Digital Discovery

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal

Cite this: DOI: 10.1039/d5dd00413f

An automated evaluation agent for Q&A pairs and reticular synthesis conditions

Nakul Rampal, †[abc] Dongrong Joe Fu, †[c] Chengbin Zhao, [abc] Hanan S. Murayshid,[d] Albatool A. Abaalkhail,[e] Nahla E. Alhazmi,[f] Majed O. Alawad,[g] Christian Borgs, *[ch] Jennifer T. Chayes*[chijk] and Omar M. Yaghi *[abcg]

We report an automated evaluation agent that can reliably assign classification labels to different Q&A pairs of both single-hop and multi-hop types, as well as to synthesis conditions datasets. Our agent is built around a suite of large language models (LLMs) and is designed to eliminate human involvement in the evaluation process. Even though we believe that this approach has broad applicability, for concreteness, we apply it here to reticular chemistry. Through extensive testing of various approaches such as DSPy and finetuning, among others, we found that the performance of a given LLM on these Q&A and synthesis conditions classification tasks is determined primarily by the architecture of the agent, where how the different inputs are parsed and processed and how the LLMs are called make a significant difference. We also found that the quality of the prompt provided remains paramount, irrespective of the sophistication of the underlying model. Even models considered state-of-the-art, such as GPT-o1, exhibit poor performance when the prompt lacks sufficient detail and structure. To overcome these challenges, we performed systematic prompt optimization, iteratively refining the prompt to significantly improve classification accuracy and achieve human-level evaluation benchmarks. We show that while LLMs have made remarkable progress, they still fall short of human reasoning without substantial prompt engineering. The agent presented here provides a robust and reproducible tool for evaluating Q&A pairs and synthesis conditions in a scalable manner and can serve as a foundation for future developments in automated evaluation of LLM inputs and outputs and more generally to create foundation models in chemistry.

Received 15th September 2025
Accepted 31st October 2025

DOI: 10.1039/d5dd00413f

rsc.li/digitaldiscovery

Introduction

Large language models (LLMs) have rapidly evolved into versatile tools, impacting numerous scientific fields and permeating essentially every domain of human knowledge.[1,2,3] In chemistry, particularly reticular chemistry,[4] there is significant potential for LLMs to assist in answering complex scientific queries and to enhance research productivity by automating routine yet important tasks.[5,6,7,8,9,10,11] However, achieving truly effective, chemistry-specific LLM agents requires specialized training data that are well-structured, contextually accurate, and comprehensible to these models.[12,13,14,15,16]

Prior research from leading organizations such as OpenAI has demonstrated that reinforcement learning with human feedback (RLHF) – a process involving iterative interactions between humans and LLMs, where human evaluators validate and provide corrective feedback on LLM-generated outputs by labeling Question and Answer (Q&A) pairs – can substantially improve model accuracy and responsiveness.[17,18] Despite its effectiveness, RLHF remains resource-intensive, often demanding considerable human effort, time, and financial resources. Consequently, implementing RLHF is challenging

[a]Department of Chemistry, University of California, Berkeley, California 94720, USA. E-mail: yaghi@berkeley.edu
[b]Kavli Energy Nanoscience Institute, University of California, Berkeley, California 94720, USA
[c]Bakar Institute of Digital Materials for the Planet, College of Computing, Data Science, and Society, University of California, Berkeley, California 94720, USA. E-mail: jchayes@berkeley.edu; borgs@berkeley.edu
[d]Artificial Intelligence & Robotics Institute, King Abdulaziz City for Science and Technology (KACST), Riyadh 11442, Saudi Arabia
[e]Center of Excellence for Advanced Materials and Manufacturing, King Abdulaziz City for Science and Technology (KACST), Saudi Arabia
[f]Hydrogen Technologies Institute, King Abdulaziz City for Science and Technology (KACST), Riyadh 11442, Saudi Arabia
[g]KACST-UC Berkeley Center of Excellence for Nanomaterials for Clean Energy Applications, King Abdulaziz City for Science and Technology, Riyadh 11442, Saudi Arabia
[h]Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, California 94720, USA
[i]Department of Mathematics, University of California, Berkeley, California 94720, USA
[j]Department of Statistics, University of California, Berkeley, California 94720, USA
[k]School of Information, University of California, Berkeley, California, 94720, USA
† These authors contributed equally.

for smaller research teams or individuals in laboratory environments, severely limiting its widespread adoption within specialized fields like reticular chemistry.

To address this, the LLM community has developed what is known as RLAIF (Reinforcement Learning with AI Feedback) where instead of humans an LLM is asked to label Q&A pairs. As in RLHF, these labeled Q&A pairs are then used to train a reward function which in turn is used to improve the model *via* reinforcement learning. In this paper, we develop this approach for Q&A pairs in the natural sciences, in particular, reticular chemistry.

As a first step in this direction, our group previously developed the RetChemQA dataset,[19] an extensive collection of question–answer (Q&A) pairs specifically tailored to reticular chemistry. This dataset aimed to mimic the quality and context-specificity of human-generated Q&A pairs by leveraging LLMs. Despite its utility, the automated generation process inherently introduced inaccuracies, necessitating rigorous human validation of each Q&A pair. Existing evaluation frameworks generally assume that the question itself is correct. However, since LLMs were used to generate the RetChemQA dataset, this assumption does not hold – the question itself can sometimes be factually incorrect or entirely out of context. This highlighted the need to generate a benchmark that explicitly accounts for question validity, in addition to answer correctness. To systematically address this issue, in RetChemQA, we developed a classification scheme categorizing each generated Q&A pair as true positive (TP), false positive (FP), true negative (TN), or false negative (FN), depending on their factual accuracy and contextual relevance. Using this scheme, approximately 2500 Q&A pairs covering both single-hop and multi-hop question types were manually evaluated by hand, highlighting the extensive labor required for manual verification.

The impracticality and inefficiency of manually evaluating such extensive datasets motivated us to develop the present automated evaluation agent named QAutoEval, which is capable of accurately assessing the correctness of Q&A pairs generated by LLMs. Automating this evaluation process not only promises significant time and resource savings but also ensures that only high-quality, validated data are utilized for further training and fine-tuning of chemistry-specific LLM agents. Ultimately, the goal of our research is to leverage automated evaluation to enhance the feasibility of reinforcement learning with AI feedback in chemistry, thus accelerating the development of robust, reliable, and domain-specific LLM systems. A broad-level schematic of our automated evaluation agent developed in this work is shown in Fig. 1, illustrating how the model systematically provides evaluations for each Q&A pair using four distinct inputs, where Input 1: main text; Input 2: SI; Input 3: structured Q&A pairs (or the synthesis conditions dataset), and Input 4: explicit evaluation criteria.

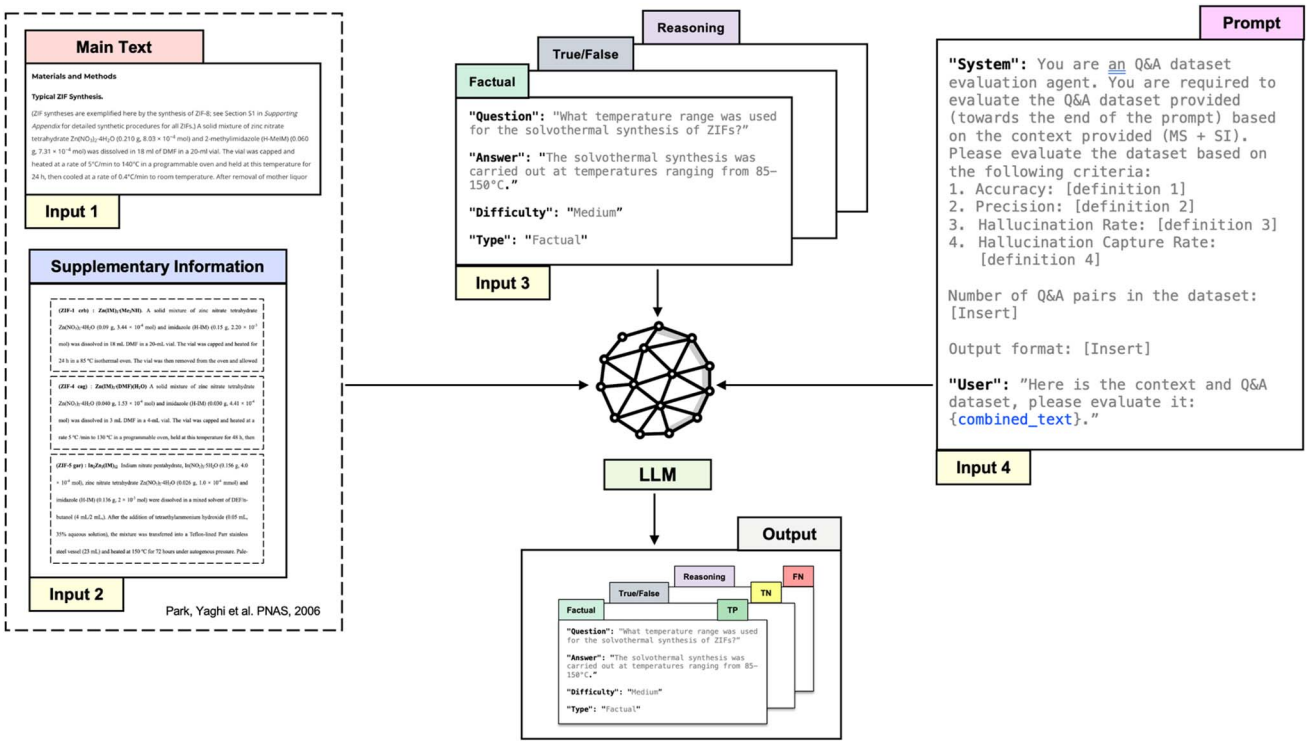

Fig. 1 Overview of the Automated Evaluation Agent, named QAutoEval. The evaluation process requires four distinct inputs. Input 1: the main text of the paper; Input 2: the SI (SI); Input 3: a pre-generated Q&A dataset comprising factual, True/False, and reasoning-type questions; and Input 4: a structured evaluation prompt containing clearly defined criteria. Together, these inputs are processed by an LLM—or a combination of multiple LLMs—which then assigns evaluation labels (TP, True Positive; FP, False Positive; TN, True Negative; FN, False Negative) to each Q&A pair based on its alignment with the provided context.

Methods

Optimizing the architecture of the agent and the prompt for Q&A evaluation tasks

Initially, we tested a straightforward prompting technique using DSPy,[20] explicitly defining TP, FP, TN and FN—please refer to our previous work on the definitions of these terms[19]—and the expected output format in the prompt (Fig. S1). However, the evaluation accuracy was inadequate due to the inherent complexity of the evaluation task, as the LLM frequently misreported the total number of questions in addition to incorrectly categorizing question types, such as factual, reasoning, or True/False questions. This was addressed by clearly dividing responsibilities between an LLM Retriever (GPT-4o-mini) to extract context from PDFs using Retrieval-Augmented Generation (RAG) and an LLM Evaluation Agent (GPT-4-Turbo) to evaluate the extracted Q&A pairs (Fig. S2). This division improved clarity, enabling accurate counting of the total number of questions. However, the categorization of the extracted Q&A pairs into factual, reasoning, or True/False questions remained incorrect. Subsequently, we implemented a DSPy-based LLM Judge Agent tasked explicitly with verifying factual correctness by presenting full context alongside the Q&A pair (Fig. S3). Unfortunately, the model exhibited substantial limitations in reliably assessing factual accuracy. Further simplification using a FactExtract Agent to shorten context worsened the overall performance (Fig. S4). In response to these limitations, we transitioned to a 'divide and conquer' strategy using structured outputs. First, we converted the JSON file into a structured DataFrame. Next, we processed the context, which included both the manuscript (MS) and SI (SI) and tasked the LLM with retrieving the relevant paragraph (at least 10 lines) for each Q&A pair. For the subsequent evaluation step, the extracted context along with the Q&A pair and a clearly defined prompt was provided to the LLM. For each Q&A pair, structured entries were created that included the Q&A itself, the extracted context, the question type, and the evaluation outcome (Fig. S5). This systematic approach markedly improved accuracy, reaching approximately 80%. Furthermore, accuracy was enhanced through iterative manual re-evaluation using the graphical user interface (GUI) – specifically OpenAI's website – by prompting the system to re-assess Q&A pairs. These iterative improvements motivated our subsequent decision to integrate a dedicated judge LLM into the evaluation framework.

Seeking to improve the accuracy further, we incorporated GPT-4o as a judge LLM within our divide and conquer framework, explicitly defining categories (TP, FP, TN, and FN) and presenting whole contexts and Q&A pairs clearly (Fig. S6). However, the GPT-4o judge model often incorrectly modified correct evaluations to incorrect and *vice versa*, indicating challenges inherent to single model judging systems.

Recognizing the limitations of standalone systems, we next attempted fine-tuning GPT-4o (snapshot from August 6th, 2024) using structured JSON inputs containing explicit roles (system, user, and assistant) and detailed context, questions, and answers. When attempting to fine-tune GPT-4o, we encountered significant limitations. Specifically, the human evaluated dataset predominantly contained Q&A pairs classified as TPs, resulting in insufficient exposure of the model to FPs, TNs, and FNs. Consequently, the fine-tuned model primarily classified evaluations as TP. To address this imbalance, we would have needed to determine how many of the remaining 90 000 Q&A pairs belonged to the FP, TN, or FN categories, requiring manual evaluation of each pair—a prohibitively labor-intensive and practically impossible task. Recognizing the impracticality of fine-tuning with such limited data, we concluded that fine-tuning the LLM was not feasible, necessitating the development of an alternative, more practical evaluation solution. To address these limitations, we advanced to a collaborative 'LLM-as-a-Judge' framework, employing a distributed approach rather than relying on a single LLM. We integrated four distinct LLMs – GPT-4o,[21] Claude 3.5 Sonnet, GPT-o1 (a model specifically optimized for reasoning tasks; preview version),[22] and Gemini 1.5-Pro.[23] Each model was independently designated as the tie-breaker with the highest weight, and the final evaluations were recorded. When using Claude as the tie-breaker, the average accuracy across single-hop and multi-hop datasets was 96.21%, while that for GPT-o1 was the same. The average TP catch rate was also comparable – 99.33% for Claude and 99.36% for GPT-o1. However, the non-TP catch rate when using GPT-o1 (36.82%) was higher by ∼10.1% compared to Claude (33.44%), which led us to favor GPT-o1 for the tie-breaker role. In comparison, Gemini's average accuracy was 95.96% with a non-TP catch rate of 30.21%, while GPT-4o's average accuracy was 95.86% with a non-TP catch rate of 36.16% (Table S3). The results of the sensitivity analysis are provided in Tables S1–S3. To illustrate, the following scenarios demonstrate how final evaluations were assigned:

Scenario 1: GPT-4o and GPT-o1 evaluated a Q&A pair as TP, while Claude 3.5 Sonnet and Gemini 1.5-Pro evaluated the same pair as FP. Given the higher weight assigned to GPT-o1, the final evaluation was TP.

Scenario 2: GPT-4o, Claude 3.5 Sonnet, and Gemini 1.5-Pro all evaluated a Q&A pair as FP, while GPT-o1 evaluated it as TP. Despite GPT-o1's higher weight, the cumulative weight of the other models resulted in a final evaluation of FP.

Using this approach and initial prompt, we achieved a high accuracy of approximately 98% across a randomly selected dataset of 50 DOIs, encompassing different question types and categories. However, we identified a significant limitation: the dataset predominantly included TP type questions, with very few FPs, TNs, and FNs. Consequently, the 'catching rate', defined as the correct identification of FPs, TNs, and FNs, was low. To address this shortcoming and enhance the capture rate for non-TP evaluations, we selected a specific DOI from our dataset—nchem.834—that contained a disproportionately high number of non-TP type questions, including nine TN Q&A pairs. When we tested the same prompt that had previously performed well on the randomly selected set of 50 DOIs, we observed a sharp drop in performance on the nchem.834 DOI. The final evaluation classified all questions as TPs, entirely missing the non-TP categories (Fig. 2). On closer inspection of the individual model outputs, we found that GPT-4o, Gemini

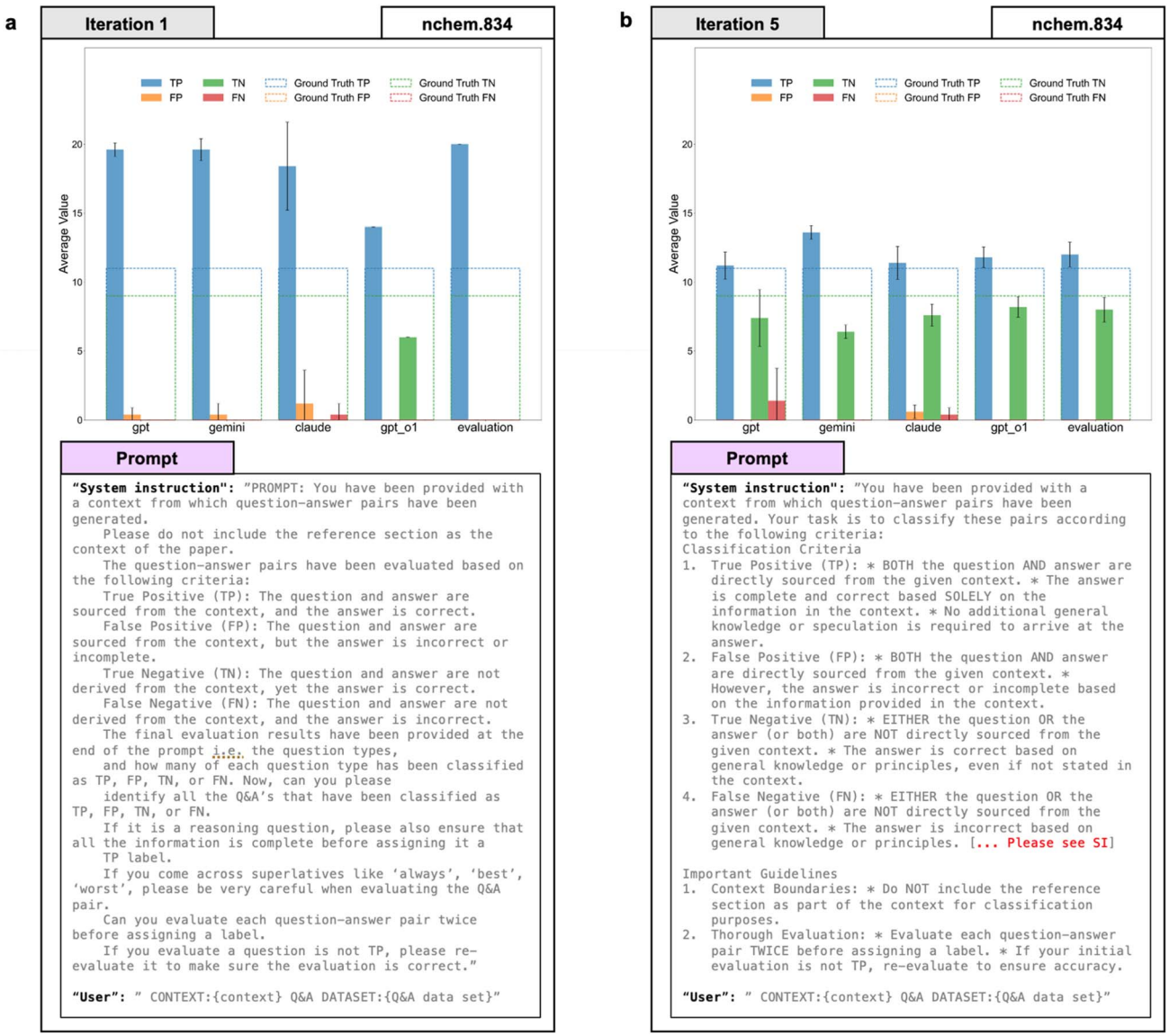

Fig. 2 Prompt optimization of the "system instruction" section to improve classification of non-TP type single-hop Q&A pairs. Comparison of evaluation results obtained for DOI nchem.834 using (a) iteration 1 and (b) iteration 5 of the evaluation prompt. Each bar chart shows the average number of Q&A pairs of each classification type (TP, FP, TN, and FN) assigned by different LLMs (GPT-4o, Gemini, Claude, and GPT-o1) and by the final evaluation. The *y*-axis represents the average number of Q&A pairs classified for each category, and the *x*-axis indicates the LLM used for the evaluation. Bar colors represent the evaluation outcomes: blue (True Positive, TP), orange (False Positive, FP), green (True Negative, TN), and red (False Negative, FN). The colored dotted lines indicate the target (ground truth) values, with colors corresponding to the respective classification outcomes. Error bars represent the standard deviation of the results obtained over three independent runs. Below each bar chart, the specific evaluation prompt used for the respective iteration is provided.

1.5-Pro, and Claude 3.5 Sonnet all overwhelmingly labeled the Q&A pairs as TPs, with only a few instances being marked as FPs. Notably, none of these three models identified any of the TNs present in the set. Claude 3.5 Sonnet also labeled a couple of examples as FNs. In contrast, GPT-o1 was the only model that approached the correct evaluation: it successfully classified 6 out of the 9 TN-type Q&A pairs correctly, with the remaining 3 misclassified as TPs. Moreover, GPT-o1 exhibited the most consistent behavior, with the lowest variance across evaluations (see Fig. 2).

In the second iteration (Fig. S7), we refined the prompt provided to the LLM to improve the accuracy of its classification outputs. Specifically, we added clarifications to prevent the LLM from misclassifying vague or implicitly stated answers as TPs. We also emphasized caution when dealing with reasoning-based questions or those containing superlatives like 'always' or 'best', which tend to lead to overconfident labeling. Finally, we included an explicit instruction to ensure that justifications such as 'not explicitly stated in the context' are not used to justify TP or FP classifications. These changes were designed to improve consistency and bring the evaluations more in line with the ground truth. On running iteration 2 of the prompt on this, we observed a reduction in the number of TP classifications generated by some of the LLMs; however, we still could

not achieve the desired outcome. Specifically, for Gemini 1.5 Pro, the model consistently produced TP outcomes without generating any FP, TN, or FN classifications. For GPT-4o, while we successfully reduced the number of TP classifications, there was a corresponding increase in FP outcomes, and we were unable to elicit any TN classifications. Claude 3.5 Sonnet showed comparatively better performance, producing a small number of TN responses in addition to fewer TPs. Overall, the final evaluation reflected an increase in TN classifications, moving closer to our target; however, a small number of FP and FN classifications remained alongside a substantial portion of TPs, highlighting the necessity for further refinement of the prompt.

In iteration 3 (Fig. S8), we introduced an explicit example instructing the models not to rely on general domain knowledge (*e.g.*, "based on general chemistry knowledge") when classifying Q&A pairs as TPs or FPs. Iteration 4 (Fig. S9) built further on this by explicitly instructing the model to avoid labeling speculative questions – those requiring general knowledge rather than context-based reasoning – as FPs, reinforcing the constraint to use only the provided context. Finally, in iteration 4* (Fig. S10), we maintained the same instructions from iteration 4 but introduced an additional verification step, employing a secondary 'checker' prompt to independently reassess and validate the classification outputs of the initial evaluation prompt. In all three iterations – iteration 3, iteration 4, and iteration 4* – we did not observe any significant improvements in performance over iteration 2. Despite our attempts to explicitly constrain the models and introduce additional verification layers, the distribution of TP, FP, TN, and FN outcomes remained largely unchanged.

Given that we had hit a dead end with optimizing the prompt by hand, we next decided to use Claude 3.5 Sonnet as a tool to optimize the prompt for us. In this prompt optimization, we provided the LLM with explicit details including the original base prompt and a structured template designed specifically to address frequent misclassification errors. The template clearly highlighted common misclassification types (TP, FP, TN, and FN), provided detailed explanations of why these classifications were incorrect, and outlined the necessary corrections to prevent these errors (Fig. 3a). Using this approach, we significantly improved the performance of the automated evaluation agent, successfully reaching close to our target human-evaluated benchmark for the DOI we were analyzing, in this case nchem.834. For reference, readers are directed to Fig. 2, where the figure on the left shows the performance achieved in iteration 1, while the figure on the right illustrates the performance obtained from iteration 5, along with the associated prompts used. From iteration 5 (Fig. S11) to iteration 7 (Fig. S13), stricter requirements were introduced based on

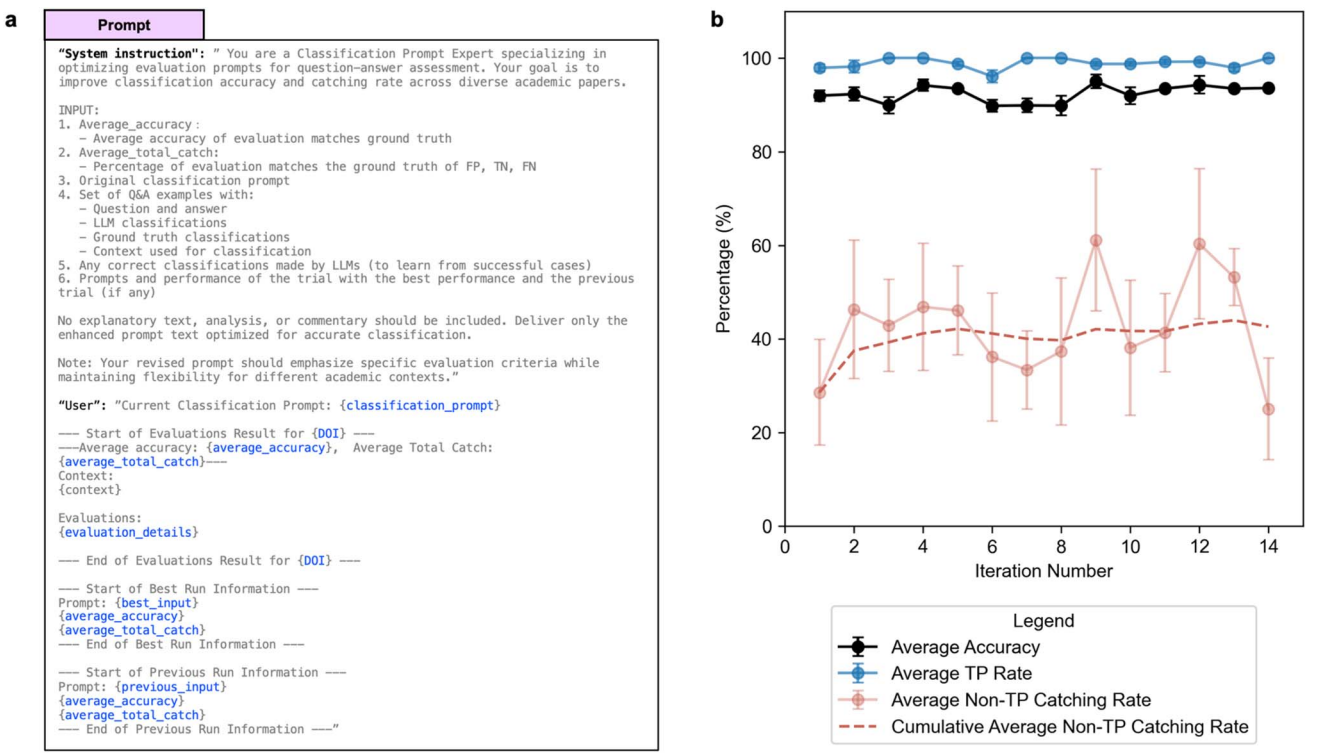

Fig. 3 Automated prompt refinement to improve the classification performance of non-TP type Q&A pairs. (a) Template utilized for prompt revision with highlighted placeholders indicating where performance metrics from prior prompts are inserted. This template was provided to the Claude API, facilitating the automated iterative refinement process illustrated in (b). (b) Average accuracy (black), average TP rate (blue), average non-TP catching rate (pink), and cumulative average non-TP catching rate (dashed line) shown as a function of iteration number. The final prompt selected for the single-hop evaluation task is the one used in iteration 9. The full prompt is shown in Fig. S15. Error bars indicate the standard deviation calculated over three independent runs for a set of seven DOIs.

Claude's suggestion, specifying that the question–answer pairs must be explicitly stated in the context, either verbatim or with minimal paraphrasing, thereby eliminating ambiguous inference. We also incorporated a clearly structured decision tree for distinguishing between TN and FN classifications, explicitly mandating verification against general scientific principles. Additionally, we removed previously ambiguous phrases like "directly sourced from context", replacing them with more precise language to minimize misinterpretation. These refinements, informed by the detailed prompt optimization template shown in Fig. S14, explicitly included the original prompt, clear descriptions of common misclassification errors, and structured examples of incorrect classifications, each accompanied by explicit explanations and corrective instructions. The use of this comprehensive template resulted in a marginal improvement in performance. A summary of the iterative prompt refinement procedure is provided in Table S4. We must add that while we were working on this project, we also tested Deepseek[24] and Grok as they had recently come out and realized that their performance in iteration 6 (Fig. S12) and iteration 7 (Fig. S13) was slightly worse off than the other models and therefore decided to proceed without them. In our previous work, we had highlighted that it was a challenge eliminating single-hop Q&A pairs when generating multi-hop Q&A datasets. We attempted to apply the same strategy used above to see if we could address this issue and improve the generation of multi-hop Q&A pairs. Using Claude as the prompt generator helped us to find a prompt that significantly improved the generation of multi-hop Q&A pairs. This prompt has been incorporated into RetChemQA and is now used for generating multi-hop Q&A pairs henceforth. The prompt is shown in Fig. S15.

We next tested the prompt obtained in iteration 6 (Fig. S12) on a set of seven DOIs (anie.200351546, adfm.202203745, anie.202306048, anie.202009613, anie.200462126, adma.200904238, and nchem.834) specifically chosen due to their high proportion of non-TP type Q&A pairs. Given the increased difficulty of accurately classifying this set, we once again leveraged Claude 3.5 Sonnet for automated prompt optimization, following the structured template depicted in Fig. 3a. This template included explicit placeholders (highlighted in blue) detailing the current classification prompt, average accuracy, total catching rate, individual DOI evaluations, and information from previous iterations. The optimization results from this process are illustrated in Fig. 3b. From this subsequent round of iterative refinement, we selected the prompt obtained at iteration 9, as it demonstrated the highest accuracy and non-TP catching rate. This prompt, chosen as the final version for integration into our automated evaluation agent, is provided in Fig. S16. A summary of the different iterations and their corresponding outcomes, including changes made, key observations, and evaluation notes, is provided in Table S1. After identifying the prompt that performed best for the single-hop Q&A evaluation task, we decided to directly test the same prompt in the multi-hop Q&A evaluation task. Surprisingly, we found that this prompt also performed exceptionally well for the multi-hop Q&A pairs, achieving high accuracy and effectively classifying the non-TP type questions. Given these results, we adopted this prompt as our final choice for the multi-hop Q&A evaluation task.

Architecture of the agent and the prompt for synthesis conditions evaluation task

To optimize the evaluation prompt specifically for assessing synthesis conditions, we adopted criteria previously defined in our earlier work.[25] These evaluation criteria include: (1) completeness, ensuring that synthesis conditions are extracted for all MOFs mentioned in the context; (2) data type, verifying that the extraction exclusively covers synthesis details and excludes any experimental characterization data (such as XRD patterns, surface area measurements, or spectroscopic information); and (3) accuracy, confirming the correct matching of extracted synthesis conditions to their corresponding MOFs. By simply providing these guidelines in the prompt, we were able to obtain good performance using our automated evaluation agent and therefore decided not to do any further prompt optimization. The final prompt used is shown in Fig. 4. Detailed descriptions of the algorithms implemented for data processing, prompt optimization, and automated evaluation are provided in the SI: Fig. S17 illustrates the algorithm for generating the "user" section of the prompt, compiling the entire

Prompt

```
"System instruction": "Please evaluate ALL the synthesis conditions extracted from the provided
context, focusing on Metal-Organic Frameworks (MOFs). Analyze according to these three criteria:
Criterion 1: Completeness
* Have synthesis conditions been included for all MOFs mentioned in the context?
Criterion 2: Data Type
* Verify that ONLY synthesis information has been extracted
* Ensure all experimental characterization data (like XRD patterns, surface area measurements,
spectroscopic data) has been excluded from the extraction
Criterion 3: Accuracy
* Confirm that all synthesis conditions details have been extracted and that they are correctly
matched to their corresponding MOFs

For each criterion, output Y if fully met for ALL MOFs, or N if not met for ANY MOF"


"User": " CONTEXT:{context}\n\nSynthesis Conditions Data::{Q&A data set}"
```

Fig. 4 The final prompt used in the automated evaluation agent for the synthesis conditions task. The prompt is used to evaluate the synthesis conditions dataset, based on three distinct criteria: completeness (criterion 1), data type (criterion 2), and accuracy (criterion 3) for 98 DOIs.

context and all relevant Q&A pairs or synthesis conditions datasets for each DOI; Fig. S18 presents the algorithm for systematically calling various LLMs *via* API for evaluation tasks; Fig. S19 details the iterative algorithm used to automate the optimization of the "system instruction" section of the prompt *via* Claude's API; and Fig. S20 describes the function responsible for summarizing evaluations, processing results from multiple LLMs through merging and weighted evaluation. All the code necessary for performing these processing steps and evaluations on both single-hop and multi-hop Q&A datasets, along with the synthesis conditions dataset and example full length prompts for each task, is publicly available at the following GitHub repository: **https://github.com/joef2002/QAutoEval**. In addition, we have also developed a user-friendly GUI app that users can download to run evaluations on their own datasets.

Results

To rigorously test the performance of the final prompt selected for evaluating single-hop and multi-hop Q&A datasets, we randomly selected 252 DOIs, each containing approximately 20 Q&A pairs. A distribution of the Q&A types in the optimization dataset (7 DOIs) and final test dataset (252 DOIs) is provided in Table S5. Upon evaluating this extensive dataset using our automated agent, we observed high accuracy rates ranging between 97 and 98%. Although these results were very encouraging, we wanted to understand the specific cases where the automated evaluation might diverge from the human-evaluated ground truth.

To implement this, we categorized the DOIs based on the percentage of matching evaluations between our automated agent and human evaluations, as depicted in Fig. 5a. In this figure, intervals on the *x*-axis represent the percentage range of Q&A pairs correctly classified by the automated evaluation compared to human evaluations. For example, the interval labeled (90, 100) indicates that precisely >90% of Q&A pairs within a DOI matched human evaluations. Correspondingly, the *y*-axis indicates the number of DOIs within each interval. Notably, we observed more than 100 DOIs in the (90, 100) interval. Upon manually examining a random subset of 20–30 DOIs within this interval, we discovered that the discrepancies were due to incorrect human evaluations rather than errors made by our agent. This indicates that the actual accuracy of our automated agent is likely higher than initially calculated. We also performed a quantitative analysis for the multi-hop Q&A pairs. On average, across all three evaluations, total Q&As = 523, total mismatches found = 45, QAutoEval correct = 30 (66.7%), human evaluation correct = 11 (24.4%), and both wrong = 4 (8.9%). A pie chart summarizing this analysis is shown in Fig. S21. The analysis also highlights common sources of misclassification, such as (i) PDFs containing more than one paper, (ii) handwritten or scanned data within papers, and (iii) incomplete or inconsistently formatted material names. These cases illustrate how ambiguous inputs and inconsistent formatting can lead to incorrect predictions despite correct model logic.

Another significant observation is that the final evaluation results from our agent (dark blue bars in Fig. 5a) show the highest frequency in the (90, 100) interval. This clearly demonstrates the benefit of combining multiple LLM outputs rather than relying on any single LLM evaluation. Interestingly, we found that GPT-o1, which was specifically designed for reasoning tasks and was thus weighted more heavily in our evaluation agent, occasionally performed worse than other models. This further underscores the importance of using an ensemble of multiple LLMs rather than relying solely on a single specialized model.

Recognizing that LLM outputs can vary between different runs, we next conducted a more detailed evaluation by selecting 26 DOIs and running the evaluation process three times for each DOI, separately analyzing both single-hop (Fig. 5b) and multi-hop Q&A pairs (Fig. 5c). In both single-hop and multi-hop scenarios, we found consistently high TP catch rates across all LLMs and notably high overall accuracy from the final ensemble evaluation. Although the non-TP catch rates were comparatively lower, they were still significantly higher than the initial results obtained when we first started our study.

For multi-hop Q&A pairs specifically (Fig. 5c), non-TP catch rates showed notably greater variability, reflected by the higher standard deviations. Among the models tested, GPT-o1 displayed the largest error bars, indicating substantial variability in its evaluations for multi-hop reasoning tasks. We hypothesize that this is because the task of evaluating multi-hop Q&As is more complex, as it requires combining and checking information from multiple sections of a paper. These findings further justify the need to employ multiple LLMs within a unified evaluation framework for reliability and consistency in Q&A pair classification, rather than depending solely on a single LLM. The results obtained from our automated evaluation agent for the synthesis conditions dataset are shown in Fig. 5d, with the corresponding prompt illustrated in Fig. 4. From these results, we clearly see varying performance among the different LLMs for each of the three evaluation criteria. For criterion 1 (Completeness) – which ensures that all MOFs mentioned in the context have their synthesis conditions extracted – we observe that Claude achieves the highest accuracy, outperforming GPT-4o, GPT-o1, and Gemini. Notably, however, the highest overall performance for criterion 1 is achieved by our final ensemble evaluation. For criterion 2 (Data Type), which confirms that only synthesis conditions and no experimental characterization details are extracted, and criterion 3 (Accuracy), which verifies correct matching of synthesis conditions to their corresponding MOFs, GPT-o1 significantly outperforms the other LLMs. These tasks inherently involve complex reasoning, given the many ways in which synthesis conditions are described or referenced across the scientific literature. Many publications, for instance, often simply cite another paper for detailed synthesis conditions rather than explicitly stating them, complicating the extraction and evaluation process. While GPT-o1 demonstrates superior performance in these reasoning-intensive criteria, we find that relying solely on a single LLM – regardless of its specialized capabilities – can limit reproducibility and consistency. Therefore, despite our final evaluation slightly

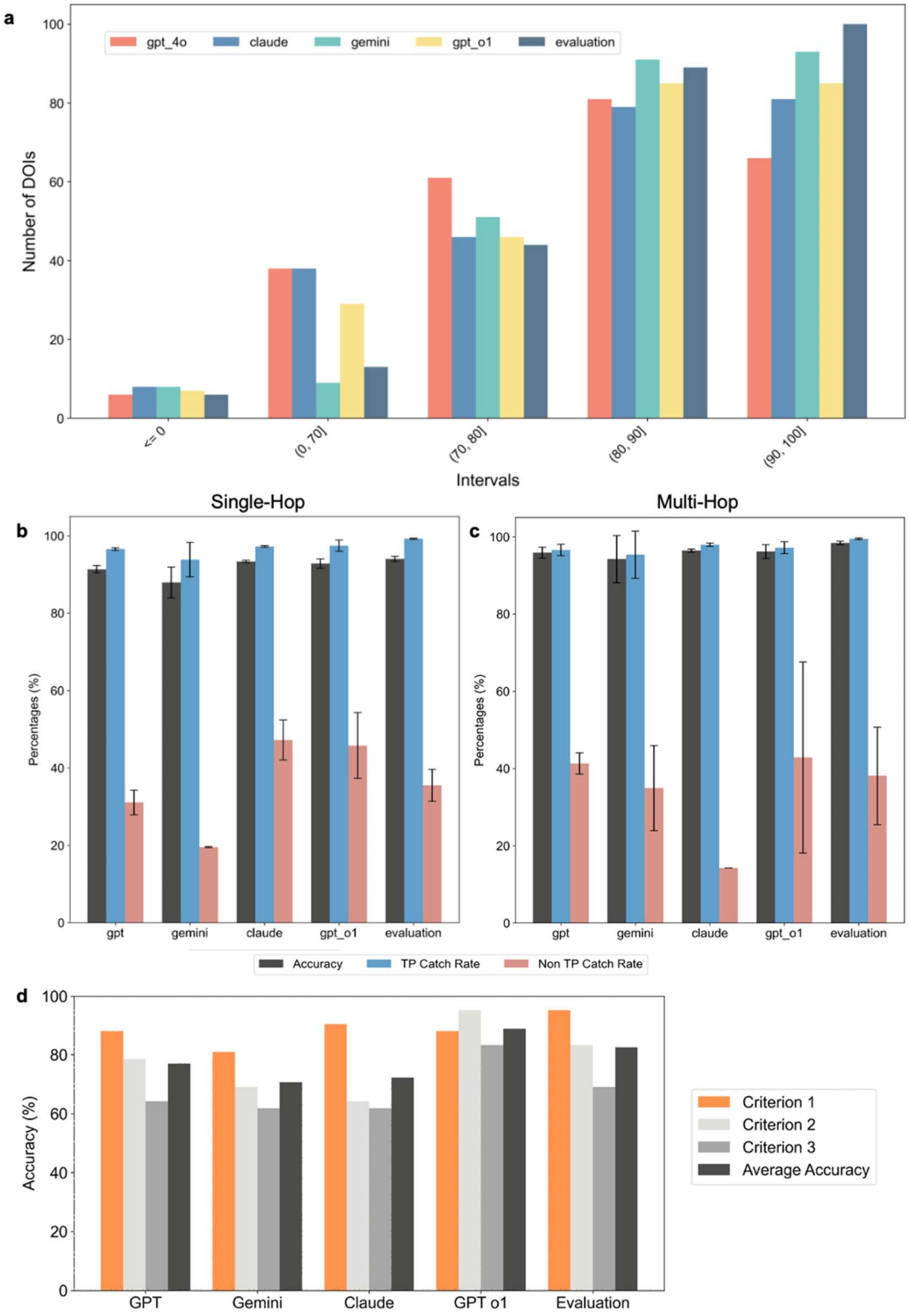

Fig. 5 Performance evaluation of the best-performing prompt on single-hop and multi-hop Q&A tasks and the synthesis conditions evaluation tasks. (a) Distribution plot showing the number of DOIs categorized by intervals representing the percentage agreement between the LLMs' evaluations and the human-generated ground truth for single-hop Q&As from 252 DOIs. (b and c) Accuracy, TP catch rate, and non-TP catch rate (%) for individual LLMs (GPT-4o, Gemini, Claude, and GPT-o1) and the final evaluation across (b) single-hop and (c) multi-hop Q&A datasets, each consisting of evaluations for 26 DOIs. Error bars indicate the standard deviation calculated over three independent runs. (d) Performance of different LLMs – GPT-4o, Gemini, Claude, and GPT-o1 – and the final evaluation in accurately assessing synthesis conditions according to these criteria. Accuracy values for each criterion, as well as the average accuracy across all criteria, are shown for comparison.

underperforming GPT-o1 in these categories, we maintain that employing a distributed evaluation model, combining outputs from multiple LLMs, is essential to ensuring robust and reproducible results across varied data sets.

To further assess the generalizability of QAutoEval, we evaluated its performance on Q&A datasets drawn from diverse areas of chemistry, including batteries,[26] biosynthesis,[27] catalysis,[28] materials chemistry,[29] synthetic organic chemistry,[30] and natural product chemistry.[31] For single-hop Q&A pairs, QAutoEval achieved ~90% accuracy, while for multi-hop Q&A pairs it achieved ~98% accuracy. These results match the performance observed for papers in reticular chemistry, demonstrating that both the framework and the underlying prompt are topic-agnostic. This confirms that QAutoEval can be reliably applied for the evaluation of datasets across multiple domains of chemistry, not just within the specialized context of reticular chemistry.

On average, the cost of running an evaluation using GPT-4o, Claude, and GPT-o1 was approximately $1.5–2 per DOI, while that of Gemini was very low and almost negligible. These estimates provide a practical reference for assessing the computational feasibility of large scale evaluations using QAutoEval.

Summary and conclusions

We have developed an automated evaluation agent (QAutoEval) that reliably assigns classification labels to single-hop and multi-hop Q&A pairs, as well as to synthesis condition datasets. A key takeaway from our work is the critical importance of the prompt and that it remains central irrespective of the model used. This observation aligns with recent studies in other fields, such as image generation, that have similarly highlighted the role of the prompt in determining the final performance of a given LLM.[32] Although GPT-o1 was considered state-of-the-art for reasoning tasks, our original prompt, which contained straightforward definitions sufficient for human evaluators, performed poorly with the model. It was only after extensive optimization, resulting in a significantly longer and highly detailed prompt aided by an LLM, that we achieved substantial improvements in classification performance. This highlights that current LLMs still lack human-level reasoning capabilities and emphasizes the need for further development in this area. Another important finding is that evaluation systems should not depend solely on a single LLM. Instead, robust evaluations require a distributed approach that leverages multiple LLMs to ensure consistency and reproducibility. Furthermore, due to inherent variability in LLM outputs, we found that performing evaluations multiple times (at least three) is essential to reliably establish accurate classifications.

We also recognize the need to generate more diverse and balanced datasets to improve evaluation robustness. In future work, we aim to expand the dataset using strategies such as adversarial generation, where prompts are designed to create more challenging or ambiguous Q&A pairs, and synthetic augmentation, where controlled prompt variations introduce a wider range of FP, TN, and FN examples. These methods can help mitigate dataset imbalance and ensure that automated evaluation systems generalize effectively across different question types and contexts.

Our agent will enable the community to move towards automated reinforcement learning systems that eliminate the need for human feedback, which is often labor-intensive, expensive, and error-prone. Additionally, we envision a future where specialized, optimized prompts become valuable intellectual property, since much of an LLM's effectiveness relies heavily on the prompt used. Ultimately, the prompt will likely emerge as the critical factor distinguishing high-performing models from weaker ones in specific tasks.

Author contributions

N. R., J. D. F., C. B., J. T. C., and O. M. Y. conceived the idea and drafted the outline. All code implementations were carried out by J. D. F. N. R. wrote the initial draft of the manuscript. J. D. F. and N. R. made and designed the figures. C. Z. helped with the evaluation of the multi-hop Q&A datasets used in this paper. All authors contributed to the review and editing of the final manuscript.

Conflicts of interest

All authors declare no competing interests.

Data availability

All code supporting this article, including processing and evaluation scripts for the single-hop and multi-hop Q&A datasets and the synthesis conditions dataset, is openly available at GitHub: **https://github.com/joef2002/QAutoEval**, with an archived version available at Zenodo (DOI: **https://doi.org/10.5281/zenodo.17479256**). A user-friendly GUI application is also provided at the same repository for running evaluations on custom datasets.

Supplementary information is available. See DOI: **https://doi.org/10.1039/d5dd00413f**.

Acknowledgements

N. R. and C. Z. acknowledge the Bakar Institute of Digital Materials for the Planet (BIDMaP) Emerging Scholars Program for the funding that supports this work.

References

1 X. Luo, *et al.*, Large language models surpass human experts in predicting neuroscience results, *Nat. Hum. Behav.*, 2024, **9**(2), 305–315, DOI: **10.1038/s41562-024-02046-9**.
2 A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan, L. Gutierrez, T. F. Tan and D. S. W. Ting, Large language models in medicine, *Nat. Med.*, 2023, **29**(8), 1930–1940, DOI: **10.1038/s41591-023-02448-8**.
3 Y. Zheng, *et al.*, Large language models for scientific discovery in molecular property prediction, *Nat. Mach. Intell.*, 2025, **7**(3), 437–447, DOI: **10.1038/s42256-025-00994-z**.
4 O. M. Yaghi, Reticular Chemistry in All Dimensions, *ACS Cent. Sci.*, 2019, **5**(8), 1295–1300, DOI: **10.1021/**

ACSCENTSCI.9B00750/ASSET/IMAGES/MEDIUM/OC9B00750_0003.GIF.

5 Z. Zheng, N. Rampal, T. J. Inizan, C. Borgs, J. T. Chayes and O. M. Yaghi, Large language models for reticular chemistry, *Nat. Rev. Mater.*, 2025, 1–13, DOI: 10.1038/s41578-025-00772-8.

6 K. M. Jablonka, P. Schwaller, A. Ortega-Guerrero and B. Smit, Leveraging large language models for predictive chemistry, *Nat. Mach. Intell.*, 2024, **6**(2), 161–169, DOI: 10.1038/s42256-023-00788-1.

7 M. Caldas Ramos, J. Christopher Collison and A. D. White, A review of large language models and autonomous agents in chemistry, *Chem. Sci.*, 2025, **16**(6), 2514–2572, DOI: 10.1039/D4SC03921A.

8 A. M. Bran, S. Cox, O. Schilter, C. Baldassari, A. D. White and P. Schwaller, Augmenting large language models with chemistry tools, *Nat. Mach. Intell.*, 2024, **6**(5), 525–535, DOI: 10.1038/s42256-024-00832-8.

9 D. A. Boiko, R. MacKnight, B. Kline and G. Gomes, Autonomous chemical research with large language models, *Nature*, 2023, **624**(7992), 570–578, DOI: 10.1038/s41586-023-06792-0.

10 Y. Zou, *et al.*, El Agente: An Autonomous Agent for Quantum Chemistry, *Matter*, 2025, **8**(7), 102263, DOI: 10.1016/j.matt.2025.102263.

11 T. Song, *et al.*, A Multiagent-Driven Robotic AI Chemist Enabling Autonomous Chemical Research On Demand, *J. Am. Chem. Soc.*, 2025, **147**(15), 12534–12545, DOI: 10.1021/JACS.4C17738/ASSET/IMAGES/LARGE/JA4C17738_0004.JPEG.

12 G. Chatziveroglou, R. Yun and M. Kelleher, Exploring LLM Reasoning Through Controlled Prompt Variations, 2025, Accessed: Apr. 26, 2025. [Online]. Available: https://arxiv.org/abs/2504.02111v1.

13 Y. Kang, W. Lee, T. Bae, S. Han, H. Jang and J. Kim, Harnessing Large Language Models to Collect and Analyze Metal-Organic Framework Property Data Set, *J. Am. Chem. Soc.*, 2025, **147**(5), 3943–3958, DOI: 10.1021/JACS.4C11085/ASSET/IMAGES/LARGE/JA4C11085_0009.JPEG.

14 X. Bai, *et al.*, Construction of a knowledge graph for framework material enabled by large language models and its application, *npj Comput. Mater.*, 2025, **11**(1), 1–9, DOI: 10.1038/s41524-025-01540-6.

15 A. Mirza, *et al.*, A framework for evaluating the chemical knowledge and reasoning abilities of large language models against the expertise of chemists, *Nat. Chem.*, 2025, **17**(7), 1027–1034, DOI: 10.1038/s41557-025-01815-x.

16 T. M. Pruyn, A. Aswad, S. T. Khan, J. Huang, R. Black and S. M. Moosavi, MOF-ChemUnity: Literature-Informed Large Language Models for Metal–Organic Framework Research, *J. Am. Chem. Soc.*, 2025, DOI: 10.1021/jacs.5c11789.

17 L. Ouyang, *et al.*, Training language models to follow instructions with human feedback, *Advances in Neural Information Processing Systems*, 2022, vol. 35, pp. 27730–27744, DOI: 10.5555/3600270.3602281.

18 P. F. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg and D. Amodei, Deep reinforcement learning from human preferences, *Adv. Neural Inf. Process Syst.*, 2017, 4300–4308, Jun. 2017, Accessed: Apr. 26, 2025. [Online]. Available: https://arxiv.org/abs/1706.03741v4.

19 N. Rampal, *et al.*, Single and Multi-Hop Question-Answering Datasets for Reticular Chemistry with GPT-4-Turbo, *J. Chem. Theory Comput.*, 2024, **20**(20), 9128–9137, DOI: 10.1021/ACS.JCTC.4C00805.

20 O. Khattab *et al.*, DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines", 2023, Accessed: Apr. 23, 2024. [Online]. Available: https://arxiv.org/abs/2310.03714v1.

21 OpenAI, *GPT-4 Technical Report*, 2023, [Online]. Available: http://arxiv.org/abs/2303.08774.

22 OpenAI *et al.*, *OpenAI o1 System Card*, 2024, Accessed: Apr. 26, 2025. [Online]. Available: https://arxiv.org/abs/2412.16720v1.

23 G. Team *et al.*, Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context", 2024, Accessed: Apr. 26, 2025. [Online]. Available: https://arxiv.org/abs/2403.05530v5.

24 DeepSeek-AI *et al.*, DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning", 2025, Accessed: Apr. 26, 2025. [Online]. Available: https://arxiv.org/abs/2501.12948v1.

25 Y. Shi, N. Rampal, C. Zhao, C. Borgs, J. T. Chayes and O. M. Yaghi, Comparison of LLMs in Extracting Synthesis Conditions and Generating Q&A Datasets for Metal-Organic Frameworks, *Digital Discovery*, 2025, **4**, 2676–2683, DOI: 10.1039/D5DD00081E.

26 I. R. Choi, *et al.*, Asymmetric ether solvents for high-rate lithium metal batteries, *Nat. Energy*, 2025, **10**(3), 365–379, DOI: 10.1038/s41560-025-01716-w.

27 M. B. Sosa, J. T. Leeman, L. J. Washington, H. V. Scheller and M. C. Y. Chang, Biosynthesis of Strained Amino Acids by a PLP-Dependent Enzyme through Cryptic Halogenation, *Angew. Chem., Int. Ed.*, 2024, **63**(31), e202319344, DOI: 10.1002/ANIE.202319344.

28 Z. Yu, *et al.*, Synthesis of Cyclopentadiene and Methylcyclopentadiene with Xylose or Extracted Hemicellulose, *Angew. Chem., Int. Ed.*, 2023, **62**(13), e202300008, DOI: 10.1002/ANIE.202300008.

29 Y. Song, *et al.*, High-Entropy Design for 2D Halide Perovskite, *J. Am. Chem. Soc.*, 2024, **146**(29), 19748–19755, DOI: 10.1021/JACS.4C01882/ASSET/IMAGES/LARGE/JA4C01882_0006.JPEG.

30 H. Lindner, W. M. Amberg and E. M. Carreira, Iron-Mediated Photochemical Anti-Markovnikov Hydroazidation of Unactivated Olefins, *J. Am. Chem. Soc.*, 2023, **145**(41), 22347–22353, DOI: 10.1021/JACS.3C09122/ASSET/IMAGES/LARGE/JA3C09122_0005.JPEG.

31 A. McDonald, *et al.*, Enzymatic epimerization of monoterpene indole alkaloids in kratom, *Nat. Chem. Biol.*, 2025, 1–10, DOI: 10.1038/s41589-025-01970-9.

32 E. Jahani *et al.*, As Generative Models Improve, People Adapt Their Prompts, 2024, Accessed: Aug. 17, 2025. [Online]. Available: https://arxiv.org/abs/2407.14333v1.

