CAN LLMS DESIGN EXPERIMENTS? BENCHMARKING EXPERIMENT DESIGN FOR AUTONOMOUS LABS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) have demonstrated strong capabilities in chemical reasoning and reaction prediction. However, knowing *what reaction to perform* does not necessarily imply knowing *how to perform it in the laboratory*. Experimental design requires translating scientific knowledge into concrete decisions about materials, operations, conditions, equipment, and execution order. Yet this capability remains largely underexplored in current LLMs. We introduce *LabProtocol*, a dataset and benchmark for learning and evaluating experimental design. LabProtocol aligns chemical reactions with real experimental procedures and represents each procedure as a sequence of structured, parameterized laboratory operations. These operations capture the materials, conditions, equipment, and temporal dependencies required to perform an experiment. Using LabProtocol, we evaluate leading LLMs on tasks ranging from complete procedure generation to single-operation completion. We find that current LLMs struggle to generate complete experimental procedures. More surprisingly, they still perform poorly when asked to predict only a single missing operation, suggesting that the limitation goes beyond long-horizon planning and reflects a lack of experimental procedural knowledge. We further fine-tune Qwen on LabProtocol and observe modest improvements in complete protocol generation but little improvement in single-operation completion. This contrast suggests that learning global protocol patterns does not necessarily yield fine-grained procedural understanding. LabProtocol provides a foundation for developing autonomous laboratory agents that translate scientific objectives into executable experiments.

1 INTRODUCTION

Large language models (LLMs) are increasingly capable of reasoning about chemistry and materials science. Recent models can answer scientific questions, predict reaction outcomes, propose synthesis routes, and assist with scientific discovery Mirza et al. (2025); Bran et al. (2024); Boiko et al. (2023). These advances raise a natural question for autonomous laboratories: *Can an LLM design the experiment needed to realize a scientific objective?*

This question goes beyond reaction prediction. Knowing that a set of reactants can produce a target product does not specify how the reaction should be carried out. A laboratory experiment requires a sequence of concrete decisions: which materials to use and in what quantities, when and how to add them, which operations to perform, what temperature and duration to use, which equipment is required, and how intermediate products should be processed Vaucher et al. (2021); Mehr et al. (2020). For example, a reaction representation may specify the reactants, products, and major conditions, while the actual experiment additionally requires operations such as addition, stirring, heating, centrifugation, washing, drying, and annealing. We refer to the ability to translate scientific objectives and reaction knowledge into such concrete procedures as *experimental design*.

Despite its importance, experimental design remains difficult to study systematically with existing scientific datasets Liu et al. (2024); O'Donoghue et al. (2023). Reaction datasets typically focus on reactants, products, and reaction conditions Lowe (2017); Kearnes et al. (2021). Synthesis planning benchmarks primarily evaluate which reactions or routes can lead to a desired product Segler et al. (2018); Genheden & Bjerrum (2022). Meanwhile, the operational knowledge required to perform these reactions is largely recorded as natural-language procedures in papers, supporting information,

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

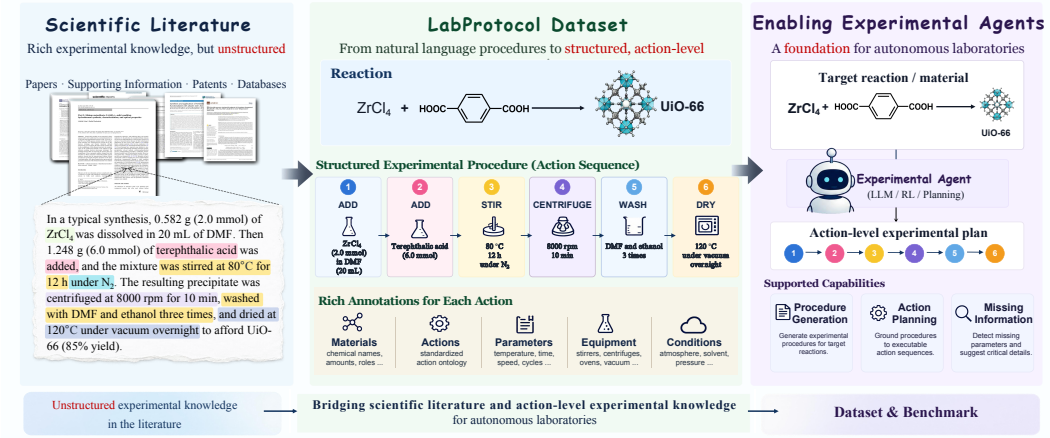


Figure 1: Overview of LabProtocol. LabProtocol converts unstructured experimental procedures from scientific literature into structured, parameterized laboratory operation sequences for learning, and evaluating reaction-conditioned procedure design.

patents, and experimental databases. Such descriptions contain rich information about operations, parameters, equipment, and temporal dependencies, but this information is rarely represented in a form suitable for training and evaluating laboratory agents. As a result, strong performance on reaction-centric benchmarks does not tell us whether a model knows how to perform the corresponding experiment.

To address this gap, we introduce **LabProtocol**, a dataset and benchmark for studying *reaction-conditioned laboratory procedure design*. LabProtocol aligns chemical reactions with their corresponding experimental procedures and represents each procedure as an ordered sequence of structured, parameterized laboratory operations. Rather than treating a protocol as free-form text, each operation explicitly specifies its action type, associated materials, operational parameters, equipment, and experimental conditions. As illustrated in Figure 1, LabProtocol transforms unstructured experimental knowledge from scientific literature into structured, action-level procedural representations, bridging reaction-level chemical descriptions and the operational knowledge required for laboratory execution. This representation exposes the procedural decisions that are otherwise implicit in natural-language experimental descriptions and enables experimental design to be evaluated at multiple levels, from generating a complete procedure to recovering an individual missing operation given its surrounding context. LabProtocol thus provides a concrete framework for studying whether language models know not only *what reaction to perform*, but also *how to carry it out*.

Using LabProtocol, we evaluate leading LLMs at two complementary levels. First, we ask models to generate a complete experimental procedure for a given reaction, testing their ability to organize a coherent sequence of laboratory operations. Second, we introduce a controlled **single-operation completion** task, where most of the procedure is provided and the model only needs to recover one missing operation. This task substantially reduces the burden of long-horizon planning. It therefore provides a diagnostic test of whether a model understands local experimental procedures themselves. Across the LLMs we evaluate, performance remains limited on complete procedure generation. More strikingly, models also struggle with single-operation completion. These results suggest that the difficulty of experimental design cannot be explained solely by long-horizon generation: current LLMs still lack fine-grained experimental procedural knowledge.

We further investigate whether this capability can be improved through direct training on LabProtocol. Fine-tuning Qwen modestly improves complete protocol generation but does not improve single-operation completion, suggesting that learning common protocol patterns alone is insufficient for acquiring robust local procedural knowledge.

Our work therefore shifts the focus from asking whether models can reason about reactions to asking whether they can design the experiments required to realize them. Our main contributions are:

- We formulate **experimental design** as an explicit learning and evaluation problem for autonomous laboratory agents, distinguishing it from reaction prediction and synthesis planning.

- We introduce **LabProtocol**, which aligns chemical reactions with real experimental procedures and represents them as structured, parameterized laboratory operations with materials, conditions, equipment, and temporal dependencies.
- We systematically evaluate leading LLMs on both complete procedure generation and controlled single-operation completion, revealing that current models struggle even when long-horizon planning is largely removed.
- We study whether experimental design can be improved through direct training on procedural data, and find that gains in complete protocol generation do not automatically translate into improved local procedural reasoning.

2 Related Work

Reaction Prediction and Synthesis Planning. Large reaction datasets have driven substantial progress in forward reaction prediction, retrosynthesis, and multi-step synthesis planning Schneider et al. (2015); Lowe (2017); Kearnes et al. (2021); Schwaller et al. (2019); Segler et al. (2018). These tasks primarily model chemical transformations through reactants, reagents, products, and reaction conditions, answering *what reaction or route can produce a target*. However, they generally do not specify the laboratory operations required to carry out the reaction.

Experimental Procedure Modeling. A closer line of work focuses on extracting and generating experimental procedures Liu et al. (2024); Zhang et al. (2025; 2026); O'Donoghue et al. (2023). The 2Actions Vaucher et al. (2020) converts natural-language synthesis descriptions into structured laboratory action sequences with associated compounds, quantities, and conditions. Vaucher et al. (2021) further studied reaction-to-procedure generation using approximately 690K standardized reaction–procedure pairs derived from patents. These works establish structured experimental procedures as a learnable representation and are the closest predecessors to LabProtocol.

LLMs for Autonomous Experimentation. Recent systems such as ChemCrow Bran et al. (2024) and Coscientist Boiko et al. (2023) combine LLM reasoning with scientific tools and laboratory automation, demonstrating the potential of language models for autonomous experimentation. LabProtocol complements these efforts by focusing on the procedural capability of the underlying models. Rather than introducing reaction-to-procedure generation itself, we use structured protocols to systematically test modern LLMs at both the complete-protocol and individual-operation levels, and study whether this capability can be improved through explicit training on experimental procedures.

3 Experimental Design

We study *reaction-conditioned experimental design*: given a desired synthesis product or material, can a language model infer the laboratory procedure required to obtain it? Unlike reaction prediction, which focuses on *what chemical transformation occurs*, our task requires generating the sequence of laboratory operations needed to realize that transformation, together with the relevant materials, conditions, and operational parameters.

We consider two complementary settings. *Complete Protocol Generation* evaluates the end-to-end ability to construct an experimental procedure from reaction-level information. *Single-Operation Completion* provides most of the procedure and asks the model to recover one missing operation, allowing us to probe local experimental procedural knowledge with substantially reduced long-horizon planning.

3.1 Problem Formulation

Let a chemical reaction be represented as $r = (\mathcal{R}, \mathcal{P})$, where $\mathcal{R}$ denotes the set of reactants and reagents and $\mathcal{P}$ denotes the target product. Additional reaction-level information, such as known conditions, may be provided when available.

The corresponding experimental protocol is represented as an ordered sequence of laboratory operations: $\Pi = (a_1, a_2, \ldots, a_T)$, where T is the number of operations in the procedure. Each operation

contains an action type and, when available, its associated materials, parameters, equipment, and experimental conditions:

$$a_t = (o_t, \mathcal{M}_t, \mathcal{Q}_t, \mathcal{E}_t, \mathcal{K}_t).$$

Here, o_t denotes the operation type, such as ADD, STIR, HEAT, or FILTER; $\mathcal{M}_t$ denotes the materials involved; $\mathcal{Q}_t$ contains operation-specific parameters; $\mathcal{E}_t$ denotes equipment; and $\mathcal{K}_t$ captures contextual conditions such as atmosphere or pressure.

Experimental design is therefore modeled as $p(\Pi \mid r)$. Importantly, this mapping is inherently one-to-many: the same reaction may be realized by multiple valid experimental procedures. We therefore do not treat the task as exact reproduction of a reference text. Instead, we evaluate whether a generated protocol captures the laboratory operations and experimental information required to carry out the reaction.

3.2 COMPLETE PROTOCOL GENERATION

Given a reaction r, the model generates a complete experimental protocol

$$\hat{\Pi} = (\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_{\hat{T}}) \sim p_\theta(\Pi \mid r).$$

This task evaluates end-to-end experimental design. A successful prediction must identify the relevant laboratory operations, arrange them in a coherent order, and associate them with appropriate experimental information. Therefore, the task requires more than identifying a plausible reaction. The model must translate reaction-level knowledge into an operational procedure that specifies how the experiment should actually be performed.

3.3 SINGLE-OPERATION COMPLETION

Complete protocol generation jointly requires experimental knowledge and long-horizon procedural planning. To isolate local procedural knowledge, we additionally introduce *Single-Operation Completion*: Given a reference protocol

$$\Pi = (a_1, \ldots, a_{t-1}, a_t, a_{t+1}, \ldots, a_T),$$

we mask one operation a_t and provide the model with the reaction and the remaining protocol:

$$\Pi_{\setminus t} = (a_1, \ldots, a_{t-1}, \square, a_{t+1}, \ldots, a_T).$$

The model is then asked to recover the missing operation:

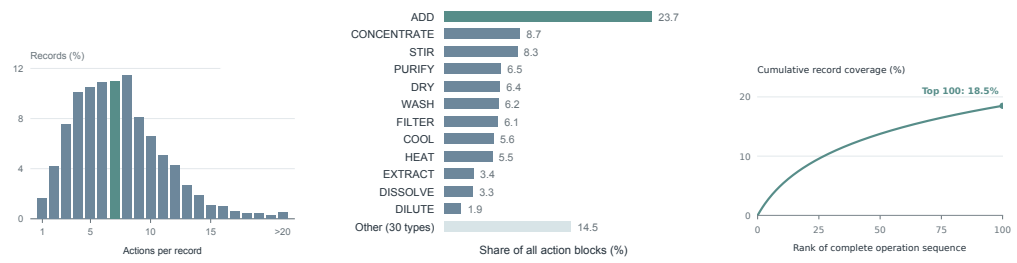
$$\hat{a}_t \sim p_\theta(a_t \mid r, \Pi_{\setminus t}).$$

Because both preceding and following operations are provided, this setting substantially reduces the need to construct a complete experimental plan from scratch. Instead, the model must identify the laboratory operation that is locally consistent with the reaction and the surrounding procedure.

4 LABPROTOCOL

Studying experimental design requires data that connect chemical transformations to the laboratory procedures used to realize them. Existing reaction datasets provide abundant information about reactants and products, but experimental procedures are typically represented as free-form text, when available at all. Such text is difficult to use directly for evaluating whether a model understands individual laboratory operations or their dependencies.

We construct **LabProtocol**, a dataset that aligns chemical reactions with experimental procedures and represents these procedures as structured sequences of laboratory operations. LabProtocol preserves multiple levels of experimental information, from the original reaction and natural-language procedure to normalized operations and their associated parameters. This design supports both end-to-end protocol generation and fine-grained analysis of experimental procedural knowledge.



(a) Procedure length (b) Operation distribution (c) Sequence diversity

Figure 2: Dataset statistics of LabProtocol. (a) Distribution of protocol lengths. (b) Distribution of normalized laboratory operation types. (c) Cumulative coverage of ranked operation sequences, illustrating procedural diversity.

4.1 Data Collection

We construct LabProtocol by integrating multiple complementary sources of experimental synthesis data, including OMG Kim et al. (2025), PcMSP Yang et al. (2022), the Solid-State Kononova et al. (2019) and Solution synthesis datasets Wang et al. (2022b), ULSA Wang et al. (2022a), and the Open Reaction Database (ORD) Kearnes et al. (2021), together with additional synthesis procedures curated from the scientific literature. We retain records that provide a reliable correspondence between a target reaction or material and a concrete experimental procedure.

Because these sources use different schemas and annotation conventions, we convert them into a unified representation by normalizing chemical entities, laboratory operations, equipment, experimental parameters, and conditions. Procedure steps are aligned to a canonical set of laboratory actions, with instruments and parameters attached whenever supported by the original source.

Importantly, *every record is manually reviewed* against its source procedure. Annotators verify the synthesis target, operation sequence, instruments, parameters, and conditions, and ambiguous or unreliable records are corrected or removed. Unreported experimental details are kept as missing rather than inferred. After integration, normalization, and manual verification, LabProtocol contains *18,729 curated experimental records*. Each record connects a chemical reaction with its corresponding experimental procedure and structured laboratory operations.

4.2 Structured Experimental Representation

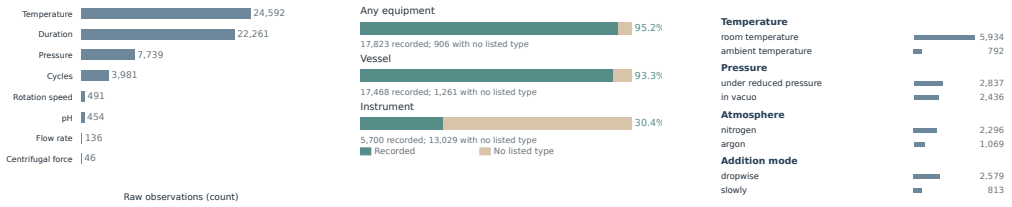
We convert each natural-language procedure into the structured representation defined in Section 3.1. The conversion consists of three steps: segmenting a procedure into atomic laboratory operations, normalizing semantically equivalent expressions into a shared operation vocabulary, and attaching operation-specific attributes extracted from the source text.

The resulting vocabulary contains *42 operation types*, covering reaction setup, mixing, thermal treatment, separation, purification, and post-processing. Each normalized operation remains aligned with its original textual span, allowing its associated materials, parameters, equipment, and conditions to be traced back to the source procedure.

We deliberately preserve incomplete reporting. Attributes absent from the original procedure are marked as missing. This produces an evidence-grounded representation that supports operation-level learning and evaluation without introducing unreported experimental details.

4.3 Dataset Statistics

LabProtocol contains *18,729 reaction–procedure records* and *42 normalized laboratory operation types*. The dataset exhibits substantial complexity at the procedure level: a protocol contains *7.4 operations on average*, with a median of *7* operations and an interquartile range of *5–9*. The 95th percentile reaches *14 operations*, while the longest protocols contain up to *70 operations*. Only



(a) Operation parameters (b) Equipment coverage (c) Experimental conditions

Figure 3: Structured operation-level information in LabProtocol: (a) quantitative experimental parameters, (b) equipment coverage across protocol records, and (c) representative qualitative experimental conditions.

1.6% of records contain a single operation. As shown in Figure 2a, most examples therefore involve multi-step experimental processes rather than isolated actions.

Figure 2b shows the distribution of normalized laboratory operations. ADD is the most frequent operation, accounting for **23.7%** of all action instances, while the dataset also contains a broad range of operations including stirring, heating, cooling, filtration, washing, drying, purification, extraction, dissolution, and concentration. This diversity reflects the multiple stages involved in experimental procedures, including preparation, reaction control, separation, workup, and product processing.

The dataset is also diverse at the sequence level. Across 18,729 records, we observe **8,282 unique operation sequences**. As shown in Figure 2c, the frequency of complete operation sequences exhibits a long tail, indicating that the corpus cannot be reduced to a small set of repeated procedural templates. Moreover, **27.4%** of records belong to sequences that occur only once.

Finally, LabProtocol retains rich execution-level information beyond operation labels. As shown in Figure 3, the dataset records quantitative operation parameters, equipment annotations, and qualitative experimental conditions. Temperature and duration are the most frequently reported parameters, with additional annotations covering pressure, operation cycles, rotation speed, pH, flow rate, and centrifugal force. Equipment and contextual conditions, such as reaction atmosphere, reduced pressure, and addition mode, are also preserved. Consequently, procedures with the same sequence of oeration types may still correspond to substantially different experiments depending on how individual steps are instantiated.

Taken together, these statistics highlight three properties of LabProtocol that are particularly relevant to reaction-conditioned procedure design: **multi-step complexity**, **procedural diversity**, and **execution-level detail**. Rather than representing a procedure as a sequence of action labels alone, LabProtocol captures both *which operations* are required and *how those operations are carried out*, providing a basis for evaluating whether models can generate complete and experimentally detailed procedures from reaction-level information.

5 CAN LLMS DESIGN EXECUTABLE EXPERIMENTS?

We evaluate whether current LLMs can translate reaction-level knowledge into laboratory procedures. We consider both Complete Protocol Generation and Single-Operation Completion, testing end-to-end experimental design and local procedural knowledge, respectively.

5.1 SETUP

Models. We evaluate representative proprietary and open-weight LLMs from the *GPT*, *Claude*, *Kimi*, *DeepSeek*, and *Qwen* families (OpenAI, 2026; Anthropic, 2026; Kimi Team, 2026; DeepSeek-AI, 2026b;a; Qwen Team, 2026). All models are evaluated on the same held-out test set with a shared task specification, operation vocabulary, and output format. Exact model versions and implementation details are provided in Appendix B.

Table 1: Comparison of different models on experimental procedure generation and sequence-level completion.

Model	Prec.	Rec.	F1	LCS	Seq. Mask
Qwen3.8-27B	47.87	44.09	45.90	40.44	18.90
GPT-5.6 Sol	43.74	65.50	52.46	59.81	41.20
Claude Opus 5.5	45.35	**66.64**	**53.97**	**61.57**	**44.60**
DeepSeek V4.1 Flash	**49.85**	45.00	47.30	41.97	27.00
DeepSeek V4 Pro	48.32	45.43	46.83	41.99	28.70
Kimi K3	41.60	60.96	49.45	56.44	31.40
Qwen3.8-27B-finetune	49.26	44.98	47.03	41.18	18.70

Table 2: Single-operation completion under different context settings. Seq. Mask provides only the operation sequence as context, whereas Full Mask provides the complete structured experimental record. Gain denotes the absolute improvement from Seq. Mask to Full Mask.

Model	Seq. Mask	Full Mask	Gain
Qwen3.8-27B	18.90	33.10	+14.20
GPT-5.6 Sol	41.20	57.20	+16.00
Claude Opus 5.5	**44.60**	**59.10**	+14.50
DeepSeek V4.1 Flash	27.00	47.20	**+20.20**
DeepSeek V4 Pro	28.70	47.70	+19.00
Kimi K3	31.40	48.30	+16.90
Qwen3.8-27B-finetune	18.70	32.90	+14.20

Evaluation. For *Complete Protocol Generation*, each model is given only the desired product and predicts an ordered sequence of laboratory operations. We report operation-level precision, recall, and F1, together with normalized longest common subsequence (LCS) to measure sequence-order agreement. For *Single-Operation Completion*, one operation in the reference protocol is replaced by `[MASK]`, and the model predicts the missing operation. We consider two context settings: **Seq. Mask**, which provides the product and surrounding operation sequence, and **Full Mask**, which additionally provides the complete structured experimental record. We report exact operation accuracy and the absolute gain from Seq. Mask to Full Mask.

Further details on prompts, masking, output constraints, and evaluation are provided in Appendix B.

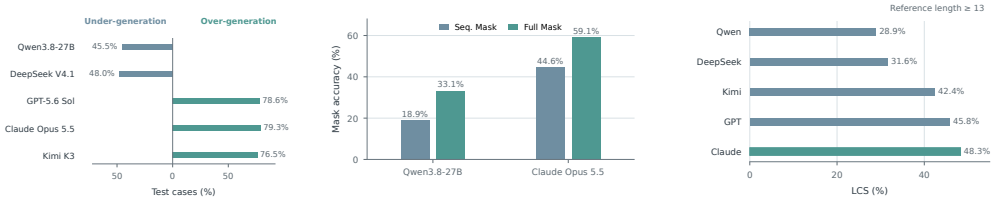
5.2 Main Results

Table 1 and Table 2 summarize the performance of current LLMs on LabProtocol. We have three observations.

Current LLMs remain limited in complete protocol generation. All evaluated models show substantial room for improvement when generating a complete experimental procedure from the desired product. The strongest model, Claude Opus 5.5, achieves an operation-level F1 of **53.97** and an LCS score of **61.57**. This indicates that even the best-performing model recovers only part of the reference operation set and sequence structure.

The results also reveal different failure profiles across models. DeepSeek V4.1 Flash achieves the highest precision (**49.85**) but substantially lower recall (**45.00**), suggesting more conservative predictions. Claude Opus 5.5, in contrast, achieves the highest recall (**66.64**) together with the strongest F1 and LCS, indicating better coverage of the reference procedure. Kimi K3 shows a similar high-recall, lower-precision pattern. These differences suggest that experimental protocol generation is not limited by a single failure mode: models vary in whether they omit necessary operations or over-generate plausible but unsupported ones.

The difficulty persists even when only one operation is missing. Poor complete-protocol performance might be attributed to long-horizon generation, since a model must predict a sequence of multiple dependent operations. However, the same models remain weak on Single-Operation Completion. Under Seq. Mask, where the surrounding operation sequence is already provided and



(a) Under-/over-generation (b) Local operation ambiguity (c) Long-protocol difficulty

Figure 4: Error analysis across three sources of difficulty: (a) under- and over-generation, (b) local operation ambiguity, and (c) degraded performance on long procedures.

only one operation must be recovered, the best accuracy is still only **44.60%**. This setting removes most of the global planning burden, yet more than half of the masked operations remain incorrectly predicted.

Richer experimental context helps, but does not close the gap. Providing the complete structured experimental record consistently improves single-operation completion across all models. The gains range from **14.2** to **20.2** percentage points, showing that models can effectively exploit additional experimental context. Nevertheless, the best Full Mask accuracy reaches only **59.10%**. Thus, missing context alone cannot explain the poor performance.

Together, these results reveal a clear *experimental design gap*. Current LLMs struggle not only to generate complete procedures, but also to recover individual operations even when most of the surrounding experimental information is provided. This suggests that the limitation extends beyond long-horizon planning to insufficient fine-grained procedural knowledge.

5.3 Error Analysis

Figure 4 summarizes the major failure modes of current LLMs. We observe three consistent patterns.

Models exhibit distinct under- and over-generation behaviors. In complete protocol generation, Qwen3.8-27B and DeepSeek V4.1 Flash tend to produce shorter sequences than the references, doing so in 45.5% and 48.0% of test cases, respectively. In contrast, GPT-5.6 Sol, Claude Opus 5.5, and Kimi K3 overgenerate in 78.6%, 79.3%, and 76.5% of cases, often inserting plausible but unsupported operations such as Stir, Cool, Quench, or Dissolve. This trade-off is reflected in the main metrics: Claude achieves the highest recall (66.64%) but lower precision (45.35%), whereas DeepSeek V4.1 Flash obtains the highest precision (49.85%) with only 45.00% recall. Exact sequence recovery is extremely rare: among 1,000 test examples, only Claude reproduces one complete reference sequence.

Local operation prediction remains highly ambiguous. Single-operation completion reveals systematic confusion between related laboratory actions. Models frequently replace Add with Stir, Dissolve, or Mix, and confuse workup operations such as Concentrate, Purify, Wash, Filter, and Dry. For example, Qwen predicts Stir instead of Add in 99 aligned cases and fails to recover any of the 226 masked Add operations under Seq. Mask. Rare but experimentally important operations, including Purge, Dilute, Distill, Separate, and Seal, also show consistently low recall. Providing the full structured record helps substantially, but does not eliminate these errors: Mask accuracy increases from 18.90% to 33.10% for Qwen and from 44.60% to 59.10% for Claude.

Longer protocols further amplify the difficulty. Performance decreases as procedural complexity increases. For protocols with at least 13 reference operations, LCS drops to 28.9% for Qwen, 31.6% for DeepSeek, 42.4% for Kimi, 45.8% for GPT, and 48.3% for Claude. Thus, the experimental design gap arises from both local and global failures: models struggle to determine *which* operation is required and to organize those operations into a coherent multi-step workflow.

6 Learning Experimental Design

The preceding results reveal a substantial gap in the procedural knowledge of general-purpose LLMs. We next ask whether direct optimization on LabProtocol is sufficient to improve this capability.

6.1 FINE-TUNING ON LABPROTOCOL

We fine-tune Qwen3.8-27B on the LabProtocol training set using GRPO while keeping the underlying model architecture unchanged. The training objective directly optimizes experimental operation prediction under the same structured action vocabulary used in our evaluation.

As shown in Table 1, fine-tuning yields a modest improvement on Complete Protocol Generation. Operation-level F1 increases from 45.90 to 47.03, while LCS increases from 40.44 to 41.18. Precision and recall also improve slightly, from 47.87/44.09 to 49.26/44.98.

However, this improvement does not transfer to Single-Operation Completion. Seq. Mask accuracy changes from 18.90% to 18.70%, while Full Mask accuracy changes from 33.10% to 32.90%. Thus, the current fine-tuning setup improves the global protocol-generation objective only marginally and does not improve the model's ability to recover a locally missing experimental operation.

6.2 WHAT DOES FINE-TUNING IMPROVE?

The contrast between the two tasks provides an informative diagnostic. Complete Protocol Generation rewards learning common operation sequences and global procedural patterns, whereas Single-Operation Completion requires identifying the locally appropriate action given its surrounding context. The observed results suggest that the current training setup primarily adapts the model toward the distribution of complete protocols, without substantially improving its fine-grained procedural reasoning. This finding also indicates that experimental design is unlikely to be solved by straightforward domain fine-tuning alone. Learning reliable laboratory procedures may require objectives that explicitly model local action dependencies, rare operations, experimental states, and operation-specific context. We leave the development of such objectives to future work.

7 DISCUSSION

Experimental design is distinct from chemical knowledge. Our results show that strong general-purpose LLMs remain limited when translating a synthesis target into an experimental procedure. More importantly, the difficulty persists when only a single operation is masked and most of the surrounding procedure is provided. Together with the consistent improvement from Seq. Mask to Full Mask, these results suggest that experimental design depends on both access to relevant experimental context and fine-grained procedural knowledge that is not reliably captured by current models.

Protocol generation is inherently one-to-many. A target material may admit multiple valid synthesis procedures, while LabProtocol typically provides an experimentally reported reference rather than an exhaustive set of valid solutions. Operation-level F1 and LCS therefore measure agreement with reported procedures rather than absolute experimental correctness. Our benchmark should consequently be interpreted as a controlled measure of procedural knowledge, rather than a direct estimate of whether a generated synthesis would succeed physically.

8 CONCLUSION

We introduced **LabProtocol**, a dataset and benchmark for studying target-conditioned experimental protocol design. By aligning synthesis targets with structured laboratory operation sequences, LabProtocol enables systematic evaluation of both complete protocol generation and local operation completion. Our evaluation of leading LLMs reveals a clear gap between general scientific reasoning and experimental procedural knowledge. Models struggle not only to generate complete procedures, but also to recover individual missing operations even when substantial procedural context is available. Richer experimental context improves performance consistently, yet leaves a large gap unresolved. Initial fine-tuning experiments further show that improving complete protocol generation does not necessarily improve local procedural reasoning. These findings highlight experimental protocol design as a distinct and challenging capability for scientific AI, and establish LabProtocol as a foundation for developing models that can progress from predicting scientific targets toward planning how they can be realized experimentally.

AI USE STATEMENT

In this work, we used generative AI tools for polishing our paper and helping us with coding. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Anthropic. Claude Opus 5.5 system card. Anthropic, September 2026. URL https://www.anthropic.com/system-cards. Available through the official model system cards catalog.

Daniil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624:570–578, 2023. doi: 10.1038/s41586-023-06792-0. URL https://www.nature.com/articles/s41586-023-06792-0.

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*, 6:525–535, 2024. doi: 10.1038/s42256-024-00832-8. URL https://www.nature.com/articles/s42256-024-00832-8.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. arXiv preprint arXiv:2606.19348, 2026a. URL https://arxiv.org/abs/2606.19348.

DeepSeek-AI. DeepSeek-V4.1-Flash: Pushing the limits of KV cache compression. Technical report, 2026b. URL https://huggingface.co/deepseek-ai/DeepSeek-V4.1-Flash/blob/main/DeepSeek_V41_Tech_Report.pdf.

Samuel Genheden and Esben Bjerrum. PaRoutes: towards a framework for benchmarking retrosynthesis route predictions. *Digital Discovery*, 1:527–539, 2022. doi: 10.1039/D2DD00015F. URL https://pubs.rsc.org/en/content/articlehtml/2022/dd/d2dd00015f.

Steven M. Kearnes, Michael R. Maser, Michael Wleklinski, Anton Kast, Abigail G. Doyle, Spencer D. Dreher, Joel M. Hawkins, Klavs F. Jensen, and Connor W. Coley. The Open Reaction Database. *Journal of the American Chemical Society*, 143(45):18820–18826, 2021. doi: 10.1021/jacs.1c09820. URL https://doi.org/10.1021/jacs.1c09820.

Heegyu Kim, Taeyang Jeon, Seungtaek Choi, Ji Hoon Hong, Dong Won Jeon, Ga-Yeon Baek, Gyeong-Won Kwak, Dong-Hee Lee, Jisu Bae, Chihoon Lee, Yoon-Seo Kim, Seon-Jin Choi, Jin-Seong Park, Sung Beom Cho, and Hyun-souk Cho. Towards fully-automated materials discovery via large-scale synthesis dataset and expert-level LLM-as-a-judge. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 1302–1312. Association for Computing Machinery, 2025. doi: 10.1145/3746252.3761359. URL https://doi.org/10.1145/3746252.3761359.

Kimi Team. Kimi K3: Open frontier intelligence. arXiv preprint arXiv:2607.24653, 2026. URL https://arxiv.org/abs/2607.24653.

Olga Kononova, Haoyan Huo, Tanjin He, Ziqin Rong, Tiago Botari, Wenhao Sun, Vahe Tshitoyan, and Gerbrand Ceder. Text-mined dataset of inorganic materials synthesis recipes. *Scientific Data*, 6:203, 2019. doi: 10.1038/s41597-019-0224-1. URL https://www.nature.com/articles/s41597-019-0224-1.

Zhiyuan Liu, Yaorui Shi, An Zhang, Sihang Li, Enzhi Zhang, Xiang Wang, Kenji Kawaguchi, and Tat-Seng Chua. ReactXT: Understanding molecular "reaction-ship" via reaction-contextualized molecule-text pretraining. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 5353–5377. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.318. URL https://aclanthology.org/2024.findings-acl.318/.

Daniel Lowe. Chemical reactions from US patents (1976-sep2016). figshare, 2017. URL https://figshare.com/articles/dataset/Chemical_reactions_from_US_patents_1976-Sep2016_/5104873. Dataset.

S. Hessam M. Mehr, Matthew Craven, Artem I. Leonov, Graham Keenan, and Leroy Cronin. A universal system for digitization and automatic execution of the chemical synthesis literature. *Science*, 370(6512):101–108, 2020. doi: 10.1126/science.abc2986. URL https://doi.org/10.1126/science.abc2986.

Adrian Mirza, Nawaf Alampara, Sreekanth Kunchapu, Martiño Ríos-García, Benedict Emoekabu, Aswanth Krishnan, Tanya Gupta, Mara Schilling-Wilhelmi, Macjonathan Okereke, Anagha Aneesh, Mehrdad Asgari, Juliane Eberhardt, Amir Mohammad Elahi, Hani M. Elbeheiry, María Victoria Gil, Christina Glaubitz, Maximilian Greiner, Caroline T. Holick, Tim Hoffmann, Abdelrahman Ibrahim, Lea C. Klepsch, Yannik Köster, Fabian Alexander Kreth, Jakob Meyer, Santiago Miret, Jan Matthias Peschel, Michael Ringleb, Nicole C. Roesner, Johanna Schreiber, Ulrich S. Schubert, Leanne M. Stafast, A. D. Dinga Wonanke, Michael Pieler, Philippe Schwaller, and Kevin Maik Jablonka. A framework for evaluating the chemical knowledge and reasoning abilities of large language models against the expertise of chemists. *Nature Chemistry*, 17(7): 1027–1034, 2025. doi: 10.1038/s41557-025-01815-x. URL https://www.nature.com/articles/s41557-025-01815-x.

Odhran O'Donoghue, Aleksandar Shtedritski, John Ginger, Ralph Abboud, Ali Ghareeb, and Samuel Rodriques. BioPlanner: Automatic evaluation of LLMs on protocol planning in biology. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2676–2694. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.162. URL https://aclanthology.org/2023.emnlp-main.162/.

OpenAI. GPT-5.6 system card. OpenAI, July 2026. URL https://deploymentsafety.openai.com/gpt-5-6.

Qwen Team. Qwen3.8-27B. Hugging Face model card, 2026. URL https://huggingface.co/Qwen/Qwen3.8-27B. Accessed September 26, 2026.

Nadine Schneider, Daniel M. Lowe, Roger A. Sayle, and Gregory A. Landrum. Development of a novel fingerprint for chemical reactions and its application to large-scale reaction classification and similarity. *Journal of Chemical Information and Modeling*, 55(1):39–53, 2015. doi: 10.1021/ci5006614. URL https://doi.org/10.1021/ci5006614.

Philippe Schwaller, Teodoro Laino, Théophile Gaudin, Peter Bolgar, Christopher A. Hunter, Costas Bekas, and Alpha A. Lee. Molecular Transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS Central Science*, 5(9):1572–1583, 2019. doi: 10.1021/acscentsci.9b00576. URL https://doi.org/10.1021/acscentsci.9b00576.

Marwin H. S. Segler, Mike Preuss, and Mark P. Waller. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature*, 555:604–610, 2018. doi: 10.1038/nature25978. URL https://www.nature.com/articles/nature25978.

Alain C. Vaucher, Federico Zipoli, Joppe Geluykens, Vishnu H. Nair, Philippe Schwaller, and Teodoro Laino. Automated extraction of chemical synthesis actions from experimental procedures. *Nature Communications*, 11:3601, 2020. doi: 10.1038/s41467-020-17266-6. URL https://www.nature.com/articles/s41467-020-17266-6.

Alain C. Vaucher, Philippe Schwaller, Joppe Geluykens, Vishnu H. Nair, Anna Iuliano, and Teodoro Laino. Inferring experimental procedures from text-based representations of chemical reactions. *Nature Communications*, 12:2573, 2021. doi: 10.1038/s41467-021-22951-1. URL https://www.nature.com/articles/s41467-021-22951-1.

Zheren Wang, Kevin Cruse, Yuxing Fei, Ann Chia, Yan Zeng, Haoyan Huo, Tanjin He, Bowen Deng, Olga Kononova, and Gerbrand Ceder. ULSA: unified language of synthesis actions for the representation of inorganic synthesis protocols. *Digital Discovery*, 1:313–324, 2022a. doi: 10.1039/D1DD00034A. URL https://pubs.rsc.org/en/content/articlelanding/2022/dd/d1dd00034a.

Zheren Wang, Olga Kononova, Kevin Cruse, Tanjin He, Haoyan Huo, Yuxing Fei, Yan Zeng, Yingzhi Sun, Zijian Cai, Wenhao Sun, and Gerbrand Ceder. Dataset of solution-based inorganic materials synthesis procedures extracted from the scientific literature. *Scientific Data*,

Table 3: Overview of the major data sources used to construct LABPROTOCOL. "Action–Instrument" indicates whether laboratory operations are explicitly associated with the devices used to perform them.

Dataset	Scale	Actions	Instrument/Device	Action–Instrument	Parameters	Original Text	Role
OMG	17,667	Yes	Yes	Partial	Rich	Available / traceable	Core synthesis data
PcMSP	305 papers	Yes	Human annotated	Human annotated	Yes	Available	High-quality supervision
Solid-State	19,488	Yes	Partial	Partial	Rich	Snippet / DOI	Large-scale extension
Solution	35,675	Yes	Partial	Partial	Rich	Traceable	Large-scale extension
ULSA	3,040	Human annotated	Limited	Partial	Yes	Available	Action ontology supervision
ORD	Large-scale	Yes	Setup / vessel	Partial	Rich	Procedure available	Cross-domain extension
Scientific literature	–	Yes	Variable	Variable	Variable	Available	Additional curated records

9:231, 2022b. doi: 10.1038/s41597-022-01317-2. URL https://www.nature.com/articles/s41597-022-01317-2.

Xianjun Yang, Ya Zhuo, Julia Zuo, Xinlu Zhang, Stephen Wilson, and Linda Petzold. PcMSP: A dataset for scientific action graphs extraction from polycrystalline materials synthesis procedure text. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 6033–6046. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-emnlp.446. URL https://aclanthology.org/2022.findings-emnlp.446/.

Yu Zhang, Ruijie Yu, Jidong Tian, Feng Zhu, Jiapeng Liu, Xiaokang Yang, Yaohui Jin, and Yanyan Xu. ChemActor: Enhancing automated extraction of chemical synthesis actions with LLM-generated data. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24291–24314. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1183. URL https://aclanthology.org/2025.acl-long.1183/.

Yuxuan Zhang, Yue Fang, Haifan Zhou, Bowen Yu, Tsz Fung Fung, Qing Liu, Christophe Len, and Hanyu Gao. ProcedureT5: adaptive experimental procedure prediction with data-augmented pre-training and multi-source data integration. *Reaction Chemistry & Engineering*, 2026. doi: 10.1039/D5RE00572H. URL https://doi.org/10.1039/D5RE00572H.

A ADDITIONAL DATASET DETAILS

We provide additional details on the construction of LABPROTOCOL, including data sources, literature curation, schema unification, laboratory action and instrument normalization, manual verification, and dataset statistics.

A.1 DATA SOURCES

LABPROTOCOL is constructed by integrating multiple complementary sources of experimental synthesis data. These sources differ substantially in scale, domain coverage, annotation granularity, and the types of experimental information they provide. Table 3 summarizes the major sources used in our data collection.

The sources are complementary rather than interchangeable. OMG provides large-scale expert-verified synthesis recipes together with equipment and experimental parameters. PcMSP contains manually annotated operation–device relations and is therefore particularly useful for grounding laboratory operations to instruments. The Solid-State and Solution synthesis datasets provide broader coverage of synthesis procedures, while ULSA contributes manually annotated laboratory actions that help establish a consistent action vocabulary. ORD provides structured reaction conditions, vessel and setup information, and experimental procedures across a broader chemistry domain.

In addition to these existing resources, we curate experimental procedures directly from the scientific literature to increase coverage of procedures that are insufficiently represented in existing datasets.

A.2 LITERATURE CURATION

We additionally screen scientific papers and their supporting information for experimental synthesis procedures. We focus on records that describe a concrete synthesis or preparation process and for which the synthesis target can be reliably associated with the corresponding experimental procedure.

Candidate procedures are primarily collected from sections such as *Experimental*, *Methods*, *Synthesis*, *Preparation*, and *Supporting Information*. We retain descriptions that contain actionable laboratory operations, such as material addition, mixing, heating, separation, washing, drying, annealing, or related synthesis steps.

We exclude records that contain only reaction equations, characterization results, qualitative descriptions of synthesis, or references to an external procedure without sufficient information to reconstruct the corresponding experiment. We also exclude cases in which the synthesis target cannot be reliably aligned with the described experimental procedure.

When the main text and supporting information describe the same experiment, the more complete description is retained while provenance links to the corresponding source documents are preserved.

A.3 UNIFIED EXPERIMENTAL REPRESENTATION

The source datasets use different schemas and annotation conventions. We therefore convert all records into a shared representation centered on an *experimental synthesis instance*. Each instance connects a target reaction or material with the procedure used to obtain it.

The unified representation contains the following major components:

- **Target**: the reaction product or synthesized material;
- **Materials**: reactants, reagents, solvents, catalysts, additives, and intermediates;
- **Procedure**: the original experimental description and its ordered synthesis steps;
- **Actions**: canonical laboratory operations associated with each procedure step;
- **Instruments**: laboratory equipment used to perform individual operations whenever reported;
- **Parameters**: quantities, temperature, duration, speed, pressure, flow rate, and other operation-specific parameters;
- **Conditions**: environmental information such as atmosphere, vacuum, humidity, or illumination;
- **Provenance**: source dataset, document identifier, DOI or record identifier, and corresponding source text.

A simplified representation of one record is shown below:

```
{
  "target": ...,
  "materials": [...],
  "procedure_text": ...,
  "steps": [
    {
      "action": ...,
      "instrument": ...,
      "parameters": {...},
      "conditions": {...},
      "source_text": ...
    }
  ],
  "source": ...
}
```

The schema preserves both the normalized representation and the source-level description. This enables every structured field to be inspected against its original experimental evidence.

Table 4: Representative laboratory actions in LABPROTOCOL.

Category	Canonical Actions
Material handling	ADD, TRANSFER, DISPENSE, POUR
Mixing	MIX, STIR, SHAKE, SONICATE
Thermal processing	HEAT, COOL, DRY, ANNEAL, CALCINE, SINTER
Separation	FILTER, CENTRIFUGE, DECANT, EXTRACT
Purification	WASH, RINSE, RECRYSTALLIZE
Solid processing	GRIND, MILL, BALL_MILL, PRESS
Atmosphere control	PURGE, EVACUATE, DEGAS, PRESSURIZE
Film processing	SPIN_COAT, DROP_CAST, DIP_COAT, DEPOSIT
Other	WAIT, AGE, COLLECT, STORE

A.4 ACTION NORMALIZATION

Experimental procedures express equivalent laboratory operations using a wide variety of linguistic forms. For example, expressions such as "was added", "was introduced", and "was charged into the flask" can describe the same underlying laboratory operation.

We normalize such expressions into a shared laboratory action vocabulary. The vocabulary is constructed using the operation annotations available in PcMSP and ULSA, together with operations observed in OMG, the Solid-State and Solution datasets, ORD, and the additionally curated literature.

The resulting ontology covers the major classes of experimental operations encountered in synthesis procedures, including material handling, mixing, thermal processing, separation, purification, solid processing, and post-processing.

Representative action categories include:

We preserve the original operation phrase in addition to the canonical action label, allowing normalized actions to be traced back to the source procedure.

A.5 INSTRUMENT AND DEVICE NORMALIZATION

Instrument descriptions are similarly heterogeneous across datasets and scientific publications. The same equipment category may be referred to using generic names, manufacturer-specific terminology, or specific model numbers.

We therefore normalize instrument mentions into canonical device categories while retaining the original equipment description whenever available. Representative categories include balances, pipettes, syringes, liquid-handling systems, magnetic stirrers, hot plates, centrifuges, sonicators, ovens, furnaces, filtration systems, rotary evaporators, vacuum systems, gloveboxes, autoclaves, ball mills, and coating systems.

Instrument normalization is performed at the category level. Specific manufacturer and model information is retained when explicitly reported but is not required for a record to be included.

Importantly, an instrument is associated with an action only when the source provides sufficient evidence for the correspondence. We do not infer a specific device solely from common laboratory practice.

A.6 ACTION–INSTRUMENT ALIGNMENT

A central feature of LABPROTOCOL is the association between laboratory actions and the instruments used to perform them.

For datasets such as PcMSP, explicit operation–device annotations are directly mapped to the unified schema. For sources that provide a procedure and equipment information but do not explicitly encode the correspondence, the alignment is established from the local experimental context and subsequently verified during manual review.

For example,

> "The suspension was centrifuged at 8000 rpm for 10 min."

is represented as

```
action      = CENTRIFUGE
instrument  = CENTRIFUGE
speed       = 8000 rpm
duration    = 10 min
```

whereas

> "The mixture was stirred overnight."

is represented as

```
action      = STIR
instrument  = unspecified
duration    = overnight
```

unless a corresponding instrument is explicitly stated elsewhere in the procedure.

A.7 Experimental Parameters and Conditions

We normalize operation-specific parameters while preserving their original values and textual expressions.

Typical parameters include mass, volume, molar amount, concentration, equivalents, temperature, duration, rotational speed, relative centrifugal force, heating rate, cooling rate, pressure, flow rate, and number of repetitions.

Different actions expose different parameter sets. For example, a centrifugation operation may contain speed, relative centrifugal force, duration, and temperature, whereas an annealing operation may contain temperature, duration, heating rate, cooling rate, and atmosphere.

Environmental conditions are represented separately from action type and include information such as nitrogen or argon atmosphere, oxygen, vacuum, controlled humidity, and illumination conditions.

Units are normalized to a consistent representation for downstream use, while the original units are preserved for traceability.

A.8 Treatment of Missing Information

Published experimental procedures frequently omit details that may be important for exact reproduction. We explicitly preserve this missingness rather than automatically completing the procedure using common laboratory knowledge or model predictions.

For instance, when a procedure states

> "The product was collected by centrifugation."

we annotate the `CENTRIFUGE` action, but centrifugation speed and duration remain unspecified unless they are reported elsewhere in the source.

Similarly, expressions such as "room temperature", "overnight", or "dropwise" are preserved as semantic experimental conditions rather than replaced by arbitrary numerical values.

This design prevents inferred experimental knowledge from being mixed with source-grounded annotations and allows the dataset to support explicit analysis of procedural completeness.

A.9 MANUAL VERIFICATION

All records in the final release of LABPROTOCOL undergo manual review against their original experimental descriptions.

During verification, annotators inspect the correspondence between the synthesis target and the associated procedure and verify the extracted sequence of laboratory operations. They additionally check the associated materials, instruments, parameters, and experimental conditions against the source text.

Incorrectly extracted fields are corrected when the source provides sufficient evidence. Records are removed when the reaction or material cannot be reliably associated with the procedure, when the procedure is too incomplete to identify the experimental process, or when unresolved ambiguity prevents reliable annotation.

Manual verification also checks that structured fields do not introduce information that is absent from the source. In particular, numerical parameters, instrument identities, and environmental conditions are never added solely on the basis of assumed laboratory practice.

A.10 HUMAN REVIEW PROTOCOL

The review interface presents annotators with both the normalized representation and the corresponding original procedure text. Annotators are asked to verify five aspects of each record:

1. whether the synthesis target is correctly identified;
2. whether the experimental procedure corresponds to the target;
3. whether the operation sequence is complete and correctly ordered;
4. whether actions are associated with the correct instruments and parameters; and
5. whether all structured information is supported by the source.

Records that fail any of these checks are corrected or removed before entering the final dataset.

For quality auditing, a subset of records is independently reviewed by multiple annotators. Disagreements are resolved by inspecting the original experimental description.

A.11 SOURCE EVIDENCE AND PROVENANCE

We retain source provenance throughout the data construction pipeline. Whenever available, each record contains the original procedure text, source dataset, document identifier, DOI or equivalent identifier, and the textual span supporting the corresponding structured information.

This design allows the normalized representation to be audited against the original experimental evidence and enables future corrections as the schema or action ontology evolves.

For sources whose redistribution policies restrict the release of raw text, we retain only permissible metadata and references to the original source in the public version of the dataset.

A.12 DEDUPLICATION

Because the same synthesis procedure may appear in multiple resources, we perform deduplication before constructing the final dataset.

Potential duplicates are identified using combinations of source identifiers, target materials or products, chemical entities, and procedure similarity. When multiple records correspond to the same experimental instance, we retain the most complete representation and merge complementary structured fields only when their provenance can be verified.

Near-duplicate procedures describing distinct synthesis conditions or different material variants are retained as separate experimental instances.

A.13 Quality Control

In addition to manual review, we perform automatic consistency checks on the normalized records.

These checks verify that action order is valid, numerical parameters have valid units, referenced materials and instruments exist in the same record, and structured fields can be traced to the corresponding experimental description.

We also check for unsupported numerical values and instrument assignments. Any structured parameter that cannot be traced to the source is flagged for manual inspection.

Records that remain inconsistent after review are excluded from the final release.

A.14 Dataset Statistics

After source integration, literature curation, normalization, deduplication, and manual verification, LabProtocol contains **18,729 curated experimental records**.

Each record represents a synthesis instance connecting a target reaction or material with its experimental procedure and the corresponding laboratory actions. Depending on the information available in the source, records may additionally contain instruments, operation-specific parameters, environmental conditions, and post-processing steps.

We additionally report the distributions of procedure length, action frequency, instrument frequency, experimental parameters, and missing information in the supplementary statistics.

A.15 Representative Record

We provide an illustrative example of the transformation from an original experimental description to the unified LabProtocol representation.

Consider the following experimental description:

> "The mixture was stirred at 80 °C for 12 h, centrifuged at 8000 rpm for 10 min, washed three times with ethanol, and dried at 120 °C."

The corresponding normalized sequence is:

```
1. STIR
   temperature = 80 degC
   duration    = 12 h

2. CENTRIFUGE
   speed       = 8000 rpm
   duration    = 10 min

3. WASH
   solvent     = ethanol
   cycles      = 3

4. DRY
   temperature = 120 degC
```

The representation preserves both the ordered experimental actions and their associated parameters while retaining the original text as evidence.

B Experimental Details

B.1 Models

We evaluate six representative proprietary and open-weight language models: **Qwen3.8-27B**, **GPT-5.6 Sol**, **Claude Opus 5.5**, **DeepSeek V4.1 Flash**, **DeepSeek V4 Pro**, and **Kimi K3**. All models are

evaluated on exactly the same test instances and under the same task specification. We use a shared operation vocabulary and enforce a constrained JSON output format to minimize differences caused by free-form generation.

Unless otherwise stated, each test instance is evaluated once. We do not use model-specific demonstrations or manually adapt prompts for individual models.

B.2 Operation Vocabulary

All tasks use the same vocabulary of 42 normalized laboratory operations:

```
ADD, ACIDIFY, MIX, HOLD, STIR, HEAT, REFLUX,
HYDROGENATE, COOL, QUENCH, PURGE, EVACUATE_PURGE,
TRANSFER, DISSOLVE, DILUTE, FILTER, WASH, DRY,
SLURRY, TRITURATE, ADSORB, PURIFY, DISTILL,
NEUTRALIZE, CONCENTRATE, CRYSTALLIZE, EXTRACT,
SEPARATE, PARTITION, CENTRIFUGE, CALCINE,
CARBONIZE, CURE, COAT, DIALYZE, DISPERSE,
IRRADIATE, PRESS, SONICATE, ADJUST_PH, DEGAS, SEAL
```

Models are required to use the operation names exactly as defined in this vocabulary. Synonyms, additional operations, natural-language explanations, and operation arguments are not permitted. This restriction allows us to measure experimental procedural knowledge independently of lexical variation. The complete vocabulary used during evaluation is fixed across all models.

B.3 Complete Protocol Generation

For Complete Protocol Generation, the model receives only the identifiers of the target product and is asked to infer a plausible ordered experimental operation sequence, including workup when necessary. The number of operations is not fixed, and repeated operations are allowed.

The input has the form

```
{
  "products": [{"identifiers": [...]}]
}
```

and the model must return exactly one JSON object:

```
{
  "operations": ["ADD", "STIR", "FILTER"]
}
```

The example above specifies only the output format; it does not constrain the number or identity of predicted operations. Models are explicitly instructed not to output materials, conditions, explanations, step numbers, or any text outside the JSON object.

This setting is intentionally challenging: the model must recover the experimental procedure from the target product rather than being given the ground-truth reaction route or partial protocol.

B.4 Single-Operation Completion

For Single-Operation Completion, we randomly replace one operation in each ground-truth operation sequence with a `[MASK]` token. We use a fixed random seed of 42 for mask selection so that all models are evaluated on identical masked positions.

Given the target product and the partially observed operation sequence, the model predicts exactly one missing operation:

```
{
  "products": [{"identifiers": [...]}],
  "masked_operation_chain":
```

```
      ["ADD", "STIR", "HEAT", "[MASK]", "FILTER"]
}
```

The required output is

```
{
  "masked_operations": ["COOL"]
}
```

Only the missing operation is returned. The model is not allowed to regenerate the full sequence or predict multiple candidate operations.

Because both operations before and after the masked position are visible, this task provides bidirectional procedural context and substantially reduces the requirement for long-horizon sequence generation.

B.5 Context Settings for Operation Completion

We consider two context settings for Single-Operation Completion.

Sequence Mask (Seq. Mask). The model is provided with the product identifiers and the masked operation sequence only. This setting evaluates whether the missing operation can be recovered primarily from the surrounding procedural structure.

Full Mask. The same operation is masked, but the model is additionally provided with the complete structured experimental record available for that instance, including the associated experimental information. The masked target and evaluation instances are otherwise unchanged.

Comparing the two settings isolates the effect of richer experimental context. We report the absolute improvement

$$\Delta_{\text{context}} = \text{Acc}_{\text{FullMask}} - \text{Acc}_{\text{SeqMask}}.$$

Providing richer experimental context consistently improves operation completion across all evaluated models. The absolute gains range from 14.2 to 20.2 percentage points, indicating that models can exploit additional experimental information. However, even under Full Mask, the best accuracy remains 59.1%, suggesting that missing context alone does not explain the difficulty of experimental operation prediction.

B.6 Evaluation Metrics

Operation Precision, Recall, and F1. For Complete Protocol Generation, we compare the predicted operation sequence with the reference sequence at the operation level and report precision, recall, and F1. These metrics measure whether the model recovers the laboratory operations present in the reference protocol, independent of exact textual realization.

Longest Common Subsequence (LCS). Operation-level F1 does not fully capture procedural ordering. We therefore also compute the longest common subsequence between the predicted and reference operation sequences. LCS rewards predictions that recover both the correct operations and their relative order, and is reported as a normalized percentage score.

Mask Accuracy. For Single-Operation Completion, the prediction is correct only if the returned operation exactly matches the masked reference operation. We report the percentage of correctly recovered masked operations. Since the output space is restricted to the fixed 42-operation vocabulary, this metric directly measures operation-level completion accuracy.

Context Gain. For the comparison between Seq. Mask and Full Mask, we report the absolute difference in mask accuracy:

$$\text{Gain} = \text{FullMask} - \text{SeqMask}.$$

C Additional Dataset Statistics and Model Analysis

We provide additional statistics on LabProtocol and supplementary analyses of model behavior. These figures distinguish dataset-level statistics from evaluation results and include only quantities supported by the available data.

Dataset statistics. Figure 5 reports the complete distribution of 42 operation types across 138,773 action blocks from 18,729 records, revealing substantial variation in operation frequency. Figure 6 summarizes the availability and normalization of eight experimental parameter fields. Here, normalization coverage denotes the fraction of recorded parameter observations with normalized values, rather than the fraction of protocols containing a parameter. Figure 7 presents conditional probabilities computed from all 120,044 adjacent-operation pairs. The matrix captures recurring local ordering patterns, while the accompanying row counts indicate the support for each conditional distribution.

Model behavior. Figure 8 summarizes the reported directional length errors in complete protocol generation. The available results show frequent over-generation for GPT, Claude, and Kimi, alongside substantial under-generation rates for Qwen and DeepSeek V4.1 Flash. Figure 9 compares Seq. Mask and Full Mask accuracy across six models. Additional experimental context improves accuracy by 14.20–20.20 percentage points, although the highest Full Mask accuracy remains 59.10%. Figure 10 reports LCS scores for reference protocols containing at least 13 operations. Because results for the other length groups are unavailable, this figure characterizes the reported long-protocol subset without establishing a trend across protocol lengths.

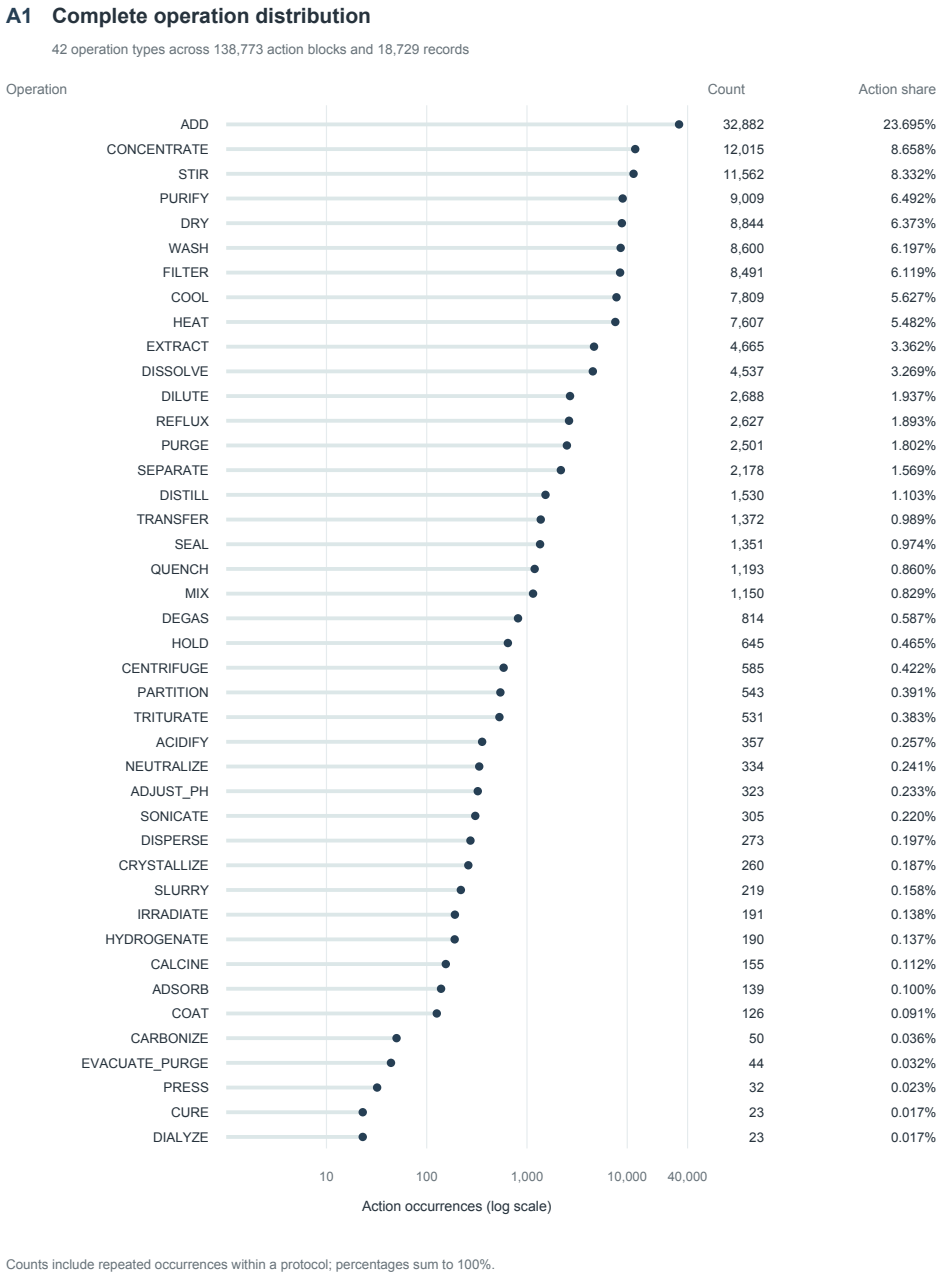


Figure 5: **Complete operation distribution in LabProtocol.** All 42 operation types are ranked by their occurrence counts across 138,773 action blocks from 18,729 records. Dots show counts on a logarithmic axis; the adjacent columns report exact counts and percentages of all action blocks. Repeated occurrences within a protocol are counted separately.

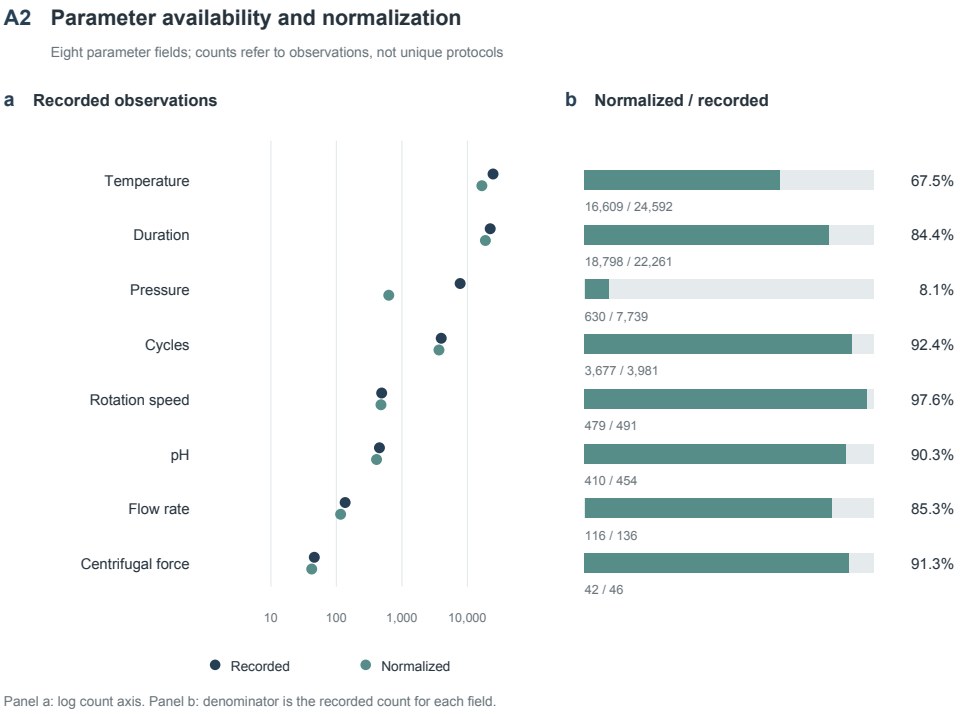


Figure 6: **Parameter availability and normalization.** (a) Recorded and normalized observation counts for eight parameter fields, shown on a logarithmic axis. (b) The fraction of recorded observations with normalized values; each numerator and denominator is shown. Counts refer to parameter observations rather than distinct protocols. Normalized counts include values flagged as outside the workbook's plausibility bounds, so normalization does not imply physical validity.

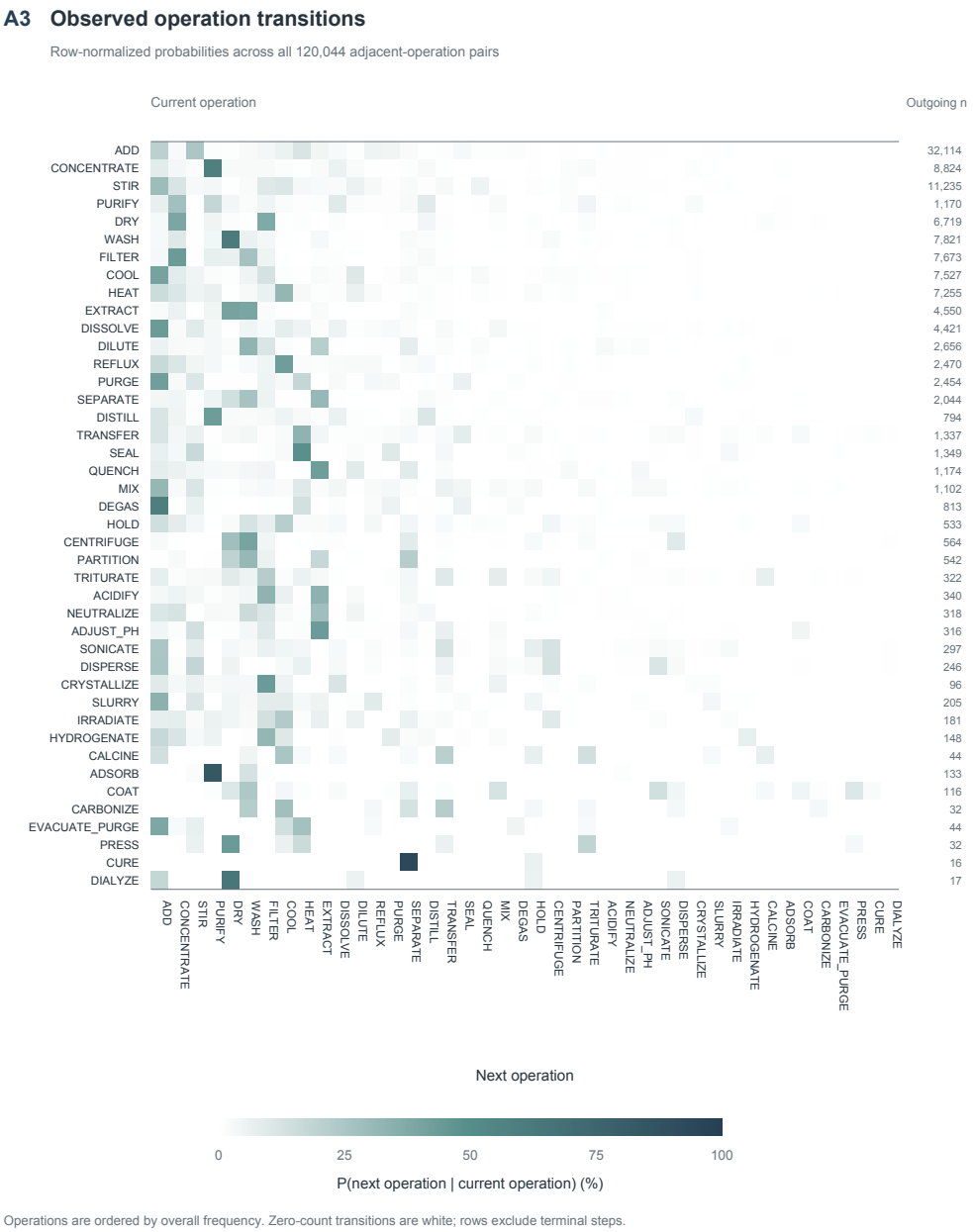


Figure 7: **Conditional operation-transition probabilities.** Each cell shows the percentage of outgoing transitions from the row operation that lead to the column operation, computed from 120,044 adjacent-operation pairs. All 42 operations are ordered by overall occurrence frequency. The right-hand column reports the outgoing-transition count supporting each row. Rows exclude terminal occurrences and are normalized over observed successors. White cells indicate zero observed transitions. Large probabilities for infrequent operations may be supported by few observations.

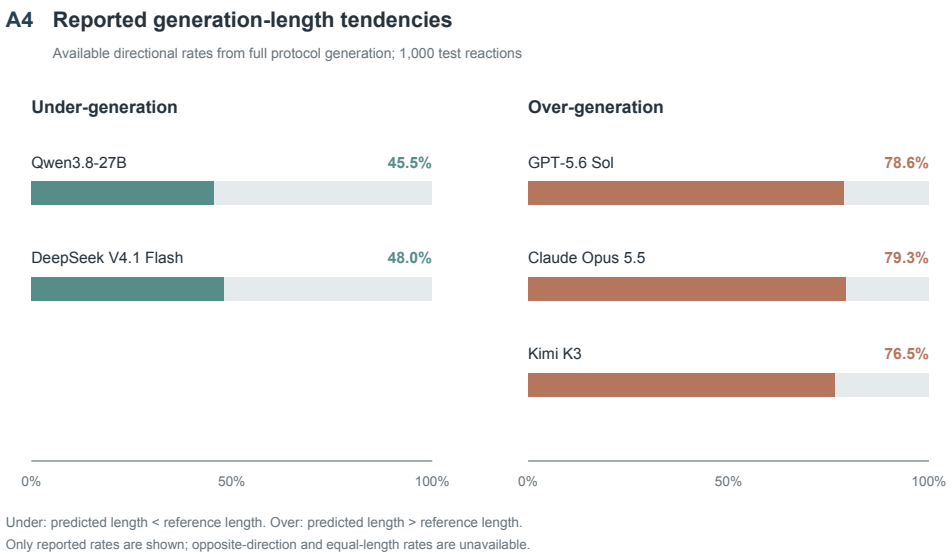


Figure 8: **Reported directional length errors.** Results are reported for complete protocol generation on 1,000 test reactions. The left panel shows the available fractions of predictions shorter than their references; the right panel shows the available fractions longer than their references. Only explicitly reported model–direction pairs are displayed. Opposite-direction and equal-length rates are unavailable, so these panels do not constitute complete distributions of prediction-length errors.

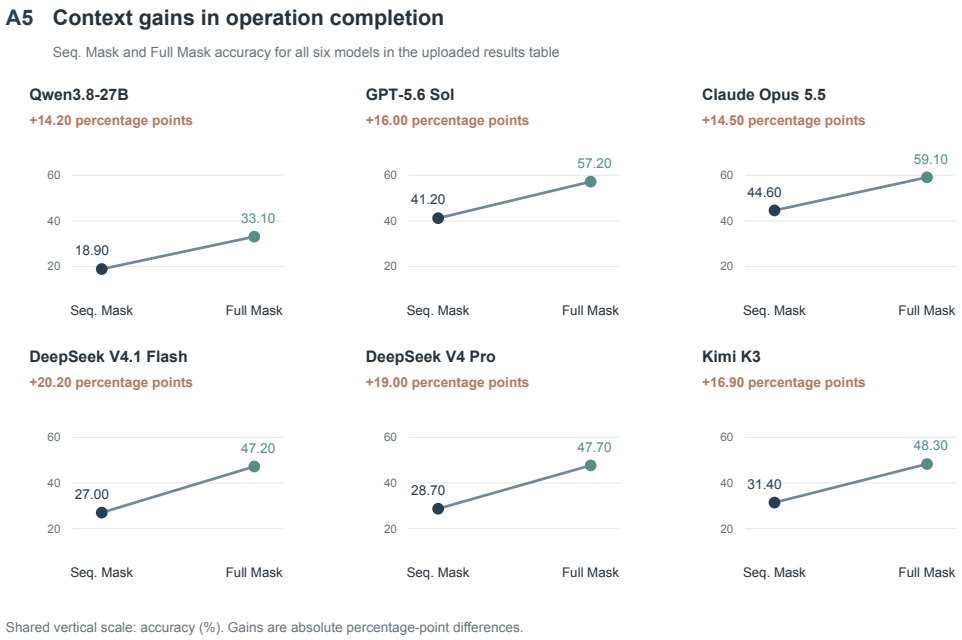


Figure 9: **Context gains in single-operation completion.** Seq. Mask and Full Mask accuracy are compared for six models using a shared vertical scale. Point labels report accuracy in percent; annotations above the slopes report absolute gains in percentage points. Full Mask improves accuracy by 14.20–20.20 percentage points. Values are taken from the supplied evaluation table; uncertainty intervals are unavailable.

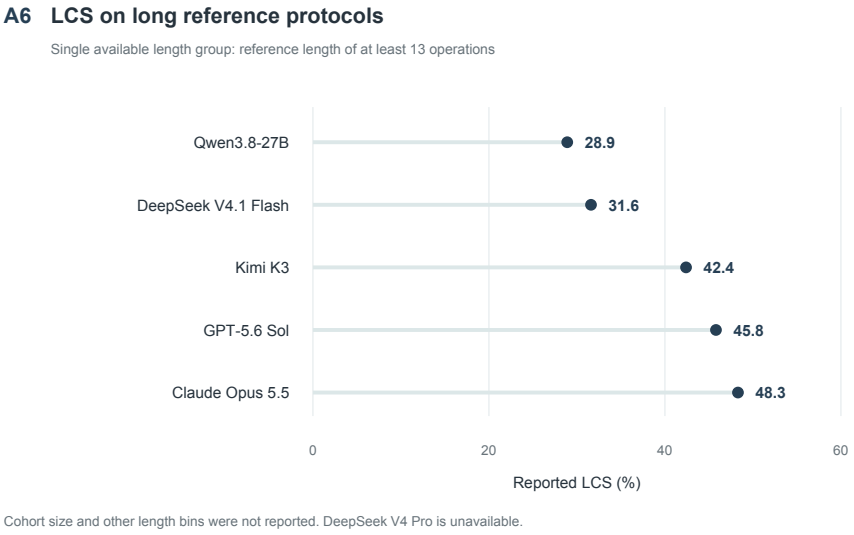


Figure 10: **LCS performance on long reference protocols.** Reported LCS scores are shown for protocols containing at least 13 operations and for the five models with available results. The subset size and scores for the 1–3, 4–6, 7–9, and 10–12 length groups were not supplied. This figure describes the available long-protocol subset and does not establish a performance trend across lengths.