in \Pi$ that maximises the worst-path objective. We achieve this through an iterative procedure: in each iteration i, we aim to find an improved policy π^{i+1} by imitating advantageous state-action pairs (s, a) experienced under policy π^i. The advantage $A^{\pi^i}(s, a)$ quantifies this, and the learning objective for π^{i+1} is formulated as:

$$J(\pi^{i+1}) = \mathbb{E}_{s\sim d_{\pi^i}(\cdot), a\sim\pi^i(\cdot|s)} \left[\exp\big(\beta A^{\pi^i}(s, a)\big) \log \pi^{i+1}(a \mid s)\right], \tag{12}$$

where $\beta > 0$ is the advantage coefficient controlling the strength of advantage weighting, and d_{π^i} is the state distribution induced by policy π^i [29]. In this case, reactions with higher advantages receive higher weights, guiding the policy toward better-than-average reactions.

Since each new policy π^{i+1} is derived by re-weighting π^i, and the initial policy π^0 restricts the support, the entire policy sequence $\{\pi^i\}_{i\geq 0}$ remains within the feasible set Π [12]. This ensures that all proposed reactions throughout training remain chemically valid.

Proposition 4. *Let π^{i+1} be the policy obtained by optimising the objective in Eq.12. Then, the updated policy is guaranteed to perform at least as well as the previous policy for all states:*

$$V^{\pi^{i+1}}(s) \geq V^{\pi^i}(s), \forall s \in \mathcal{S}. \tag{13}$$

The proof is provided in Appendix B.4.

Proposition 4 guarantees monotonic improvement under the exact policy update:

$$\pi^{i+1}(a \mid s) \propto \pi^i(a \mid s) \exp\big(\beta A^{\pi^i}(s, a)\big), \tag{14}$$

which re-weights the previous policy by the exponentiated advantages, thereby increasing the probability of actions that yield higher returns than the current expectation.

For policy optimisation, we use the current policy π^i to interact with the retrosynthesis environment and collect data to learn the next iteration's policy π^{i+1}. Through iterative weighted imitation, the policy increasingly favours high-quality reactions that lead to successful synthetic routes with shorter paths, while maintaining chemical plausibility by respecting the support of the pre-trained model.

4 The InterRetro Algorithm

We now present our algorithm InterRetro for retrosynthesis planning. Section 4.1 introduces how the single-step model interacts with the tree MDP and collects trajectories. Section 4.2 describes how to learn a value function on the worst-path objective and how to fine-tune the policy to reproduce advantageous decisions. Finally, Section 4.3 provides more implementation details for reproducibility.

4.1 Environment Interactions

The EXPLORE procedure in Algorithm 1 shows how InterRetro constructs a synthetic tree by interacting with the tree MDP. Starting from a target molecule m, the single-step model proposes a reaction a to decompose the molecule into a set of reactants $\mathcal{S}_r$. These reactants will be attached to the parent node m in the synthetic tree. Among them, the non-building blocks are then placed in a collection (e.g., a queue) for further expansion. Each subsequent round pops a molecule from the collection to continue the decomposition process. The interaction terminates when the collection becomes empty (meaning all leaf nodes are building blocks) or when a predefined maximum number of steps is reached.

Algorithm 1 Interactive retrosynthesis planning (InterRetro).

Input: pre-trained one-step policy π_θ, value function V_ϕ, training set $\mathcal{D}$, replay buffer $\mathcal{B}$.

def EXPLORE(π_θ, m):
- $\texttt{tree} \leftarrow \texttt{Tree}(\texttt{root} = m)$
- $\texttt{q} \leftarrow \{m\}$
- $\texttt{step} \leftarrow 0$
- **while** $\texttt{q} \neq \emptyset$ **and** $\texttt{step} < \texttt{max\_steps}$ **do**
  - $s \leftarrow \texttt{q.pop()}$
  - $a, \mathcal{S}_r \leftarrow \pi_\theta.\texttt{get\_reactants}(s)$
  - $\texttt{tree.add\_branch}(s, a, \mathcal{S}_r)$
  - `# Add non-building blocks`
  - $\texttt{q} \leftarrow \texttt{q} \cup \{\, s' \in \mathcal{S}_r \mid s' \notin \mathcal{S}_{bb} \,\}$
  - $\texttt{step} \leftarrow \texttt{step} + 1$
- **end while**
- **return** tree

def INTERRETRO(π_θ, m):
- **for** $i = 1, \ldots, I$ **do**
  - **while** $\mathcal{D}$ is not empty **do**
    - $m \leftarrow \mathcal{D}.\texttt{pop()}$
    - $\texttt{tree} \leftarrow$ EXPLORE(π_θ, m)
    - $\texttt{brs} \leftarrow \{\}$
    - **for each** subtree $\tau \in \texttt{tree}$ **do**
      - **if** τ is successful **then**
        - $\texttt{brs} \leftarrow \texttt{brs} \cup$ ALLBRANCHES(τ)
      - **end if**
    - **end for**
    - $\mathcal{B}.\texttt{append}(\texttt{brs})$
    - $\texttt{branches} \leftarrow \mathcal{B}.\texttt{sample()}$
    - $V_\phi.\texttt{update}(\texttt{branches})$ ▷ Eq. 15
    - $\pi_\theta.\texttt{update}(V_\phi, \texttt{branches})$ ▷ Eq. 16
  - **end while**
- **end for**

4.2 Value Function and Policy Learning

The EXPLORE procedure constructs synthetic trees by interacting with the tree MDP. We extract all branches $(s, a, \mathcal{S}_r)$ within successful subtrees and store them in a first-in first-out replay buffer $\mathcal{B}$ for value function and policy learning (see INTERRETRO in Algorithm 1).

To learn the value function network V_ϕ with parameter ϕ, we minimise the mean squared error between the predicted value $V_\phi(s)$ and its Bellman target derived from Proposition 2:

$$\mathcal{L}(\phi) = \mathbb{E}_{(s,a,\mathcal{S}_r)\sim\mathcal{B}}[(V_\phi(s) - (r(s) + \gamma(1 - r(s)) \min_{s' \in \mathcal{S}_r} V_{\phi^-}(s')))^2], \tag{15}$$

where V_{ϕ^-} is a target network updated slowly for stability.

The policy network π_θ with pre-trained parameter θ is updated using the weighted imitation learning objective from Eq. 12, implemented as the following loss function:

$$\mathcal{L}(\theta) = -\mathbb{E}_{(s,a,\mathcal{S}_r)\sim\mathcal{B}}[\exp_{\text{clip}}(\beta A_\phi(s, a)) \log \pi_\theta(a \mid s)], \tag{16}$$

where $\beta > 0$ is the advantage coefficient which controls the imitation strength on the past successful experience, and $\exp_{\text{clip}}(\cdot)$ is the exponential function with a clipped output range $(0, C]$ for numerical stability. The advantage is estimated by the value function network, according to Eq. 8:

$$A_\phi(s, a) = r(s) + \gamma(1 - r(s)) \min_{s' \in \mathcal{T}(s,a)} V_\phi(s') - V_\phi(s). \tag{17}$$

This joint optimisation of the value and policy networks enables InterRetro to fine-tune the single-step model, increasing the probability of reproducing high-advantage past decisions as indicated by the value function.

4.3 Implementation Details

We choose Graph2Edits as the single-step model due to its strong performance and flexibility in single-step retrosynthesis prediction [38]. Graph2Edits represents a molecule through a graph and predicts a sequence of graph edits to transform it to reactants. These graph edits can delete bonds, modify bond types, alter atoms, or attach leaving groups (see Appendix C for more details). The single-step model and our proposed value function encode the molecule using a message passing neural network [33] and forward the latent code into linear layers with ReLU as the activation function. For training, we run 6 parallel exploration processes and collect 36 synthetic trees per iteration. The networks are updated 5 times per iteration using data from a compact replay buffer of maximal $20,000$ branches, chosen to reduce CPU memory usage and maintain close alignment between the data-collecting policy π^i and the updated policy π^{i+1}. Our models are trained on a single NVIDIA RTX A5000 GPU and, without pre-training the single-step model, require approximately 48 hours to fully converge. The code has been open-sourced[1].

[1] GitHub repository: https://github.com/MianchuWang/InterRetro.

Table 1: Performance evaluation on three benchmarks. The evaluation metrics include the success rate under different test molecules with different budgets of model calls, which are direct generation (DG), 100, 200 and 500 model calls. The DG columns are single-step model's results without search.

		Retro*-190				ChEMBL-1000				GDB17-1000			
Single-step	**Search**	DG	100	200	500	DG	100	200	500	DG	100	200	500
Template	MCTS	20.00	43.68	47.37	62.63	32.00	45.60	68.80	71.90	3.00	3.20	3.70	4.50
Template	Retro*	20.00	38.42	58.42	75.26	32.00	69.10	72.00	74.70	3.00	5.40	6.60	7.50
LocalRetro	MCTS	22.10	44.21	57.36	62.10	47.30	62.70	69.10	75.00	4.60	14.00	16.70	20.30
LocalRetro	Retro*	22.10	58.94	64.73	73.68	47.30	74.80	80.40	82.40	4.60	18.90	22.20	28.80
MEGAN	Retro*	8.42	60.52	62.10	73.15	38.00	71.70	75.40	79.00	6.20	37.60	45.70	57.20
Graph2Edits	Retro*	16.84	41.05	50.00	56.31	47.10	68.70	78.80	80.70	5.90	18.20	24.00	32.20
Self-improve	Retro*	–	67.37	83.16	94.74	–	–	–	81.10	–	–	–	15.00
PDVN	Retro*	–	93.68	97.37	98.95	–	–	–	83.50	–	–	–	26.90
RetroCaptioner	Retro*	5.26	68.94	72.63	85.26	3.90	72.60	76.50	78.70	3.20	56.20	68.20	75.20
DreamRetroer		32.10	78.94	88.42	90.52	31.10	78.10	81.70	83.10	4.20	27.36	28.97	33.20
InterRetro	MCTS	**95.78**	89.47	98.94	**100.00**	**93.10**	78.40	89.30	97.50	**89.00**	80.80	96.10	**99.50**
InterRetro	Retro*	**95.78**	**96.31**	**100.00**	**100.00**	**93.10**	**91.40**	**96.20**	**98.20**	**89.00**	**83.80**	**96.50**	97.20

5 Experimental Results

In this section, we aim to answer the following questions: (1) What are the advantages of our proposed method compared to the SOTA algorithms? (2) How does each component of the method contribute to the performance? Additionally, we examine the real-world feasibility of the proposed synthetic routes and present illustrative examples in Appendix A.

5.1 Benchmark Results

The proposed method is trained by creating synthetic routes for nearly $300k$ molecules in the USPTO-50k dataset with commercially available building blocks from the *eMolecules* dataset[2]. We evaluate performance on three benchmarks of increasing difficulty: Retro*-190 [1], ChEMBL-1000 [10, 35], and GDB17-1000 [10, 18], where the suffix indicates the dataset size. We compare against established retrosynthesis planning methods: MCTS, Retro* [1], Self-improve [9], PDVN [10], and GraphRetro [31]. Additionally, we include two single-step methods, MEGAN [19] and Graph2Edits [38], combined with Retro* as the search algorithm, and two recently proposed methods, DreamRetroer [36] and RetroCaptioner [11]. Baseline results are produced from their official implementations or the Syntheseus project [13].

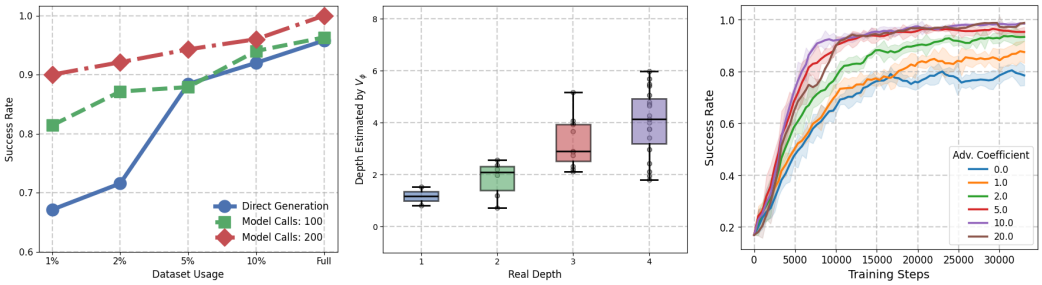
Success Rate. Success rate measures the percentage of target molecules that can be successfully decomposed into building blocks. In Table 1, we firstly compare methods under different model-call budgets: 100, 200, and 500. Our proposed InterRetro significantly outperforms SOTA algorithms across all three test sets. With 500 model calls, InterRetro achieves 100%, 98.2%, and 99.5% success rates on Retro*-190, ChEMBL-1000, and GDB17-1000, respectively. All synthetic routes generated by InterRetro on Retro*-190 using 500 model calls are provided in the supplementary materials.

Most notably, InterRetro maintains exceptional performance on the challenging GDB17-1000 benchmark, where prior approaches such as PDVN and Self-improve struggle (26.9% and 15.0% respectively). This dramatic performance gap suggests that our worst-path objective effectively handles complex molecules that require precise reaction planning.

Furthermore, InterRetro can directly generate high-quality synthetic routes without any search algorithm, as shown in the Direct Generation (DG) columns. Our search-free performance (95.78%, 93.10%, and 89.00% across the three benchmarks) substantially exceeds even the search-based performance of competing methods. This demonstrates that our self-imitation learning approach successfully transfers planning capabilities to inference time, effectively eliminating the computational bottleneck of real-time search.

Route Length. Route length is the number of reactions needed to synthesise a target molecule. Route length could be a concern with our worst-path objective since it does not consider the width of the tree and the total number of reactions required. In Table 2, we compare the route length with other baselines on the 138 target molecules that all methods can resolve. Our method outperforms the SOTA by 4.85%.

[2]eMolecules: https://downloads.emolecules.com/free/.



(a) (b) (c)

Figure 4: Experimental figures. **(a)** Performance under different training data usage and computation budgets. **(b)** Statistics on estimated depth of synthetic trees. **(c)** Ablations on the advantage coefficient.

Sample Efficiency. Sample efficiency refers to the amount of training data required to learn an effective policy. We evaluate the policy using subsets of the training data at 1%, 2%, 5%, 10%, and 100%, corresponding to approximately $3k$, $6k$, $15k$, $30k$, and $300k$ molecules. As shown in Figure 4a, InterRetro achieves near SOTA performance (92%) with just 10% of the training set when directly generating synthetic trees, and reaches SOTA performance when combined with 200 Retro* search iterations. Notably, when trained on the full dataset, InterRetro outperforms most existing methods in direct generation, achieving a success rate of 95.78%. Similar trends are observed on the ChEMBL-1000 and GDB17-1000 benchmarks.

Table 2: The average length of the routes on the Retro*-190 test set.

Algorithm	Average Length
Retro*	5.83
RetroGraph	5.63
PDVN	4.83
MEGAN	4.12
Graph2Edits	4.38
InterRetro	**3.92**

5.2 Ablation Studies

Value Estimation. We proposed a value function to estimate the worst-path return. The worst-path return indicates the estimated depth of the synthetic tree: $\text{depth}(s) = \frac{\log V_\phi(s)}{\log \gamma}$. In Figure 4b, we investigate the distribution of the estimated depth, corresponding to the real depth by direct generation. We found that the value function can reflect the difficulties to synthesise the molecules, while it shows better capability on the molecules that require a less deep synthetic route.

Advantage Coefficient. The hyperparameter β controls the imitation strength on the past successful experience. A large β means a focused imitation on the action with a high advantage. Figure 4c shows the learning curve on the Retro*-190 test set with $\beta \in \{0, 1, 2, 5, 10, 20\}$. The figure shows that the DG performance increases from 16% to around 80% by uniformly imitating successful past experiences. With the advantage weighting, the performance soars to a success rate of more than 90% when β increases to 2, and more than 95% with $\beta \geq 10$.

6 Conclusions and Discussion

We introduced InterRetro, a novel approach that frames retrosynthesis planning as worst-path optimisation in tree-structured MDPs. Our weighted self-imitation algorithm enables single-step models to generate high-quality synthetic routes without search at inference time, achieving SOTA performance in success rate, route length, and sample efficiency.

Further investigation can proceed in three main directions. First, the worst-path objective could be extended to incorporate additional real-world constraints, such as reaction conditions and the cost of building blocks. Second, and perhaps most importantly, our method and other contemporary work assume that all proposed reactions are feasible in real-world settings — an assumption that does not always hold. Although we mitigate this by imitating only actions supported by pre-trained models, computational approaches cannot guarantee practical feasibility without experimental validation. Future work can therefore focus on integrating reaction feasibility checks into InterRetro. Furthermore,

the "weakest-link" principle underlying our method is broadly applicable beyond chemistry, extending to sequential decision problems where overall success depends on the least reliable component — for example, in robust project planning where delays in any critical task impact the entire timeline, or in multi-agent systems where team performance is constrained by the least capable agent.

Broader Impacts. The retrosynthesis community has recently open-sourced many high-quality models capable of suggesting synthetic routes for a vast number of molecules — including both beneficial drugs and potentially harmful compounds. To promote safe and ethical research, we propose that access to such models or their source code be governed by a Responsible Use License, under which researchers acknowledge the responsible and lawful application of the technology.

Acknowledgments

We acknowledge support from a UKRI Turing AI Acceleration Fellowship (EPSRC EP/V024868/1).

References

[1] Binghong Chen, Chengtao Li, Hanjun Dai, and Le Song. Retro*: Learning retrosynthetic planning with neural guided A* search. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1608–1616. PMLR, Jul 2020.

[2] Shuan Chen and Yousung Jung. Deep retrosynthetic reaction prediction using local reactivity and global attention. *JACS Au*, 1(10):1612–1620, Oct 2021.

[3] Connor W Coley, Luke Rogers, William H Green, and Klavs F Jensen. Computer-assisted retrosynthesis based on molecular similarity. *ACS Cent. Sci.*, 3(12):1237–1245, Dec 2017.

[4] Jingxin Dong, Mingyi Zhao, Yuansheng Liu, Yansen Su, and Xiangxiang Zeng. Deep learning in retrosynthesis planning: datasets, models and tools. *Briefings in Bioinformatics*, 23(1):bbab391, Sep 2021.

[5] Yuqiang Han, Xiaoyang Xu, Chang-Yu Hsieh, Keyan Ding, Hongxia Xu, Renjun Xu, Tingjun Hou, Qiang Zhang, and Huajun Chen. Retrosynthesis prediction with an iterative string editing model. *Nature Communications*, 15(1):6404, Jul 2024.

[6] Abraham Heifets and Igor Jurisica. Construction of new medicines via game proof search. *Proceedings of the AAAI Conference on Artificial Intelligence*, 26(1):1564–1570, Sep. 2021.

[7] Siqi Hong, Hankz Hankui Zhuo, Kebing Jin, Guang Shao, and Zhanwen Zhou. Retrosynthetic planning with experience-guided Monte Carlo tree search. *Communications Chemistry*, 6(1):120, Jun 2023.

[8] Ross Irwin, Spyridon Dimitriadis, Jiazhen He, and Esben Jannik Bjerrum. Chemformer: a pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology*, 3(1):015022, Jan 2022.

[9] Junsu Kim, Sungsoo Ahn, Hankook Lee, and Jinwoo Shin. Self-improved retrosynthetic planning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5486–5495. PMLR, Jul 2021.

[10] Guoqing Liu, Di Xue, Shufang Xie, Yingce Xia, Austin Tripp, Krzysztof Maziarz, Marwin Segler, Tao Qin, Zongzhang Zhang, and Tie-Yan Liu. Retrosynthetic planning with dual value networks. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 22266–22276. PMLR, Jul 2023.

[11] Xiaoyi Liu, Chengwei Ai, Hongpeng Yang, Ruihan Dong, Jijun Tang, Shuangjia Zheng, and Fei Guo. Retrocaptioner: beyond attention in end-to-end retrosynthesis transformer via contrastively captioned learnable graph representation. *Bioinformatics*, 40(9):btae561, Sep 2024.

[12] Yixiu Mao, Hongchang Zhang, Chen Chen, Yi Xu, and Xiangyang Ji. Supported trust region optimization for offline reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 23829–23851. PMLR, Jul 2023.

[13] Krzysztof Maziarz, Austin Tripp, Guoqing Liu, Megan Stanley, Shufang Xie, Piotr Gaiński, Philipp Seidl, and Marwin H. S. Segler. Re-evaluating retrosynthesis algorithms with Syntheseus. *Faraday Discuss.*, 256:568–586, 2025.

[14] Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. AWAC: Accelerating online reinforcement learning with offline datasets, 2021.

[15] Junhyuk Oh, Yijie Guo, Satinder Singh, and Honglak Lee. Self-imitation learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3878–3887. PMLR, Jul 2018.

[16] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning, 2019.

[17] Milo Roucairol and Tristan Cazenave. Comparing search algorithms on the retrosynthesis problem. *Molecular Informatics*, 43(7):e202300259, 2024.

[18] Lars Ruddigkeit, Ruud van Deursen, Lorenz C Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. *J. Chem. Inf. Model.*, 52(11):2864–2875, Nov 2012.

[19] Mikołaj Sacha, Mikołaj Błaż, Piotr Byrski, Paweł Dąbrowski-Tumański, Mikołaj Chromiński, Rafał Loska, Paweł Włodarczyk-Pruszyński, and Stanisław Jastrzębski. Molecule edit graph attention network: Modeling chemical reactions as sequences of graph edits. *J. Chem. Inf. Model.*, 61(7):3273–3284, Jul 2021.

[20] Nadine Schneider, Nikolaus Stiefl, and Gregory A Landrum. What's what: The (nearly) definitive guide to reaction role assignment. *J. Chem. Inf. Model.*, 56(12):2336–2346, Dec 2016.

[21] John S Schreck, Connor W Coley, and Kyle J M Bishop. Learning retrosynthetic planning through simulated experience. *ACS Cent. Sci.*, 5(6):970–981, Jun 2019.

[22] Julian Schrittwieser, Thomas K Hubert, Amol Mandhane, Mohammadamin Barekatain, Ioannis Antonoglou, and David Silver. Online and offline reinforcement learning by planning with a learned model. In *Advances in Neural Information Processing Systems*, 2021.

[23] Marwin H S Segler, Mike Preuss, and Mark P Waller. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature*, 555(7698):604–610, Mar 2018.

[24] Noah Siegel, Jost Tobias Springenberg, Felix Berkenkamp, Abbas Abdolmaleki, Michael Neunert, Thomas Lampe, Roland Hafner, Nicolas Heess, and Martin Riedmiller. Keep doing what worked: Behavior modelling priors for offline reinforcement learning. In *International Conference on Learning Representations*, 2020.

[25] Vignesh Ram Somnath, Charlotte Bunne, Connor Coley, Andreas Krause, and Regina Barzilay. Learning graph models for retrosynthesis prediction. In *Advances in Neural Information Processing Systems*, volume 34, pages 9405–9415. Curran Associates, Inc., 2021.

[26] Zhengkai Tu and Connor W Coley. Permutation invariant Graph-to-Sequence model for template-free retrosynthesis and reaction prediction. *J. Chem. Inf. Model.*, 62(15):3503–3513, Aug 2022.

[27] Mianchu Wang, Yue Jin, and Giovanni Montana. Learning on one mode: Addressing multi-modality in offline reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[28] Mianchu Wang, Rui Yang, Xi Chen, Hao Sun, Meng Fang, and Giovanni Montana. GOPlan: Goal-conditioned offline reinforcement learning by planning with learned models. *Transactions on Machine Learning Research*, 2024.

[29] Qing Wang, Jiechao Xiong, Lei Han, peng sun, Han Liu, and Tong Zhang. Exponentially weighted imitation learning for batched historical data. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[30] Yu Wang, Chao Pang, Yuzhe Wang, Junru Jin, Jingjie Zhang, Xiangxiang Zeng, Ran Su, Quan Zou, and Leyi Wei. Retrosynthesis prediction with an interpretable deep-learning framework based on molecular assembly tasks. *Nature Communications*, 14(1):6155, Oct 2023.

[31] Shufang Xie, Rui Yan, Peng Han, Yingce Xia, Lijun Wu, Chenjuan Guo, Bin Yang, and Tao Qin. RetroGraph: Retrosynthetic planning with graph search. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 2120–2129, New York, NY, USA, 2022. Association for Computing Machinery.

[32] Chaochao Yan, Qianggang Ding, Peilin Zhao, Shuangjia Zheng, JINYU YANG, Yang Yu, and Junzhou Huang. RetroXpert: Decompose retrosynthesis prediction like a chemist. In *Advances in Neural Information Processing Systems*, volume 33, pages 11248–11258. Curran Associates, Inc., 2020.

[33] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, Andrew Palmer, Volker Settels, Tommi Jaakkola, Klavs Jensen, and Regina Barzilay. Analyzing learned molecular representations for property prediction. *J. Chem. Inf. Model.*, 59(8):3370–3388, Aug 2019.

[34] Yemin Yu, Ying Wei, Kun Kuang, Zhengxing Huang, Huaxiu Yao, and Fei Wu. Grasp: Navigating retrosynthetic planning with goal-driven policy. In *Advances in Neural Information Processing Systems*, volume 35, pages 10257–10268. Curran Associates, Inc., 2022.

[35] Barbara Zdrazil, Eloy Felix, Fiona Hunter, Emma J Manners, James Blackshaw, Sybilla Corbett, Marleen de Veij, Harris Ioannidis, David Mendez Lopez, Juan F Mosquera, Maria Paula Magarinos, Nicolas Bosc, Ricardo Arcila, Tevfik Kizilören, Anna Gaulton, A Patrícia Bento, Melissa F Adasme, Peter Monecke, Gregory A Landrum, and Andrew R Leach. The chembl database in 2023: a drug discovery platform spanning multiple bioactivity data types and time periods. *Nucleic Acids Research*, 52(D1):D1180–D1192, 11 2023.

[36] Xuefeng Zhang, Haowei Lin, Muhan Zhang, Yuan Zhou, and Jianzhu Ma. A data-driven group retrosynthesis planning model inspired by neurosymbolic programming. *Nature Communications*, 16(1):192, Jan 2025.

[37] Dengwei Zhao, Shikui Tu, and Lei Xu. Efficient retrosynthetic planning with MCTS exploration enhanced A* search. *Communications Chemistry*, 7(1):52, Mar 2024.

[38] Weihe Zhong, Ziduo Yang, and Calvin Yu-Chian Chen. Retrosynthesis prediction using an end-to-end graph generative architecture for molecular graph editing. *Nature Communications*, 14(1):3009, May 2023.

[39] Zipeng Zhong, Jie Song, Zunlei Feng, Tiantao Liu, Lingxiang Jia, Shaolun Yao, Tingjun Hou, and Mingli Song. Recent advances in deep learning for retrosynthesis. *WIREs Computational Molecular Science*, 14(1):e1694, 2024.

A Case Studies

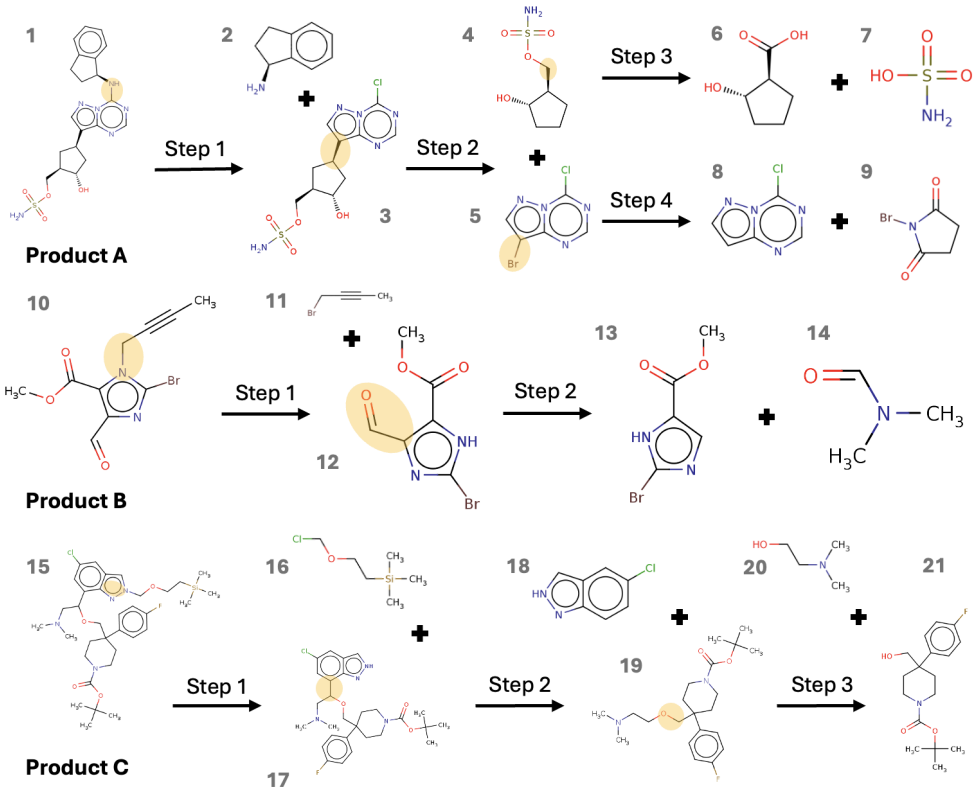


Figure 5: Predicted synthetic routes on three randomly selected molecules in Retro*-190. The yellow circles highlight the reaction centres.

In Figure 5, we randomly select three targets from the Retro*-190 benchmark and illustrate their complete synthetic trees as directly generated by InterRetro in a single forward pass. (i) For the sulfonylated chiral cyclohexanol derivative, InterRetro proposes a four-step route involving: a sulfonylation of a chiral alcohol to introduce the sulfonate ester, an electrophilic bromination of the diazine core, a C(sp^3)–C(sp^2) cross-coupling to build the fused triazolopyrimidine system, and an N-arylation to complete the final structure. (ii) For the alkyne-substituted diazine, InterRetro constructs the target via two steps: first an acylation of the brominated azine to give the N-formyl intermediate, followed by an SN2 N-propargylation. (iii) For the complex triazolopyridine-fused drug-like scaffold, InterRetro suggests a three-step route: a benzylic etherification that assembles the protected piperidine core, a C(sp^3)–C(sp^2) cross-coupling to append the triazolopyridine fragment, and an N-alkylation that installs the silyl-protected side chain. Across all three cases, InterRetro applies feasible chemical transitions and terminates in commercially available building blocks. A full list of the synthetic routes on Retro*-190 is attached in the supplementary materials.

B Proofs and Further Theory

B.1 Proof of Proposition 1

Proof. Recall that

$$Q^{\pi}(s,a) \;=\; \mathbb{E}_{\tau\sim\pi}\Big[\min_{p\in P(\tau)} \sum_{t=0}^{T} \gamma^{t}\, r\big(s_t\big) \;\Big|\; s_0 = s,\; a_0 = a\Big].$$

Since the transition function $\mathcal{T}$ is deterministic, taking action a in s immediately leads to the set of next states $\mathcal{T}(s,a)$. Observe that $r(s)$ is 1 if and only if s is a building block (in which case no further transitions occur), and 0 otherwise. We prove the desired equality by a simple case distinction:

1. Case $\boldsymbol{r(s) = 1}$ (i.e., s is a building block). In this case, once we reach s, the path terminates and collects reward 1. Hence

$$Q^\pi(s,a) \;=\; \underbrace{r(s)}_{=1} \;=\; r(s) + \gamma\,\big(1 - r(s)\big) \min_{s' \in \mathcal{T}(s,a)} V^\pi(s') \quad \text{(since } 1 - r(s) = 0\text{)}.$$

2. Case $\boldsymbol{r(s) = 0}$ (i.e., s is not a building block). Any path extending from (s,a) must proceed into one of the next states $s' \in \mathcal{T}(s,a)$. Because we are taking a "worst-path" (minimum-return) perspective, the worst-case continuation value from s under action a is

$$\min_{s' \in \mathcal{T}(s,a)} V^\pi(s'),$$

and each step is discounted by γ. Therefore,

$$Q^\pi(s,a) \;=\; \underbrace{r(s)}_{=0} + \gamma \min_{s' \in \mathcal{T}(s,a)} V^\pi(s') \;=\; r(s) \;+\; \gamma\,\big(1 - r(s)\big) \min_{s' \in \mathcal{T}(s,a)} V^\pi(s').$$

Combining both cases completes the proof:

$$Q^\pi(s,a) \;=\; r(s) \;+\; \gamma\,\big(1 - r(s)\big) \min_{s' \in \mathcal{T}(s,a)} V^\pi(s').$$

□

B.2 Proof of Proposition 2

Proof. Recall that

$$V^\pi(s) \;=\; \mathbb{E}_{\tau \sim \pi}\Big[\min_{p \in P(\tau)} \sum_{t=0}^{T} \gamma^t\, r\big(s_t\big) \;\Big|\; s_0 = s\Big].$$

When we are in state s, the next action a is sampled from the policy $\pi(\cdot \mid s)$. Because the environment is deterministic, taking (s,a) leads to the set of next states $\mathcal{T}(s,a)$. By the definition of the worst-path criterion (the inner minimum), we have

$$Q^\pi(s,a) \;=\; r(s) \;+\; \gamma\,(1 - r(s)) \min_{s' \in \mathcal{T}(s,a)} V^\pi(s'),$$

and

$$V^\pi(s) \;=\; \sum_{a \in \mathcal{A}} \pi(a \mid s)\, Q^\pi(s,a).$$

Substitute $Q^\pi(s,a)$ into the sum:

$$V^\pi(s) \;=\; \sum_{a \in \mathcal{A}} \pi(a \mid s)\Big[r(s) \;+\; \gamma\,(1 - r(s)) \min_{s' \in \mathcal{T}(s,a)} V^\pi(s')\Big].$$

Since $\sum_{a \in \mathcal{A}} \pi(a \mid s) = 1$, we can factor out $r(s)$ to obtain

$$V^\pi(s) \;=\; r(s) \underbrace{\sum_{a \in \mathcal{A}} \pi(a \mid s)}_{=1} + \gamma\,(1 - r(s)) \sum_{a \in \mathcal{A}} \pi(a \mid s) \min_{s' \in \mathcal{T}(s,a)} V^\pi(s').$$

Hence,

$$V^\pi(s) \;=\; r(s) \;+\; \gamma\,\big(1 - r(s)\big) \sum_{a \in \mathcal{A}} \pi(a \mid s) \min_{s' \in \mathcal{T}(s,a)} V^\pi(s'),$$

completing the proof. □

B.3 Proof of Proposition 3

We consider the space $\mathcal{V}$ of all bounded real-valued functions $V : \mathcal{S} \to \mathbb{R}$ defined over the state space $\mathcal{S}$. For any function $V \in \mathcal{V}$, its L_∞-norm (or max-norm) is given by $\|V\|_\infty = \sup_{s\in\mathcal{S}} |V(s)|$. The space $(\mathcal{V}, \|\cdot\|_\infty)$ is a complete metric space (a Banach space).

The Bellman optimality operator $B^* : \mathcal{V} \to \mathcal{V}$ for the worst-path objective, with discount factor $\gamma \in (0, 1)$, is defined for any value function $V \in \mathcal{V}$ and any state $s \in \mathcal{S}$ as:

$$(B^*V)(s) = r(s) + \gamma(1 - r(s)) \max_{a\in\mathcal{A}} \left[\min_{s'\in\mathcal{T}(s,a)} V(s') \right]. \tag{18}$$

Note that if $s \in \mathcal{S}_{bb}$, then $r(s) = 1$, so $1 - r(s) = 0$, and $(B^*V)(s) = r(s)$. If $s \notin \mathcal{S}_{bb}$, then $r(s) = 0$, so $1 - r(s) = 1$, and $(B^*V)(s) = \gamma \max_{a\in\mathcal{A}} \left[\min_{s'\in\mathcal{T}(s,a)} V(s')\right]$. The optimal value function V^* is the unique fixed point of this operator, i.e., $V^* = B^*V^*$.

Proposition 5. *The Bellman optimality operator B^* defined in Eq.* (18) *is a γ-contraction mapping with respect to the L_∞-norm. That is, for any $V_1, V_2 \in \mathcal{V}$:*

$$\|B^*V_1 - B^*V_2\|_\infty \le \gamma\|V_1 - V_2\|_\infty.$$

Proof. Let $V_1, V_2 \in \mathcal{V}$ be two arbitrary bounded value functions. We consider any state $s \in \mathcal{S}$.

Case 1: $s \in \mathcal{S}_{bb}$. In this scenario, $(B^*V_1)(s) = r(s)$ and $(B^*V_2)(s) = r(s)$, as the second term in Eq. (18) (involving the maximisation) vanishes because $1 - r(s) = 0$. Thus, $|(B^*V_1)(s) - (B^*V_2)(s)| = 0$. The contraction inequality $0 \le \gamma\|V_1 - V_2\|_\infty$ therefore holds trivially.

Case 2: $s \notin \mathcal{S}_{bb}$. In this case, $r(s) = 0$, so $1 - r(s) = 1$. Then,

$$(B^*V_1)(s) = \gamma \max_{a\in\mathcal{A}} \left[\min_{s'\in\mathcal{T}(s,a)} V_1(s') \right],$$

$$(B^*V_2)(s) = \gamma \max_{a\in\mathcal{A}} \left[\min_{s'\in\mathcal{T}(s,a)} V_2(s') \right].$$

Therefore,

$$|(B^*V_1)(s) - (B^*V_2)(s)| = \gamma \left| \max_{a\in\mathcal{A}} \left[\min_{s'\in\mathcal{T}(s,a)} V_1(s') \right] - \max_{a\in\mathcal{A}} \left[\min_{s'\in\mathcal{T}(s,a)} V_2(s') \right] \right|.$$

Let $f_V(a) = \min_{s'\in\mathcal{T}(s,a)} V(s')$. The expression is $\gamma|\max_a f_{V_1}(a) - \max_a f_{V_2}(a)|$. Using the property that for any functions g_1, g_2, $|\max_x g_1(x) - \max_x g_2(x)| \le \sup_x |g_1(x) - g_2(x)|$, we have:

$$\left| \max_{a\in\mathcal{A}} f_{V_1}(a) - \max_{a\in\mathcal{A}} f_{V_2}(a) \right| \le \sup_{a\in\mathcal{A}} |f_{V_1}(a) - f_{V_2}(a)|$$

$$= \sup_{a\in\mathcal{A}} \left| \min_{s'\in\mathcal{T}(s,a)} V_1(s') - \min_{s'\in\mathcal{T}(s,a)} V_2(s') \right|.$$

Using the property that for any functions h_1, h_2, $|\min_y h_1(y) - \min_y h_2(y)| \le \sup_y |h_1(y) - h_2(y)|$:

$$\left| \min_{s'\in\mathcal{T}(s,a)} V_1(s') - \min_{s'\in\mathcal{T}(s,a)} V_2(s') \right| \le \sup_{s'\in\mathcal{T}(s,a)} |V_1(s') - V_2(s')|.$$

Combining these,

$$\begin{aligned} |(B^*V_1)(s) - (B^*V_2)(s)| &\le \gamma \sup_{a\in\mathcal{A}} \left[\sup_{s'\in\mathcal{T}(s,a)} |V_1(s') - V_2(s')| \right] \\ &\le \gamma \sup_{s''\in\mathcal{S}} |V_1(s'') - V_2(s'')| \quad (\text{as } \mathcal{T}(s,a) \subseteq \mathcal{S} \text{ and we take sup over } a) \\ &= \gamma\|V_1 - V_2\|_\infty. \end{aligned}$$

This inequality holds for any $s \notin \mathcal{S}_{bb}$.

Combining Case 1 and Case 2, for all $s \in \mathcal{S}$:

$$|(B^*V_1)(s) - (B^*V_2)(s)| \le \gamma\|V_1 - V_2\|_\infty.$$

Taking the supremum over all $s \in \mathcal{S}$ on the left side:

$$\|B^*V_1 - B^*V_2\|_\infty = \sup_{s\in\mathcal{S}} |(B^*V_1)(s) - (B^*V_2)(s)| \leq \gamma \|V_1 - V_2\|_\infty.$$

Since $0 < \gamma < 1$, B^* is a γ-contraction mapping in $(\mathcal{V}, \|\cdot\|_\infty)$. The space $\mathcal{V}$ of bounded real-valued functions on $\mathcal{S}$, equipped with the L_∞-norm, is a complete metric space (a Banach space). Therefore, by the Banach fixed-point theorem, B^* has a unique fixed point $V^* \in \mathcal{V}$ such that $V^* = B^*V^*$. Furthermore, for any initial bounded value function $V_0 \in \mathcal{V}$, the sequence $V_{k+1} = B^*V_k$ (i.e., value iteration) converges to V^*. The existence of a unique V^* implies the existence of at least one stationary deterministic optimal policy π^* such that for $s \notin \mathcal{S}_{bb}$:

$$\pi^*(s) \in \arg\max_{a\in\mathcal{A}} \left[\min_{s'\in\mathcal{T}(s,a)} V^*(s') \right].$$

□

B.4 Proof of Proposition 4

Proof. We begin by recalling a standard identity from policy improvement theory, which relates the value difference between two policies π^{i+1} and π^i to the expected advantage under π^{i+1}:

$$V^{\pi^{i+1}}(s) - V^{\pi^i}(s) = \mathbb{E}_{a\sim\pi^{i+1}(\cdot|s)} \left[A^{\pi^i}(s,a) \right],$$

where $A^{\pi^i}(s,a) = Q^{\pi^i}(s,a) - V^{\pi^i}(s)$ is the advantage function under the behaviour policy π^i.

The update rule defined by the weighted imitation objective yields a new policy π^{i+1} that is proportional to the exponentiated advantage:

$$\pi^{i+1}(a \mid s) = \frac{\pi^i(a \mid s) \cdot \exp(\beta A^{\pi^i}(s,a))}{Z(s)},$$

where $Z(s) = \sum_{a'} \pi^i(a' \mid s) \cdot \exp(\beta A^{\pi^i}(s,a'))$ is a normalising constant. This update ensures that π^{i+1} remains in the same support set as π^i.

Substituting into the value difference identity, we get:

$$V^{\pi^{i+1}}(s) - V^{\pi^i}(s) = \sum_a \pi^{i+1}(a \mid s) \cdot A^{\pi^i}(s,a) = \frac{\sum_a \pi^i(a \mid s) \cdot \exp(\beta A^{\pi^i}(s,a)) \cdot A^{\pi^i}(s,a)}{\sum_a \pi^i(a \mid s) \cdot \exp(\beta A^{\pi^i}(s,a))}.$$

This expression is a weighted average of the advantages $A^{\pi^i}(s,a)$, where the weights are strictly positive and increasing in the advantage. Therefore, unless all advantages are exactly zero, the expectation is strictly non-negative:

$$V^{\pi^{i+1}}(s) \geq V^{\pi^i}(s), \quad \forall s \in \mathcal{S}.$$

□

C Graph2Edits as Single-step Model

Graph2Edits [38] considers one-step retrosynthesis as a graph-editing game. Starting from the product molecule, the model autoregressively predicts a short sequence of primitive edits — Delete Bond, Change Bond, Change Atom, Attach Leaving Group, and finally a Terminate token. At each step the current intermediate graph is embedded by a directed message-passing neural network (D-MPNN), whose atom- and bond-level embeddings are fed to three linear heads that score every possible bond edit, atom edit and the termination symbol. The highest-scoring edit is applied to yield the next intermediate, and the process repeats until Terminate is chosen, at which point the intermediates have been fully converted to the predicted reactants. This edit vocabulary, mined from training data, covers 99.9% of USPTO-50k reactions and encodes stereochemistry as well as functional leaving groups.

Trained with teacher forcing, the system already achieves a 55% top-1 exact-match accuracy on USPTO-50k and maintains high validity and diversity even for long or stereochemically rich reactions; these strengths make it an ideal single-step model inside our multi-step InterRetro framework.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state four contributions — worst-path optimisation framework, a self-imitation algorithm with monotonic improvement guarantees, a search-free retrosynthesis approach, and SOTA empirical results — and every one of these is delivered and substantiated in the main text and experiments.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 explicitly notes that InterRetro and contemporary works assume the suggested reactions are experimentally feasible. We recognise this may be violated, and recommend adding feasibility filters in future work.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All propositions are stated with assumptions and conditions in the main text, and formal proofs are provided in Appendix B.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper gives dataset names/sizes, model-call budgets, hyper-parameters, and a code repository link that contains the full training and evaluation code.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is provided and all datasets used (USPTO-50k, eMolecules, Retro*-190, ChEMBL-1000, GDB17-1000) are public.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimiser, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The core details are provided in the main text and the full details can be found in the attached code.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We quantify the impact of stochasticity by repeating every experiment with multiple random seeds. Figure 4c reports the mean performance (solid line) together with a shaded band that marks the 95% confidence interval computed across 4 different seeds.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In the experiments, we state that InterRetro was trained over two days using six parallel processes on an NVIDIA RTX A5000 GPU.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses only public reaction datasets and releases code under an anonymised repository during review phase; no personal data or sensitive content is involved.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The broad impact has been discussed in the main text.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The dataset and the code from others are properly cited. The license and terms of use explicitly mentioned in the code and properly respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The newly released InterRetro codebase is documented in the repository, including usage instructions and dependency list.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.