CATAPATH: EXPERT-AUGMENTED PATH SEARCH FOR CONSTRAINT-AWARE SCIENTIFIC REASONING IN CATALYST DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) have shown promise as scientific assistants, yet their use in open-ended material design remains limited by violations of domain-specific constraints. We study rational catalyst design as a challenging testbed for scientific reasoning, where an agent must generate chemically coherent intermediate decisions. We propose CataPath, an expert-augmented LLM agent that formulates catalyst design as search over a chemically informed action space. CataPath uses Monte Carlo Tree Search to explore design trajectories and a hybrid reward mechanism that combines property-prediction signals with an Expert Preference Reward Model trained from domain feedback. To support this framework, we curate CataDesignQR, a dataset of literature-grounded design goals, verified solutions, and expert-scored reasoning trajectories. We evaluate CataPath on two catalyst-related material-design settings, CO_2 reduction electrocatalyst generation and organic structure-directing agent optimization. Experiments show that CataPath improves generation validity, proxy effectiveness, and expert alignment over other baselines. These results suggest that process-level search with expert-informed rewards can improve the reliability of LLM agents for scientific design tasks.

1 INTRODUCTION

Scientific discovery increasingly requires models that can make intermediate decisions under domain constraints and revise hypotheses based on feedback. This challenge appears across biomedicine (Fan et al., 2023; Luo et al., 2025), chemistry (Jiang et al., 2024; Zheng et al., 2025), and materials science (Ma et al., 2025; Wang et al., 2025c), where a valid solution must satisfy not only logical coherence but also strict physical and chemical feasibility constraints. Such problems pose a fundamental challenge: how can Large Language Models (LLMs) act as reliable scientific reasoning agents when correctness cannot be reduced to exact-match answers?

Rational catalyst design provides a representative and demanding testbed for this problem. For example, designing electrocatalysts for the CO_2 Reduction Reaction (CO_2RR) is critical for carbon-neutral chemical production, but it requires navigating a large design space involving surface structures, alloy compositions, and interface engineering (M. Bran et al., 2024; Kang & Kim, 2024; Mirza et al., 2024). A successful design must balance multiple objectives, including catalytic activity, selectivity, and synthetic feasibility. As a result, identifying a valid candidate is difficult even for domain experts, and remains challenging for automated systems (Tshitoyan et al., 2019).

LLMs have shown promise as scientific assistants, demonstrating capabilities in scientific text mining (Dagdelen et al., 2024; Chen et al., 2024), experimental design (Zhou et al., 2025), and tool-augmented reasoning (Bran et al., 2023; Ding et al., 2025). However, there remains a substantial gap between retrieving scientific knowledge and generating scientifically valid designs. Standard LLM generation is optimized only for next-token prediction. Consequently, early errors in a reasoning chain can propagate and lead to hallucinated conclusions (Suvarna et al., 2023). For example, in catalyst design, LLMs may generate a linguistically plausible proposal that violates thermodynamic constraints.

A promising direction is to guide LLMs through explicit intermediate reasoning steps. Yet building such a reasoning framework for catalyst design raises two central challenges: 1) the reasoning

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.

process must be structured around domain-meaningful actions. Unlike general-purpose planning tasks, catalyst design involves multiple chemically grounded operations. Without an action space aligned with material structures and properties, LLM agents may explore irrelevant or infeasible reasoning paths; 2) the search process requires reliable feedback. Scientific design problems often lack a single gold answer (Wei et al., 2025). Existing approaches commonly rely on external databases or property predictors to constrain LLM exploration (Sprueill et al., 2023; 2024; Hu et al., 2025). These signals are limited by the coverage and accuracy of the underlying predictors, and they often fail to capture tacit expert knowledge about practical design constraints.

In this work, we study catalyst design as a constraint-sensitive scientific reasoning problem for LLM agents. We formulate it as a structured search problem over a chemically informed action space. We introduce **CataPath**, an expert-augmented framework that uses Monte Carlo Tree Search (MCTS) to explore reasoning trajectories. Each action corresponds to a domain-specific design strategy, and each state records the current prompt, generated material candidate, and accumulated reasoning history. To guide the search, CataPath uses a hybrid reward mechanism that combines two complementary sources of feedback. The first is a property-prediction signal, which provides dense estimates of task-specific material properties such as Faradaic efficiency or binding energy. The second is an Expert Preference Reward Model (EPRM), trained from expert-scored reasoning trajectories to approximate domain judgments that are difficult to encode with numerical predictors alone. To support training and evaluation, we curate **CataDesignQR**, a dataset of catalyst design goals, verified solutions, and expert-scored reasoning trajectories. The dataset covers two complementary settings: de novo generation of CO_2RR electrocatalysts and structural optimization of Organic Structure-Directing Agents (OSDAs). The contributions of this work can be summarized as follows:

1. We propose **CataPath**, a MCTS-based agent over a chemically informed action space, enabling LLMs to explore, evaluate, and refine catalyst design trajectories in a structured manner.
2. We introduce a hybrid reward mechanism that integrates dense property-prediction feedback with an Expert Preference Reward Model trained from expert-scored trajectories.
3. We curate **CataDesignQR**, a benchmark containing design goals, verified solutions, and expert-scored reasoning trajectories for CO_2RR electrocatalyst generation and OSDA optimization.
4. Experiments across two catalyst-design tasks show that CataPath improves generation validity, proxy effectiveness, and expert alignment over other baselines.

2 Related Work

2.1 LLMs for Material Discovery

The application of LLMs to scientific discovery has made substantial progress, evolving from passive text generators to autonomous agents capable of complex reasoning (Ansari & Moosavi, 2024; Kang & Kim, 2024; Darvish et al., 2025). In chemistry and biomedicine, systems such as MedAgents (Tang et al., 2024) and DRUGAGENT (Liu et al., 2024b) demonstrate that LLMs can coordinate specialized tools to solve complex scientific tasks. Existing approaches generally fall into two lines: training domain-specific models with scientific corpora or chemical representations (Zhang et al., 2024; Xie et al., 2024), and adapting general-purpose LLMs to interact with external databases, simulators, or chemical APIs (Chiang et al., 2024; Buehler, 2024). Despite these advances, most LLM-based material discovery systems still rely on open-ended generation or tool orchestration, with limited control over the intermediate reasoning process. Recent search-based methods incorporate external feedback into LLM inference (Sprueill et al., 2023; 2024; Hu et al., 2025), but they are often constrained by the coverage of property predictors or simulation tools and may not capture tacit expert judgments about feasibility. Existing MCTS-based scientific agents demonstrate the value of search, but their settings typically provide a relatively well-defined success condition (Shi et al., 2025; Wuwu et al., 2025). Catalyst-related material design is different: multiple candidates may be plausible, while intermediate steps must remain chemically meaningful and practically feasible. CataPath addresses this setting by combining a domain-specific action space, the process-level trajectory reward, and a hybrid reward.

2.2 Reasoning Trajectory Datasets

Recent progress in LLM reasoning has shown that supervising intermediate reasoning processes can improve model reliability beyond final-answer supervision (Lightman et al., 2024; Setlur et al., 2025). Datasets with chain-of-thought, solution traces, process rewards, and preference annotations have been widely used in mathematical problem solving, question answering, tool use, and decision making (Yao et al., 2023; Song et al., 2025; Jiao et al., 2024; Song et al., 2024; Xi et al., 2025). These datasets provide process-level signals that help models learn how to decompose problems, explore alternatives, and recover from intermediate errors. However, reasoning trajectory datasets for scientific discovery remain limited (Chen et al., 2025). Unlike mathematical reasoning, scientific design problems often lack a unique gold answer, and intermediate steps must satisfy domain-specific constraints. Existing scientific and chemical datasets typically focus on input-output prediction or question answering, making them less suitable for training or evaluating LLM agents that search through complex scientific decision spaces (Hao et al., 2026; M. Bran et al., 2024; Wang et al., 2025a). CataDesignQR is designed to fill this gap for catalyst design. Instead of only pairing design goals with final candidates, it records structured reasoning trajectories over chemically meaningful actions and associates them with expert feedback. These trajectories support both reward-model training and benchmark evaluation, enabling us to study whether LLM agents can generate scientifically plausible solutions through constrained intermediate reasoning.

3 The CataDesignQR Dataset

We introduce CataDesignQR, a dataset curated to capture process-oriented reasoning in material science. As illustrated in Fig. 1, the dataset covers two complementary settings: the de novo design of CO_2RR electrocatalysts and the structural optimization of OSDAs. A defining feature of CataDesignQR is the formalization of chemical knowledge into a discrete action space $\mathscr{A}$, which transforms scientific open-ended generation into a structured reasoning process suitable for LLM agents. The complete strategy set is provided in Appendix A.3.

3.1 CO_2RR Electrocatalyst Design

The CO_2RR subset focuses on the design of electrocatalysts for carbon dioxide reduction. We first construct a knowledge base of catalytic design strategies to reflect the trial-and-error reasoning process of human researchers, who optimize catalytic activity and selectivity through structural and compositional regulation (Deng et al., 2022). Through expert annotation, we organize broad regulation strategies into six categories, including alloy, defect, atomic-level dispersion, surface modification, composite construction, and structure control. These categories and their detailed keywords constitute the discrete action space for this task.

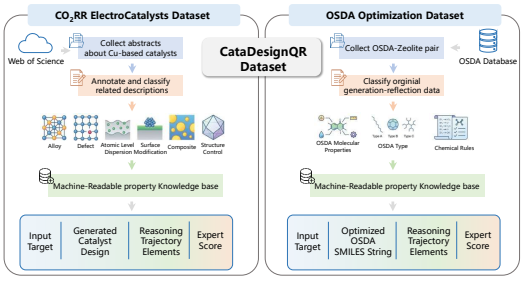

Figure 1: The construction pipeline for CataDesignQR, illustrating the extraction of action spaces and the collection of expert-annotated reasoning trajectories.

Based on literature-verified design goals, we prompt LLMs with different strategy combinations to generate candidate catalysts and corresponding reasoning trajectories. Each trajectory records the sequence of design actions, generated material candidates, and intermediate reasoning context. Domain experts then assign feedback scores according to the chemical validity, synthetic feasibility, and practical plausibility of the generated candidates. These expert scores provide process-level supervision for learning expert-aligned reward signals.

3.2 OSDA Structural Optimization

The OSDA subset focuses on optimizing organic structure-directing agents for zeolite synthesis, which is treated as catalyst-related molecular optimization. Building upon previous agentic frameworks (Hu et al., 2025), we refine the original generation-evaluation-reflection workflow into a more structured reasoning process.

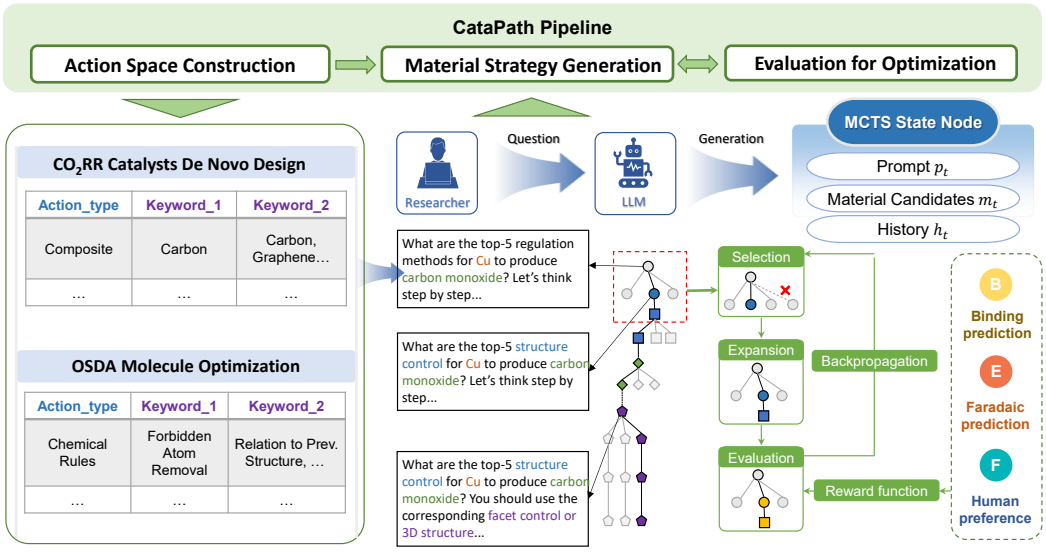

Figure 2: The CataPath pipeline. CataPath transforms catalyst design into constrained trajectory search over a chemically informed action space for CO_2RR electrocatalyst design and OSDA molecule optimization. Each MCTS state contains the prompt p_t, generated material candidate m_t, and reasoning history h_t. The LLM expands the tree under selected design actions, while a hybrid reward evaluates each state using property-prediction signals and expert-preference feedback.

The OSDA action space covers constraints and operations related to molecular type, molecular properties, chemical validity, synthetic feasibility, and binding energy optimization. Given an initial OSDA molecule and a target zeolite framework, LLMs generate optimized SMILES strings under selected strategy constraints. We also collect the generated molecules, reasoning trajectories, and expert feedback scores, enabling the dataset to support both reward-model training and evaluation of constrained molecular optimization.

3.3 Data Split and Leakage Control

We separate expert-scored reasoning trajectories from held-out reference solutions through a design-goal-level split. Specifically, all trajectories associated with the same design goal are assigned to the same split. We use an 80/20 design-goal-level split; within the training split, 10% of preference pairs are reserved only for EPRM validation. In our experiments, expert-scored trajectories from the training split are used to train the EPRM described in Sec. 4.3, while held-out design goals are reserved for final evaluation. During evaluation, CataPath receives only the design goal and task-specific constraints as input. The corresponding literature-grounded reference solutions and expert scores are used only for post-hoc evaluation. The scale and composition of CataDesignQR are summarized in Table 1. Detailed annotation procedures and consistency checks are provided in Appendix A.4.

Table 1: Statistics of the CataDesignQR Dataset.

Metric	CO_2RR	OSDA	Total
Design Goals (Queries)	456	205	661
Total Trajectories	2,250	656	2,906
Avg. Elements per Trajectory	4.9	3.2	–
Action Space Size ($\lvert\mathscr{A}\rvert$)	152	38	–

4 CataPath Methods

4.1 Problem Formulation

As shown in Fig. 2, we formulate catalyst design as a sequential decision-making problem over chemically grounded reasoning steps. Given a design goal, CataPath searches for a trajectory $\tau = \{s_0, a_0, s_1, \ldots, s_T\}$ that leads to a candidate material satisfying the target properties and domain constraints. We model this process as a Markov Decision Process (MDP) defined by $(\mathscr{S}, \mathscr{A}, \mathscr{P}, \mathscr{R})$.

State space. Each state $s_t \in \mathscr{S}$ represents the current reasoning status and is defined as $s_t = (p_t, m_t, h_t)$. Here, p_t is the prompt used to query the LLM, m_t is the current candidate generated by the LLM, and h_t records the accumulated reasoning history, including previous constraints, actions, and generated candidates. For CO_2RR electrocatalyst design, m_t is a textual catalyst design; for OSDA optimization, m_t is a SMILES string. At test time, the initial state s_0 contains only the held-out design goal and task-specific constraints.

Action space. The action space $\mathscr{A}$ is the discrete strategy set constructed in Sec. 3. Each action $a_t \in \mathscr{A}$ corresponds to a chemically meaningful design operation. In the CO_2RR task, an action selects a regulation keyword such as alloying, defect engineering, or surface modification. In the OSDA task, an action selects an optimization operation related to molecular validity, synthetic feasibility, or binding-energy improvement. Additional analyses in Appendix D.4 show that the predefined action space covers 87.2% of sampled free-form proposals.

Transition dynamics. Given a state s_t and an action a_t, CataPath first updates the prompt by injecting the selected action and accumulated history into a task-specific template:

$$p_{t+1} = \text{Update}(p_t, a_t, h_t). \tag{1}$$

The LLM then generates the next candidate conditioned on the updated prompt p_{t+1}:

$$m_{t+1} \sim \pi_{\text{LLM}}(\cdot \mid p_{t+1}, h_t). \tag{2}$$

The history is updated as $h_{t+1} = h_t \cup \{(a_t, m_{t+1})\}$, and the next state is $s_{t+1} = (p_{t+1}, m_{t+1}, h_{t+1})$. The prompt templates for the two tasks are provided in Appendix E.

Reward and objective. The reward function $\mathscr{R}(s_t)$ evaluates the quality of an intermediate state according to both task-specific property estimates and expert-aligned feasibility judgments. The hybrid reward is described in Sec. 4.3. The objective of CataPath is to find a terminal trajectory whose final candidate maximizes the search reward, where s_T is the terminal state:

$$\tau^* = \arg\max_{\tau} \mathscr{R}(s_T). \tag{3}$$

4.2 MCTS-based Trajectory Search

To explore the combinatorial space of chemically valid design strategies, CataPath performs Monte Carlo Tree Search over reasoning trajectories. Each node corresponds to a reasoning state s_t, and each edge corresponds to a selected design action a_t. Since full LLM-based rollouts are computationally expensive, we use a reward-guided MCTS variant that evaluates newly expanded states directly with the hybrid reward model.

Selection. Starting from the root state s_0, CataPath recursively selects child nodes until reaching a leaf node. At each state, the next action is chosen according to the Upper Confidence Bound for Trees (UCT) criterion:

$$a_t = \arg\max_{a \in \mathscr{A}(s_t)} \left[Q(s_t, a) + c_{\text{puct}} \sqrt{\frac{\ln N(s_t)}{1 + N(s_t, a)}} \right], \tag{4}$$

where $\mathscr{A}(s_t)$ denotes the set of valid actions under the current reasoning history, $Q(s_t,a)$ is the estimated value of taking action a at state s_t, $N(s_t)$ and $N(s_t,a)$ are visit counts, and c_{puct} controls the exploration-exploitation trade-off.

Expansion. When the search reaches a non-terminal leaf node, CataPath selects an unvisited valid action $a_t \in \mathscr{A}(s_t)$ and applies the transition function described in Sec. 4.1. The prompt is updated with the selected action and accumulated history, and the LLM generates a new candidate m_{t+1}, forming a child state s_{t+1}. Actions that violate task-specific constraints or duplicate invalid reasoning paths are filtered before expansion.

Evaluation. Instead of performing random rollouts to terminal states, CataPath evaluates each newly expanded state immediately, where $\mathscr{R}$ is the hybrid reward defined in Sec. 4.3:

$$V(s_{t+1}) \approx \mathscr{R}(s_{t+1}). \tag{5}$$

Backpropagation. The estimated value $v = V(s_{t+1})$ is propagated from the expanded node back to the root. For each visited state-action pair along the path, CataPath updates the visit count and running value estimate. After a fixed search budget or stopping criterion is reached, the algorithm returns the candidate associated with the highest search value.

$$Q(s,a) \leftarrow Q(s,a) + \frac{v - Q(s,a)}{N(s,a)}. \tag{6}$$

4.3 Hybrid Reward Mechanism

CataPath uses a hybrid reward to guide MCTS toward candidates that are both functionally promising and chemically feasible:

$$\mathscr{R}(s_{t+1}) = \lambda \hat{\mathscr{R}}_{\text{prop}}(s_{t+1}) + (1-\lambda)\hat{\mathscr{R}}_{\text{expert}}(s_{t+1}), \tag{7}$$

where $\hat{\mathscr{R}}_{\text{prop}}$ is the normalized property-prediction reward, $\hat{\mathscr{R}}_{\text{expert}}$ is the normalized expert-preference reward, and λ controls their relative contribution. Implementation details and hyperparameter settings are provided in Appendix B.

Property-Prediction Reward. The property-prediction reward provides dense feedback on task-specific material properties. For CO_2RR electrocatalyst design, we use a Faradaic efficiency predictor based on a copper-based material knowledge graph (Gao et al., 2023), which combines CatBERT (Lee et al., 2020) with VGAE representations (Kipf & Welling, 2016). For OSDA structural optimization, we use the binding-energy predictor from hu2025osda, which combines SMILES-BERT (Wang et al., 2019) and CGCNN (Xie & Grossman, 2018) features. These predictors are treated as fixed external evaluators during CataPath search. The motivation for selecting these properties is discussed in Appendix A.1.

Expert Preference Reward Model. Property predictors provide useful theoretical signals, but they may miss practical constraints such as synthesizability and expert design intuition. We therefore introduce an Expert Preference Reward Model (EPRM) to estimate whether a generated reasoning state is consistent with expert judgment.

The EPRM is trained only on expert-scored trajectories from the training split of CataDesignQR. During evaluation, it is frozen and used as a fixed reward model. Instead of directly regressing scalar expert scores, we formulate EPRM training as pairwise preference learning. Given a context and two candidate reasoning steps, the model learns to assign a higher reward to the expert-preferred one. To improve robustness, we include hard negative pairs in which a candidate receives favorable property-prediction scores but low expert feasibility scores. This encourages the EPRM to penalize designs that are theoretically attractive yet practically unrealistic. Details of preference-pair construction, negative mining, and the expert feedback platform are provided in Appendix B.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Evaluation protocol. We evaluate CataPath on two tasks in CataDesignQR: CO_2RR electrocatalyst generation and zeolite-related OSDA molecular optimization. The CO_2RR task is used for the main results, ablation studies, and external analyses, while the OSDA task tests adaptation to a different molecular output space and constraint structure. We use GPT-4 (Achiam et al., 2023) as the base LLM for design generation and Galactica-1.3B (Taylor et al., 2022) as the backbone for EPRM. All task-level results use the held-out design-goal split in Sec. 3.3. The EPRM is trained only on training-split expert trajectories and remains frozen during evaluation.

Baselines. We compare CataPath with two groups of baselines, with detailed settings provided in Appendix D.1: (i) General prompting and agentic reasoning strategies based on GPT-4: Chain-of-Thought (CoT) (Kojima et al., 2022), CoT with Self-Consistency (Wang et al., 2022), Tree-of-Thoughts (ToT) (Yao et al., 2023), ReAct (Yao et al., 2022), Graph of Thoughts (GoT) (Besta et al., 2024), Algorithm of Thoughts (AoT) (Sel et al., 2023), and Language Agent Tree Search (LATS) (Zhou et al., 2023); (ii) Text-based generation models, including general-purpose LLMs and specialized molecular models. To test backbone robustness, we compare direct generation with CataPath using DeepSeek-V3 and Qwen3-32B-Thinking in Table 9. Full adaptation details and prompts are provided in Appendix D.1. We further discuss methodological differences from ChemReasoner (Sprueill et al., 2024), since their original quantum-chemical feedback pipelines are not directly aligned with our benchmark setting.

Metrics. Evaluation metrics are tailored to each task. For CO_2RR de novo design, we use three complementary criteria: reasonableness, proxy effectiveness, and feasibility. Reasonableness measures whether generated strategies match the target regulation category, proxy effectiveness estimates alignment with literature-verified high-performance catalysts using a domain encoder, and feasibility is assessed by domain experts based on chemical validity and synthetic plausibility. For OSDA optimization, we follow established benchmarks and report validity, BLEU, molecular fingerprint similarities, Energy Distance, and KL divergence. Detailed metric definitions and validations are provided in Appendix D.2.

5.2 MAIN RESULTS

5.2.1 CO_2RR ELECTROCATALYST GENERATION

Table 2 reports the results on CO_2RR electrocatalyst de novo generation. CataPath achieves the best performance across all three metrics, improving reasonableness, effectiveness, and feasibility over both direct LLM generation and prompting-based reasoning strategies. These results suggest that constraint-aware trajectory search is more effective than unconstrained reasoning prompts for scientific design tasks (Ye et al., 2023).

General-purpose LLMs improve as model capability increases, with DeepSeek-V3 (Liu et al., 2024a) outperforming GPT-family baselines. However, their feasibility scores remain substantially lower than their reasonableness scores, indicating that fluent and chemically plausible descriptions do not necessarily correspond to experimentally realistic designs. Prompting and agentic reasoning methods such as ToT-BFS and ReAct further improve performance by exploring multiple reasoning paths, but they still lack task-specific intermediate verification. In contrast, CataPath enables the agent to prune infeasible paths and prioritize candidates that better align with both proxy effectiveness and expert feasibility judgments.

5.2.2 OSDA STRUCTURAL OPTIMIZATION

Table 3 reports the results on OSDA molecule optimization for the LTA zeolite framework. CataPath achieves the strongest performance on validity and most structural similarity metrics, including BLEU, Morgan, MACCS, and RDK fingerprints. Compared with MolT5 and BioT5, LLM-based methods substantially improve validity and distributional alignment. However, direct LLM generation still falls behind agent-based methods, indicating the importance of iterative optimization.

Table 2: Performance comparison on CO_2RR electrocatalyst de novo generation over three independent runs. The best scores are in bold.

Method	Reasonableness	Effectiveness	Feasibility
General LLMs			
GPT-4	$0.685_{\pm 0.010}$	$0.622_{\pm 0.013}$	$0.412_{\pm 0.017}$
GPT-4o	$0.753_{\pm 0.009}$	$0.762_{\pm 0.011}$	$0.524_{\pm 0.015}$
GPT-5.5	$0.858_{\pm 0.009}$	$0.842_{\pm 0.011}$	$0.636_{\pm 0.015}$
DeepSeek-V3	$0.821_{\pm 0.008}$	$0.791_{\pm 0.010}$	$0.553_{\pm 0.014}$
Prompting strategies (based on GPT-4)			
CoT	$0.742_{\pm 0.015}$	$0.704_{\pm 0.017}$	$0.487_{\pm 0.020}$
CoT w/Self-consistency	$0.781_{\pm 0.012}$	$0.768_{\pm 0.014}$	$0.542_{\pm 0.018}$
ToT-BFS	$0.832_{\pm 0.011}$	$\underline{0.812}_{\pm 0.013}$	$0.615_{\pm 0.016}$
ReAct	$\underline{0.845}_{\pm 0.012}$	$0.796_{\pm 0.015}$	$\underline{0.655}_{\pm 0.017}$
GoT	$\underline{0.852}_{\pm 0.011}$	$0.829_{\pm 0.014}$	$\underline{0.664}_{\pm 0.010}$
AoT	$\underline{0.834}_{\pm 0.009}$	$0.801_{\pm 0.017}$	$\underline{0.641}_{\pm 0.017}$
LATS	$\underline{0.869}_{\pm 0.014}$	$0.853_{\pm 0.015}$	$\underline{0.706}_{\pm 0.018}$
CataPath	$\mathbf{0.904}_{\pm 0.007}$	$\mathbf{0.913}_{\pm 0.008}$	$\mathbf{0.789}_{\pm 0.012}$

Compared with the strong OSDA Agent baseline, CataPath obtains higher BLEU, Morgan, and RDK scores while maintaining the same validity and MACCS performance. The improvement suggests that CataPath better preserves task-relevant substructural features of experimentally verified OSDAs. The inference-time logging indicates that CataPath expands 43.6 CO_2RR nodes and 31.4 OSDA nodes per query on average, with effective branching factors of 4.82 and 3.61 in Appendix D.4.

Table 3: Performance comparison on OSDA molecule optimization for the small-pore LTA zeolite framework over three independent runs. The best scores are in bold.

Method	Validity ↑	BLEU ↑	Morgan ↑	MACCS ↑	RDK ↑	ED ↓	KL Divergence ↓
Molecule Models							
BioT5	$0.000_{\pm 0.000}$	$0.189_{\pm 0.006}$	$0.148_{\pm 0.005}$	$0.552_{\pm 0.010}$	$0.326_{\pm 0.009}$	$277.6_{\pm 8.4}$	$7.820_{\pm 0.214}$
MolT5	$0.007_{\pm 0.002}$	$0.179_{\pm 0.007}$	$0.136_{\pm 0.006}$	$0.382_{\pm 0.012}$	$0.142_{\pm 0.010}$	$130.7_{\pm 5.9}$	$3.145_{\pm 0.126}$
General LLMs							
GPT-4	$0.432_{\pm 0.018}$	$0.513_{\pm 0.012}$	$0.319_{\pm 0.010}$	$0.742_{\pm 0.011}$	$0.500_{\pm 0.013}$	$4.431_{\pm 0.142}$	$0.659_{\pm 0.023}$
GPT-4o	$0.439_{\pm 0.016}$	$0.521_{\pm 0.011}$	$\underline{0.322}_{\pm 0.009}$	$0.747_{\pm 0.010}$	$0.505_{\pm 0.012}$	$4.189_{\pm 0.128}$	$0.611_{\pm 0.020}$
GPT-5.5	$0.502_{\pm 0.015}$	$0.548_{\pm 0.011}$	$0.348_{\pm 0.010}$	$0.761_{\pm 0.009}$	$\mathbf{0.533}_{\pm 0.011}$	$3.780_{\pm 0.135}$	$\mathbf{0.548}_{\pm 0.022}$
Agent-based LLM (GPT-4)							
OSDA Agent	$\mathbf{1.000}_{\pm 0.000}$	$\underline{0.554}_{\pm 0.010}$	$0.300_{\pm 0.008}$	$\mathbf{0.779}_{\pm 0.007}$	$0.518_{\pm 0.009}$	$\mathbf{2.667}_{\pm 0.084}$	$\underline{0.590}_{\pm 0.018}$
CataPath	$\mathbf{1.000}_{\pm 0.000}$	$\mathbf{0.602}_{\pm 0.008}$	$\mathbf{0.372}_{\pm 0.007}$	$\mathbf{0.779}_{\pm 0.006}$	$\underline{0.523}_{\pm 0.008}$	$\underline{2.682}_{\pm 0.079}$	$0.603_{\pm 0.016}$

5.3 Ablation Study

We conduct ablation studies to examine two key components of CataPath: the structured action space and the hybrid reward mechanism.

Effect of structured action space. We first remove the predefined chemical action space and allow the LLM to generate optimization plans freely. As shown in Fig. 3, this free-form variant performs worse on both CO_2RR catalyst design and OSDA optimization. Without chemically grounded actions, the agent more frequently produces irrelevant or weakly constrained reasoning paths, making the search less efficient and less reliable. In contrast, the structured action space narrows the exploration space and guides MCTS toward chemically meaningful design decisions.

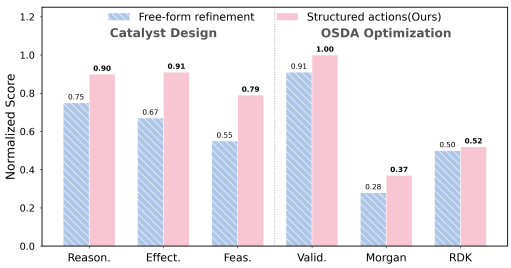

Figure 3: Effect of the structured action space on CO_2RR catalyst design and OSDA optimization.

Effect of hybrid reward. We further isolate the two reward components by removing either the property-prediction reward or the expert-preference reward. As shown in Fig. 4, the full hybrid reward consistently outperforms single-reward variants. Without the expert-preference reward, the search tends to exploit property predictors and generate candidates with high predicted scores but weaker synthesizability or chemical plausibility. Without the property-prediction reward, the agent becomes more conservative and produces feasible candidates that are less optimized for the target property. These results show that the two rewards are complementary: property prediction drives functional optimization, while expert preference regularizes the search toward practical feasibility.

5.4 Qualitative and External Validation

Fig. 5 visualizes a representative CataPath trajectory for CO_2RR electrocatalyst design. Along the search path, the generated content gradually shifts from broad regulation concepts to more specific catalyst design strategies. For example, when refining actions related to atomic-level dispersion, CataPath encourages the LLM to elaborate concrete candidates such as multi-atom configurations and atomic clusters. This illustrates how constrained trajectory search helps maintain reasoning continuity and progressively specialize the design.



Figure 4: Effect of different reward components.

Beyond qualitative inspection, we provide external validation in Appendix D.3. We perform DFT calculations for potentially active CO_2RR surfaces proposed by CataPath and conduct wet-lab characterization for a top-ranked Cu-Sn bimetallic nanostructure. These results provide case-level evidence that CataPath can generate practically actionable candidates.

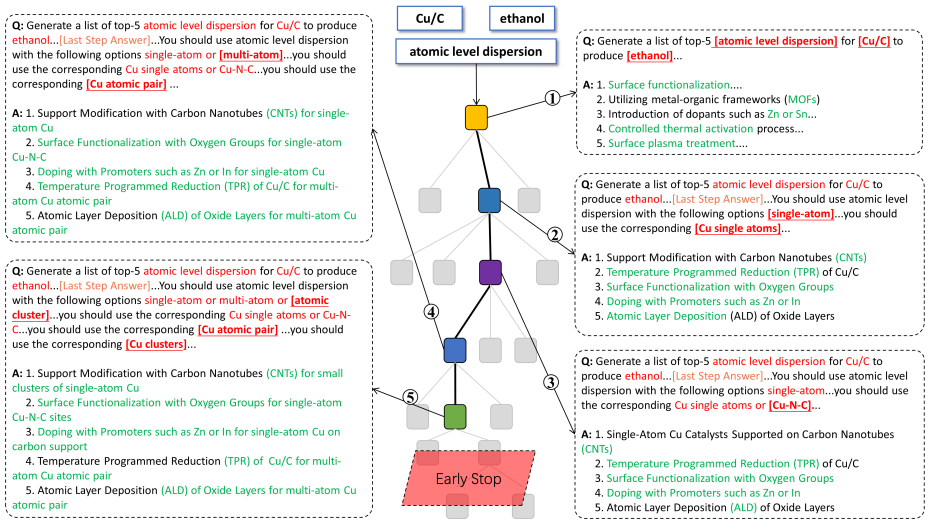

Figure 5: Representative CataPath reasoning trajectory leading to the highest-reward node.

6 Conclusion

We present CataPath, an expert-augmented LLM agent that searches chemically structured reasoning trajectories using MCTS and a hybrid property-prediction and expert-preference reward. We also introduce CataDesignQR, a dataset of literature-grounded design goals, verified solutions, and expert-scored reasoning trajectories for CO_2RR electrocatalyst generation and OSDA optimization. Experiments show that CataPath improves generation validity, proxy effectiveness, and expert alignment across the evaluated tasks. Broader validation across catalyst families and experimental conditions remains future work.

AI USE STATEMENT

In this work, we used generative AI models as the systems under study. LLMs were used to generate catalyst-design candidates, OSDA molecular candidates, and intermediate reasoning trajectories during the CataPath experiments and baseline evaluations. These model-generated outputs are treated as experimental model outputs rather than as experimental measurements.

Additionally, we used AI-assisted tools for language polishing, manuscript organization, and LaTeX formatting. AI-assisted text and LaTeX edits were checked against the source manuscript, experimental records, and cited literature. All reported results, analyses, claims, and final manuscript content were reviewed and approved by the authors, who take responsibility for the final content of this work.

ETHICS STATEMENT

This work uses domain-expert judgments to assess generated catalyst designs, OSDA molecules, and reasoning trajectories. The annotation protocol focuses on the scientific properties of generated candidates and does not report personal or sensitive information about the annotators. Final expert-feasibility assessments are conducted separately from EPRM training and are evaluated without revealing the generating method.

REPRODUCIBILITY STATEMENT

The paper describes the CataDesignQR construction, task definitions, data splits, and leakage-control protocol in Section 3 and Appendix A. The MCTS algorithm and hybrid reward mechanism are described in Section 4 and Appendix B. Search-efficiency experiments and ablations are provided in Appendix C, while baseline implementations, evaluation metrics, external validation protocols, and robustness experiments are detailed in Appendix D. The prompt templates used for CO_2RR and OSDA generation are provided in Appendix E. Together, these sections specify the action spaces, model settings, reward definitions, evaluation procedures, and reported statistics needed to reproduce the experiments. Anonymous source code, prompts, evaluation scripts, and the processed data needed for reproduction are provided as supplementary material.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Mehrad Ansari and Seyed Mohamad Moosavi. Agent-based learning of materials datasets from the scientific literature. *Digital Discovery*, 3(12):2607–2617, 2024.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 17682–17690, 2024.

Andres M Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Chemcrow: Augmenting large-language models with chemistry tools. *arXiv preprint arXiv:2304.05376*, 2023.

Markus J Buehler. Generative retrieval-augmented ontologic graph and multiagent strategies for interpretive large language model-based materials design. *ACS Engineering Au*, 4(2):241–277, 2024.

Xueqing Chen, Yang Gao, Ludi Wang, Wenjuan Cui, Jiamin Huang, Yi Du, and Bin Wang. Large language model enhanced corpus of co2 reduction electrocatalysts and synthesis procedures. *Scientific Data*, 11(1):347, 2024.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, et al. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations*, volume 2025, pp. 96934–96990, 2025.

Yuan Chiang, Elvis Hsieh, Chia-Hong Chou, and Janosh Riebesell. Llamp: Large language model made powerful for high-fidelity materials knowledge retrieval and distillation. *arXiv preprint arXiv:2401.17244*, 2024.

John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S Rosen, Gerbrand Ceder, Kristin A Persson, and Anubhav Jain. Structured information extraction from scientific text with large language models. *Nature communications*, 15(1):1418, 2024.

Kourosh Darvish, Marta Skreta, Yuchi Zhao, Naruki Yoshikawa, Sagnik Som, Miroslav Bogdanovic, Yang Cao, Han Hao, Haoping Xu, Alán Aspuru-Guzik, et al. Organa: A robotic assistant for automated chemistry experimentation and characterization. *Matter*, 8(2), 2025.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. Rlprompt: Optimizing discrete text prompts with reinforcement learning. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pp. 3369–3391, 2022.

Keyan Ding, Jing Yu, Junjie Huang, Yuchen Yang, Qiang Zhang, and Huajun Chen. Scitoolagent: a knowledge-graph-driven scientific agent for multitool integration. *Nature Computational Science*, 5(10):962–972, 2025.

David Dohan, Winnie Xu, Aitor Lewkowycz, Jacob Austin, David Bieber, Raphael Gontijo Lopes, Yuhuai Wu, Henryk Michalewski, Rif A Saurous, Jascha Sohl-Dickstein, et al. Language model cascades. *arXiv preprint arXiv:2207.10342*, 2022.

Yi Du, Ludi Wang, Mengyi Huang, Dongze Song, Wenjuan Cui, and Yuanchun Zhou. Autodive: An integrated onsite scientific literature annotation tool. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 76–85, 2023.

Carl Edwards, Tuan Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, and Heng Ji. Translation between molecules and natural language. *arXiv preprint arXiv:2204.11817*, 2022.

Hehe Fan, Linchao Zhu, Yi Yang, and Mohan Kankanhalli. Pointlistnet: Deep learning on 3d point lists. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17692–17701, 2023.

Yang Gao, Ludi Wang, Xueqing Chen, Yi Du, and Bin Wang. Revisiting electrocatalyst design by a knowledge graph of cu-based catalysts for co2 reduction. *ACS Catalysis*, 13(13):8525–8534, 2023.

Li Hao, He Cao, Bin Feng, Daniel Shao, Robert Tang, Zhiyuan Yan, Yonghong Tian, Li Yuan, and Yu Li. Beyond chemical qa: Evaluating llm's chemical reasoning with modular chemical operations. *Advances in Neural Information Processing Systems*, 38, 2026.

Zhaolin Hu, Yixiao Zhou, Zhongan Wang, Xin Li, Weimin Yang, Hehe Fan, and Yi Yang. Osda agent: Leveraging large language models for de novo design of organic structure directing agents. In *The Thirteenth International Conference on Learning Representations*, 2025.

Zach Jensen, Soonhyoung Kwon, Daniel Schwalbe-Koda, Cecilia Paris, Rafael Gómez-Bombarelli, Yuriy Román-Leshkov, Avelino Corma, Manuel Moliner, and Elsa A Olivetti. Discovering relationships between osdas and zeolites through data mining and generative neural networks. *ACS central science*, 7(5):858–867, 2021.

Yuanyuan Jiang, Guo Zhang, Jing You, Hailin Zhang, Rui Yao, Huanzhang Xie, Liyun Zhang, Ziyi Xia, Mengzhe Dai, Yunjie Wu, et al. Pocketflow is a data-and-knowledge-driven structure-based molecular generative model. *Nature Machine Intelligence*, 6(3):326–337, 2024.

Fangkai Jiao, Chengwei Qin, Zhengyuan Liu, Nancy Chen, and Shafiq Joty. Learning planning-based reasoning by trajectories collection and process reward synthesizing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 334–350, 2024.

Yeonghun Kang and Jihan Kim. Chatmof: an artificial intelligence system for predicting and generating metal-organic frameworks using large language models. *Nature communications*, 15 (1):4705, 2024.

Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

Younghoo Lee, Joshua Saxe, and Richard Harang. Catbert: Context-aware tiny bert for detecting social engineering emails. *arXiv preprint arXiv:2010.03484*, 2020.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024a.

Sizhe Liu, Yizhou Lu, Siyu Chen, Xiyang Hu, Jieyu Zhao, Yingzhou Lu, and Yue Zhao. Drugagent: Automating ai-aided drug discovery programming through llm multi-agent collaboration. *arXiv preprint arXiv:2411.15692*, 2024b.

Ziming Luo, Zonglin Yang, Zexin Xu, Wei Yang, and Xinya Du. Llm4sr: A survey on large language models for scientific research. *arXiv preprint arXiv:2501.04306*, 2025.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535, 2024.

Yujie Ma, Yang Gao, Ludi Wang, Ming Chen, Wenjuan Cui, Bin Wang, and Yi Du. Accelerating materials discovery through active learning: Methods, challenges and opportunities. *The Innovation Informatics*, 1(1):100013–1, 2025.

Adrian Mirza, Nawaf Alampara, Sreekanth Kunchapu, Martiño Ríos-García, Benedict Emoekabu, Aswanth Krishnan, Tanya Gupta, Mara Schilling-Wilhelmi, Macjonathan Okereke, Anagha Aneesh, et al. Are large language models superhuman chemists? *arXiv preprint arXiv:2404.01475*, 2024.

Qizhi Pei, Wei Zhang, Jinhua Zhu, Kehan Wu, Kaiyuan Gao, Lijun Wu, Yingce Xia, and Rui Yan. Biot5: Enriching cross-modal integration in biology with chemical knowledge and natural language associations. *arXiv preprint arXiv:2310.07276*, 2023.

Daniel Schwalbe-Koda and Rafael Gómez-Bombarelli. Supramolecular recognition in crystalline nanocavities through monte carlo and voronoi network algorithms. *The Journal of Physical Chemistry C*, 125(5):3009–3017, 2021.

Daniel Schwalbe-Koda, Soonhyoung Kwon, Cecilia Paris, Estefania Bello-Jurado, Zach Jensen, Elsa Olivetti, Tom Willhammar, Avelino Corma, Yuriy Román-Leshkov, Manuel Moliner, et al. A priori control of zeolite phase competition and intergrowth with high-throughput simulations. *Science*, 374(6565):308–315, 2021.

Bilgehan Sel, Ahmad Al-Tawaha, Vanshaj Khattar, Ruoxi Jia, and Ming Jin. Algorithm of thoughts: Enhancing exploration of ideas in large language models. *arXiv preprint arXiv:2308.10379*, 2023.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. In *International Conference on Learning Representations*, volume 2025, pp. 60808–60838, 2025.

Zijing Shi, Meng Fang, and Ling Chen. Monte carlo planning with large language model for text-based game agents. In *International Conference on Learning Representations*, volume 2025, pp. 52924–52944, 2025.

Mingyang Song, Zhaochen Su, Xiaoye Qu, Jiawei Zhou, and Yu Cheng. Prmbench: A fine-grained and challenging benchmark for process-level reward models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25299–25346, 2025.

Yifan Song, Weimin Xiong, Xiutian Zhao, Dawei Zhu, Wenhao Wu, Ke Wang, Cheng Li, Wei Peng, and Sujian Li. Agentbank: Towards generalized llm agents via fine-tuning on 50000+ interaction trajectories. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 2124–2141, 2024.

Henry Sprueill, Carl Edwards, Mariefel Olarte, Udishnu Sanyal, Heng Ji, and Sutanay Choudhury. Monte carlo thought search: Large language model querying for complex scientific reasoning in catalyst design. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 8348–8365, 2023.

Henry W Sprueill, Carl Edwards, Khushbu Agarwal, Mariefel V Olarte, Udishnu Sanyal, Conrad Johnston, Hongbin Liu, Heng Ji, and Sutanay Choudhury. Chemreasoner: Heuristic search over a large language model's knowledge space using quantum-chemical feedback. *arXiv preprint arXiv:2402.10980*, 2024.

Manu Suvarna, Alain Claude Vaucher, Sharon Mitchell, Teodoro Laino, and Javier Pérez-Ramírez. Language models and protocol standardization guidelines for accelerating synthesis planning in heterogeneous catalysis. *Nature communications*, 14(1):7964, 2023.

Xiangru Tang, Anni Zou, Zhuosheng Zhang, Ziming Li, Yilun Zhao, Xingyao Zhang, Arman Cohan, and Mark Gerstein. Medagents: Large language models as collaborators for zero-shot medical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 599–621, 2024.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022.

Vahe Tshitoyan, John Dagdelen, Leigh Weston, Alexander Dunn, Ziqin Rong, Olga Kononova, Kristin A Persson, Gerbrand Ceder, and Anubhav Jain. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature*, 571(7763):95–98, 2019.

Ludi Wang, Yang Gao, Xueqing Chen, Wenjuan Cui, Yuanchun Zhou, Xinying Luo, Shuaishuai Xu, Yi Du, and Bin Wang. A corpus of co2 electrocatalytic reduction process extracted from the scientific literature. *Scientific Data*, 10(1):175, 2023.

Ludi Wang, Xueqing Chen, Yi Du, Yuanchun Zhou, Yang Gao, and Wenjuan Cui. Catalm: empowering catalyst design through large language models. *International Journal of Machine Learning and Cybernetics*, 16(5):3681–3691, 2025a.

Ludi Wang, Dongze Song, Qiang Cui, Xueqing Chen, Yuanchun Zhou, Wenjuan Cui, and Yi Du. Autodive+: An adaptive model enhanced multimodal online annotation tool. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 2919–2922, 2025b.

Sheng Wang, Yuzhi Guo, Yuhong Wang, Hongmao Sun, and Junzhou Huang. Smiles-bert: large scale unsupervised pre-training for molecular property prediction. In *Proceedings of the 10th ACM international conference on bioinformatics, computational biology and health informatics*, pp. 429–436, 2019.

Xinming Wang, Jian Xu, Aslan H Feng, Yi Chen, Haiyang Guo, Fei Zhu, Yuanqi Shao, Minsi Ren, Hongzhu Yi, Sheng Lian, et al. The hitchhiker's guide to autonomous research: A survey of scientific agents. *TechRxiv.August 07, 2025. DOI:10.36227/techrxiv175459840.02185500/V1*, 2025c.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Jiaqi Wei, Yuejin Yang, Xiang Zhang, Yuhan Chen, Xiang Zhuang, Zhangyang Gao, Dongzhan Zhou, Guangshuai Wang, Zhiqiang Gao, Juntai Cao, et al. From ai for science to agentic science: A survey on autonomous scientific discovery. *arXiv preprint arXiv:2508.14111*, 2025.

Qingpo Wuwu, Chonghan Gao, Tianyu Chen, Yihang Huang, Yuekai Zhang, Jianing Wang, Jianxin Li, Haoyi Zhou, and Shanghang Zhang. Pinnsagent: Automated pde surrogation with large language models. *arXiv preprint arXiv:2501.12053*, 2025.

Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Xin Guo, Dingwen Yang, Chenyang Liao, Wei He, et al. Agentgym: Evaluating and training large language model-based agents across diverse environments. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27914–27961, 2025.

Tian Xie and Jeffrey C Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Physical review letters*, 120(14):145301, 2018.

Tong Xie, Yuwei Wan, Yixuan Liu, Yuchen Zeng, Shaozhou Wang, Wenjie Zhang, Clara Grazian, Chunyu Kit, Wanli Ouyang, Dongzhan Zhou, et al. Darwin 1.5: Large language models as materials science adapted learners. *arXiv preprint arXiv:2412.11970*, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, et al. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420*, 2023.

Di Zhang, Wei Liu, Qian Tan, Jingdan Chen, Hang Yan, Yuliang Yan, Jiatong Li, Weiran Huang, Xiangyu Yue, Wanli Ouyang, et al. Chemllm: A chemical large language model. *arXiv preprint arXiv:2402.06852*, 2024.

Yizhen Zheng, Huan Yee Koh, Jiaxin Ju, Anh TN Nguyen, Lauren T May, Geoffrey I Webb, and Shirui Pan. Large language models for scientific discovery in molecular property prediction. *Nature Machine Intelligence*, pp. 1–11, 2025.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models. *arXiv preprint arXiv:2310.04406*, 2023.

Jiacheng Zhou, Bo Xiao, Qi Liu, Lifeng Liu, and Lei Zhang. Matpc: Prompting large language model, crystal structure prediction, and first-principles for semantic-driven material design. *ACS Applied Materials & Interfaces*, 17(31):44528–44540, 2025.

A CataDesignQR Dataset Construction

A.1 Background and Motivation

A.1.1 CO_2RR Electrocatalysts De Novo Design

The electrochemical CO_2 reduction reaction (CO_2RR) is a promising route for converting carbon dioxide into value-added chemical feedstocks using renewable electricity. Efficient CO_2RR catalysts are essential for improving activity, selectivity, and stability, especially when targeting multi-carbon products such as ethylene and ethanol. However, CO_2RR involves complex multi-electron transfer pathways and competing side reactions such as hydrogen evolution, making catalyst design a challenging multi-objective optimization problem. A desirable catalyst must tune intermediate binding energies while remaining thermodynamically stable and synthetically feasible.

CO_2RR catalyst design is also difficult from a generative modeling perspective. Unlike small organic molecules, which can often be represented as discrete strings, heterogeneous electrocatalysts involve surface structures, alloy compositions, defects, coordination environments, and interface effects. These factors create a large and highly constrained design space that is difficult to explore through direct open-ended generation. As a result, autonomous de novo design requires more than pattern matching over known materials: the model must make chemically coherent intermediate decisions and satisfy multiple domain constraints throughout the design process. This motivates our formulation of CO_2RR catalyst design as a structured reasoning and search problem over chemically meaningful actions.

A.1.2 OSDA Molecule Optimization

Organic Structure-Directing Agents (OSDAs) play a central role in zeolite synthesis. Zeolites are crystalline microporous materials widely used in industrial catalysis, adsorption, and separation. During hydrothermal synthesis, OSDA molecules act as organic templates that guide the assembly of inorganic precursors through non-covalent interactions. Their size, shape, charge distribution, and binding affinity influence the resulting zeolite topology, pore dimensions, framework composition, and the accessibility of framework active sites. These structural features subsequently affect reactant diffusion, confinement effects, acid-site environments, and product selectivity in catalytic reactions. Therefore, OSDA optimization provides an indirect but important route to catalyst-related material design. If an OSDA is incompatible with the target framework, synthesis may yield amorphous products, undesired phases, or zeolites with unsuitable pore architectures.

OSDA optimization is challenging because suitable molecules must satisfy multiple constraints, including chemical validity, synthetic feasibility, framework compatibility, and favorable binding interactions. Traditional discovery often relies on trial-and-error experiments or computationally expensive simulations, which are difficult to scale across the large space of possible organic candidates. Meanwhile, data scarcity limits the effectiveness of standard generative models, especially when optimizing for specific zeolite frameworks. These challenges make OSDA optimization a suitable testbed for constrained molecular reasoning, where an LLM agent must make chemically valid structural modifications while preserving task-specific functionality. This motivates our use of a structured action space and expert-informed feedback for guiding OSDA molecule optimization.

A.1.3 Motivation

The two tasks above share a common challenge: valid material design requires models to make intermediate decisions that satisfy domain-specific constraints, rather than simply generate plausible final outputs. To support this form of process-level scientific reasoning, we introduce **CataDesignQR**. The dataset organizes catalyst and OSDA design problems around structured reasoning trajectories, where each step corresponds to a chemically meaningful action and is associated with expert feedback. These trajectories provide supervision for learning expert-aligned reward signals and enable systematic evaluation of whether LLM agents can search through constrained scientific design spaces.

A.2 Data Source

We construct CataDesignQR from four public data sources covering CO_2RR electrocatalysis and OSDA-guided zeolite synthesis. These sources provide literature-grounded design goals, verified catalyst or OSDA records, and property-related references for building reasoning trajectories and evaluation targets.

- **ElectroCatalytic Reduction 1 (Wang et al., 2023)**[1]: An open-source corpus of electrocatalytic CO_2 reduction records extracted from scientific literature (Du et al., 2023).
- **ElectroCatalytic Reduction 2 (Chen et al., 2024)**[2]: A text-mining dataset focused on copper-based CO_2RR electrocatalysts (Wang et al., 2025b). It includes catalyst materials, regulation strategies, target products, Faradaic efficiency values, and synthesis-related conditions.
- **Jensen dataset (Jensen et al., 2021)**[3]: An experimental zeolite synthesis dataset containing verified synthesis routes that link OSDA molecules, synthesis gel compositions, and resulting zeolite crystalline phases.
- **OSDB database (Schwalbe-Koda et al., 2021)**[4]: A computational database that provides thermodynamic interaction information between organic templates and porous zeolite frameworks.

A.3 Reasoning Strategy Set

The reasoning strategy set defines the action space used by CataPath. It maps abstract design objectives to discrete, domain-grounded operations that can be selected during trajectory search. For CO_2RR electrocatalyst design, the action space covers six regulation categories, including alloy, defect, atomic-level dispersion, surface modification, composite construction, and structure control. These categories reflect common experimental strategies for tuning catalyst composition, active sites, and surface morphology. For OSDA optimization, the action space consists of molecular modification operations related to chemical validity, synthetic feasibility, and binding-energy improvement. These actions constrain the LLM agent to search through chemically meaningful intermediate steps rather than unconstrained free-form generation. Table 4 and Table 5 summarize the two action spaces.

A.4 Data Split and Leakage Control Details

CataDesignQR is designed to serve two roles: providing process-level supervision for reward-model training and supporting held-out evaluation of constraint-aware scientific reasoning. To avoid leakage between these two roles, we organize and split the dataset at the design-goal level.

Each design goal contains an input query, the target material or molecular setting, and the desired product or zeolite framework. For CO_2RR electrocatalyst design, a design goal specifies the base material, target reduction product, and regulation category. For OSDA optimization, a design goal specifies the initial OSDA molecule and the target zeolite framework. For each design goal, CataDesignQR includes multiple LLM-generated reasoning trajectories, expert feedback scores, and literature-grounded reference solutions.

We use an 80/20 design-goal-level split for training and final task evaluation. All trajectories associated with the same design goal are assigned to the same split. This prevents the Expert Preference Reward Model (EPRM) from observing alternative reasoning paths for a held-out test query during training. The 80% training split is used to construct expert-scored trajectories and preference pairs for EPRM training, while the 20% held-out design-goal split is reserved exclusively for final evaluation of CataPath and all baselines. Within the training split, we further reserve 10% of the constructed preference pairs as an EPRM validation set for model selection and diagnostic evaluation. During evaluation, CataPath receives only the design goal and task-specific constraints as

[1] https://doi.org/10.57760/sciencedb.07106

[2] https://doi.org/10.57760/sciencedb.13290

[3] https://github.com/olivettigroup/OSDA_Generator

[4] https://zeodb.mit.edu/index

Table 4: Overview of the domain-specific action space for CO_2RR electrocatalyst design.

Defined Action Categories	Keywords	Action numbers	Classification basis	Examples
Alloy	CO-producing metal	5	main product	Ag, Sn, Pd
	HCOOH-producing metal	3		
	H_2-producing metal	3		
Defect	grain boundary	1	defect type	grain boundary, lattice defect, under-coordinated Cu lattice defects
	lattice defect	2		
	cation vacancy	1		
	anion vacancy	2		
	doping	5		
Atomic-level dispersion	single-atom	7	existence form	Cu-N-4 sites, CuO_5 sites, Ni-Cu atomic pair
	multi-atom	3		
	atomic cluster	1		
Surface modification	nonmetal	6	modified substance	phosphate, tricomponent copolymer, dihexadecyl dimethyl ammonium
	ionic liquid	1		
	polymer	6		
	organic molecular compound	7		
Composite	carbon	9	substance containing	Co_3O_4, $Al(OH)_3$, reduced graphene oxide (rGO)
	metal	14		
	metal oxide	15		
	metal hydroxide	3		
	metal sulfide	3		
Structure control	composite control	2	control form	nanoporous structure, hierarchical core/shell structure, dendritic structures with low porosity
	phase control	3		
	facet control	7		
	size control	3		
	morphology control	18		
	3D structure	6		
	core-shell structure	4		
	dendritic structure	7		
	porous structure	5		

Table 5: Overview of the domain-specific action space for OSDA molecule optimization.

Defined Action Categories	Keywords	Action numbers	Classification basis	Examples
Chemical Rules	Forbidden Atom Removal	6	Chemical rules & stability limits	Remove O/S/P atoms, remove carbonyl (C=O), expand 3-membered rings
	Unstable Group Deletion	4		
	C/N Ratio Adjustment	2		
	Ring Size Correction	2		
	Amine Type Correction	1		
	Flexibility Limit	1		
	Valency Check	1		
	Charge Balance	1		
Synthetic Feasibility	Complexity Reduction	3	Synthetic accessibility	Simplify fused rings, remove chiral centers, prune branched chains
	Symmetry Enhancement	1		
	Substituent Simplification	2		
	Fragment Replacement	5		
Binding Energy	Void Filling	2	Energy minimization & shape matching	Elongate alkyl chain, add bulky adamantyl group, introduce cyclic rigidity, align aspect ratio
	Shape Matching	2		
	Rigidity Control	2		
	Hydrophobicity Tuning	2		
	Charge Positioning	1		

input. The corresponding reference solutions and expert scores are not provided to the LLM agent, the search algorithm, or the reward model during generation. They are used only after generation for automatic evaluation or expert assessment.

We further apply task-specific deduplication checks. For the CO_2RR subset, we remove duplicate query–material–target tuples across splits and check overlapping catalyst–product–regulation tuples before evaluation. This reduces the possibility that a model is evaluated on a catalyst design pattern that has already appeared in reward-model training. For the OSDA subset, we canonicalize generated and reference SMILES strings and remove duplicated OSDA molecules and framework–molecule pairs across splits. These checks ensure that test instances correspond to unseen design goals.

The property-prediction models used in the hybrid reward are treated as fixed external evaluators. They are not updated with held-out CataDesignQR evaluation trajectories, expert scores, or reference solutions. Therefore, the reward signal available to CataPath at test time consists only of predictions from fixed property models and the EPRM trained on the training split. The literature-grounded references in the held-out split are used exclusively for post-hoc evaluation.

A.5 Quality Assurance Details

To ensure the reliability of expert feedback, we use a standardized annotation and quality-control protocol. We recruit five domain experts, all with doctoral-level training in materials science, catalysis, or organic synthesis. Experts evaluate generated catalyst designs and OSDA molecules according to three criteria: chemical validity, thermodynamic plausibility, and practical synthesizability. These criteria are used to assign feedback scores that later serve as process-level supervision for the Expert Preference Reward Model.

Each generated candidate is independently reviewed by at least two experts in a double-blind setting, where annotators do not know the generating method or each other's scores. When the assigned scores differ substantially, the case is reviewed by an additional senior expert to reach an adjudicated label. We measure annotation consistency using **Fleiss' Kappa** and obtain a **score of 0.82**, indicating substantial inter-annotator agreement. This protocol reduces individual bias and improves the reliability of the expert-scored trajectories used in CataDesignQR.

B Hybrid Reward Mechanism Details

This section provides implementation details for the hybrid reward used in CataPath. Consistent with Sec. 4.3, each newly generated state s_{t+1} is first assigned a state-level reward by combining the property-prediction signal and the expert-preference signal:

$$\mathscr{R}(s_{t+1}) = \lambda \hat{\mathscr{R}}_{\text{prop}}(s_{t+1}) + (1-\lambda)\hat{\mathscr{R}}_{\text{expert}}(s_{t+1}), \tag{8}$$

where $\hat{\mathscr{R}}_{\text{prop}}$ and $\hat{\mathscr{R}}_{\text{expert}}$ are rescaled to comparable ranges within each task. In our experiments, λ controls the trade-off between task-specific property optimization and expert-aligned feasibility.

During MCTS, the value of an expanded node is estimated from the reward of its successor states with depth discounting:

$$V_{\text{update}}(s_t, a_t) = \sum_{s' \in \text{Succ}(s_t, a_t)} \gamma^{d(s')} \mathscr{R}(s'), \tag{9}$$

where $\text{Succ}(s_t, a_t)$ denotes the successor states expanded from action a_t, $d(s')$ is the depth of successor state s', and γ is the depth-discount factor. We set $\gamma = 0.9$ and $c_{\text{puct}} = 15$ in the UCT selection policy. The maximum search depth $d_{\max}$ and node budget N_p control the depth and branching of the search tree. The reward trade-off parameter is set to $\lambda = 0.4$, corresponding to a larger weight on expert-preference feedback for feasibility regularization.

B.1 Property Prediction Model

This section describes the fixed property-prediction models used to compute $\hat{\mathscr{R}}_{\text{prop}}$ in the hybrid reward. These models provide dense task-specific feedback during MCTS search and are not updated with held-out CataDesignQR evaluation trajectories.

Faradaic Efficiency Estimation Model. For CO_2RR electrocatalyst design, we use a knowledge-enhanced Faradaic efficiency prediction model (Gao et al., 2023). The model connects textual descriptions of catalyst materials and regulation strategies with quantitative CO_2RR performance records. It combines semantic features extracted by CatBERT (Lee et al., 2020) with graph-structured representations learned by a Variational Graph Auto-Encoder (VGAE) (Kipf & Welling, 2016) over a copper-based catalyst knowledge graph. The fused representation is passed to a regression head to estimate Faradaic efficiency. In CataPath, the predicted value is rescaled to a comparable reward range and used as the property-prediction signal for CO_2RR candidate states. The model is trained on 10,316 electrocatalyst entities and reports an R^2 of 0.84 in the original benchmark.

Binding Energy Estimation Model. For OSDA structural optimization, we use the binding-energy surrogate model from hu2025osda. The model represents OSDA molecules with a pre-trained SMILES-BERT encoder (Wang et al., 2019) and represents zeolite framework structures with Crystal

Graph Convolutional Networks (CGCNN) (Xie & Grossman, 2018). These molecular and framework embeddings are fused through an attention-based module and passed to a regression head to estimate OSDA–zeolite binding energy. During search, candidate molecules are docked into zeolite pores with the VOID package to obtain plausible binding poses (Schwalbe-Koda & Gómez-Bombarelli, 2021). The lowest predicted binding energy is used as the property signal, with lower energy corresponding to a higher reward after rescaling. The surrogate model is trained on approximately 78,000 OSDA-zeolite pairs and reports an R^2 of 0.89 in the original benchmark.

B.2 Expert Preference Reward Model

B.2.1 Human feedback annotation platform

To collect process-level expert feedback for EPRM training, we develop an interactive annotation platform for MCTS-guided catalyst design. Taking the CO_2RR task as an example, each annotation session starts from a design goal consisting of a base material, target product, and regulation category. The platform then presents LLM-generated candidate responses along the current search trajectory, together with editable action constraints drawn from the predefined strategy set.(Dohan et al., 2022).

As shown in Fig. 6, during each interaction round, experts review the generated candidate and assign a score from 1 to 5 according to chemical validity, synthetic feasibility, and practical plausibility. Experts may also modify the selected constraints to guide the next generation step. When the expert considers the current candidate satisfactory, or when further refinement no longer improves the result, the session is terminated. The platform records the full interaction trace, including the design goal, selected actions, prompt constraints, generated candidates, expert scores, and final decision.

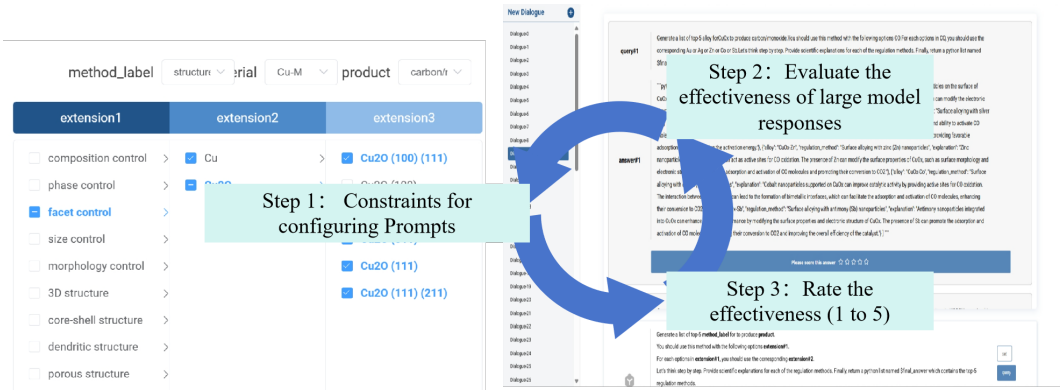

Figure 6: Interface and workflow of the human feedback annotation platform.

To improve annotation efficiency, the platform supports targeted expert intervention during search. When the score difference between the two highest-valued candidate nodes is below a threshold, the system requests expert scoring to resolve uncertain preferences. Experts can also manually correct a trajectory when the search path diverges from their domain judgment. In total, we collect 2,906 reasoning trajectories with expert feedback, which are used to construct the process-level supervision for CataDesignQR and the preference pairs for EPRM training.

Fig. 7 summarizes the distribution of expert scores under different input conditions. The statistics show how feedback varies across product categories, regulation strategies, and base materials. For example, trajectories associated with monoxide production, structure-control strategies, and Cu-based materials tend to receive higher expert scores. Among the analyzed combinations, **(Cu-M, monoxide, alloy)** receives the strongest average feedback. These statistics provide additional evidence that expert preferences are sensitive to both task goals and selected design actions, supporting the use of action-conditioned preference modeling in EPRM.

B.2.2 Data Collection and Pre-processing

The human feedback collected by the platform contains two complementary types of supervision. The first type is the expert score assigned to each generated state node in the MCTS trajectory. These

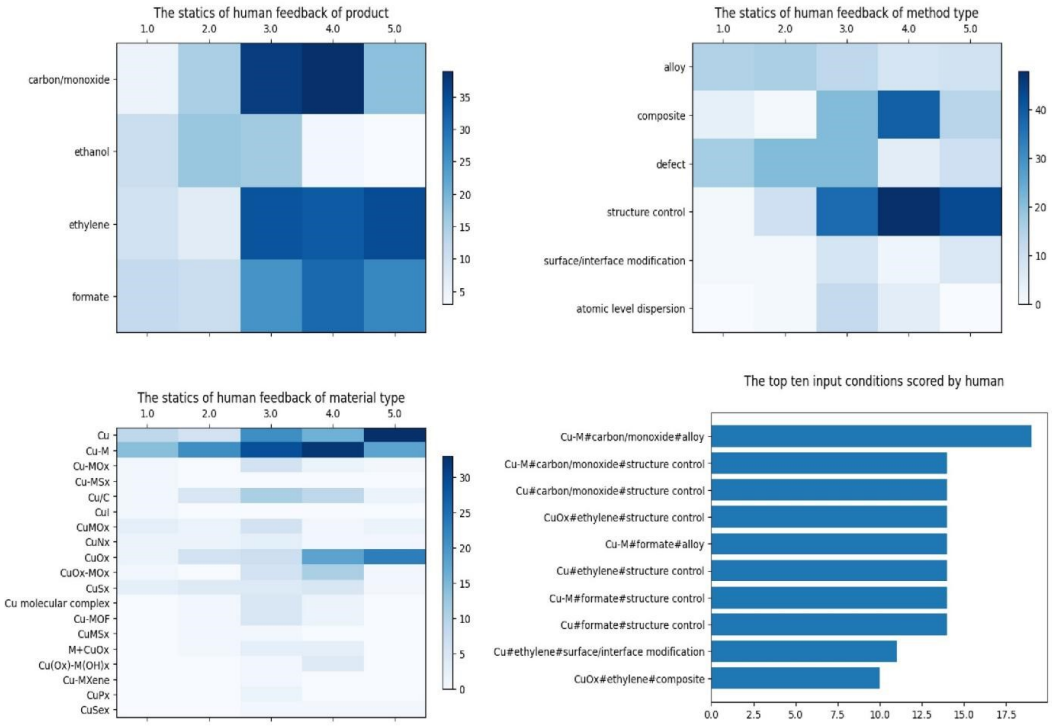

Figure 7: Distribution of expert feedback scores under different input conditions.

node-level scores are used to construct training signals for the EPRM. Since the same node may be reviewed by multiple experts, we aggregate repeated annotations by averaging their scores. Each trajectory is stored with its design goal, action sequence, generated candidates, and corresponding expert scores.

The second type is the action-selection preference reflected in expert constraint edits during interaction. Before each new generation step, experts may add, remove, or modify action constraints from the predefined strategy set. We record these behavioral signals as $[\texttt{id}, \texttt{attribute_list}, \texttt{previous_score}, \texttt{action}]$, where `id` identifies the annotation session, `attribute_list` denotes the selected constraints in the current prompt, `previous_score` is the expert score of the preceding response, and `action` records the constraint-editing operation. These records capture how experts adjust the reasoning trajectory after observing intermediate outputs.

During pre-processing, we remove incomplete interaction records, normalize expert scores to a comparable range, and align each score with its corresponding state-action context. The resulting data are used in two ways: node-level scores support preference-pair construction for EPRM training, while action-selection statistics provide priors for pruning unlikely actions in the efficiency optimization experiments. Details of the search-efficiency analysis are provided in Appendix C.

B.2.3 Pairwise Sample Construction

The EPRM training data are constructed from expert-annotated trajectories in CataDesignQR. Each annotated step contains a design context c, a selected action or modification a, the generated candidate, and an expert score $s \in [1,5]$. We convert scalar expert scores into pairwise preference samples. For each design context, we compare candidate actions that share the same input condition and construct a preference pair when their expert-score difference exceeds a margin δ:

$$\mathscr{D} = \{(c, a_w, a_l) \mid s(a_w) - s(a_l) > \delta\}, \tag{10}$$

where a_w denotes the expert-preferred action and a_l denotes the less preferred action. We set $\delta = 1$ in our experiments to retain pairs with clear preference signals and reduce noise from marginal score differences.

To make the reward model sensitive to chemically deceptive candidates, we further apply hard negative mining. Specifically, we include pairs where the disfavored action a_l achieves a high property-prediction score but receives a low expert feasibility score. By contrasting these hard negatives with expert-preferred actions, the EPRM learns to complement property predictors and penalize candidates that optimize proxy objectives while violating practical design constraints.

After applying the margin filter, we obtain 8,742 preference pairs from the expert-scored trajectories. Among them, 1,386 pairs are identified as hard-negative pairs, where the disfavored candidate receives a relatively high property-prediction score but a low expert feasibility score. We split these preference pairs into 90% for EPRM training and 10% for EPRM validation. Importantly, all preference pairs are constructed only from the training design-goal split described in Appendix A.4.

B.3 EPRM Training and Evaluation

The EPRM is initialized from the pre-trained scientific language model Galactica-1.3B (Taylor et al., 2022). We append a linear reward head to the final hidden representation to produce a scalar reward $r_\theta(c,a)$ for a context-action pair. The model is trained with the Bradley–Terry pairwise ranking objective:

$$\mathscr{L}(\theta) = -\mathbb{E}_{(c,a_w,a_l)\sim\mathscr{D}}\left[\log\sigma\left(r_\theta(c,a_w) - r_\theta(c,a_l)\right)\right], \tag{11}$$

where a_w and a_l denote the expert-preferred and less preferred actions, respectively, and σ is the sigmoid function.

We evaluate EPRM on the 10% held-out preference-pair validation set constructed from training-split design goals. This validation set is used only for reward-model diagnostics and is disjoint from the final held-out design-goal split used to evaluate CataPath. Pairwise accuracy is defined as the percentage of cases where the model assigns a higher reward to the expert-preferred action. EPRM achieves a pairwise accuracy of 81.4%, indicating that it captures a substantial portion of expert preference patterns. We also compute the Spearman rank correlation between predicted rewards and original expert scores, obtaining a correlation of 0.68. These diagnostic results suggest that EPRM provides a useful feasibility-oriented reward signal for CataPath, while final task performance is evaluated separately on the held-out design-goal split described in Sec. 3.3.

C Efficiency Optimization Experiments

C.1 Search Strategy

CataPath searches over a large discrete action space, where exhaustive expansion can quickly become computationally expensive. To improve search efficiency, we introduce two lightweight mechanisms: expert-prior action pruning and semantic-convergence early stopping.

Expert-prior action pruning. During the expansion phase, CataPath filters candidate actions using statistics collected from the expert feedback platform. Actions that are rarely selected or consistently assigned low expert scores under similar design contexts are assigned lower expansion priority. This biases the search toward operations that are more likely to be useful for the current material, target product, and regulation category. In this way, the search can allocate more budget to promising reasoning paths.

Semantic-convergence early stopping. We further terminate branches that show little semantic progress across consecutive states. For CO_2RR design, we encode the textual descriptions of the parent and child candidates using CatBERT and compute their cosine similarity. If the similarity exceeds a predefined threshold, the child node is treated as semantically redundant and the branch is stopped. For OSDA optimization, an analogous check can be applied using molecular similarity between consecutive SMILES strings. This criterion prevents repeated LLM calls when newly generated candidates make only negligible changes to the current state.

Table 6: Ablation results for different maximum search depths.

Tree depth	CataPath			CataPath with efficiency optimization		
	Reasonableness	Effectiveness	Time	Reasonableness	Effectiveness	Time
$d_{\max} = 3$	0.7872	0.7954	87 min	0.7754	0.7616	67 min
$d_{\max} = 5$	0.8469	0.8856	103 min	0.8212	0.8759	89 min
$d_{\max} = 7$	0.8516	0.8907	124 min	0.8492	0.8849	96 min
$d_{\max} = 9$	0.8664	0.8982	136 min	0.8731	0.9021	116 min
$d_{\max}$ unlimited[1]	0.8802	0.9043	162 min	N/A	N/A	N/A

[1] Due to the setup of the efficiency-optimization experiment, the unlimited $d_{\max}$ for CataPath is 10, whereas the unlimited $d_{\max}$ for CataPath with efficiency optimization is 9.

Together with the maximum depth $d_{\max}$ and node budget N_p, these mechanisms reduce unnecessary tree expansion while maintaining the ability to search along deeper, chemically meaningful trajectories.

C.2 Additional Ablation Study

Effect of efficiency optimization. Table 6 shows that efficiency optimization consistently reduces search time while maintaining comparable performance. For example, at $d_{\max} = 9$, the optimized version achieves slightly higher reasonableness and effectiveness than the original CataPath, while reducing runtime from 136 minutes to 116 minutes. Across different depths, the optimized search avoids unnecessary expansion by using expert-prior pruning and semantic-convergence stopping, allowing more computation to be allocated to useful reasoning paths.

Effect of search depth. Increasing $d_{\max}$ generally improves both reasonableness and effectiveness, indicating that deeper search helps CataPath refine catalyst designs through more informative intermediate steps. However, the runtime also increases substantially as the search tree grows. Very shallow depths, such as $d_{\max} = 3$, limit the model's ability to explore meaningful combinations of regulation strategies, while larger depths provide better accuracy at higher computational cost. These results suggest that $d_{\max}$ controls an important trade-off between reasoning quality and search efficiency.

D Experimental Details

D.1 Baseline Methods

We compare CataPath with three groups of baselines: prompting-based reasoning methods, general-purpose LLMs, and domain-specific molecular generation models. Unless otherwise specified, GPT-4 is used as the base model for prompting and agentic reasoning baselines to ensure a comparable backbone.

Chain-of-Thought (CoT). CoT prompting encourages the LLM to generate intermediate reasoning steps before producing the final answer (Kojima et al., 2022; Wei et al., 2022). We implement this baseline by appending "Let's think step by step" to the standard task prompt. The model generates a single reasoning trace and final candidate without external feedback or search.

CoT with Self-Consistency. Self-Consistency improves CoT by sampling multiple reasoning paths and aggregating their outputs (Wang et al., 2022). For each query, we sample 10 CoT responses under the same prompt template and select the final answer by majority voting.

Tree-of-Thoughts (ToT). ToT formulates LLM reasoning as a search process over intermediate textual states (Yao et al., 2023). We implement ToT with breadth-first search, using a branching factor

of $b = 6$ and a maximum depth of $d_{\max} = 5$. To make the comparison computationally controlled, we limit the total number of generated nodes to approximately $N_p = 501$ per query.

ReAct. ReAct interleaves reasoning and action generation, allowing the model to iteratively update its plan based on intermediate thoughts (Yao et al., 2022). In our setting, ReAct reasons about the current chemical constraints and proposes regulation or optimization actions before producing the final candidate. Unlike CataPath, ReAct does not use a chemically predefined action space or hybrid reward feedback.

General-purpose LLMs. We evaluate direct generation with GPT-3.5-turbo, GPT-4, GPT-4o (Achiam et al., 2023), and DeepSeek-V3 (Liu et al., 2024a). These models are prompted with the same task descriptions and output requirements as CataPath but generate candidates directly without MCTS or reward-guided refinement.

Domain-specific molecular models. For OSDA optimization, we also compare against MolT5 (Edwards et al., 2022) and BioT5 (Pei et al., 2023), which are text-to-text models adapted to molecular representation and generation tasks. These baselines test whether molecular pretraining alone is sufficient for constrained OSDA optimization.

OSDA Agent. We include OSDA Agent (Hu et al., 2025) as a strong agent-based baseline for OSDA structural optimization. It follows a generation-evaluation-reflection workflow using LLM feedback and task-specific evaluation signals. Compared with OSDA Agent, CataPath differs by explicitly searching over a chemically informed action space and using the hybrid reward to guide intermediate trajectory selection.

GoT. Graph of Thoughts (GoT) generalizes chain- and tree-structured reasoning by representing intermediate thoughts as a graph, allowing candidate ideas to be generated, refined, merged, or discarded (Besta et al., 2024). In our setting, GoT generates multiple candidate regulation strategies, refines promising thoughts, and aggregates compatible strategies into a final catalyst-design proposal. Unlike CataPath, GoT does not use a chemically predefined action space or hybrid property-prediction and expert-preference reward.

AoT. Algorithm of Thoughts (AoT) guides the LLM to follow algorithmic reasoning patterns through carefully designed demonstrations, enabling it to enumerate, evaluate, and refine candidate solutions within the prompt (Sel et al., 2023). In our setting, AoT is prompted with examples showing how to explore, reject, and refine catalyst regulation or molecular-optimization strategies before producing the final candidate.

LATS. Language Agent Tree Search (LATS) combines LLM reasoning with tree search, value estimation, self-reflection, and external feedback (Zhou et al., 2023). In our setting, each node represents a partial catalyst-design trajectory, and LATS expands the tree by generating regulation or optimization actions with the LLM. Candidate nodes are evaluated using LLM-based value estimation together with property-prediction feedback. To ensure a fair comparison, LATS does not use CataPath's Expert Preference Reward Model.

We do not directly reproduce ChemReasoner because it relies on task-specific quantum-chemical feedback pipelines and catalyst settings that are not directly aligned with CataDesignQR. Instead, we compare with ToT and ReAct as general search-style LLM baselines and provide a methodological comparison with ChemReasoner in Table 7.

Table 7: Methodological comparison between CataPath and related search-based LLM reasoning methods.

Method	Search-based	Domain Action Space	Property Feedback	Expert Preference Reward	Trajectory Dataset
ToT (Yao et al., 2023)	✓	–	–	–	–
ReAct (Yao et al., 2022)	✓	–	–	–	–
MCTSci (Sprueill et al., 2023)	✓	–	task-specific	–	–
ChemReasoner (Sprueill et al., 2024)	✓	heuristic	quantum-chemical feedback	–	–
OSDA Agent (Hu et al., 2025)	✓	task workflow	binding-energy feedback	–	–
CataPath	✓	✓	✓	✓	✓

D.2 Evaluation Metrics

D.2.1 CO_2RR Electrocatalyst De Novo Design

For CO_2RR electrocatalyst generation, we evaluate each method from three complementary perspectives: *Reasonableness*, *Effectiveness*, and *Feasibility*. These metrics are designed to assess whether a generated catalyst design follows the required regulation category, aligns with literature-verified high-performance catalysts, and satisfies expert judgments of practical plausibility. Fig. 8 illustrates the evaluation protocol.

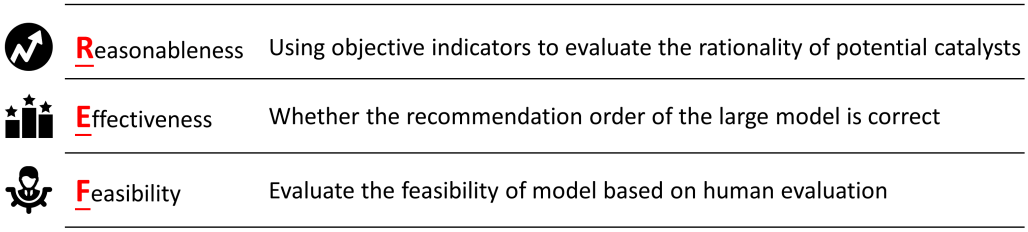

Figure 8: Evaluation protocol for Reasonableness, Effectiveness, and Feasibility in CO_2RR electrocatalyst design.

Reasonableness. Reasonableness measures whether the generated regulation strategies match the target action category specified by the design goal. For each query, the model is required to generate five candidate strategies. We use a prototype-network classifier trained on electrocatalytic CO_2RR literature data (Wang et al., 2023) to classify each generated strategy into one of the predefined regulation categories. A strategy is counted as correct if its predicted category matches the target category in the prompt. The reasonableness score is computed as:

$$\text{Reasonableness} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{K}\sum_{j=1}^{K}\mathbb{I}[\hat{c}_{i,j} = c_i], \tag{12}$$

where N is the number of evaluation queries, $K = 5$ is the number of generated strategies per query, c_i is the target regulation category for query i, $\hat{c}_{i,j}$ is the classifier-predicted category of the j-th generated strategy, and $\mathbb{I}[\cdot]$ is the indicator function. A higher score indicates better alignment between generated strategies and the required design category. The prototype-network classifier reaches 0.914 accuracy and 0.899 macro-F1 on source-paper-level held-out literature validation. On 300 generated outputs, its agreement with adjudicated expert labels is 0.873, compared with inter-expert $\kappa = 0.887$. We therefore use Reasonableness as an automated regulation-category consistency signal.

Effectiveness. Effectiveness measures whether the generated catalyst design is semantically aligned with literature-verified high-performance catalysts in the held-out test set $\mathscr{D}_{\text{test}}$. Because valid catalyst descriptions may differ lexically while expressing similar structural design principles, we use embedding similarity. For each evaluation query, we encode the generated candidate $\hat{y}_i$ and the corresponding reference design y_i with a domain-specific CatBERT encoder $\phi(\cdot)$, and compute:

$$\text{Effectiveness} = \frac{1}{N}\sum_{i=1}^{N}\text{CosSim}\left(\phi(\hat{y}_i), \phi(y_i)\right), \tag{13}$$

where N is the number of evaluation queries. This metric captures whether the generated design is close to experimentally verified catalyst designs in the learned chemical-text representation space. We use it as a proxy for task-specific effectiveness rather than a direct measurement of catalytic activity, since final catalytic performance still requires experimental validation.

Feasibility. Feasibility evaluates whether a generated catalyst design is chemically plausible and practically synthesizable. We use blind expert evaluation for this dimension. For each generated

candidate $\hat{y}_i$, experts assign a score based on chemical logic, thermodynamic plausibility, and synthesis feasibility. The feasibility score is computed as:

$$\text{Feasibility} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{K}\sum_{k=1}^{K} S(e_k, \hat{y}_i), \tag{14}$$

where N is the number of evaluation queries, $K = 5$ is the number of experts, and $S(e_k, \hat{y}_i)$ denotes the normalized score assigned by expert e_k to candidate $\hat{y}_i$. To reduce evaluation bias, the expert panel used for final feasibility assessment is separate from the experts involved in EPRM data collection. The evaluation is conducted in a double-blind setting, where experts do not know the identity of the generating method or the scores assigned by other annotators.

D.2.2 Evaluation Metrics for OSDA Optimization

For OSDA optimization, we follow established zeolite synthesis benchmarks and evaluate generated molecules from three perspectives: chemical validity, structural similarity to verified OSDAs, and distributional alignment with the known OSDA chemical space. The reference molecules are derived from OSDB and the Jensen dataset, which contain validated OSDA-zeolite pairs from computational and experimental sources.

Validity. Validity measures whether a generated molecule satisfies general chemical rules and OSDA-specific empirical constraints. These constraints include valency validity, acceptable ring sizes, C/N ratio ranges, rotatable-bond limits, and the absence of unstable functional groups. The validity score is defined as:

$$V = \frac{N_{\text{valid}}}{N_{\text{total}}}, \tag{15}$$

where N_{valid} is the number of generated molecules that pass all validity checks, and N_{total} is the total number of generated candidates.

Structural Similarity. Structural similarity evaluates how closely generated OSDAs resemble verified reference molecules. We report BLEU for SMILES-level sequence overlap and Tanimoto similarity over Morgan, MACCS, and RDKit fingerprints for molecular substructure and topology. For each generated molecule $m_{\text{gen},i}$, we compute its maximum similarity to the reference set:

$$S = \frac{1}{N}\sum_{i=1}^{N}\max_{j}\text{Sim}\left(F(m_{\text{gen},i}), F(m_{\text{ref},j})\right), \tag{16}$$

where $F(\cdot)$ denotes the corresponding feature extractor or fingerprint function, $m_{\text{ref},j}$ is a reference OSDA molecule, and $\text{Sim}(\cdot,\cdot)$ is BLEU or Tanimoto similarity depending on the metric.

Distributional Alignment. Distributional alignment measures whether generated molecules follow the broader property distribution of known OSDAs. We use WHIM descriptors to represent molecular geometry and physicochemical properties such as size, shape, and symmetry. Let P be the generated distribution and Q be the reference distribution. We report KL divergence and Energy Distance (ED), where ED is computed as:

$$\text{ED}(P,Q) = 2\mathbb{E}\|X - Y\| - \mathbb{E}\|X - X'\| - \mathbb{E}\|Y - Y'\|, \tag{17}$$

where $X, X' \sim P$ and $Y, Y' \sim Q$ are independent samples. Lower KL divergence and ED indicate closer distributional alignment with verified OSDA molecules.

Table 8: Potentially active CO_2RR surfaces proposed by CataPath, including elemental compositions, Miller indices, and DFT-calculated adsorption energies [eV]. The calculations follow the setup of tran2018active.

Formula	Miller index	Energy	Formula	Miller index	Energy	Formula	Miller index	Energy
Al_2Cu	[2,1,0]	-0.75	$NiZnCu_2$	[1,0,0]	-0.68	Al_2Cu_6	[1,2,0]	-0.65
Al_2Cu_6	[1,2,1]	-0.69	Pd_4Cu_{16}	[0,0,1]	-0.71	$AlCu_3$	[1,1,0]	-0.72
Al_3Cu_2	[1,1,1]	-0.82	PdCu	[1,0,0]	-0.66	Au_2Cu_2	[1,0,0]	-0.69
Al_5Cu_5	[1,1,-2]	-0.83	$PdCu_3$	[1,0,0]	-0.66	Au_3Cu	[2,1,1]	-0.64
Al_6Cu_2	[1,0,1]	-0.58	Pd_2CuIn	[1,0,0]	-0.79	$AuCu_3$	[2,1,1]	-0.75
AlCu	[1,1,1]	-0.80	PtCu	[0,0,1]	-0.72	Cu	[1,1,0]	-0.63
$AlCu_3$	[2,1,0]	-0.75	$PtCu_3$	[1,0,0]	-0.69	Ga_2Cu	[1,0,0]	-0.64
Au_2AlCu	[1,1,1]	-0.73	$SbCuNi_2$	[1,0,0]	-0.66	Ga_2Cu	[1,1,1]	-0.66
Au_2Cu_2	[0,0,1]	-0.67	Si_2Cu_6	[2,1,1]	-0.71	Ga_2Cu_6	[1,0,1]	-0.75
Au_3Cu	[2,1,0]	-0.61	$SiCu_3$	[0,0,1]	-0.65	Ge_2Cu_6	[2,1,0]	-0.72
$AuCu_3$	[1,1,0]	-0.68	$Sn_{10}Cu_{12}$	[1,0,0]	-0.68	Si_2Cu_6	[1,0,2]	-0.73
Cu	[2,1,1]	-0.69	$Sn_{16}Cu_{20}$	[0,0,1]	-0.65	$SiCu_3$	[1,0,0]	-0.69
$CuAlPt_2$	[1,1,0]	-0.77	Sn_2Cu_2	[2,1,1]	-0.61	$Sn_{10}Cu_{12}$	[0,1,0]	-0.64
Ga_2Cu	[1,0,1]	-0.67	$Zn_4Au_2Cu_2$	[1,1,1]	-0.68	$Zn_4Au_2Cu_2$	[1,2,1]	-0.63
Ga_2Cu_6	[1,0,0]	-0.69	$ZnAu_2Cu$	[2,1,0]	-0.72			
$Ge_4Co_4Cu_2$	[1,0,0]	-0.71	$Zn_{16}Cu_{10}$	[1,1,0]	-0.82			
Ge_2Cu_6	[1,0,2]	-0.71	ZnCu	[1,0,0]	-0.61			
$Ge_4Fe_4Cu_2$	[1,0,0]	-0.70	Al_2Cu	[1,1,0]	-0.66			

D.3 Case Study

We provide five representative case studies to examine how CataPath generates catalyst design strategies under different regulation categories, including alloying, defect engineering, surface modification, composite construction, and structure control. These cases illustrate whether the search process can connect a design goal with chemically meaningful intermediate actions and final candidate strategies.

For example, when asked to generate defect-regulation strategies for CuO_x catalysts toward ethylene production, CataPath proposes candidates such as oxygen vacancies, copper cation vacancies, and under-coordinated Cu lattice defects. These suggestions are consistent with common defect-engineering principles for tuning adsorption and activation behavior in oxide-derived copper catalysts. In another case, for structure control of CuS_x toward formate production, CataPath ranks "Size Control with Nanoparticle Thickness" as a high-scoring strategy and explains that smaller nanoparticles can expose more low-coordinated surface atoms, which may facilitate reactant adsorption and activation. These examples suggest that CataPath can often specialize broad regulation categories into more concrete catalyst design hypotheses.

At the same time, the case studies also reveal limitations. For surface modification of Cu toward ethanol production, one generated response suggests "Chemical Vapor Deposition (CVD) of 1-Octadecanethiol Films". While 1-octadecanethiol self-assembled monolayers can be chemically meaningful for surface functionalization, CVD is not an appropriate preparation route in this context. This example indicates that CataPath may still combine a plausible modifier with an unsuitable synthesis procedure. Future work should improve synthesis-route verification and incorporate more explicit procedural constraints into the action space or reward model.

We further provide external validation for selected CO_2RR candidates generated by CataPath. First, we aggregate potentially active catalyst surfaces proposed by the model and perform DFT calculations to estimate their adsorption energetics. The elemental compositions, Miller indices, and calculated adsorption energies are reported in Table 8. Second, we conduct wet-lab characterization of a top-ranked Cu-Sn bimetallic nanostructure suggested by CataPath. The structural and electrochemical characterization results are shown in Fig. 9, 10, 11, and 12. These results provide case-level evidence that some CataPath-generated designs are synthetically accessible and experimentally relevant.

method_type: alloy

Initial Question:
Generate a list of top-5 **alloy** for **Cu-M** to produce **formate**. Take a deep breath and let's think step-by-step.
CataPath Answer:
1. Cu-Cd Alloy with Facet Engineering.
- Explanation: Facet engineering involves controlling the crystal facets exposed on the surface of the catalyst. For Cu-Cd alloys, optimizing the (111) facet can enhance the adsorption of CO_2 and the subsequent reduction steps, leading to an increased selectivity for formate production. Cd helps in stabilizing the active sites and reducing the overpotential required for the reaction.

2. Cu-In Alloy with Facet Engineering.
- Explanation: Similar to Cd, In can stabilize certain crystal facets of Cu, particularly the (100) and (110) facets, which are known to facilitate the formation of intermediates that favor formate production. Facet engineering can increase the exposure of these facets, thus improving catalytic performance.

3. Cu-Pd Alloy with Interfacial Engineering.
- Explanation: Interfacial engineering involves modifying the interface between Cu and Pd to enhance catalytic activity. Pd can improve the electronic properties of Cu by altering its electron density and facilitating the activation of CO_2. This method can increase the number of active sites and improve the efficiency of electron transfer, favoring the formation of formate over hydrogen.

4. Cu-Ni Alloy with Strain Engineering.
- Explanation: Strain engineering involves inducing strain in the Cu lattice by alloying with Ni. The strain can alter the electronic properties and the binding energies of reaction intermediates on the catalyst surface. For Cu-Ni alloys, applying tensile strain can enhance CO_2 adsorption and activation, leading to higher formate selectivity.

5. Cu-Ti Alloy with Defect Engineering.
- Explanation: Defect engineering introduces controlled defects into the Cu-Ti alloy structure to create active sites for CO_2 reduction. Ti can introduce oxygen vacancies and other defects that act as catalytic sites for the formation of formate. These defects can improve the binding of CO_2 and lower the activation energy required for the reaction.

method_type: defect

Initial Question:
Generate a list of top-5 **defect** for **CuO_x** to produce **ethylene**. Take a deep breath and let's think step-by-step.

CataPath Answer:
1. Oxygen Vacancy (O^2-Vacancy).
- Regulation Method: Oxygen Vacancy Engineering.
- Explanation: Oxygen vacancies can create active sites for ethylene production by modifying the electronic structure and surface reactivity of CuO_x, facilitating the adsorption and activation of reactants.

2. Copper Cation Vacancy (Cu^{2+} Vacancy).
- Regulation Method: Copper Cation Vacancy Control.
- Explanation: Vacancies in the copper cation sites alter the redox properties and surface chemistry of CuO_x, influencing the activation energy for ethylene formation and enhancing catalytic activity.

3. Under-Coordinated Cu Lattice Defects.
- Regulation Method: Surface Engineering for Under-Coordinated Cu Sites.
- Explanation: Lattice defects where copper atoms are under-coordinated can act as active sites for ethylene synthesis, promoting adsorption and facilitating the desorption of ethylene.

4. Lattice Strain.
- Regulation Method: Lattice Strain Induction.
- Explanation: Introducing controlled lattice strain in CuO_x can modify the geometric and electronic properties of the catalyst, enhancing the interaction with reactants and improving ethylene selectivity and yield.

5. Surface Facet Engineering.
- Regulation Method: Surface Facet Control
- Explanation: Tailoring the surface facets of CuO_x can optimize the binding energies of intermediates involved in ethylene production, thereby improving catalytic performance and selectivity.

method_type: surface modification

Initial Question:
Generate a list of top-5 **surface modification** for **Cu** to produce **ethanol**. Take a deep breath and let's think step-by-step.

CataPath Answer:
1. Functionalization with Phosphate-Based Compounds.
- Explanation: Phosphate-based compounds can effectively modify the surface of Cu, creating sites that promote the adsorption and activation of CO_2, a key intermediate in the electrochemical reduction to ethanol. The phosphate groups can enhance the interaction between the catalyst surface and the reactants, improving selectivity and activity towards ethanol production.

2. Surface Functionalization with Hydrobromic Acid (HBr).
- Explanation: HBr can introduce bromine-containing functional groups on the Cu surface, which can alter the electronic properties of the catalyst and facilitate the selective adsorption and activation of CO_2 towards ethanol formation. The bromine-containing groups may also help stabilize key intermediates during the reaction.

3. Layer-by-Layer Assembly of Polyethyleneimine (PEI) and Phosphate-Based Compounds.
- Explanation: The layer-by-layer assembly of PEI, a cationic polymer, and phosphate-based compounds can create a hierarchical, functional interface on the Cu surface. The PEI layer can provide a favorable environment for CO_2 adsorption and activation, while the phosphate groups can further enhance the selectivity towards ethanol production.

4. Chemical Vapor Deposition (CVD) of 1-Octadecanethiol Films.
- Explanation: The CVD of 1-octadecanethiol, a long-chain aliphatic thiol, can form a self-assembled monolayer (SAM) on the Cu surface. The hydrophobic nature of the 1-octadecanethiol SAM can selectively adsorb and activate CO_2 over other intermediates, guiding the reaction towards ethanol formation.

5. Functionalization with Dihexadecyl Dimethyl Ammonium Bromide (DHDAB).
- Explanation: DHDAB, a quaternary ammonium salt with long-chain alkyl groups, can be used to functionalize the Cu surface. The cationic head group of DHDAB can interact with the Cu surface, while the hydrophobic alkyl chains can create a favorable environment for the adsorption and activation of CO_2 towards ethanol production.

method_type: composite

Initial Question:
Generate a list of top-5 **composite** for **Cu/C** to produce **carbon monoxide**. Take a deep breath and let's think step-by-step.

CataPath Answer:
1. Cu/Nitrogen-Doped Carbon Nanotube Composite (Cu/N-CNT).
- Regulation Method: Incorporation of nitrogen-doped carbon nanotubes enhances the catalytic activity of copper for the production of carbon monoxide. Nitrogen doping can modify the electronic structure of the carbon nanotubes, promoting stronger metal-support interactions and improving the catalytic efficiency of Cu/C.

2. Cu/Graphene Oxide Composite (Cu/GO).
- Regulation Method: Graphene oxide serves as a support material that can provide abundant oxygen-containing functional groups and a large surface area. These features enhance the dispersion of copper nanoparticles and facilitate the catalytic conversion of carbon-containing compounds to carbon monoxide.

3. Cu/Nitrogen-Doped Graphene Composite (Cu/N-Graphene).
- Regulation Method: Similar to nitrogen-doped carbon nanotubes, nitrogen-doped graphene modifies the electronic structure near the copper active sites. This modification can enhance the adsorption and activation of reactants, thereby improving the selectivity and activity of Cu/C towards carbon monoxide production.

4. Cu/Graphitic Carbon Nitride (g-C3N4) Composite (Cu/g-C3N4).
- Regulation Method: Graphitic carbon nitride (g-C3N4) acts as a metal-free semiconductor support for copper nanoparticles. The conjugated structure and nitrogen-containing functional groups of g-C3N4 can facilitate charge transfer and modify the chemical environment around Cu particles, thereby promoting efficient carbon monoxide generation.

5. Cu/Carbon Nanotube Composite (Cu/CNT).
- Regulation Method: Carbon nanotubes provide a stable and conductive support for copper nanoparticles. The one-dimensional structure of CNTs offers high surface area and excellent mechanical properties, which can enhance the dispersion and stability of Cu nanoparticles, leading to improved catalytic performance in carbon monoxide production.

method_type: structure control

Initial Question:
Generate a list of top-5 **structure control** for **CuS_x** to produce **formate**. Take a deep breath and let's think step-by-step.

CataPath Answer:
1. Size Control with Nanoparticle Thickness.
- Explanation: Reducing the thickness of CuS_x nanoparticles can enhance the specific surface area and expose more active sites, which can improve the catalytic activity towards formate production. Smaller nanoparticles also have a higher proportion of low-coordinated surface atoms, which can facilitate the adsorption and activation of reactants.

2. Morphology Control with Porous Structures.
- Explanation: Introducing porous structures in CuS_x can increase the surface area and provide more active sites for the CO_2 reduction reaction. The porous network can also facilitate the mass transport of reactants and products, enhancing the overall catalytic performance.

3. Morphology Control with Nanorod Structures.
- Explanation: Synthesizing CuS_x in the form of nanorods can promote the exposure of

specific crystal facets that are more selective towards formate production. The anisotropic morphology can also affect the adsorption and activation of reactants, leading to improved catalytic selectivity.

4. Dendritic Structure with Dendritic Catalysts on Mesh Supports.
- Explanation: Incorporating a dendritic structure in the CuS_x catalyst, supported on a mesh substrate, can create a hierarchical architecture with high surface area and efficient mass transport. The dendritic structure can enhance the exposure of active sites and improve the accessibility of reactants, leading to enhanced formate production.

5. Size Control with Monolayer Thickness.
- Explanation: Preparing CuS_x catalysts with a monolayer thickness can maximize the surface-to-volume ratio, allowing for efficient utilization of the active sites. This can result in improved catalytic activity and selectivity towards formate production.

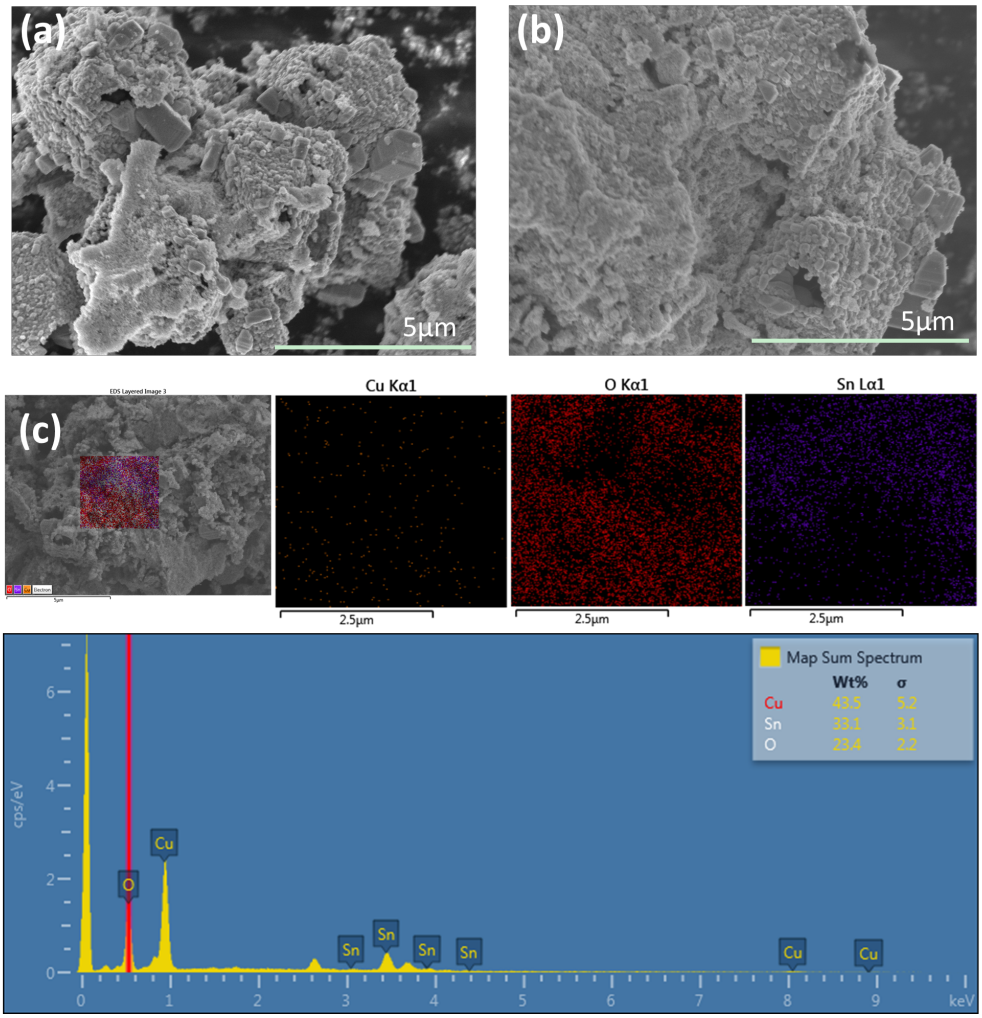

Figure 9: (a-b) SEM images and (c) EDS mapping of Cu-Sn bimetallic nanostructure.

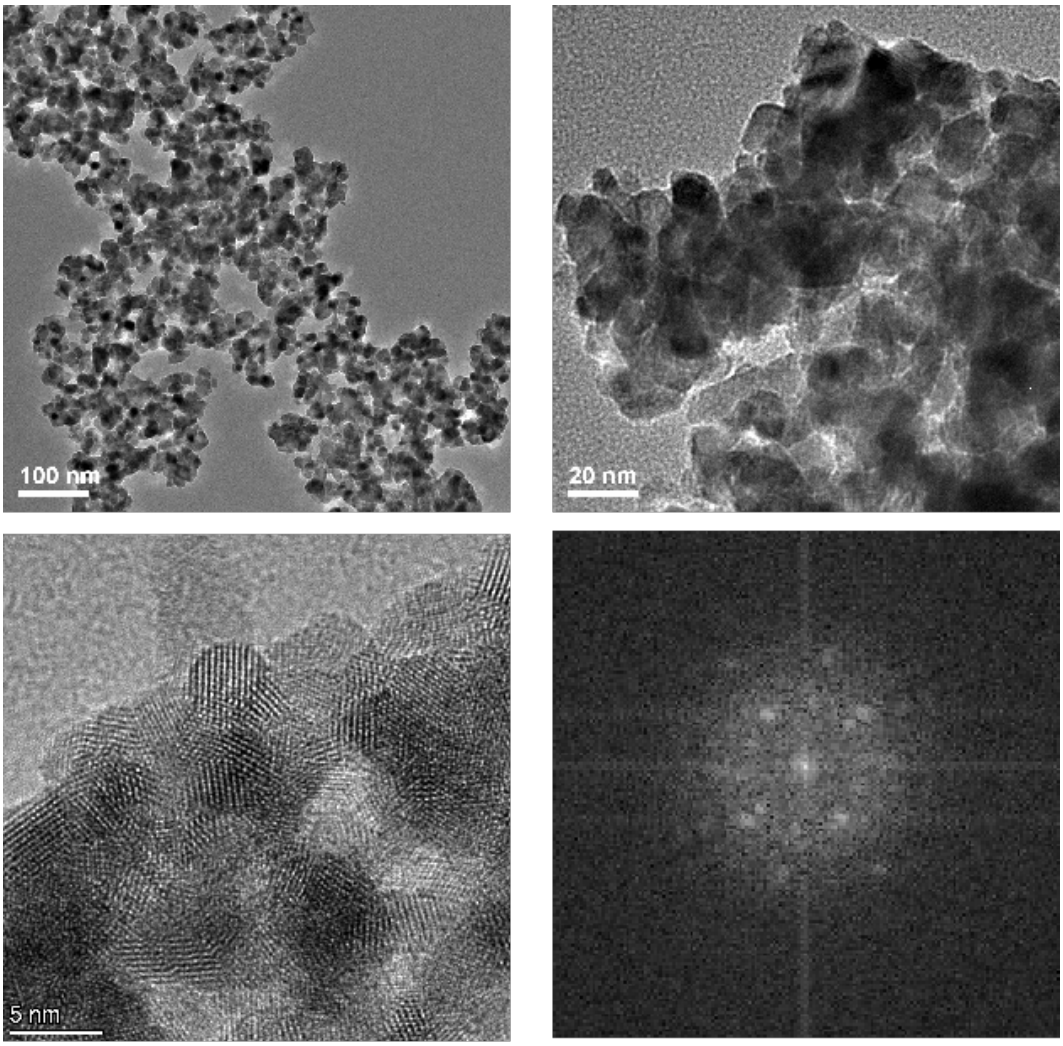

Figure 10: HRTEM images and FFT pattern of Cu-Sn bimetallic nanostructure.

D.4 Additional Validation and Robustness

We report the additional validation and robustness experiments. Reference solutions used in the literature-matching analysis are used only for post-hoc evaluation and are withheld from EPRM training and MCTS search.

Backbone robustness. To evaluate whether the performance gains depend on a particular LLM backbone, we instantiate CataPath with DeepSeek-V3 and Qwen3-32B-Thinking. For each backbone, we compare direct generation with CataPath-guided search. The chemically informed action space, MCTS search budget, reward definitions, prompts, and evaluation protocol are kept fixed.

As shown in Table 9, CataPath consistently improves Reasonableness, Effectiveness, and Expert Feasibility for all three evaluated backbones. The largest absolute gain within each backbone is in Expert Feasibility, suggesting that expert-guided search helps improve practical plausibility across both open-weight and closed-source models. These results support the applicability of CataPath across the tested backbones.

Held-out literature matching. We evaluate whether generated strategies recover independently documented catalyst-design principles. For each held-out CO_2RR design goal, every method receives the same input query and returns a ranked list of candidate regulation strategies. A candidate is

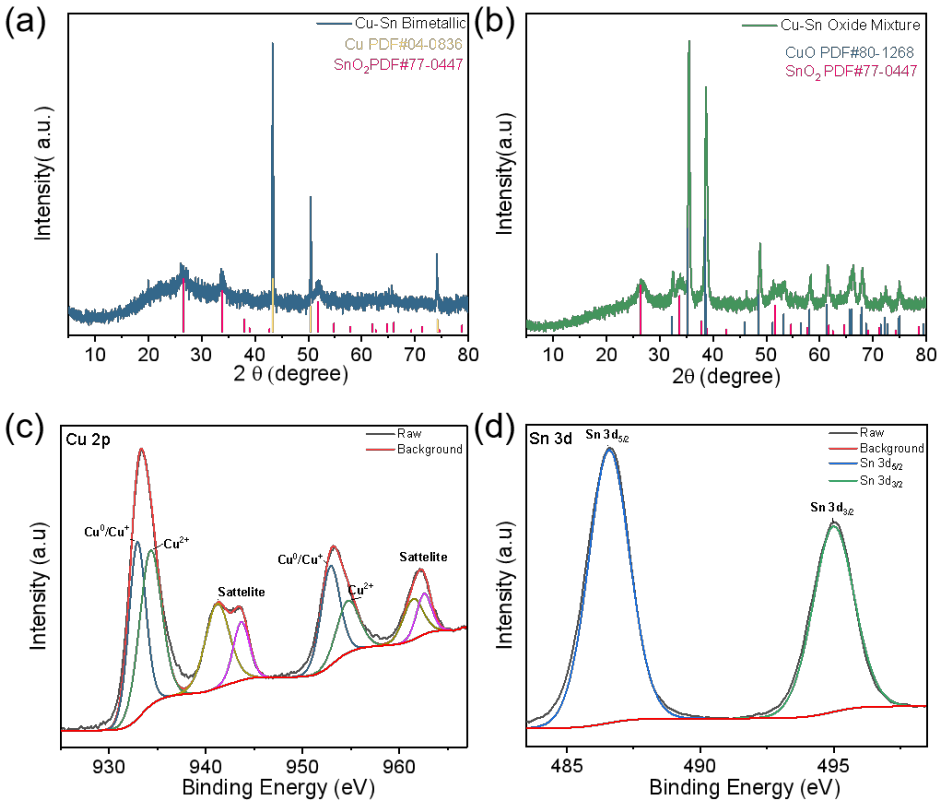

Figure 11: XRD patterns of (a) Cu-Sn bimetallic nanostructure and (b) Cu-Sn mixture. (c) Cu 2p and (d) Sn 3d XPS spectra of Cu-Sn bimetallic nanostructure.

Table 9: Backbone robustness on CO_2RR electrocatalyst generation. Each backbone is evaluated with direct generation and CataPath. Orange values indicate absolute improvements over direct generation.

	CO_2RR electrocatalyst generation		
Model	**Reasonableness ↑**	**Effectiveness ↑**	**Expert Feasibility ↑**
Open-weight models			
DeepSeek-V3	0.821	0.791	0.553
+ CataPath	**0.923** (+0.102)	**0.935** (+0.144)	**0.812** (+0.259)
Qwen3-32B-Thinking	0.804	0.776	0.536
+ CataPath	**0.887** (+0.083)	**0.895** (+0.119)	**0.751** (+0.215)
Closed-source models			
GPT-4	0.685	0.622	0.412
+ CataPath	**0.904** (+0.219)	**0.913** (+0.291)	**0.789** (+0.377)

counted as a match when it agrees with the reference in target material, target product, regulation category, and core design mechanism. Two domain experts independently assess each match, and disagreements are resolved by a third expert. Top-1 measures whether the highest-ranked candidate matches the reference, while Top-5 measures whether at least one of the five highest-ranked candidates matches.

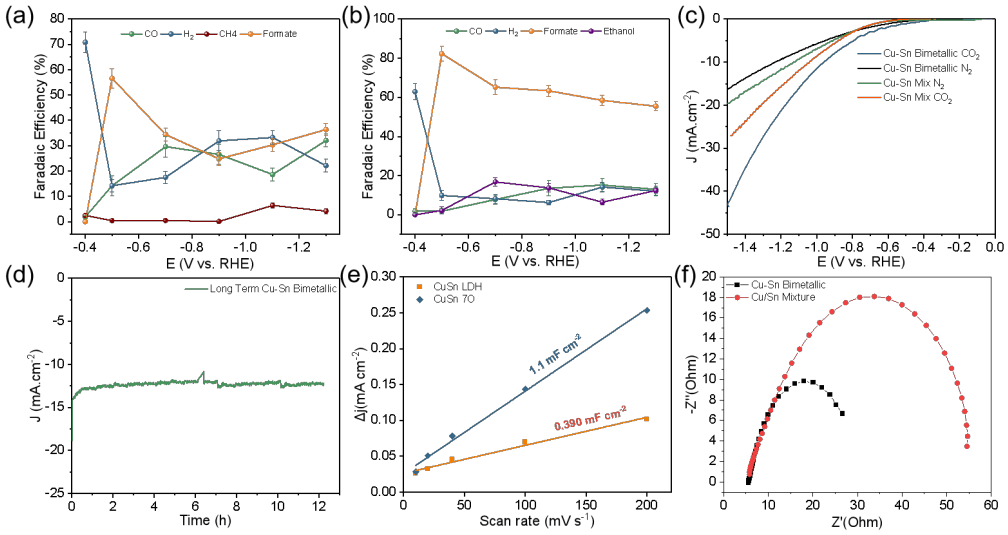

Figure 12: (a) FE of Cu-Sn mixture at different potentials. (b) FE of Cu-Sn bimetallic nanostructure at different potentials. (c) LSV curves of Cu-Sn mixture and Cu-Sn bimetallic in N_{2-} and CO_{2-} saturated 0.5 M $KHCO_3$ solution. (d) Long-term stability test of Cu-Sn bimetallic nanostructure. (e) Linear fitting of double layer capacitance current vs. scan rate to estimate ECSA. (f) Nyquist plot of Cu-Sn bimetallic nanostructure at -0.6 V.

Table 10: The Held-out literature matching rates. The final column reports the fraction of candidates judged chemically valid by experts but not matching the held-out reference strategy.

Method	Top-1	Top-5	Expert-valid but non-matching
GPT-4o	0.34	0.58	0.21
DeepSeek-V3	0.39	0.63	0.23
GPT-4 CoT + Self-Consistency	0.41	0.65	0.24
GPT-4 ToT-BFS	0.43	0.67	0.25
GPT-4 ReAct	0.46	0.70	0.26
CataPath	**0.58**	**0.81**	**0.29**

As shown in Table 10, CataPath achieves the highest Top-1 and Top-5 matching rates, exceeding ReAct by 0.12 and 0.11, respectively. The results indicate that CataPath more frequently recovers documented design principles under the held-out protocol. The expert-valid but non-matching rate also shows that some generated strategies are chemically plausible even when they differ from the specific literature reference.

Action-space coverage. We assess whether the predefined action space covers strategies proposed by strong LLMs without explicit action constraints. We sample 100 held-out CO_2RR design goals and request five free-form regulation strategies per goal from GPT-4o, GPT-5.2, and DeepSeek-V3, yielding 500 proposals per backbone. Each proposal is mapped to the existing action-space taxonomy when possible. Proposals that cannot be mapped are independently judged by domain experts as either chemically meaningful or invalid. The results are shown in Table 11.

Overall, 87.2% of the sampled free-form proposals can be mapped to the existing action space, while 5.1% are unmapped but judged chemically valid. This result suggests that literature mining, semantic clustering, and expert verification could be used to expand the action space.

Inference-time MCTS statistics. We record inference-time search behavior on held-out test queries for both tasks. The statistics include the average search depth, number of expanded nodes, effective branching factor, number of LLM calls, runtime, and early-stopping rate. The results are summarized

Table 11: Coverage of the predefined action space on free-form proposals from held-out CO_2RR design goals.

Backbone	Mapped to existing action space	Unmapped but expert-valid	Unmapped and invalid
GPT-4o	438/500 (87.6%)	23/500 (4.6%)	39/500 (7.8%)
GPT-5.2	446/500 (89.2%)	21/500 (4.2%)	33/500 (6.6%)
DeepSeek-V3	424/500 (84.8%)	32/500 (6.4%)	44/500 (8.8%)
Overall	**1308/1500 (87.2%)**	**76/1500 (5.1%)**	**116/1500 (7.7%)**

Table 12: Inference-time search statistics for CataPath.

Statistic	CO_2RR	OSDA
Average search depth	6.73	5.42
Average expanded nodes per query	43.6	31.4
Average effective branching factor	4.82	3.61
Average GPT calls per query	44.1	31.9
Average runtime per query	1.34 min	0.96 min
Early-stopped branches	26.7%	21.4%

in Table 12. The effective branching factor is computed from the number of expanded child nodes divided by the number of expanded parent nodes.

CataPath performs deeper and broader searches for CO_2RR than for OSDA, which is consistent with the larger CO_2RR action space and more diverse regulation strategies. The effective branching factors remain substantially smaller than the raw action-space sizes because task-specific constraints, expert-prior pruning, and semantic-convergence stopping filter low-value expansions.

E PROMPT TEMPLATE

Here are the prompt templates we build during the construction of MCT in CataPath framework.

<LLM-based CO_2RR electrocatalyst design prompt>Prompt

You are an expert in electrocatalysis and material science. Please complete the catalyst design task based on the following query and historical reasoning constraints:
1. Task Definition:
- Objective: Generate a list of top-5 candidate catalyst regulation strategies for the specific base material and target product.
- Requirement: Provide a scientific explanation for each candidate, detailing why it improves catalytic performance.
- Approach: Take a deep breath and let's think step-by-step.

2. Input Information:
- Base Material: {material} (e.g., Cu, CuO_x)
- Target Product: {product} (e.g., Ethanol, Formate)
- Current Regulation Category: {action_type} (e.g., atomic level dispersion, defect)

3. History & Constraints (MCTS Trajectory):
- Last Step Answer / Context: {history_constraints}
(e.g., "You should use atomic level dispersion with the following options single-atom... you should use the corresponding Cu single atoms...")

4. Output Format:
- Please list exactly 5 distinct strategies.
- Format: [Number]. [Specific Strategy Name] - [Explanation]

5. User Query Construction:
"Generate a list of top-5 {action_type} for {material} to produce {product}. {history_constraints} Take a deep breath and let's think step-by-step."

<LLM-based OSDA molecule optimization prompt>Prompt

You are an expert in zeolite synthesis and organic chemistry. Please optimize the structure of the Organic Structure-Directing Agent (OSDA) based on the following instructions:
1. Task Definition:
- Objective: Modify the current OSDA molecule structure to satisfy the specific optimization strategy while maintaining chemical validity.
- Requirement: Output the optimized molecule in SMILES format and provide a brief explanation of the structural change.

2. Input Information:
- Current Molecule (SMILES): {current_SMILES}
- Target Zeolite Framework: {zeolite_framework} (e.g., LTA, CHA)

3. Optimization Strategy (MCTS Action):
- Action Category: {action_category} (e.g., Binding Energy, Chemical Validity and Synthetic Feasibility)
- Specific Strategy Keyword: {action_keyword}
(e.g., Void Filling, Shape Matching)

4. Constraints & History:
- Previous Modifications: {history_constraints}
- Chemical Rules: Ensure the generated SMILES string is valid, stable, and synthetically feasible.

5. User Query Construction:
"The current OSDA molecule is {current_smiles}. Please apply the {action_keyword} strategy to optimize its structure for the {zeolite_framework} framework. {history_constraints} Take a deep breath and let's think step-by-step."