ESCALATE ONLY WHEN NECESSARY: MARGINAL-VALUE-AWARE FEEDBACK ACQUISITION FOR RETROSYNTHETIC PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrosynthetic planning decomposes a target molecule into building blocks, and recent LLM-based planners enrich this search with external feedback. Such feedback, however, varies substantially in fidelity and acquisition cost, creating a trade-off between decision reliability and resource consumption. We formulate feedback acquisition as *adaptive escalation*, where a planner progressively acquires higher-fidelity feedback only when its expected benefit justifies the additional cost. To learn such policies, we propose **Marginal-Value-Aware Advantage Estimation (MAVE)**. MAVE models the reward-cost relationship across successive escalation levels, using its slope to estimate the current marginal value and its curvature to capture how this value evolves with further resource investment. MAVE incorporates both signals into advantage estimation, enabling marginal-value-aware credit assignment for *just-enough* escalation policy while discouraging premature stopping and over-escalation. Across three bencharks, MAVE achieves the highest success rates with hierarchical feedback, while reducing average feedback cost by 64.9% under the yield-only setting. Code is available at `https://anonymous.4open.science/r/MAVE-retro/`.

1 INTRODUCTION

Retrosynthetic planning traces a target molecule back to simpler, commercially available precursors through successive chemical transformations. Deep learning propels computer-aided synthesis planning (CASP) towards scalable multi-step search Segler et al. (2018). Recent LLM-based planners augment retrosynthetic reasoning with human feedback and specialized tools, acquiring additional evidence and improve planning reliability Bran et al. (2023); Sathyanarayana et al. (2025).

The feedback sources available in retrosynthetic planning, however, differ substantially in fidelity and acquisition cost, creating an inherent trade-off between decision reliability and resource consumption. As illustrated in Fig. 1a, higher-fidelity feedback, such as wet-lab validation, can provide high-confidence evidence of reaction feasibility but may be unnecessary once existing information suffices for subsequent decisions. Conversely, lower-cost feedback, such as automated literature search, enables rapid assessment but may offer limited guidance under high uncertainty. This trade-off motivates **adaptive escalation**, in which a planner progressively acquires higher-fidelity feedback only when accumulated evidence remains inadequate and further acquisition is worthwhile. A familiar analogue is clinical diagnosis Gorry & Barnett (1968), where clinicians escalate from routine examinations to costly imaging or invasive tests as diagnostic uncertainty persists. Whereas one-shot routing commits to an acquisition choice upfront Ong et al. (2025), adaptive escalation trades additional lightweight queries for a lower risk of invoking burdensome feedback unnecessarily. Existing retrosynthetic methods either model query costs with single feedback source Yuan et al. (2024), or incorporate multiple tools and human feedback without accounting for heterogeneous acquisition costs Liu et al. (2025); Sathyanarayana et al. (2025); Baker et al. (2025). Adaptive escalation remains largely unexplored in retrosynthetic planning, leaving open how to learn policies that weigh the expected benefit of further feedback against its acquisition cost.

To address this challenge, we propose **M**arginal-**V**alue-**A**ware Advantage **E**stimation (**MAVE**), a method for learning adaptive escalation policies. Because escalation is incremental, the decision to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

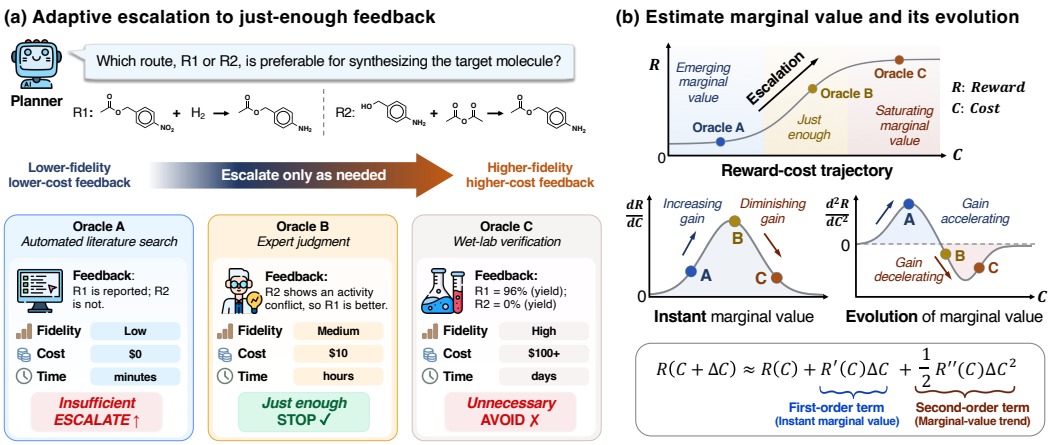

Figure 1: **Adaptive escalation and marginal-value-aware learning.** (a) The planner starts with lower-cost feedback and escalates only when existing evidence is insufficient, stopping once enough evidence is obtained. (b) Reward and acquisition cost form a reward-cost trajectory across escalation levels. Its slope captures current marginal value, while curvature indicates whether this value is increasing or diminishing, guiding just-enough escalation policy.

escalate depends not only on the next source's benefit relative to its cost, but on how that benefit evolves with further resource investment. As shown in Fig. 1b, task rewards and acquisition costs trace out a reward-cost trajectory across successive escalation levels, whose slope captures the current marginal value and curvature indicates whether this value is increasing or diminishing. MAVE folds both signals into advantage estimation, distinguishing emerging reward gains that warrant further escalation from diminishing gains that indicate approaching saturation. Beyond uncertainty- or utility-based assessments of whether additional resources are worthwhile at a given stage Gupta et al. (2024); Damani et al. (2025); Hashemi et al. (2026); Ong et al. (2025), MAVE accounts for the potential benefits of continued acquisition. This guides the policy toward *just-enough* escalation, avoiding premature stopping that sacrifices reliability and over-escalation that wastes resources.

Our contributions are threefold. (1) *Adaptive escalation for retrosynthesis:* We formulate feedback acquisition as sequential stop-or-escalate descisions over a hierarchy of sources with heterogeneous fidelity and cost, invoking stronger feedback only when needed. (2) *Marginal-value-aware policy learning:* We propose **MAVE**, which models how marginal value evolves with further resource investment and incorporates it into advantage estimation for adaptive escalation. (3) *Empirical evaluation:* Across three benchmarks, MAVE attains success rates of 92.6%, 86.8%, and 84.4% under hierarchical feedback, while reducing average feedback cost by 64.9% in the yield-only setting.

2 Related Work

Feedback-augmented retrosynthetic planning. Recent work has advanced retrosynthetic planning through LLM reasoning Wang et al. (2025); Li et al. (2025); Zhang et al. (2025); Song et al. (2025); Wang et al. (2026). Beyond internal reasoning, planners exploit the interactive capabilities of LLMs to solicit external support. ChemCrow, LARC, and Retro-R1 augment planning with chemistry tools, constraint evaluation, and expert feedback, respectively Bran et al. (2023); Baker et al. (2025); Liu et al. (2025). RetroAgent further couples chemistry tools with structured search memory to reason over intermediate states Zhu et al. (2026). Synthegy incorporates chemist-specified synthesis strategies Bran et al. (2025), while MBRPS retrieves reaction evidence through human judgments Ma et al. (2025). Despite growing use of external support, few studies account for the cost of acquiring feedback. DeepRetro reports LLM evaluation cost Sathyanarayana et al. (2025), while ARP learns whether to query reaction qualities under fixed cost Yuan et al. (2024). Yet heterogeneous feedback with distinct fidelity and acquisition cost remains largely unexplored.

Adaptive escalation and cost-sensitive acquisition. Adaptive escalation is closely related to sequential cost-sensitive acquisition, where a policy decides whether current information is sufficient or additional resources should be acquired. Classical cascades and active feature acquisition

Table 1: Comparison of representative feedback-augmented retrosynthetic planning methods.

	ChemCrow	Synthegy	LARC	RetroR1	RetroAgent	DeepRetro	ARP	**Ours**
LLM-mediated	✓	✓	✓	✓	✓	✓		✓
Active learning							✓	✓
Cost-aware						✓	✓	✓
Multi-fidelity	✓				✓			✓
Adaptive escalation								✓

Table 2: Escalation hierarchy of external feedback, with multiple options available at each level.

Level	**Type**	**Information provided**
L_0	None	**No query:** No external feedback is acquired.
L_1	Structural	**Reaction class:** Suggests a reaction class for the current product. **Reaction center:** Identifies atoms or bonds associated with a plausible disconnection. **Activity assessment:** Assesses local functional-group reactivity and compatibility. **Bond disconnection:** Suggests a plausible bond for disconnection.
L_2	Evaluative	**Reaction yield:** Returns a normalized estimate of reaction outcome quality. **Reaction feasibility:** Returns whether a candidate reaction is feasible. **Selectivity assessment:** Evaluates the expected selectivity of a candidate reaction.
L_3	Comparative	**Reaction comparison:** Compares candidate reactions and returns a preference. **Route comparison:** Compares partial routes and returns a preference.
L_4	Strategic	**Multi-step strategy:** Provides high-level guidance over multiple synthesis steps. **Goal suggestion:** Proposes intermediate subgoals or purchasable building blocks.

learn instance-dependent stop/acquire policies that balance predictive quality against the cost of further observations Trapeznikov et al. (2012); Shim et al. (2018); Valancius et al. (2024); Norcliffe et al. (2025), while multi-fidelity learning selects among observations of different fidelity and cost Takeno et al. (2020); Wu et al. (2023). More recently, model cascades and learning-to-defer extend cost-sensitive acquisition to stronger models or human experts Mozannar & Sontag (2020); Gupta et al. (2024); Chang et al. (2026), while cost-aware retrieval learns adaptive information acquisition through resource-aware advantage estimation Hashemi et al. (2026).

3 Preliminaries

3.1 Multi-step retrosynthetic planning

Multi-step retrosynthetic planning is formulated as a finite-horizon Markov decision process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$. At step t, the state $s_t \in \mathcal{S}$ contains the current set of unsolved molecules. A reaction action $a_t \in \mathcal{A}$ is selected from candidates proposed by a single-step retrosynthesis model. The deterministic transition $s_{t+1} = \mathcal{P}(s_t, a_t)$ replaces the selected product with its predicted reactants. A trajectory succeeds when all remaining molecules are purchasable building blocks. The policy π maximizes the task reward $\sum_{t=0}^{H} \mathcal{R}(s_t, a_t)$ over a planning horizon H.

3.2 Feedback-augmented retrosynthetic planning

Feedback-augmented planning allows the planner to acquire external evidence before committing to a reaction action. The standard MDP is extended as $\langle \mathcal{S}, \mathcal{A}, \mathcal{Q}, \mathcal{P}, \mathcal{R}, \mathcal{O}, \mathcal{C} \rangle$, where $\mathcal{Q}$ denotes the query space, $\mathcal{O}$ the feedback oracle, and $\mathcal{C}$ the acquisition cost function. At state s_t, the planner submits a query $q_t \in \mathcal{Q}$ and receives feedback $f_t \sim \mathcal{O}(q_t)$ from the oracle. Each query incurs an acquisition cost $c_t = \mathcal{C}(q_t)$. The planner then determines the reaction action a_t conditioned on both s_t and the acquired feedback f_t. Over a trajectory, the planner maximizes cumulative task reward while accounting for feedback cost, as in Eq. 1, where λ controls the trade-off between planning performance and feedback cost.

$$a_t \sim \pi\big(\cdot \mid s_t, f_t\big), \quad \max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=1}^{H} \mathcal{R}(s_t, a_t) - \lambda \sum_{t=1}^{H} c_t\right]. \tag{1}$$

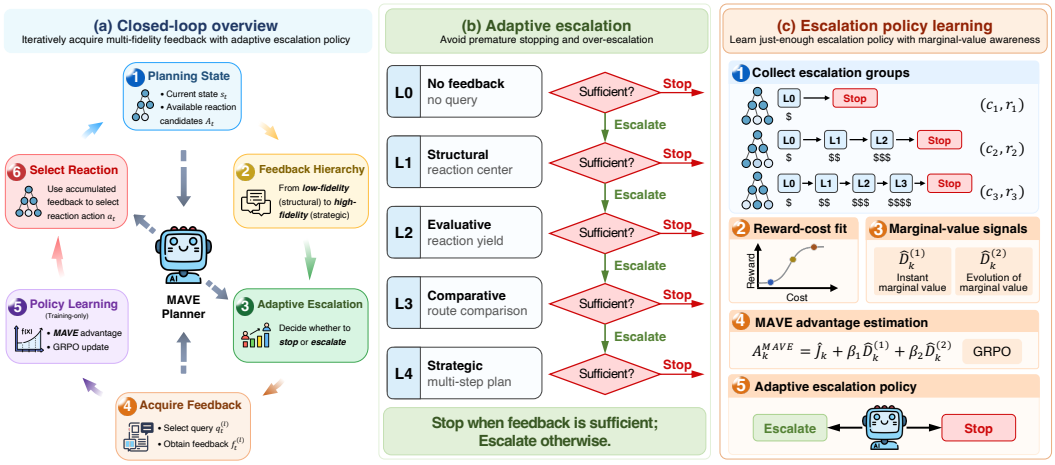

Figure 2: **Overview of adaptive feedback escalation with MAVE.** (a) Closed-loop retrosynthetic planning adaptively acquires feedback before selecting each reaction. (b) The planner traverses an ordered feedback hierarchy, stopping when the accumulated evidence is sufficient and escalating otherwise. (c) MAVE learns the escalation policy from collected escalation trajectories by fitting the reward-cost relation and estimating first- and second-order marginal-value signals, which are incorporated into advantage estimation to support just-enough escalation.

3.3 Adaptive Feedback Escalation

As illustrated in Fig. 2b, adaptive feedback escalation introduces a sequential acquisition process within each planning step. Feedback sources are organized into ordered levels $\ell \in \{0, \dots, L\}$, where $\ell = 0$ denotes no feedback and higher levels generally provide more informative feedback at greater acquisition cost. Each level may contain multiple feedback options, partitioning the query space as $\mathcal{Q} = \bigcup_{\ell=1}^{L} \mathcal{Q}^{(\ell)}$, where $\mathcal{Q}^{(\ell)}$ denotes the options available at level ℓ. The feedback hierarchy used in our experiments is summarized in Tab. 2, with the information provided at each feedback level detailed in Appendix A.1. When level ℓ is accessed at planning step t, the planner selects $q_t^{(\ell)} \in \mathcal{Q}^{(\ell)}$ and receives the associated feedback $f_t^{(\ell)} \sim \mathcal{O}(q_t^{(\ell)})$ and cost $c_t^{(\ell)} = \mathcal{C}(q_t^{(\ell)})$. After escalation reaches level ℓ, feedback and acquisition cost accumulate as

$$f_t^{(\le \ell)} = \left(f_t^{(1)}, \dots, f_t^{(\ell)}\right), \qquad c_t^{(\le \ell)} = \sum_{j=1}^{\ell} c_t^{(j)}, \tag{2}$$

Starting from $\ell = 0$, the escalation policy μ jointly decides whether to stop or escalate and, upon escalation, which feedback option to acquire at the next level. Because each level may contain multiple feedback options, an escalation action is represented directly a query from $\mathcal{Q}^{(\ell+1)}$:

$$e_t^{(\ell)} \sim \mu\left(\cdot \mid s_t,\, f_t^{(\le \ell)}\right), \qquad e_t^{(\ell)} \in \{\textsc{Stop}\} \cup \left\{\textsc{Escalate}(q) \;\middle|\; q \in \mathcal{Q}_{\ell+1}\right\}. \tag{3}$$

If $e_t^{(\ell)} = \textsc{Stop}$ or the maximum level L is reached, the planner commits to a reaction action according to Eq. 1, with accumulated feedback $f_t = f_t^{(\le \ell)}$ and acquisition cost $c_t = c_t^{(\le \ell)}$. Otherwise, if $e_t^{(\ell)} = \textsc{Escalate}(q_t^{(\ell+1)})$, the planner issues $q_t^{(\ell+1)}$, acquires the corresponding feedback, and advances to level $\ell + 1$. The resulting problem is therefore not a one-shot selection of a feedback source, but a sequence of stop-or-escalate decisions under progressively accumulated evidence.

4 Method

We introduce *Marginal-Value-Aware Advantage Estimation* (**MAVE**) for learning adaptive feedback escalation policies. Fig. 2a illustrates the overall closed-loop planning framework, in which feedback is adaptively acquired before each retrosynthetic reaction decision. MAVE models the local reward-cost relationship and derives first- and second-order marginal-value signals (Sec. 4.1). These signals reshape advantage estimation to provide marginal-value-aware credit assignment (Sec. 4.2). We then use group-sampled roll-outs and GRPO to learn a just-enough escalation policy (Sec. 4.3).

Algorithm 1: Group-sampled adaptive escalation roll-out.

Require: planning state s_t; reaction space $\mathcal{A}_t$; escalation policy μ; reaction policy π; group size K.
for $k = 1$ **to** K **do**
  Initialize $\ell \leftarrow 0$, $f_k \leftarrow \varnothing$, $c_k \leftarrow 0$
  while $\ell < L$ **do**
    Sample escalation action $e_k^{(\ell)} \sim \mu(\cdot \mid s_t, f_k)$
    if $e_k^{(\ell)} = \textsc{Stop}$ **then**
      break ; `// current evidence is sufficient`
    $e_k^{(\ell)} = \textsc{Escalate}(q_k^{(\ell+1)})$
    Acquire $f_k^{(\ell+1)} \sim \mathcal{O}(q_k^{(\ell+1)})$; `// obtain stronger feedback`
    $f_k \leftarrow (f_k, f_k^{(\ell+1)})$; `// accumulate acquired evidence`
    $c_k \leftarrow c_k + \mathcal{C}(q_k^{(\ell+1)})$; `// accumulate acquisition cost`
    $\ell \leftarrow \ell + 1$
  Sample reaction action $a_k \sim \pi(\cdot \mid s_t, f_k)$
  Continue planning to termination and obtain r_k
return group $\{(c_k, r_k)\}_{k=1}^K$;

4.1 Reward-cost trajectory approximation

The marginal value of feedback depends on the current planning context. We therefore estimate the reward-cost relation from a group of K complete escalation roll-outs sampled from the same context. After each roll-out terminates, we obtain its downstream return r_k and the feedback acquisition cost c_k accumulated during the current escalation process. These roll-outs yield a set of reward-cost observations $\{(c_k, r_k)\}_{k=1}^K$, characterizing how downstream planning performance varies with feedback expenditure under the current context. We fit these observations with a smooth mapping $\phi(c;\theta)$, where ϕ specifies the curve form and θ parameterizes the mapping. Since only a few observations are available for each context, we fit them with a simple low-dimensional parametric mapping $\phi(c;\theta)$. We impose a non-decreasing, saturating structure, reflecting that additional feedback may improve downstream planning performance while its marginal benefit eventually diminishes. In the main experiments, we instantiate ϕ with a tanh mapping,

$$\phi(c;\theta) = a + b\tanh[\kappa(c-\tau)] \quad b > 0,\ \kappa > 0. \tag{4}$$

MAVE is not tied to tanh mapping; other differentiable mappings, such as sigmoid, polynomials, or neural networks, can also be used. We illustrate and compare alternative mappings in Appendix B and Appendix D.1. The parameters are fitted using the completed roll-outs in the current group:

$$\theta^* = \arg\min_\theta \sum_{k=1}^{K} \left(r_k - \phi(c_k;\theta)\right)^2 + \eta\|\theta\|_2^2. \tag{5}$$

Marginal-value signals. As illustrated in Fig. 2c, we derive the first- and second-order marginal-value signals from the cost-aware objective in Eq. 1:

$$D_k^{(1)} = \phi'(c_k;\theta^*), \qquad D_k^{(2)} = \phi''(c_k;\theta^*). \tag{6}$$

$D_k^{(1)}$ measures the instantaneous marginal value of further escalation, while $D_k^{(2)}$ captures its evolution, indicating whether the marginal value is increasing or diminishing. They jointly distinguish emerging reward gains from approaching saturation.

4.2 Marginal-value-aware advantage estimation

For each escalation roll-out group sample, we first compute its realized cost-aware outcome

$$J_k = r_k - \lambda c_k. \tag{7}$$

While J_k evaluates the realized trade-off between downstream return and feedback cost, it does not reveal whether further escalation remains valuable. MAVE therefore augments this outcome with the marginal-value signals $D_k^{(1)}$ and $D_k^{(2)}$. We then normalize them within each roll-out group:

$$\widehat{X}_k = \frac{X_k - \mathrm{mean}(X)}{\mathrm{std}(X) + \epsilon}, \qquad X \in \{J, D^{(1)}, D^{(2)}\}, \tag{8}$$

where ϵ is a small constant for numerical stability. The MAVE advantage is then defined as

$$A_k^{\mathrm{MAVE}} = \widehat{J}_k + \beta_1 \widehat{D}_k^{(1)} + \beta_2 \widehat{D}_k^{(2)}, \tag{9}$$

where β_1 and β_2 control the contributions of the first- and second-order marginal-value signals. The three terms provide complementary credit: $\widehat{J}_k$ reflects the realized reward-cost outcome, $\widehat{D}_k^{(1)}$ captures the instantaneous value of further escalation, and $\widehat{D}_k^{(2)}$ indicates whether this value is increasing or diminishing. This encourages escalation when further investment is still beneficial and discourages over-escalation when additional feedback yields diminishing returns.

4.3 Training and inference

We alternate between optimizing the escalation policy μ and the reaction policy π, while keeping the other policy fixed. Both share the same backbone with task-specific prompts.

Escalation policy. When optimizing μ, we sample a group of K complete escalation roll-outs from each planning context under the fixed reaction policy π. After escalation terminates, each branch continues through downstream retrosynthetic planning to obtain its return r_k, together with the feedback acquisition cost c_k incurred during the current escalation process. The resulting reward-cost observations $\{(c_k, r_k)\}_{k=1}^K$ are used to construct the MAVE advantages following Eqs. 9. We then optimize μ with group relative policy optimization (GRPO), replacing the conventional group-relative outcome advantage with A_k^{MAVE}. The rollout-level advantage is assigned to all escalation decisions within the corresponding roll-out.

Reaction policy. When optimizing π, the escalation policy μ is fixed. We collect retrosynthetic planning roll-outs and optimize π with standard GRPO using downstream task returns. Given a planning state and its acquired feedback, π learns a distribution over candidate reactions.

Escalation-guided inference. We emphasize no reward-cost fitting or marginal-value estimation is required at inference. At inference time, MAVE is integrated into a standard MCTS retrosynthetic planner Hong et al. (2021); Yuan et al. (2024). For each frontier molecule selected by MCTS, the escalation policy μ repeatedly decides whether to STOP or acquire higher-level feedback. After escalation terminates, the reaction policy π proposes candidate reactions conditioned on the accumulated feedback, and their policy probabilities are used as MCTS priors. The details about training and inference is provided in Appendix B.

5 Experiments

5.1 Experimental setup

Benchmarks. We conduct experiments on three benchmarks with varying levels of difficulty. USPTO-190 was introduced by Chen et al. (2020), which covers 190 hard targets and has been widely adopted in prior retrosynthetic studies. Wuxi-expert-873 is a proprietary benchmark introduced from Yuan et al. (2024), comprising 873 test targets with referential retrosynthetic routes extracted by experts from the literature. Additionally, we release a new benchmark, PDBbind-160, constructed from the open-source PDBbind dataset, whose targets exhibit high functional-group complexity along with potentially competitive reaction sites. Overall, the retrosynthetic difficulty increases from USPTO-190 to Wuxi-expert-873 to PDBbind-160.

Retrosynthetic planner baselines. We compare MAVE with three groups of retrosynthetic planners. (i) *Specialized planners*. **Retro*+** extends the neural-guided A*-like search of Retro* Chen et al. (2020) with self-improvement on successful planning trajectories Kim et al. (2021). **EG-MCTS** Hong et al. (2023) replaces online MCTS roll-outs with an experience-guidance network learned from prior search experience. **GRASP** Yu et al. (2022) formulates retrosynthetic planning as a goal-driven actor-critic problem and uses learned policy and value functions to guide MCTS. (ii) *LLM-mediated planners*. **Llama-3-8B-Instruct** Grattafiori et al. (2024) serves as the general-purpose LLM baseline, while **MolInst-Molecule-8B** Fang et al. (2024) is its chemistry-oriented counterpart instruction-tuned on Mol-Instructions. **LLM-Syn-Planner** uses an off-the-shelf LLM with route-level search and structured evaluation. (iii) *Feedback-augmented planners*. **DeepRetro** Sathyanarayana et al. (2025) combines LLM reasoning, retrosynthetic engines, and iterative tool or expert feedback. **ARP** Yuan et al. (2024) selectively queries external reaction-quality

Table 3: Comparison with retrosynthetic baselines across three benchmarks. Results are averaged per sample: Success rate ($[0, 1]$), Route quality ($[0, 1]$), Query rate ($[0, 1]$), and Query cost. Best success rates and route qualities are **bolded**, while the top-3 results are underlined.

Method	Feedback setting	USPTO-190				Wuxi-expert-873				PDBbind-160			
		Success rate ↑	Route quality ↑	Query rate ↓	Query cost ↓	Success rate ↑	Route quality ↑	Query rate ↓	Query cost ↓	Success rate ↑	Route quality ↑	Query rate ↓	Query cost ↓
Non-LLM methods													
Retro*+	Yield only	72.1%	0.671	100%	106.2	68.5%	0.680	100%	102.3	57.5%	0.652	100%	102.9
EG-MCTS	Yield only	85.3%	0.558	100%	104.1	75.4%	0.612	100%	110.5	62.5%	0.599	100%	107.9
GRASP	Yield only	84.7%	0.602	100%	101.5	79.5%	0.665	100%	109.4	83.1%	0.610	100%	111.0
LLM-mediated methods													
Llama-3-8B-Instruct	None	25.3%	**0.735**	0.0%	0.0	19.1%	0.700	0.0%	0.0	15.0%	0.685	0.0%	0.0
MolInst-Molecule-8B	None	52.1%	0.641	0.0%	0.0	34.9%	0.756	0.0%	0.0	33.1%	0.655	0.0%	0.0
LLM-Syn-Planner	None	60.5%	0.610	0.0%	0.0	55.0%	0.635	0.0%	0.0	42.5%	0.578	0.0%	0.0
Feedback-augmented methods													
DeepRetro	None	78.9%	0.640	0.0%	0.0	70.2%	0.706	0.0%	0.0	55.0%	0.594	0.0%	0.0
DeepRetro	Yield only	77.9%	0.686	100%	104.4	69.0%	0.787	100%	103.2	55.6%	0.632	100%	109.9
ARP (GRASP)	None	85.3%	0.654	0.0%	0.0	74.2%	0.792	0.0%	0.0	68.1%	0.619	0.0%	0.0
ARP (GRASP)	Yield only	86.8%	0.711	86.4%	93.2	80.4%	**0.826**	91.0%	100.4	75.6%	0.650	96.0%	99.2
MAVE (ours)	None	89.5%	0.680	0.0%	0.0	78.2%	0.734	0.0%	0.0	76.3%	0.666	0.0%	0.0
MAVE (ours)	Yield only	86.3%	0.710	34.2%	34.3	80.1%	0.808	41.0%	41.8	73.1%	0.695	25.6%	26.8
MAVE (ours)	Hierarchy	**92.6%**	0.699	43.4%	70.4	**86.8%**	0.806	55.7%	89.6	**84.4%**	**0.710**	58.8%	95.1

Table 4: Algorithm comparison across three benchmarks under the *Hierarchy* setting. All methods share the same planner and feedback hierarchy. One-shot depth routing fixes the acquisition depth upfront, whereas sequential methods adaptively decide whether to continue after each level. Best success rates and route qualities are **bolded**, with the top-2 underlined.

Algorithm	Feedback setting	USPTO-190				Wuxi-expert-873				PDBbind-160			
		Success rate ↑	Route quality ↑	Query rate ↓	Query cost ↓	Success rate ↑	Route quality ↑	Query rate ↓	Query cost ↓	Success rate ↑	Route quality ↑	Query rate ↓	Query cost ↓
One-shot depth routing													
A^2FM	Hierarchy	86.8%	0.672	43.7%	47.8	80.3%	0.808	56.4%	63.5	79.4%	0.640	59.2%	68.7
RouteLLM	Hierarchy	91.6%	0.684	52.3%	108.0	83.7%	0.748	64.6%	142.1	81.9%	0.625	72.1%	154.3
HybridLLM	Hierarchy	**92.6%**	0.640	84.8%	126.5	84.5%	0.766	90.6%	177.4	83.1%	0.646	67.5%	158.2
Sequential escalation													
MoT Cascade	Hierarchy	90.5%	0.696	63.8%	98.7	85.7%	**0.815**	75.0%	110.6	83.8%	0.682	58.1%	79.8
LM Cascade	Hierarchy	88.4%	0.641	31.0%	37.2	84.1%	0.787	42.0%	54.6	76.3%	0.644	52.4%	58.9
MAVE (ours)	Hierarchy	**92.6%**	**0.699**	43.4%	70.4	**86.8%**	0.806	55.7%	89.6	**84.4%**	**0.710**	58.8%	95.1

feedback to guide planning. For LLM-mediated baselines, we standardize the backbone as Llama-3-8B-Instruct whenever applicable, together with the planning interface and prompt format.

Routing and escalation baselines. To isolate the benefit of sequential escalation, both routing and escalation methods use the same ordered feedback hierarchy, detailed in Appendix C. One-shot depth routing commits upfront to a maximum depth $d \in \{0, \ldots, L\}$ and acquires all feedback up to d, whereas sequential escalation decides whether to continue after each observed level. (i) *One-shot depth routing*. **RouteLLM** Ong et al. (2025) learns a preference-based router over resources with different quality-cost profiles; **HybridLLM** Ding et al. (2024) routes queries according to predicted difficulty and target quality; and **A^2FM** Chen et al. (2025) performs cost-aware adaptive mode selection. (ii) *Sequential escalation*. **MoT Cascade** Yue et al. (2024) escalates based on answer consistency, while **LM Cascade** Gupta et al. (2024) uses uncertainty and learned deferral signals.

Feedback-augmented retrosynthetic environment. We use eMolecules as the purchasable building-block database. Methods that require an external single-step retrosynthesis model uniformly use the template-based predictor from Chen et al. (2020). For each product, the top-50 predicted reactions are retained. In the main text, we report results with a budget of at most 100 single-step model calls, providing a resource-constrained setting for evaluating feedback use. Results under larger budgets of 200, 500, and 1,000 calls are reported in the Appendix D.2. **Feedback setting** specifies the available feedback sources. *None* uses no feedback, *Yield only* uses a single reaction-yield source, *Hierarchy* use the multi-fidelity sources organized as an ordered hierarchy.

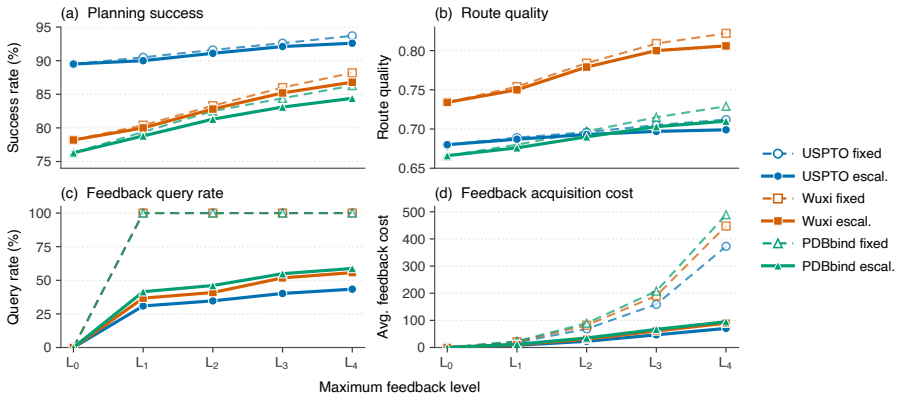

Figure 3: Effect of the maximum accessible feedback level. Fixed-depth acquisition obtains all feedback up to the prescribed level, whereas MAVE adaptively stops when sufficient evidence is available. (a) Success rate, (b) route quality, (c) query rate, and (d) average feedback cost. MAVE retains most performance gains while avoiding unnecessary high-level feedback.

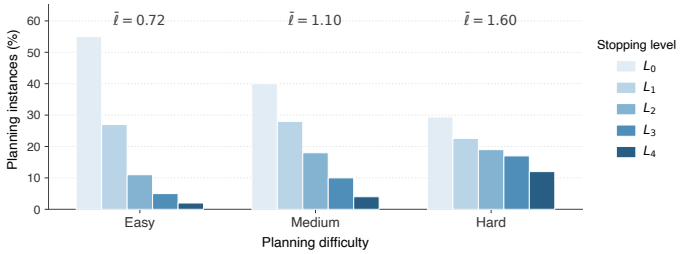

Figure 4: Percentage of planning instances stopping at each feedback level by difficulty. As difficulty increases, MAVE stops at deeper levels, raising the mean stopping depth from 0.72 to 1.60. This indicates difficulty-adaptive feedback escalation.

Evaluation metrics. Three metrics are used to quantify planning performance: (i) **Success rate** is the fraction of solved targets for which at least one retrosynthetic route is found. (ii) **Route quality** evaluates the quality of the returned route based on per-reaction quality scores, following Yuan et al. (2024), with details provided in Appendix D.4. (iii) **Query rate** is the fraction of planning iterations that invoke external feedback, and **Query cost** is the cumulative feedback acquisition cost over a planning episode, averaged across target molecules.

5.2 Overall Performance and Allocation Comparison

System-level comparison with retrosynthetic planners. Tab. 3 compares complete planning systems across non-LLM, LLM-mediated, and feedback-augmented methods. Since these methods differ in feedback access and allocation, the results reflect overall system performance rather than the allocation strategy alone, which we isolate in the controlled comparison below. With hierarchical feedback, MAVE achieves success rates of 92.6%, 86.8%, and 84.4% on USPTO-190, Wuxi-expert-873, and PDBbind-160, exceeding the strongest non-MAVE system-level results by 5.8, 6.4, and 1.3 percentage points. These gains are obtained while querying feedback on only 43.4%, 55.7%, and 58.8% of planning steps, at average costs of 70.4, 89.6, and 95.1, and route quality remains competitive across benchmarks, reaching 0.806 on Wuxi-expert-873 and the best result of 0.710 on PDBbind-160. To compare feedback efficiency under the same feedback source, we further consider the yield-only setting. Compared with ARP, MAVE reduces average feedback cost by 64.9%, indicating more selective use of the same feedback source. The cost reduction is not accompanied by a uniform success gain, reflecting the known quality-success trade-off. Yield feedback favors well-precedented high-yield disconnections, which raises route quality but can prune lower-yield steps needed to reach purchasable building blocks under a tight search budget. This motivates the hierarchy setting, where heterogeneous feedback improves both metrics simultaneously.

Table 5: Ablation of marginal-value-aware credit assignment under the *Hierarchy* setting. All variants use the same sequential escalation architecture and training setup, differing only in the advantage formulation. Best results are **bolded**; second-best results are underlined.

Advantage	USPTO-190				Wuxi-expert-873				PDBbind-160			
	Success rate ↑	Route quality ↑	Query rate ↓	Query cost ↓	Success rate ↑	Route quality ↑	Query rate ↓	Query cost ↓	Success rate ↑	Route quality ↑	Query rate ↓	Query cost ↓
J	88.9%	0.675	51.2%	79.5	82.5%	0.777	63.1%	91.2	80.0%	0.660	66.0%	108.1
$J + D^{(1)}$	90.5%	**0.702**	40.8%	73.4	84.5%	0.803	51.8%	95.3	81.3%	0.698	54.4%	98.6
$J + D^{(2)}$	90.0%	0.684	49.4%	77.5	85.1%	0.789	59.5%	93.0	83.1%	0.681	62.0%	101.1
$J + D^{(1)} + D^{(2)}$	**92.6%**	0.699	43.4%	70.4	**86.8%**	**0.806**	55.7%	89.6	**84.4%**	**0.710**	58.8%	95.1

Controlled comparison of feedback-allocation algorithms. Tab. 4 compares allocation strategies under the same planner and feedback hierarchy, with one-shot depth routers given access to the same cumulative feedback as sequential methods. MAVE achieves success rates of 92.6%, 86.8%, and 84.4% across the three benchmarks, matching the strongest baseline on USPTO-190 and outperforming the strongest non-MAVE allocation baseline by 1.1 and 0.6 percentage points on Wuxi-expert-873 and PDBbind-160, respectively. It also achieves the highest route quality on PDBbind-160 (0.710). Although MAVE does not minimize acquisition cost, it queries selectively and outperforms consistency- and uncertainty-based cascades, indicating that the gain comes from conditioning escalation on intermediate feedback rather than from cumulative feedback access alone.

5.3 Analysis of Adaptive Escalation Behavior

Effect of maximum feedback depth. Fig. 3 compares fixed-depth acquisition with adaptive escalation under the same maximum accessible feedback level. For a maximum depth L_ℓ, the fixed strategy acquires feedback up to L_ℓ at every planning step. Increasing the accessible feedback depth generally improves planning performance across all three benchmarks. MAVE retains most of these gains, with modest reductions in success rate and route quality relative to fixed-depth acquisition. Meanwhile, its feedback cost grows substantially more slowly as the maximum depth increases. These results demonstrate that adaptive stopping preserves much of the benefit of deeper feedback while substantially reducing acquisition cost.

Difficulty-dependent escalation behavior. Target difficulty is defined independently using the required single-step model calls of Retro*+, grouped into Easy, Medium, and Hard (Appendix D.3). Fig. 4 shows that MAVE adapts its stopping depth to planning difficulty. Easy instances are predominantly resolved at shallow levels, whereas harder instances increasingly continue to L_2-L_4. Accordingly, the average stopping depth rises from 0.72 for Easy cases to 1.60 for Hard cases. This shift indicates that MAVE allocates deeper feedback selectively, stopping early when limited evidence is sufficient while reserving more costly feedback for more difficult planning states.

Ablation of marginal-value-aware credit assignment. Tab. 5 shows that both marginal-value signals improve success rates and route quality over the cost-aware outcome objective J. Compared with $D^{(2)}$ alone, $D^{(1)}$ yields higher route quality and lower query rates, whereas $D^{(2)}$ achieves higher success rates on two benchmarks. Combining both terms achieves the highest success rates and lowest acquisition costs across all three benchmarks, despite slightly higher query rates than $J + D^{(1)}$. This pattern is consistent with $D^{(2)}$ refining escalation depth: lightweight feedback can be acquired when further value is emerging, while unnecessary progression to costly higher levels is avoided. These results support the complementary roles of the two signals in learning effective and cost-efficient escalation policy.

6 Conclusion

We formulate feedback acquisition in retrosynthetic planning as adaptive escalation and introduce MAVE for marginal-value-aware policy learning. MAVE incorporates the slope and curvature of the reward-cost trajectory into advantage estimation to support just-enough feedback acquisition. Across three benchmarks, MAVE improves planning performance while avoiding unnecessary feedback usage and adapting escalation depth to planning difficulty. These results demonstrate the value of marginal-value-aware credit assignment for efficient adaptive escalation.

IMPACT STATEMENT

This work aims to advance machine learning for cost-aware and interactive retrosynthetic planning with multi-fidelity feedback. More efficient feedback allocation may improve the reliability of computer-aided synthesis planning, accelerate chemical discovery, and reduce unnecessary computational or experimental effort. At the same time, advances in synthesis planning could potentially lower the barrier to designing or synthesizing harmful compounds. Our experiments rely on surrogate feedback oracles in a controlled evaluation environment and do not provide laboratory protocols or actionable guidance for hazardous synthesis. Further safeguards and domain-specific validation would therefore be necessary before deployment in real-world chemical discovery workflows.

AI USE STATEMENT

In this work, we used generative AI tools for writing assistance, specifically to improve the grammar and clarity of author-written text. We have not used generative AI tools for research ideation, method formulation, experimental design, or the generation and analysis of experimental results, and retrieval and discovery of related work is not applicable to this work. Additionally, we used generative AI tools for coding assistance on auxiliary scripts. We have reviewed all AI-assisted work: AI-edited text was checked line by line by the authors against the intended technical content, and AI-assisted code was read and tested for correctness by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The MAVE source code and both the Wuxi-expert-873 and PDBbind-160 benchmarks are available at `https://anonymous.4open.science/r/MAVE-retro`. The method is specified in Sec. 4, with the roll-out procedure in Alg. 1 and the inference-time planner in Alg. 2. The oracle system and acquisition-cost schedule are described in App. A.1 and Tab. 6, hyperparameters in Tab. 7, alternative reward-cost mappings in App. B, baseline adaptations in App. C, and the route-quality metric and difficulty grouping in App. D.4 and D.3.

REFERENCES

Frazier N. Baker, Daniel Adu-Ampratwum, Reza Averly, Botao Yu, Huan Sun, and Xia Ning. Larc: Towards human-level constrained retrosynthesis planning through an agentic framework, 2025. URL https://arxiv.org/abs/2508.11860.

Andres M Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Chemcrow: Augmenting large-language models with chemistry tools, 2023. URL https://arxiv.org/abs/2304.05376.

Andres M Bran, Theo A Neukomm, Daniel P Armstrong, Zlatko Jončev, and Philippe Schwaller. Chemical reasoning in llms unlocks strategy-aware synthesis planning and reaction mechanism elucidation, 2025. URL https://arxiv.org/abs/2503.08537.

Raeyoung Chang, Dongwook Kwon, Jisoo Lee, and Nikhil Verma. CascadeDebate: Multi-agent deliberation for cost-aware LLM cascades. In Yunyao Li, Georg Rehm, and Mei Tu (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 1329–1340, San Diego, California, USA, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-394-4. doi: 10.18653/v1/2026.acl-industry.93. URL https://aclanthology.org/2026.acl-industry.93/.

Binghong Chen, Chengtao Li, Hanjun Dai, and Le Song. Retro*: Learning retrosynthetic planning with neural guided a* search, 2020. URL https://arxiv.org/abs/2006.15820.

Qianben Chen, Jingyi Cao, Jiayu Zhang, Tianrui Qin, Xiaowan Li, King Zhu, Dingfeng Shi, He Zhu, Minghao Liu, Xiaobo Liang, Xin Gui, Ge Zhang, Jian Yang, Yuchen Eleanor Jiang, and Wangchunshu Zhou. A^2fm: An adaptive agent foundation model for tool-aware hybrid reasoning, 2025. URL https://arxiv.org/abs/2510.12838.

Mehul Damani, Idan Shenfeld, Andi Peng, Andreea Bobu, and Jacob Andreas. Learning how hard to think: Input-adaptive allocation of lm computation. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 102783–102802, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/ff414825df833edb8b1839e3d5d495e9-Paper-Conference.pdf.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks Lakshmanan, and Ahmed H Awadallah. Hybrid llm: Cost-efficient and quality-aware query routing. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 41348–41366, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/b47d93c99fa22ac0b377578af0a1f63a-Paper-Conference.pdf.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. In *ICLR*. OpenReview.net, 2024. URL https://openreview.net/pdf?id=Tlsdsb6l9n.

Jonathan Goodman. Computer software review: Reaxys. *Journal of Chemical Information and Modeling*, 49(12):2897–2898, 2009. doi: 10.1021/ci900437n. URL https://doi.org/10.1021/ci900437n. PMID: 19961179.

G. Anthony Gorry and G. Octo Barnett. Sequential diagnosis by computer. *JAMA*, 205(12):849–854, 09 1968. ISSN 0098-7484. doi: 10.1001/jama.1968.03140380053012. URL https://doi.org/10.1001/jama.1968.03140380053012.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, and et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. Language model cascades: Token-level uncertainty and beyond. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 4147–4180, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/11f5520daf9132775e8604e89f53925a-Paper-Conference.pdf.

Helia Hashemi, Victor Ruehle, and Saravan Rajmohan. Cost-aware retrieval-augmentation reasoning models with adaptive retrieval depth. In *2026 The Web Conference*, April 2026.

Siqi Hong, Hankz Hankui Zhuo, Kebing Jin, and Zhanwen Zhou. Retrosynthetic planning with experience-guided monte carlo tree search. *CoRR*, abs/2112.06028, 2021. URL https://arxiv.org/abs/2112.06028.

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, June 2023. ISSN 2399-3669. doi: 10.1038/s42004-023-00911-8. URL https://doi.org/10.1038/s42004-023-00911-8.

Shu Jiang, Zhuosheng Zhang, Hai Zhao, Jiangtong Li, Yang Yang, Bao-Liang Lu, and Ning Xia. When smiles smiles, practicality judgment and yield prediction of chemical reaction via deep chemical language processing. *IEEE Access*, 9:85071–85083, 2021. doi: 10.1109/ACCESS.2021.3083838.

Junsu Kim, Sungsoo Ahn, Hankook Lee, and Jinwoo Shin. Self-improved retrosynthetic planning. *CoRR*, abs/2106.04880, 2021. URL https://arxiv.org/abs/2106.04880.

Xinyi Li, Sai Wang, Yutian Lin, Yu Wu, and Yi Yang. Retro-expert: Collaborative reasoning for interpretable retrosynthesis, 2025. URL https://arxiv.org/abs/2508.10967.

Wei Liu, Jiangtao Feng, Hongli Yu, Yuxuan Song, Yuqiang Li, Shufei Zhang, LEI BAI, Wei-Ying Ma, and Hao Zhou. Retro-r1: Llm-based agentic retrosynthesis. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi,

and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 70709–70737. Curran Associates, Inc., 2025. doi: 10.52202/085713-2378. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/6697768644aeabfd163726c983f75795-Paper-Conference.pdf.

Qinyu Ma, Yuhao Zhou, and Jianfeng Li. Automated retrosynthesis planning of macromolecules using large language models and knowledge graphs. *Macromolecular Rapid Communications*, pp. 2500065, 2025.

John Mayfield, Daniel Lowe, and Roger Sayle. Pistachio: Search and faceting of large reaction databases. In *ABSTRACTS OF PAPERS OF THE AMERICAN CHEMICAL SOCIETY*, volume 254. AMER CHEMICAL SOC 1155 16TH ST, NW, WASHINGTON, DC 20036 USA, 2017.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7076–7087. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/mozannar20b.html.

Alexander Luke Ian Norcliffe, Changhee Lee, Fergus Imrie, Mihaela Van Der Schaar, and Pietro Lio. Stochastic encodings for active feature acquisition. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 46784–46814. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/norcliffe25a.html.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, M Kadous, and Ion Stoica. Routellm: Learning to route llms from preference data. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 34433–34448, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/5503a7c69d48a2f86fc00b3dc09de686-Paper-Conference.pdf.

Shreyas Vinaya Sathyanarayana, Sharanabasava D. Hiremath, Rahil Shah, Rishikesh Panda, Rahul Jana, Riya Singh, Rida Irfan, Ashwin Murali, and Bharath Ramsundar. Deepretro: Retrosynthetic pathway discovery using iterative llm reasoning, 2025. URL https://arxiv.org/abs/2507.07060.

Philippe Schwaller, Alain C Vaucher, Teodoro Laino, and Jean-Louis Reymond. Prediction of chemical reaction yields using deep learning. *Machine Learning: Science and Technology*, 2(1): 015016, mar 2021. doi: 10.1088/2632-2153/abc81d. URL https://doi.org/10.1088/2632-2153/abc81d.

Marwin Segler, Mike Preuss, and Mark Waller. Planning chemical syntheses with deep neural networks and symbolic ai. *Nature*, 555:604–610, 03 2018. doi: 10.1038/nature25978.

Hajin Shim, Sung Ju Hwang, and Eunho Yang. Joint active feature acquisition and classification with variable-size set encoding. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/e5841df2166dd424a57127423d276bbe-Paper.pdf.

Xiaozhuang Song, Xuanhao Pan, Xinjian Zhao, Hangting Ye, Shufei Zhang, Jian Tang, and Tianshu Yu. Aot*: Efficient synthesis planning via llm-empowered and-or tree search, 2025. URL https://arxiv.org/abs/2509.20988.

Shion Takeno, Hitoshi Fukuoka, Yuhki Tsukada, Toshiyuki Koyama, Motoki Shiga, Ichiro Takeuchi, and Masayuki Karasuyama. Multi-fidelity Bayesian optimization with max-value entropy search and its parallelization. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9334–9345. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/takeno20a.html.

Kirill Trapeznikov, Venkatesh Saligrama, and David Castañón. Multi-stage classifier design. In Steven C. H. Hoi and Wray Buntine (eds.), *Proceedings of the Asian Conference on Machine Learning*, volume 25 of *Proceedings of Machine Learning Research*, pp. 459–474, Singapore Management University, Singapore, 04–06 Nov 2012. PMLR. URL https://proceedings.mlr.press/v25/trapeznikov12.html.

Michael Valancius, Maxwell Lennon, and Junier Oliva. Acquisition conditioned oracle for nongreedy active feature acquisition. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 48957–48975. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/valancius24a.html.

Haorui Wang, Jeff Guo, Lingkai Kong, Rampi Ramprasad, Philippe Schwaller, Yuanqi Du, and Chao Zhang. LLM-augmented chemical synthesis and design decision programs. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 62980–63001. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/wang25ag.html.

YiFei Wang, Qizhi Pei, Jiangtao Feng, Yuntian Shi, Yi Duan, Lihao Wang, Lei Bai, Lijun Wu, Wei-Ying Ma, and Hao Zhou. r^3: End-to-end reasoning-based planning for multi-step retrosynthesis via reinforcement learning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 37618–37632, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1745. URL https://aclanthology.org/2026.acl-long.1745/.

Dongxia Wu, Ruijia Niu, Matteo Chinazzi, Yian Ma, and Rose Yu. Disentangled multi-fidelity deep Bayesian active learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 37624–37634. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/wu23p.html.

Yemin Yu, Ying Wei, Kun Kuang, Zhengxing Huang, Huaxiu Yao, and Fei Wu. Grasp: Navigating retrosynthetic planning with goal-driven policy. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 10257–10268. Curran Associates, Inc., 2022.

Luotian Yuan, Yemin Yu, Ying Wei, Yongwei Wang, Zhihua Wang, and Fei Wu. Active retrosynthetic planning aware of route quality. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=h7DGnWGeos.

Murong Yue, Jie Zhao, Min Zhang, Liang Du, and Ziyu Yao. Large language model cascades with mixture of thought representations for cost-efficient reasoning. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 21691–21728, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/5de11e930c1bbfda5d4fc9d2b0924032-Paper-Conference.pdf.

Situo Zhang, Hanqi Li, Lu Chen, Zihan Zhao, Xuanze Lin, Zichen Zhu, Bo Chen, Xin Chen, and Kai Yu. Reasoning-driven retrosynthesis prediction with large language models via reinforcement learning, 2025. URL https://arxiv.org/abs/2507.17448.

Yanqiao Zhu, Jingru Gan, Xiaoqi Sun, Fang Sun, Yidan Shi, Md Mofijul Islam, Chao Shang, Wenhao Gao, Connor W. Coley, Yizhou Sun, and Wei Wang. Retroagent: Harnessing LLMs to search over structured memory for agentic retrosynthesis planning. In *Conference on Language Modeling (COLM)*, 2026.

A APPENDIX

A.1 MULTI-FIDELITY SURROGATE ORACLE SYSTEM

Overview. We construct a controlled multi-fidelity oracle system for studying feedback allocation in retrosynthetic planning. Given a query q, each oracle returns query-specific feedback together with an acquisition cost $c(q)$. Here, fidelity refers to the scope and decision informativeness of feedback, rather than strictly to predictive accuracy. The oracle costs are defined on a common normalized scale that reflects the relative acquisition burden of the surrogate backends, rather than literal laboratory monetary or time costs. Consequently, the simulated cost ordering need not coincide with real-world experimental expense; for example, surrogate yield prediction is inexpensive in our environment even though experimental yield measurement may be costly in practice. All compared allocation methods use the same underlying feedback sources and cost specification.

Feedback hierarchy. We organize the oracle interfaces into four operational levels according to the decision context and information granularity of the feedback. Here, fidelity refers to progressively broader decision context and richer planning information, rather than strictly higher predictive accuracy.

- L_1 **(Structural).** Local guidance on the current molecule, including activity assessment, reaction class, reaction center, and bond disconnection.
- L_2 **(Evaluative).** Assessment of an individual candidate reaction, including reaction yield, feasibility, and selectivity.
- L_3 **(Comparative).** Relative assessment among alternative reactions or partial routes, including reaction and route comparison.
- L_4 **(Strategic).** Search-level guidance over multiple synthesis steps, including goal suggestion and multi-step strategy.

Relative acquisition-cost scale. Because the surrogate backends involve heterogeneous computational and annotation processes, their raw acquisition burdens are not directly comparable in a single physical unit. We therefore express oracle costs on a common dimensionless scale and set $b_{\mathrm{yield}} = 1$ as a numerical reference point, which defines the unit of the scale rather than a physical cost.

For each feedback type, we assign a base cost representing its typical acquisition burden. Base costs specify a controlled relative-cost schedule reflecting the relative acquisition workload of different surrogate interfaces, rather than empirical measurements of absolute real-world cost. The actual cost of a query is then slightly adjusted according to its query-specific complexity, so queries of the same type may incur slightly different costs. Higher hierarchy levels are assigned larger base costs overall. Query cost is determined independently of the downstream planning reward.

Base costs reflect the typical acquisition burden of the surrogate interfaces, while the actual cost of each query varies within $\pm 20\%$ of its base value according to query-specific complexity. At L_1, the adjustment mainly depends on the number of competing reactive centers or plausible disconnections. At L_2, cost varies with reaction complexity, including the number of atoms, reactants, and reagents involved in the candidate reaction. At L_3, the adjustment additionally accounts for the similarity between the reactions or partial routes being compared, together with their aggregate molecular and reagent complexity. At L_4, cost reflects the workload of the depth-limited traversal, including its wall-clock search time, AND-OR traversal search-space size. Thus, queries of the same feedback type may incur different costs while remaining centered around the corresponding base cost.

Surrogate backends. Each oracle maps a query to a compact feedback signal through a dedicated surrogate backend. Only the returned feedback is exposed to the planner; internal backend states are not accessible.

Structural feedback (L_1). Structural feedback is derived from the reaction-template space using a round-trip model as the common scoring backend. Applicable templates are enumerated at the current node and ranked by their round-trip scores. The resulting high-scoring candidates are then used to derive reaction class, reaction center, bond disconnection, and activity assessment, providing complementary structural cues for the subsequent reaction decision.

Table 6: Acquisition costs of the multi-fidelity surrogate oracle system. Costs are expressed on a dimensionless relative scale, with $b_{\text{yield}} = 1$ as the numerical reference. Query-specific costs vary within $\pm 20\%$ of the corresponding base cost according to acquisition complexity; the ranges do not represent random sampling.

Level	Feedback type	Base cost	Cost range
L_0	None	0.00	0.00
L_1	Activity assessment	0.10	0.08-0.12
	Reaction class	0.25	0.20-0.30
	Reaction center	0.30	0.24-0.36
	Bond disconnection	0.35	0.28-0.42
L_2	Reaction feasibility	0.80	0.64-0.96
	Reaction yield	1.00	0.80-1.20
	Selectivity assessment	1.20	0.96-1.44
L_3	Reaction comparison	2.00	1.60-2.40
	Route comparison	3.50	2.80-4.20
L_4	Goal suggestion	5.00	4.00-6.00
	Multi-step strategy	10.00	8.00-12.00

Evaluative feedback (L_2). Evaluative feedback operates on individual candidate reactions. Reaction yield is predicted by a pretrained yield model and normalized to $[0, 1]$. For the remaining signals, a round-trip model serves as the common backend: reaction feasibility returns a binary feasibility judgment, while selectivity assessment compares competing reactive sites. These interfaces provide candidate-level assessments without considering the broader search trajectory.

Comparative feedback (L_3). Comparative feedback operates over multiple alternatives. Reaction comparison takes two candidate reactions and returns a relative preference based on their predicted yields. Route comparison evaluates partial synthesis routes from the current search frontier using the round-trip model and returns a relative preference between them. Thus, L_3 provides comparative guidance across alternatives rather than evaluating a single candidate in isolation.

Strategic feedback (L_4). Strategic feedback provides guidance beyond an individual reaction. A depth-limited traversal of the environment-defined AND-OR search space is used to identify informative intermediate subgoals and recurring multi-step transformation patterns. Goal suggestion returns candidate intermediates or subgoals that may guide subsequent search. Multi-step strategy instead emulates high-level guidance from a human expert: rather than specifying a unique synthesis route, it provides one or more plausible strategic directions, such as preferred transformation orderings, functional-group manipulations, or intermediate choices. To reflect the inherent ambiguity of such expert advice, the returned strategy may contain multiple alternatives with different levels of relevance and may include uncertain or weakly informative suggestions. Only the resulting textual guidance is exposed to the planner; the underlying search trees are not accessible.

Oracle information boundary. Higher-level oracles may use additional computation that is not directly available to the planning policy, reflecting their role as external information sources. Such computation remains internal to the oracle and is accessed only through the corresponding query interface after incurring its acquisition cost. In particular, the L_4 traversal is depth-limited and does not use ground-truth complete synthesis routes or terminal building-block reachability labels. All compared allocation methods use the same oracle interfaces and cost specification.

B MAVE Implementation Details

MAVE hyperparameters. The main implementation and optimization hyperparameters are summarized in Tab. 7.

Alternative reward-cost mappings. The main experiments use the tanh reward-cost mapping described in Sec. 4.1. To examine whether MAVE depends on this particular functional form, we replace it with several differentiable mappings while keeping the remaining training and planning settings unchanged. For each fitted mapping, MAVE uses

$$D^{(1)}(c) = \phi'(c), \qquad D^{(2)}(c) = \phi''(c), \tag{10}$$

such that the first-order term measures the current net marginal value and the second-order term captures the local change in marginal value.

Tanh mapping. The default mapping is

$$\phi(c) = a + b \tanh z, \qquad z = \kappa(c - \tau), \tag{11}$$

with $b > 0$ and $\kappa > 0$. Its derivatives are

$$\phi'(c) = b\kappa \operatorname{sech}^2(z), \tag{12}$$

$$\phi''(c) = -2b\kappa^2 \operatorname{sech}^2(z) \tanh(z). \tag{13}$$

Sigmoid mapping. We alternatively use

$$\phi(c) = a + b\sigma(z), \qquad z = \kappa(c - \tau), \tag{14}$$

where $\sigma(z) = 1/(1 + \exp(-z))$. Let $p = \sigma(z)$. The corresponding derivatives are

$$\phi'(c) = b\kappa p(1 - p), \tag{15}$$

$$\phi''(c) = b\kappa^2 p(1 - p)(1 - 2p). \tag{16}$$

Polynomial mapping. We further consider a quadratic mapping,

$$\phi(c) = a_0 + a_1 c + a_2 c^2, \tag{17}$$

whose derivatives have the closed forms

$$\phi'(c) = a_1 + 2a_2 c, \tag{18}$$

$$\phi''(c) = 2a_2. \tag{19}$$

Tiny neural mapping. Finally, we consider a lightweight neural mapping to relax the assumption of a predefined analytic functional form. Because each local reward-cost relation is estimated from only a small roll-out group, we deliberately use a low-capacity network rather than a fully parameterized MLP. The cost is first normalized within the fitting range as

$$\tilde{c} = \frac{c - c_{\min}}{c_{\max} - c_{\min} + \epsilon}. \tag{20}$$

We then use two fixed sigmoid basis units,

$$h_j(\tilde{c}) = \sigma[\alpha(\tilde{c} - \tau_j)], \qquad j \in \{1, 2\}, \tag{21}$$

and define

$$\phi_{\mathrm{NN}}(c) = a + \sum_{j=1}^{2} w_j h_j(\tilde{c}), \qquad w_j = \operatorname{softplus}(\tilde{w}_j). \tag{22}$$

We fix the hidden-unit centers to $\tau_1 = 1/3$ and $\tau_2 = 2/3$ and use $\alpha = 6$. Only $a, \tilde{w}_1, \tilde{w}_2$ are fitted, resulting in three trainable parameters. The positive output weights preserve a non-decreasing reward-cost relation while allowing the local slope and curvature to vary flexibly.

Let

$$\Delta_c = c_{\max} - c_{\min} + \epsilon. \tag{23}$$

The derivatives with respect to the original cost variable are

$$\phi'_{\mathrm{NN}}(c) = \frac{\alpha}{\Delta_c} \sum_{j=1}^{2} w_j h_j(1 - h_j), \tag{24}$$

$$\phi''_{\mathrm{NN}}(c) = \frac{\alpha^2}{\Delta_c^2} \sum_{j=1}^{2} w_j h_j(1 - h_j)(1 - 2h_j). \tag{25}$$

These derivatives are evaluated at the observed costs and passed to the same MAVE advantage estimator as the analytic mappings.

Algorithm 2: Cost-aware escalation-guided MCTS inference.

Require: s_0, μ, π, $\mathcal{O}$, single-step model, L, B, B_f.
Initialize AND-OR tree $\mathcal{T}$, search statistics, feedback cache $\mathcal{F}$, and cost $C \leftarrow 0$
while B *remains* **do**
  // **Selection**
  Select path $\mathcal{P}$ from s_0 to a frontier molecule using Eq. 27
  if *the frontier is terminal* **then**
    Set terminal value v
  else
    // **Expansion**
    Generate candidate reactions $\mathcal{A}(s)$
    Initialize feedback f and level $\ell \leftarrow 0$
    while $\ell < L$ **do**
      Select STOP or a feedback query with μ
      if STOP **then break**
      if *the query is not cached* **then**
        if $C + c(q) > B_f$ **then break**
        Query $\mathcal{O}$ and cache the feedback
        $C \leftarrow C + c(q)$
      Update f and ℓ
    Score $\mathcal{A}(s)$ with π
    Initialize P and Q using Eq. 28
    Expand $\mathcal{T}$ with reaction and precursor nodes
    Set v using Eq. 29
  // **Backup**
  Back-propagate v along $\mathcal{P}$ using Eq. 30
  if *a complete route is found* **then**
    return the synthesis route
return FAILURE

Reward propagation. To propagate downstream planning outcomes to earlier decisions, we compute the return backward along each completed retrosynthetic trajectory with a discount factor γ. For an escalation roll-out initiated at planning step t and terminating at step T, its downstream return is

$$r_k = \sum_{j=t}^{T} \gamma^{j-t} R(s_j, a_j). \tag{26}$$

This return is used to construct the MAVE advantage for the corresponding escalation roll-out. The same discounted return is used for optimizing the reaction policy π, while feedback acquisition cost is accounted for separately through c_k.

MCTS-guided inference. At inference time, the learned escalation and reaction policies are integrated into an AND-OR Monte Carlo tree search (MCTS), following the standard selection-expansion-backup procedure used in retrosynthetic planning. MCTS determines which branch to explore, while μ controls feedback acquisition and π evaluates candidate reactions after escalation. Reward-cost fitting and marginal-value estimation are used only during training and are not required at inference.

During *selection*, MCTS recursively traverses the search tree according to the PUCT criterion

$$a^* = \arg\max_{a \in \mathcal{A}(s)} \left[Q(s,a) + c_{\text{puct}} P(s,a) \frac{\sqrt{N(s)}}{1 + N(s,a)} \right], \tag{27}$$

where $Q(s,a)$ is the current search value, $P(s,a)$ is the reaction prior, and $N(s,a)$ and $N(s)$ denote the corresponding visit counts. For reactions with multiple unsolved precursors, all precursor molecules are retained as AND children and must eventually be solved for the reaction to form a complete route.

During *expansion*, the single-step model first generates candidate reactions for the selected frontier molecule. The escalation policy μ then sequentially decides whether to stop or acquire feedback

Table 7: Implementation and training hyperparameters of MAVE.

Hyperparameter	**Value**
Backbone model	Llama-3-8B-Instruct
Maximum feedback level L	4
Escalation group size K	4
Cost coefficient λ	0.5
First-order weight β_1	1.0
Second-order weight β_2	1.0
Fit regularization η	0.001
Reward-cost mapping	Tanh
Learning rate for μ	3×10^{-5}
Learning rate for π	3×10^{-5}
GRPO clipping range	0.2
KL coefficient	0.001
Alternating interval	1
Training epochs	10000
Batch size	6
Discount factor γ	0.95

from the next level, subject to the feedback budget. Previously acquired feedback is cached and reused without additional acquisition cost. After escalation terminates, the reaction policy π scores the candidates conditioned on the accumulated feedback. The resulting probabilities are used both as MCTS priors and as heuristic initial values for newly expanded edges:

$$P(s,a) = \pi(a \mid s, f), \qquad Q_0(s,a) = P(s,a). \tag{28}$$

Accordingly, the value of a newly expanded non-terminal frontier is initialized by the policy score,

$$v(s) = \max_{a \in \mathcal{A}(s)} P(s,a), \tag{29}$$

while solved and dead terminal nodes are assigned values of 1 and 0, respectively. We use these policy-derived values only as heuristic initialization rather than as an additional learned value function. During *backup*, the leaf value is propagated toward the root with length discounting. For each traversed edge, the visit count is incremented and its search value is updated using:

$$N(s,a) \leftarrow N(s,a) + 1, \qquad Q(s,a) \leftarrow Q(s,a) + \frac{\gamma v - Q(s,a)}{N(s,a)}. \tag{30}$$

The updated value is then propagated to the next edge toward the root. Selection, expansion, and backup are repeated until a complete route to purchasable building blocks is found or the search budget is exhausted. Alg. 2 summarizes the complete procedure.

C Routing and Escalation Baselines

We adapt five representative routing and escalation methods to the same retrosynthetic planning environment used by MAVE. All methods share the same backbone initialization, single-step reaction model, MCTS procedure, feedback hierarchy, acquisition-cost specification, and evaluation budgets, while retaining their respective allocation principles.

For one-shot routing, we formulate feedback allocation as *depth routing*. Given a planning state, the router selects a maximum acquisition depth $d \in \{0, \ldots, L\}$ before observing any feedback. All feedback up to the selected depth is then acquired and provided jointly to the reaction policy. Sequential methods instead traverse the same hierarchy progressively, deciding after each observed level whether to stop or continue escalation. Thus, both paradigms have access to the same cumulative feedback for a given depth; they differ in whether the acquisition depth is committed to upfront or adapted to intermediate evidence. All method-specific thresholds are tuned on the validation set and fixed during evaluation.

These baselines cover complementary allocation mechanisms. RouteLLM performs preference-based one-shot depth routing, HybridLLM uses quality-aware depth routing, and A^{2}FM performs

cost-aware depth selection. MoT Cascade and LM Cascade instead make sequential stopping decisions based on prediction consistency and predictive uncertainty, respectively.

C.1 One-shot Depth Routing

RouteLLM. We construct a RouteLLM-style preference router over candidate acquisition depths $d \in \{0, \dots, L\}$. Pairwise supervision compares the downstream planning returns obtained at two depths:

$$y_{ij} = \mathbb{I}[\bar{r}(s, d_j) > \bar{r}(s, d_i)], \tag{31}$$

where $\bar{r}(s, d)$ denotes the average downstream return when acquiring feedback up to depth d at planning state s. The learned preference model scores candidate depths and selects one before observing any feedback. Feedback up to the selected depth is then acquired jointly before invoking the reaction policy π.

HybridLLM. We adapt HybridLLM as a quality-aware one-shot depth router. Given the current planning context, the router estimates the expected benefit of additional feedback and maps the resulting score to a maximum acquisition depth $d \in \{0, \dots, L\}$ using thresholds calibrated on the validation set. Feedback up to the selected depth is then acquired jointly. The depth is fixed upfront and cannot be revised after intermediate feedback is observed.

A^{2}FM. We adapt the cost-aware mode-selection principle of A^{2}FM by treating each acquisition depth as an execution mode. For depth d, the cost-adjusted objective is

$$J_d = r_d - \lambda c_d, \qquad c_d = \sum_{\ell=1}^{d} c^{(\ell)}, \tag{32}$$

where r_d is the downstream planning return and c_d is the cumulative feedback cost up to depth d. The allocator selects a maximum depth according to its estimated cost-adjusted utility before observing any feedback. Unlike sequential escalation, the acquisition decision is not revised as feedback is accumulated.

C.2 Sequential Escalation

MoT Cascade. We extend the consistency-based stopping principle of MoT Cascade to the multi-level feedback hierarchy. At level ℓ, multiple reaction predictions are sampled from π under the currently accumulated feedback. Their agreement is measured as

$$S_\ell = \frac{1}{M} \max_a \sum_{m=1}^{M} \mathbb{I}[a_m = a], \tag{33}$$

where M denotes the number of sampled predictions. The cascade stops when $S_\ell \geq \tau_{\text{con}}$; otherwise, it acquires feedback from the next level and repeats the decision. Additional reaction-policy samples used to estimate consistency are counted toward the common inference budget but do not incur oracle acquisition cost.

LM Cascade. We instantiate an LM-Cascade-style uncertainty baseline over the same feedback hierarchy. At each level, predictive uncertainty is estimated from the reaction distribution produced by π under the accumulated feedback. We use normalized reaction entropy,

$$U_\ell = -\frac{\sum_{a \in \mathcal{A}(s)} \pi(a \mid s, f^{(\leq \ell)}) \log \pi(a \mid s, f^{(\leq \ell)})}{\log |\mathcal{A}(s)|}. \tag{34}$$

The cascade stops when $U_\ell \leq \tau_{\text{unc}}$ and otherwise proceeds to the next feedback level. This task-specific adaptation replaces token-level language uncertainty with reaction-level predictive uncertainty while retaining the original uncertainty-based deferral principle.

Table 8: Robustness to alternative reward-cost mappings on PDBbind-160. All variants use the same MAVE training and planning setup, differing only in the mapping used to estimate the local reward-cost relation. Best and second-best planning results are bolded and underlined, respectively.

Mapping	Success rate ↑	Route quality ↑	Query rate ↓	Query cost ↓
Tanh	**84.4%**	**0.710**	58.8%	95.1
Sigmoid	<u>83.8%</u>	<u>0.707</u>	59.6%	96.4
Polynomial	81.9%	0.694	62.7%	101.8
Tiny neural	82.5%	0.699	61.3%	99.6

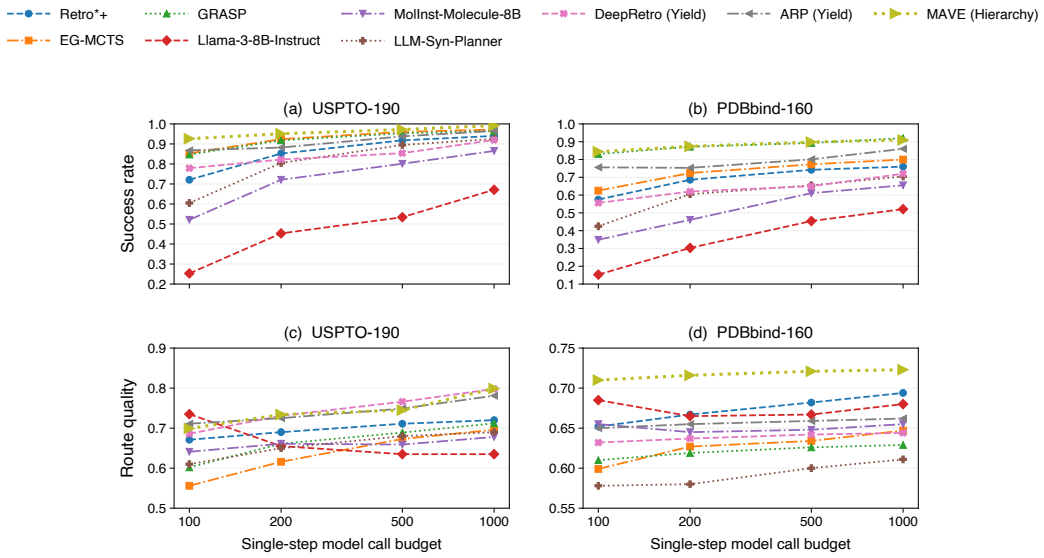

Figure 5: Retrosynthetic performance under 100, 200, 500, 1000 single-step model call budgets.

D Additional Experiments

D.1 Alternative Mappings

Alternative reward-cost mappings. Tab. 8 evaluates the sensitivity of MAVE to the reward-cost mapping on PDBbind-160. Tanh performs best overall, while sigmoid yields comparable results, indicating that MAVE is not tied to a specific smooth saturating form. The quadratic mapping is weaker, likely because its constant curvature provides limited flexibility for modeling marginal-value evolution. For the tiny neural mapping, we fit a lightweight local function independently for each roll-out group rather than training a global network. Normalized costs are passed through two fixed sigmoid basis units, and only three parameters are fitted from the current group's reward-cost observations. The fitted mapping is used solely to compute the local first- and second-order derivatives for MAVE advantage estimation and is discarded afterward; no neural reward-cost model is used at inference. Its slightly weaker performance suggests that, under small local roll-out groups, additional flexibility does not necessarily improve derivative estimation.

D.2 Larger Search Budgets

The main experiments use a budget of 100 single-step model calls to evaluate methods under constrained planning resources. We further increase the budget to 200, 500, and 1,000 calls to examine whether the observed trends persist with additional planning capacity. As shown in Fig. 5, increasing the planning budget generally improves the success rate of search-based retrosynthetic methods, while route quality changes more moderately. The gains gradually diminish at larger budgets, indicating saturation of the search process. MAVE remains competitive across all three benchmarks as the budget increases, suggesting that its effectiveness is not specific to the 100-call setting.

D.3 Definition of Planning Difficulty

We define planning difficulty independently of MAVE using the search effort of Retro*+ as a reference planner. For each target i, we record the number of single-step model calls n_i required by Retro*+ to identify its first complete retrosynthetic route. This provides an operational measure of planning difficulty under a fixed reference planner and avoids defining difficulty from MAVE's own search or escalation behavior. Targets are grouped according to the required search effort:

$$\text{Difficulty}(i) = \begin{cases} \text{Easy}, & n_i < 100, \\ \text{Medium}, & 100 \le n_i < 500, \\ \text{Hard}, & n_i \ge 500. \end{cases} \tag{35}$$

Targets not solved by Retro*+ before reaching the Hard-category threshold are also assigned to the Hard group. The resulting labels are determined independently of MAVE and remain fixed throughout the analysis.

D.4 Route Quality Evaluation

Following ARP Yuan et al. (2024), we first score each returned synthesis tree by recursively aggregating per-reaction quality scores. For a purchasable molecule m, we set $Q(m) = 1$. For a non-terminal molecule m synthesized through reaction r,

$$Q(m) = q(r) \min_{m' \in \mathcal{C}(r)} Q(m'), \tag{36}$$

where $q(r) \in [0, 1]$ is the reaction quality score and $\mathcal{C}(r)$ is the set of reactant molecules. This recursion multiplies reaction scores along each branch and takes the minimum across required branches. Thus, the score of a synthesis tree $\mathcal{T}$ is

$$S(\mathcal{T}) = \min_{p \in \mathcal{P}(\mathcal{T})} \prod_{r \in p} q(r), \tag{37}$$

where $\mathcal{P}(\mathcal{T})$ contains all root-to-leaf reaction paths.

To account for differences in attainable route scores across targets, we normalize the returned route's score by the optimal score for the same target. Let Ω_i denote the complete synthesis routes within the defined reaction search space for target i. We identify the optimal route through exhaustive traversal:

$$\mathcal{T}_i^\star \in \arg\max_{\mathcal{T} \in \Omega_i} S(\mathcal{T}). \tag{38}$$

For the route $\widehat{\mathcal{T}}_i$ returned by the planner, the reported route-quality metric is

$$Q_{\text{route},i} = \frac{S(\widehat{\mathcal{T}}_i)}{S(\mathcal{T}_i^\star)}. \tag{39}$$

Provided that $S(\mathcal{T}_i^\star) > 0$ and both routes belong to the same search space, this metric lies in $[0, 1]$, with higher values indicating routes closer to the target-specific optimum.

Per-reaction scores are estimated using a surrogate reaction-yield predictor Jiang et al. (2021), trained on more than 48 million reactions aggregated from USPTO Schwaller et al. (2021), Pistachio Mayfield et al. (2017), CJHIF Jiang et al. (2021), and Reaxys Goodman (2009). The same predictor provides reaction-yield feedback in our environment.