ProtocoLLM: Automatic Evaluation Framework of LLMs on Domain-Specific Scientific Protocol Formulation Tasks

Anonymous EMNLP submission

Abstract

Automated generation of scientific protocols executable by robots can significantly accelerate scientific research processes. Large Language Models (LLMs) excel at Scientific Protocol Formulation Tasks (SPFT), but the evaluation of their capabilities rely on human evaluation. Here, we propose a flexible, automatic framework to evaluate LLMs' capability on SPFT: *ProtocoLLM*[1]. This framework prompts the target model and GPT-4 to extract pseudocode from biology protocols using only predefined lab actions and evaluates the output of target model using LLAM-EVAL, the pseudocode generated by GPT-4 serving as a baseline and Llama-3 acting as the evaluator. Our adaptable prompt-based evaluation method, LLAM-EVAL, offers significant flexibility in terms of evaluation model, material, criteria, and is free of cost. We evaluate GPT variations, Llama, Mixtral, Gemma, Cohere, and Gemini. Overall, we find that GPT and Cohere is a powerful scientific protocol formulators. We also introduce BIOPROT 2.0, a dataset with biology protocols and corresponding pseudocodes, which can aid LLMs in formulation and evaluation of SPFT. Our work is extensible to assess LLMs on SPFT across various domains and other fields that require protocol generation for specific goals.

1 Introduction

Laboratory automation is essential for accelerating scientific research processes. However, most contemporary laboratories use manual labor, especially in the field of biology. This not only constrains the scope for scalability, but also introduces potential vulnerabilities in reproducibility (Kwok, 2010).

One of the barriers for automation in biology is the reliance on manual experiments when validating scientific protocols. Traditionally, trial-and-error approach has been employed to formulate a protocol to achieve a certain goal. As a breakthrough, LLMs have demonstrated remarkable capabilities in formulating precise experimental protocols across diverse fields (White et al., 2023; Jablonka et al., 2023). These protocols comprise pseudocodes with actionable sequences that can be executed by machines which can be automated. Yet, efforts in biology to utilize LLMs for pseudocode formulation are yet to achieve desired outcomes (Inagaki et al., 2023). These works rely on human evaluations, and objective evaluation methods for protocol formulation are nonexistent. Therefore, it is necessary to establish an automated evaluation framework on formulating protocols to move beyond manual labor.

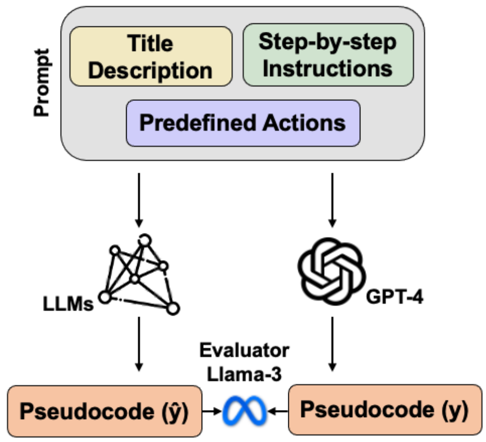

Figure 1: **Overview of the *ProtocoLLM* Framework.** A protocol containing a title, descriptions, step-by-step instructions, and predefined biology lab actions is given to both a target model and GPT-4 for pseudocode generation. Then, Llama-3 evaluates these outputs considering the target model's pseudocode as the prediction ($\hat{y}$) and GPT-4's as a baseline (y).

Previous work suggests a framework to assess the capabilities of LLMs on SPFT: *BioPlanner* (O'Donoghue et al., 2023). This method outlines three primary steps: (i) extracting pseudofunctions

[1]The dataset and code are available here.

and pseudocode[2] from a protocol using an evaluator, (ii) using the target model to produce pseudocode given the pseudofunctions, and (iii) evaluating the pseudocode generated in step (ii) against the original pseudocode in (i). Using this framework, they performed evaluation exclusively on GPTs (Brown et al., 2020; OpenAI, 2023).

We highlight the following key observations: (1) Various representations of pseudofunctions corresponding to identical experimental actions, causes performance degradation and inconsistency of the evaluation framework. (2) The repertoire of actions executed in biology labs is confined to a finite set of actions. (3) High values in traditional automatic metrics (i) does not necessarily imply human-perceived good quality in scientific protocols. (4) The use of automatic metrics (i) requires manual labor, which limits the transition to fully automatic evaluation.

Here, we propose an evaluation framework that evaluates the capabilities of LLMs in SPFT: *ProtocoLLM* (Figure 1). First, we define a set of actions in advance (Table 1), which eliminates individual action (pseudofunction) extraction step and variations of actions on each occasion. Second, we independently zero-shot prompted the target model and GPT-4 (OpenAI, 2023) to extract pseudocode from biology protocols, only using predefined actions as pseudofunctions. Lastly, we use LLAM-EVAL to evaluate the response, treating the target model's pseudocode as a prediction ($\hat{y}$) and that of GPT-4's as a baseline (y). LLAM-EVAL offers significant flexibility in terms of evaluation model, material, and criteria. This approach is inspired by the automated extraction of chemical synthesis actions from experimental procedures[3] (Vaucher et al., 2020). We compared multiple LLMs to our framework, including GPT variations (Brown et al., 2020; OpenAI, 2023), Llama, Mixtral, Gemma, Cohere, and Gemini (Google, 2024). We find that GPT-4o and Cohere+ is a powerful scientific protocol formulator.

We also introduce BIOPROT 2.0, a larger dataset with scientific protocols and the corresponding pseudocodes that can aid LLMs in formulation and evaluation of SPFT.

Overall, we make the following contributions:

1. We propose *ProtocoLLM*: a flexible, automatic framework for evaluating LLMs on SPFT using domain knowledge and LLAM-EVAL.
2. We propose LLAM-EVAL, an evaluation method that uses a form-filling paradigm offering significant flexibility in terms of evaluation model, material, and criteria.
3. We introduce the BIOPROT 2.0 dataset, featuring protocols and corresponding pseudocode for evaluating and aiding LLMs on SPFT.

2 Related Works

Task-specific Evaluation LLMs have been evaluated based on their performance in specific tasks. Information extraction abilities were measured by the generated quality of summaries (Durmus et al., 2020; Wang et al., 2020), paper reviews (Zhou et al., 2024), question correction (Fan et al., 2024), or combination of a few tasks (Labrak et al., 2024). However, these studies do not provide comprehensive evaluations and only assess very limited aspects, thus limiting their generalizability to other abilities or tasks.

LLM Evaluation on SPFT Recent work proposes a three-step framework (Section 1) for the evaluation of scientific protocols in biology: *BioPlanner* (O'Donoghue et al., 2023). This work evaluates GPT's performance in three tasks: next-step prediction, pseudocode generation, and pseudofunction retrieval. It employs statistical scoring methods including Levenshtein distance ($\mathcal{L}_d$) and BLEU (Papineni et al., 2002) to measure the relevance between a baseline and generated protocols, despite their modest correlation with human judgments.

Domain-specific LLMs in Science A Large number of LLMs have been trained, fine-tuned, or augmented for domain-specific uses. ChemBERTa/-2 (Chithrananda et al., 2020; Ahmad et al., 2022), MatSciBERT (Gupta et al., 2021), MaterialsBERT (Shetty et al., 2023), Chemcrow (Bran et al., 2023), and LLM augmentation methods for various experiment-related tasks (Guo et al., 2023) has been introduced in chemistry. BioGPT (Luo et al., 2022), BioBERT (Lee et al., 2019), CamemBERT-bio (Touchent et al., 2024), BlueBERT (Peng et al., 2019), PubmedBERT (Gu et al., 2020), BioMegatron (Shin et al., 2020), and

[2] Pseudofunctions represent laboratory actions, while pseudocode embodies protocols composed of these pseudofunctions.

[3] A set of actions in chemistry labs were defined prior to the pseudocode extraction process.

Action Name	Description
`Transfer`	Move substances between containers using lab equipment, such as pipettes.
`Centrifuge`	Spin at high speed to separate mixture components by density.
`Vortex`	Mix solutions by creating a vortex for even distribution.
`SetTemp`	Set specific temperatures for reactions or processes.
`Wait`	Period of inactivity to allow reactions or condition stabilization.
`Wash`	Rinse materials, often with solvents to remove contaminants.
`Measure`	Quantify substances or properties using instruments.
`Microscopy`	Use a microscope to observe and analyze cell morphology and structures.
`CellDetachment`	Release adherent cells from a culture surface using enzymatic or mechanical methods.
`CellCount`	Determine the number of cells in a sample using a hemocytometer or automated counter.
`InvalidAction`	Undefined action due to documentation error or ambiguity.
`OtherLanguage`	Text in non-English, indicating translation need.
`NoAction`	Text not corresponding to any defined action.
`PCR`	Amplify DNA segments through Polymerase Chain Reaction.
`Gel`	Separate molecules by size in a gel with electric field.
`Culture`	Grow cells in lab to study behavior or for experimentation.
`Dilute`	Reducing the concentration of a solution by adding solvent.

Table 1: **Predefined Set of Actions.** List of actions performed in biological experiments and the corresponding descriptions. Actions above the line represent the basic actions, with the last three specifically designated for instances where a new protocol introduces an undefined action. Actions below represent the coarse-grained actions.

ProtoCode (Jiang et al., 2024) has been introduced in biology.

Evaluating LLMs with LLMs Evaluation of LLMs encompasses a dual-method approach:

(i) Statistical scoring: BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), METEOR (Banerjee and Lavie, 2005), Levenshtein Distance

(ii) Model-based scoring: G-Eval (Liu et al., 2023), Prometheus (Kim et al., 2023), BLEURT (Sellam et al., 2020), Natural Language Inference (NLI)

(iii) Combination of (i) and (ii): GPTScore (Fu et al., 2023), SelfCheckGPT (Manakul et al., 2023), BERTScore (Zhang et al., 2020), SciBERTScore (O'Donoghue et al., 2023), WMD (Kusner et al., 2015), MoverScore (Zhao et al., 2019), Question Answer Generation (QAG) Score

In tasks where reasoning is involved, (ii)(iii) outperforms (i). Previous work adopted (i) with (iii) being minimal (O'Donoghue et al., 2023). In this work, we adopt the notion of G-Eval (Liu et al., 2023), a framework for evaluating LLM-generated text, which prompts GPT with text and criteria, then scores based on its output.

3 Methods

The *ProtocoLLM* framework can evaluate the capability of LLMs on SPFT in three steps (Figure 2): (1) prompt the target LLM to generate pseudocode based on the given protocol, (2) repeat previous step for GPT-4, and (3) LLAM-EVAL for evaluation. To utilize this framework, we curated protocols in biology (Section 3.1), predefined actions performed in biology labs (Section 3.2), prompted LLMs for pseudocode generation (Section 3.3), and prompted Llama-3 for evaluation (LLAM-EVAL) (Section 3.6).

3.1 Data Curation of Protocols in Biology

Each protocol is composed of three core elements: a title, description, and experimental steps. We curated the dataset through a process of collection and refinement. We collected a set of keywords relevant to biology. Then, we used a scoring system based on the number of keywords included in the description of each protocol from *protocols.io*[4] (Teytelman et al., 2016). We refined the dataset collected in the previous step using automated and manual methods. (Appendix A.1.)

3.2 Defining Actions

The defined actions are composed of two parts: (i) **basic actions** corresponding to a single action which can be performed directly in biology labs, and (ii) **coarse-grained actions** which corresponds to a large set of basic actions repeated throughout various protocols. Defined actions were reviewed by experts with intensive experiences

[4]A platform for reproducible protocol sharing provides access to more than 15k publicly available protocols, and has no limitations regarding the use of LLMs.

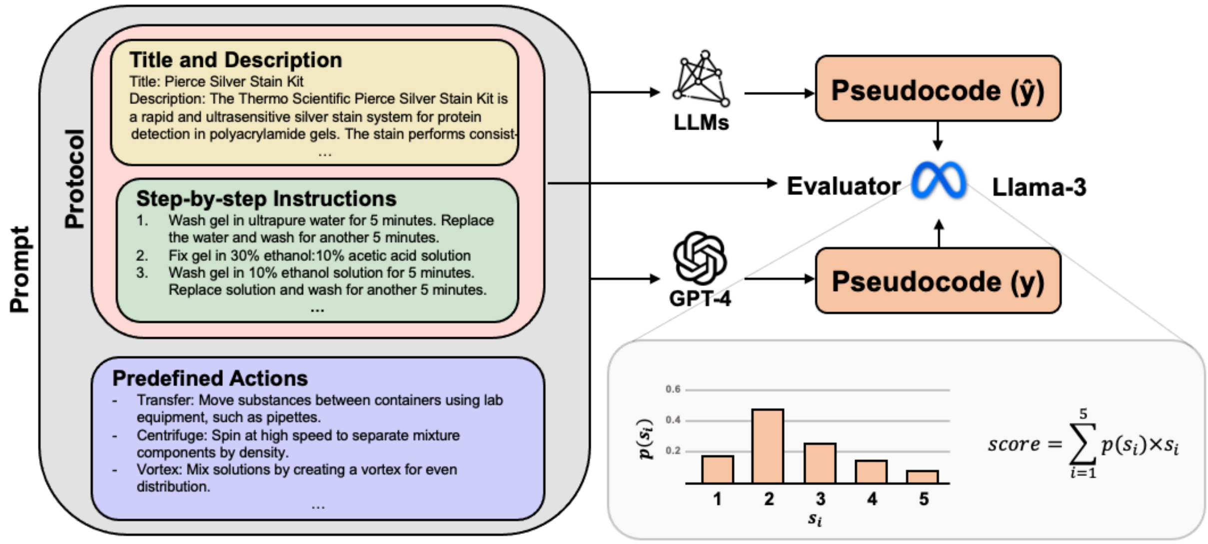

Figure 2: **The *ProtocoLLM* Framework.**

in biology experiments. The target model specifies the arguments for each action on each occasion.

Basic Actions Since the repertoire of actions executed in biology labs is confined to a finite set of actions, we defined a set of actions performed in biology labs prior to the extraction of pseudocode from protocols (Table 1). We performed a comprehensive literature review to define the set of basic actions performed in biology labs.

Coarse-grained Actions We observed that a series of complex, repetitive actions can be effectively encapsulated and described by a single, comprehensive action. For instance, the process of diluting a solution is conceptually straightforward and can possibly defined by basic actions. However, this involves intricate calculations and logical reasoning, which can result in performance degradation by calculation mistakes and posing variations in representations of an identical process. To this end, we coarse-grained these complex set of actions into a singular action.

3.3 Prompting Pseudocode Generation

To evaluate the target LLMs on SPFT, we prompted the models to generate pseudocode based on a protocol collected at Section 3.1. Models are instructed to use only the actions defined in Section 3.2 as the function name. However, they were allowed to define the arguments for each pseudofunction as needed for each occasion. If applicable, the fixed prompt, including the instructions and predefined actions, was provided in the system message, while the protocol was included in the user message. In this work, we prompted GPT-3 (Brown et al., 2020), GPT-4 (OpenAI, 2023), Gemini (Google, 2024), Claude3 (Anthropic, 2023), and Cohere. Below is the prompt for generating pseudocode based on the given protocol. Note that actions and corresponding descriptions presented in Table 1 are placed at *{actions}*.

You are an AI that generates Python pseudocode for biology protocols. This pseudocode must accurately describe a complete scientific protocol to obtain a result. You will be provided with the title, description, and steps of the biology protocol, and your task is to convert it to Python pseudocode.

You may define the arguments on your own. You must ONLY use these functions.

{actions}

Do NOT provide any captions. ONLY present the pseudocode and pseudofunctions used inside the code. Present the pseudofunctions at the beginning and then the pseudocode. Do NOT provide any descriptions inside the code.

title: {title}

description: {description}

steps: {steps}

3.4 Metrics and Evaluation

We observe that using automatic metrics (i) necessitates manual annotation of functions and pseudocodes each time, which significantly hampers the automation of the evaluation process. Moreover, evaluating the function and input[5] separately falls short of flexible and comprehensive evaluation in a protocol manner.

To this end, we propose LLAM-EVAL, an automatic, flexible prompt-based framework to evaluate the quality of LLM responses. This framework requires three elements: two input texts (one serving as the baseline and the other as the target) and an evaluator LLM: Llama-3[6]. This method encompasses predefining a set of scores[7] $S = \{s_1, s_2, ..., s_n\}$, prompting Llama-3 to rate the outcomes of a target LLM with that of GPT-4 in the scale of S, calculating the probability of each score $p(s_i)$, and calculating the final score as following. This method is inspired by G-Eval (Liu et al., 2023).

$$\text{score} = \sum_{i=1}^{n} s_i p(s_i)$$

Llama-3 is prompted to evaluate according to one criterion at a time. The original prompts targeting summarizing tasks are modified to perform evaluation on SPFT. In this work, we evaluate the pseudocode generated by the target LLM based on six criteria: the four original criteria used in G-Eval (Liu et al., 2023) (`Coherence`, `Consistency`, `Fluency`, and `Relevance`) and two criteria we propose (`Precision`, and `Coverage`), considering the context of SPFT. For example, the definition of `Coherence` is:

Coherence (1-5) - the overall quality of all lines in the pseudocode. The target pseudocode should not be a rough overview but should provide a precise description of a baseline pseudocode.

The definitions of other criteria in prompts can be found at Appendix A.2. To automatically implement chain-of-thoughts (CoT) in the evaluation process, we instructed GPT-4 to create specific evaluation steps for each criterion. GPT is capable of producing these evaluation steps by itself (Liu et al., 2023). GPT-4 was given a task and evaluation criteria, then prompted to generate the evaluation steps using a form-filling paradigm. An example prompt containing GPT-4 generated instructions for evaluation can be found at Appendix A.2. We also implemented an automatic feedback loop to regenerate the response up to five or ten times if the output did not contain scores. We evaluated using two baselines: the GPT-generated pseudocode and the original protocol.

This approach is not constrained by the output structure of the target models, eliminates the need for manual annotation efforts during the parsing process as required in reference-based metrics, enables a comprehensive evaluation, and thereby makes *ProtocoLLM* significantly more flexible and automatic.

To ensure compatibility, we also use conventional reference-based metrics: Normalized Levenshtein distance ($\mathcal{L}_{dn}$) for function names, BLEU (Papineni et al., 2002), precision, recall, and SciBERTScore (O'Donoghue et al., 2023) for function inputs. SciBERTScore is calculated using the encoded **pred**icted $\mathcal{E}(a_i^{\text{pred}})$ and **b**ase**l**ine values $\mathcal{E}(a_i^{\text{BL}})$ using the SciBERT (Beltagy et al., 2019) sentence encoder $\mathcal{E}$.

$$\text{SciBERTScore} = \frac{1}{N}\sum_{i=0}^{N} \frac{\langle \mathcal{E}(a_i^{\text{pred}}), \mathcal{E}(a_i^{\text{BL}})\rangle}{\|\mathcal{E}(a_i^{\text{pred}})\|\|\mathcal{E}(a_i^{\text{BL}})\|}$$

3.5 Evaluator LLM Selection

To select a specific LLM as an evaluator, we propose *self-self comparison task* as a baseline, where an LLM generates a pseudocode[8] for a protocol and then evaluates the score using the same LLM against the generated pseudocode. For example, this means evaluating GPT-4 generated pseudocode against the same pseudocode using GPT-4. Our assumption was that the score should be close to the maximum[9] when the baseline and target pseudocode are the same. Our goal was to select the model with the best results as the evaluator. We evaluated each model based on six criteria in Section 3.4. More details in Appendix A.3.

3.6 Evaluating LLMs using LLAM-EVAL

Using LLAM-EVAL, we evaluate across three tasks for each model: (1) GPT-4 generated pseudocode as a baseline with predefined actions given in

[5] Input refers to the function parameters and arguments.
[6] Llama3-70b
[7] s_1=1 and s_n=5 is set in this work.

[8] Pseudocode with pseudofunctions defined at the beginning to be precise.
[9] maximum score $s_n = 5$ in this work

prompt, (2) the same task with no predefined actions, (3) the original protocol as a baseline with predefined actions. We evaluate GPT variations (Brown et al., 2020; OpenAI, 2023), Llama, Mixtral, Gemma, Cohere, and Gemini (Google, 2024). Details are in Appendix A.4.

3.7 Implementation Details

To ensure a fair evaluation of LLMs, we considered additional factors that may affect performance and present several settings. We consider that LLMs tend to perform better when the actions are presented in the same order as in the protocol. While previous work extracted different actions from each protocol[10], we predefined the actions which is equivalent to shuffling. Also, while using LLAM-EVAL, we encountered instances where the output was a sentence instead of a score (number). To address this issue, we modified the parameters, dataset, and prompts. Further details are in Appendix A.5.

4 Analysis

4.1 Evaluator LLM Selection

Llama-3 achieved the highest scores across all six tasks, while there were small differences across models (Table 2). We chose Llama-3 as an evaluator, which is free of cost to date. Note that evaluations for other models not presented in the table were not feasible, as numerical responses were not generated. More details are in Appendix A.3.

4.2 Evaluating LLMs on SPFT

Our results show that GPT-4o and Cohere+ is a powerful protocol formulator (Table 3). We found our work compatible to previous work (O'Donoghue et al., 2023).

Is applying domain knowledge an effective strategy for evaluation? We applied domain knowledge by predefining the finite set of actions performed in biology labs. To evaluate the efficacy of this method, we compare the responses generated with predefined actions included in the prompts to those generated without them (Table 4). The performance is enhanced for most models, with the exception of the *Recall*. Further research should be conducted to explore these findings.

[10]In previous work, this required shuffling, as LLMs presented the pseudofunctions in the same order as in the protocol.

Can the original protocol itself serve as a baseline? Evaluation of LLMs in SPFT in previous work requires manual processes and pseudocode extraction step in SPFT. However, evaluation using the original protocol itself completely eliminates the manual processes of pseudofunction evaluation and the GPT-generated pseudocode extraction step, thereby enhancing flexibility and automation. To this end, we evaluate using the original protocol as a baseline. While scores obtained using this approach is not close to the maximum score (Table 3), we observe that the relative ranking of the models remains relevant to the results of using the pseudocode as a baseline.

Will LLM as an evaluator prefer responses from itself? It is reported that LLM as an evaluator prefer responses from itself over human responses in text summarization tasks (Liu et al., 2023). Therefore, a potential concern is that the evaluator may prefer outputs from itself regardless of its quality. While results in Table 2 and 4 address this concern, Table 3 shows that Llama-3 as an evaluator does not prefer its outputs over that of GPT-4. Our results suggest that GPT's preference for its own responses in previous work (Liu et al., 2023) may be a phenomenon unique to GPT.

4.3 The BIOPROT 2.0 Dataset

We introduce BIOPROT 2.0, a dataset with scientific protocols and the corresponding pseudocodes with a larger number of datapoints. Previous work highlights that a dataset with these two components can aid protocol formulation of LLMs (O'Donoghue et al., 2023). The pseudocode extracted from protocols are only composed of pseudofunctions (actions) predefined above the previous step, as each model was prompted to use only the provided functions but to define the arguments on their own. The summary of generated pseudocode are in Table 5. This dataset can be used to formulate scientific protocols to achieve a prompted goal using a toolformer like (Schick et al., 2023) chain-of-thought LLM agent (Wei et al., 2023).

5 Conclusion

We introduce *ProtocoLLM*, a flexible and automatic framework designed to evaluate LLMs' capabilities on Scientific Protocol Formulation Tasks (SPFT).

Models	Original Criteria				New Criteria	
	Coherence	Consistency	Fluency	Relevance	Precision	Coverage
GPT-4o	4.95 ± 0.26	4.98 ± 0.25	4.95 ± 0.27	4.93 ± 0.44	4.97 ± 0.23	4.95 ± 0.08
GPT-4	4.98 ± 0.23	4.99 ± 0.19	4.99 ± 0.19	4.99 ± 0.14	4.99 ± 0.18	4.99 ± 0.17
GPT-3.5	4.96 ± 0.23	4.97 ± 0.21	4.77 ± 0.52	4.96 ± 0.25	4.95 ± 0.30	4.99 ± 0.12
Llama-3	$\mathbf{5.00} \pm 0.02$	$\mathbf{5.00} \pm 0.00$	$\mathbf{5.00} \pm 0.00$	$\mathbf{5.00} \pm 0.00$	$\mathbf{5.00} \pm 0.00$	$\mathbf{5.00} \pm 0.06$

Table 2: ***Self-Self Comparison Task* Results**: We report the mean and standard deviation of scores over ten runs. Values in bold indicate the highest scores for each criterion. Higher values for all metrics represent better performance. Note that a larger dataset was used for this task. Details in Appendix A.3.

Models	Prompt		Original Criteria				New Criteria		
	Ac	Pr	Coherence	Consistency	Fluency	Relevance	Precision	Coverage	Average
GPT-4o	✓	✗	$\mathbf{4.10} \pm 0.79$	$\mathbf{3.80} \pm 0.85$	$\mathbf{3.86} \pm 0.67$	$\mathbf{4.32} \pm 0.71$	$\mathbf{4.02} \pm 0.65$	$\mathbf{4.26} \pm 0.73$	**4.06**
	✗	✗	$\mathbf{4.28} \pm 0.50$	$\mathbf{3.94} \pm 0.64$	$\mathbf{4.04} \pm 0.37$	$\mathbf{4.45} \pm 0.54$	$\mathbf{4.18} \pm 0.41$	$\mathbf{4.39} \pm 0.50$	**4.21**
	✓	✓	$\mathbf{4.29} \pm 0.57$	$\underline{4.73} \pm 0.50$	4.42 ± 0.53	$\mathbf{4.75} \pm 0.48$	$\mathbf{3.90} \pm 0.48$	$\mathbf{4.67} \pm 0.56$	$\underline{4.46}$
GPT-4 (Baseline)	✓	✗	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.08	5.00 ± 0.00	4.99 ± 0.11	5.00 ± 0.00	5.00
	✗	✗	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.04	5.00 ± 0.03	5.00 ± 0.03	5.00 ± 0.00	5.00
	✓	✓	4.32 ± 0.53	4.70 ± 0.58	4.53 ± 0.51	4.75 ± 0.44	3.99 ± 0.29	4.67 ± 0.48	4.49
GPT-3.5	✓	✗	3.61 ± 0.97	3.51 ± 1.02	3.58 ± 0.85	4.11 ± 0.78	3.82 ± 0.73	3.90 ± 0.83	3.75
	✗	✗	3.83 ± 0.82	3.71 ± 0.81	3.76 ± 0.68	4.19 ± 0.64	3.96 ± 0.57	3.97 ± 0.71	3.90
	✓	✓	4.13 ± 0.65	$\mathbf{4.76} \pm 0.49$	$\underline{4.48} \pm 0.52$	$\underline{4.69} \pm 0.49$	3.79 ± 0.58	$\underline{4.49} \pm 0.67$	4.39
Llama3-8b	✓	✗	2.25 ± 1.00	1.93 ± 0.99	2.27 ± 0.83	2.39 ± 1.08	2.61 ± 0.96	2.56 ± 1.09	2.33
	✗	✗	2.90 ± 0.89	2.69 ± 0.92	3.02 ± 0.88	3.41 ± 0.81	3.47 ± 0.70	3.19 ± 0.82	3.12
	✓	✓	2.80 ± 1.02	3.00 ± 1.26	3.10 ± 1.00	3.39 ± 1.09	2.93 ± 0.92	3.27 ± 1.06	3.08
Llama3-70b	✓	✗	3.61 ± 0.94	3.14 ± 1.10	3.53 ± 0.82	3.73 ± 0.97	3.72 ± 0.70	3.77 ± 0.79	3.58
	✗	✗	$\underline{3.98} \pm 0.64$	$\underline{3.72} \pm 0.75$	3.92 ± 0.49	$\underline{4.20} \pm 0.57$	$\underline{4.03} \pm 0.36$	$\underline{4.09} \pm 0.53$	$\underline{3.99}$
	✓	✓	4.02 ± 0.75	4.17 ± 0.98	4.15 ± 0.66	4.37 ± 0.74	3.78 ± 0.59	4.25 ± 0.69	4.12
Mixtral	✓	✗	3.41 ± 1.03	2.90 ± 1.13	3.57 ± 0.83	3.36 ± 1.14	3.68 ± 0.77	3.54 ± 0.93	3.41
	✗	✗	3.95 ± 0.68	3.68 ± 0.79	3.94 ± 0.53	4.18 ± 0.66	4.05 ± 0.43	4.00 ± 0.61	3.97
	✓	✓	4.06 ± 0.69	4.32 ± 0.84	4.28 ± 0.59	4.37 ± 0.71	3.88 ± 0.44	4.31 ± 0.70	4.21
Gemma-7b	✓	✗	3.06 ± 0.97	2.81 ± 1.03	3.47 ± 0.86	3.52 ± 0.93	3.55 ± 0.78	3.19 ± 0.89	3.27
	✗	✗	2.93 ± 0.85	2.66 ± 0.88	3.63 ± 0.76	3.61 ± 0.71	3.66 ± 0.61	3.06 ± 0.80	3.26
	✓	✓	3.81 ± 0.75	4.13 ± 0.83	4.25 ± 0.61	4.26 ± 0.75	3.76 ± 0.60	3.94 ± 0.79	4.02
Cohere+	✓	✗	$\underline{3.95} \pm 0.74$	$\underline{3.63} \pm 0.87$	$\underline{3.87} \pm 0.60$	$\underline{4.11} \pm 0.74$	$\underline{3.98} \pm 0.50$	$\underline{4.07} \pm 0.63$	$\underline{3.94}$
	✗	✗	3.97 ± 0.60	3.71 ± 0.73	$\underline{3.95} \pm 0.46$	4.15 ± 0.56	$\underline{4.03} \pm 0.38$	4.04 ± 0.50	3.98
	✓	✓	4.44 ± 0.52	4.63 ± 0.61	$\mathbf{4.53} \pm 0.52$	4.73 ± 0.47	4.04 ± 0.30	4.66 ± 0.49	**4.50**
Cohere	✓	✗	3.51 ± 0.91	3.06 ± 1.02	3.56 ± 0.74	3.66 ± 0.87	3.71 ± 0.63	3.70 ± 0.76	3.53
	✗	✗	3.71 ± 0.68	3.44 ± 0.83	3.83 ± 0.56	4.05 ± 0.53	3.94 ± 0.41	3.84 ± 0.56	3.80
	✓	✓	$\underline{3.98} \pm 0.63$	4.11 ± 0.87	4.14 ± 0.51	4.29 ± 0.63	3.83 ± 0.48	4.24 ± 0.64	4.10
Gemini-1.0	✓	✗	2.77 ± 1.09	2.30 ± 1.08	2.90 ± 0.95	2.80 ± 1.10	3.13 ± 0.92	3.15 ± 1.01	2.84
	✗	✗	3.46 ± 0.93	3.22 ± 1.01	3.59 ± 0.83	3.89 ± 0.77	3.80 ± 0.69	3.66 ± 0.79	3.60
	✓	✓	3.37 ± 0.93	3.68 ± 1.11	3.73 ± 0.80	3.87 ± 0.87	3.42 ± 0.78	3.86 ± 0.84	3.66
Gemini-2.0	✓	✗	3.09 ± 1.05	2.53 ± 1.10	3.75 ± 0.70	2.98 ± 1.08	3.63 ± 0.73	3.43 ± 0.89	3.24
	✗	✗	3.88 ± 0.82	3.61 ± 0.91	4.11 ± 0.60	4.13 ± 0.73	4.14 ± 0.54	3.93 ± 0.73	3.97
	✓	✓	3.80 ± 0.80	3.95 ± 0.97	4.30 ± 0.58	4.18 ± 0.72	3.80 ± 0.49	4.11 ± 0.68	4.02
Gemini-1.5	✓	✗	3.02 ± 1.05	2.48 ± 1.02	3.10 ± 0.93	2.97 ± 1.07	3.32 ± 0.84	3.42 ± 0.93	3.05
	✗	✗	4.12 ± 0.66	3.86 ± 0.72	4.03 ± 0.55	4.33 ± 0.62	4.13 ± 0.50	4.21 ± 0.59	4.11
	✓	✓	3.34 ± 0.95	3.62 ± 1.04	3.76 ± 0.76	3.81 ± 0.86	3.36 ± 0.77	3.80 ± 0.84	3.61

Table 3: ***ProtocoLLM* Evaluation Results** of three tasks for each model: (1) GPT-4 generated pseudocode as a baseline with predefined actions given in prompt, (2) the same task with no predefined actions, (3) the original protocol as a baseline with predefined actions. '**Ac**' and '**Pr**' represent whether the predefined **ac**tions and the original **pr**otocol were given for evaluation, respectively. We report the mean, standard deviation, and average of scores over five runs. The best and second best performance besides a baseline (GPT-4) for each criterion and task is bolded and underlined, respectively. The scores range from a minimum of 1 to a maximum of 5. Higher values for all metrics represent better performance.

Models	Actions	Precision	Recall	SciBERT	BLEU	$\mathcal{L}_{dn}$
GPT-4o	✓	0.581 ± 0.390	0.548 ± 0.414	0.783 ± 0.111	0.102 ± 0.189	$\underline{0.216} \pm 0.110$
	✗	0.600 ± 0.375	0.620 ± 0.373	$\mathbf{0.778} \pm 0.103$	0.118 ± 0.188	0.214 ± 0.106
GPT-4 (baseline)	✓	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.905 ± 0.198	0.055 ± 0.129
	✗	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.911 ± 0.173	0.021 ± 0.043
GPT-3.5	✓	0.817 ± 0.308	0.425 ± 0.404	0.766 ± 0.115	0.102 ± 0.205	$\mathbf{0.205} \pm 0.117$
	✗	0.732 ± 0.357	0.572 ± 0.378	0.742 ± 0.099	0.099 ± 0.178	$\mathbf{0.200} \pm 0.106$
Llama3-8b	✓	0.763 ± 0.323	0.708 ± 0.411	0.801 ± 0.128	0.135 ± 0.329	0.413 ± 0.351
	✗	0.759 ± 0.322	0.570 ± 0.352	0.744 ± 0.100	0.075 ± 0.174	0.242 ± 0.133
Llama3-70b	✓	0.825 ± 0.319	$\mathbf{0.917} \pm 0.220$	$\mathbf{0.883} \pm 0.136$	$\mathbf{0.563} \pm 0.464$	0.287 ± 0.203
	✗	0.812 ± 0.268	$\mathbf{0.769} \pm 0.260$	0.772 ± 0.097	$\mathbf{0.161} \pm 0.210$	0.206 ± 0.095
Mixtral	✓	0.855 ± 0.280	0.605 ± 0.393	0.784 ± 0.120	0.135 ± 0.288	0.603 ± 0.366
	✗	0.754 ± 0.291	$\underline{0.735} \pm 0.290$	0.771 ± 0.093	0.130 ± 0.215	0.499 ± 0.261
Gemma-7b	✓	$\underline{0.911} \pm 0.249$	0.641 ± 0.406	0.838 ± 0.139	0.205 ± 0.342	0.243 ± 0.130
	✗	$\mathbf{0.849} \pm 0.261$	0.651 ± 0.337	$\underline{0.775} \pm 0.116$	0.092 ± 0.180	0.221 ± 0.096
Cohere+	✓	0.646 ± 0.373	0.548 ± 0.352	0.767 ± 0.110	0.075 ± 0.172	0.363 ± 0.300
	✗	0.600 ± 0.366	0.604 ± 0.368	0.744 ± 0.100	0.095 ± 0.153	0.325 ± 0.265
Cohere	✓	0.645 ± 0.361	0.551 ± 0.380	0.717 ± 0.097	0.077 ± 0.193	0.360 ± 0.247
	✗	$\underline{0.767} \pm 0.295$	0.630 ± 0.314	0.750 ± 0.099	0.091 ± 0.165	$\underline{0.204} \pm 0.105$
Gemini-1.0	✓	0.852 ± 0.319	0.867 ± 0.313	$\underline{0.875} \pm 0.133$	$\underline{0.444} \pm 0.497$	0.410 ± 0.699
	✗	0.758 ± 0.319	0.584 ± 0.360	0.765 ± 0.111	0.112 ± 0.211	0.247 ± 0.182
Gemini-2.0	✓	$\mathbf{0.942} \pm 0.147$	0.878 ± 0.288	0.843 ± 0.165	0.342 ± 0.415	0.381 ± 0.254
	✗	0.736 ± 0.350	0.651 ± 0.339	0.758 ± 0.104	0.128 ± 0.197	0.308 ± 0.268
Gemini-1.5	✓	0.889 ± 0.258	$\underline{0.896} \pm 0.202$	0.814 ± 0.163	0.355 ± 0.461	0.371 ± 0.217
	✗	0.628 ± 0.377	0.682 ± 0.367	0.773 ± 0.101	$\underline{0.135} \pm 0.205$	0.214 ± 0.116

Table 4: **Evaluation Results Using Reference-Based Metrics.** Comparison with and without predefined actions given in prompts. We report mean and standard deviation of scores over five runs. The best and second best performance for each criterion is bolded and underlined, respectively. Except for $\mathcal{L}_{dn}$, higher values for all metrics represent better performance.

Statistic	Value ($m \pm \sigma$)
# of protocols	300
Tokens / protocol	812.3 ± 469.9
# of steps	14.81 ± 10.74
Tokens / step	54.28 ± 42.41
Tokens / description	139.0 ± 135.7
Tokens / generated pseudocode	623.8 ± 223.2
# of lines / generated pseudocode	83.06 ± 28.89
# of pseudofunctions / edited pseudocode	10.28 ± 6.582

Table 5: **Statistics of BIOPROT 2.0.** 'Edited Pseudocode' refers to the pseudocode that was reformatted, while preserving its content, to obtain the scores presented in Table 4.

This framework prompts the target model and GPT-4 to extract pseudocode from biology protocols using only predefined lab actions, then evaluates the target model's output using LLAM-EVAL, with the GPT-4 generated pseudocode as a baseline and Llama-3 as the evaluator. Our prompt-based evaluation method, LLAM-EVAL, provides significant flexibility in terms of evaluation models, materials, criteria, and is free of cost. We assess various models, including GPT variants, Llama, Mixtral, Gemma, Cohere, and Gemini, and find GPT and Cohere to be particularly effective in formulating scientific protocols. Additionally, we present BIOPROT 2.0, a dataset containing biology protocols and corresponding pseudocodes, which supports LLMs in the formulation and evaluation of SPFT. Our work is extensible to the assessment of LLMs on SPFT across various domains and other fields that require protocol generation for specific goals.

6 Limitations

We recognize several limitations. The predefined actions may not encompass all actions performed in a biology labs. The definitions of predefined actions may be incomplete. To precisely define an action, it is necessary to define not only the function but also the function arguments. The number of protocols in BIOPROT 2.0 may be insufficient for evaluation purposes. The performance of *ProtocoLLM* may decline outside of biology. Addressing this requires redefining domain-specific actions and exploring other LLMs for diverse fields. Future work should investigate these cross-disciplinary implications. LLMs are continuously evolving due to regular updates. The LLMs used for evaluation in this work might become unavailable in the future. Upgraded versions of LLMs may result in performance degradation and metrics may differ from those obtained using previous models. Due to selecting Llama-3 as the evaluator, our results may be susceptible to its biases and hallucinations. The outcomes when evaluated with models other than Llama-3 are unknown. Future work should investigate the outcomes using different LLMs as an evaluator. Using an API of LLMs as an evaluator, such GPT, is often not free of charge and can be costly. We used GPT-4 generated responses as a baseline; however, it may not accurately represent the ground truth. Future work should explore the implications of employing alternative resources (e.g., manually annotated pseudocodes, responses generated by other models) as the baseline. We observed basic actions classified as `NoAction` in minor cases. It has been reported that GPT prefers outputs from LLMs, which also produced our evaluation materials including all ground truth and target pseudocodes. This can potentially influence the scores. The four criteria mentioned in G-Eval may not sufficiently fulfill the role of evaluating protocols where real-world validation is crucial. Also, applying these criteria originally designed for summarization tasks may be inappropriate for evaluating SPFT. Even if the protocol pseudocode is successfully synthesized, real-world experiments may fail depending on the person performing the protocol or the condition of the physical equipment, especially in cases that are more complex than stem cell culture or require delicate manual work and experience.

Ethical Considerations

The use of manually verified protocols in LLMs is strictly prohibited for generating false protocols on platforms like STAR Protocols (Cell Press) and Nature Protocols. Numerous sites also prohibit the use of these protocols in conjunction with any form of AI tool. Our framework can be applied to the protocols of these sites. Although we have endeavored to exclude protocols that can create dangerous substances, there remains the potential for generating protocols that inadvertently produce hazardous products or byproducts.

Acknowledgements

We would like to thank Karim Md.Adnan for assisting us with the action defining process.

References

Walid Ahmad, Elana Simon, Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. 2022. Chemberta-2: Towards chemical foundation models.

Anthropic. 2023. The claude 3 model family: Opus, sonnet, haiku. Technical report, Anthropic. Accessed: 2024-05-16.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

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

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. 2020. Chemberta: Large-scale self-supervised pretraining for molecular property prediction.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Yuchen Fan, Yantao Liu, Zijun Yao, Jifan Yu, Lei Hou, and Juanzi Li. 2024. Evaluating generative language models in information extraction as subjective question correction.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire.

Gemini Team Google. 2024. Gemini: A family of highly capable multimodal models.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-specific language model pretraining for biomedical natural language processing.

Taicheng Guo, Kehan Guo, Bozhao Nan, Zhenwen Liang, Zhichun Guo, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2023. What can large language models do in chemistry? a comprehensive benchmark on eight tasks.

Tanishq Gupta, Mohd Zaki, N. M. Anoop Krishnan, and Mausam. 2021. Matscibert: A materials domain language model for text mining and information extraction.

T Inagaki, Akari Kato, Koichi Takahashi, Haruka Ozaki, and Genki N. Kanda. 2023. Llms can generate robotic scripts from goal-oriented instructions in biological laboratory automation.

Kevin Maik Jablonka, Qianxiang Ai, Alexander Al-Feghali, Shruti Badhwar, Joshua D. Bocarsly, Andres M. Bran, Stefan Bringuier, L. Catherine Brinson, Kamal Choudhary, Defne Circi, Sam Cox, Wibe A. de Jong, Matthew L. Evans, Nicolas Gastellu, Jerome Genzling, María Victoria Gil, Ankur K. Gupta, Zhi Hong, Alishba Imran, Sabine Kruschwitz, Anne Labarre, Jakub Lála, Tao Liu, Steven Ma, Sauradeep Majumdar, Garrett W. Merz, Nicolas Moitessier, Elias Moubarak, Beatriz Mouriño, Brenden Pelkie, Michael Pieler, Mayk Caldas Ramos, Bojana Ranković, Samuel G. Rodriques, Jacob N. Sanders, Philippe Schwaller, Marcus Schwarting, Jiale Shi, Berend Smit, Ben E. Smith, Joren Van Herck, Christoph Völker, Logan Ward, Sean Warren, Benjamin Weiser, Sylvester Zhang, Xiaoqi Zhang, Ghezal Ahmad Zia, Aristana Scourtas, K. J. Schmidt, Ian Foster, Andrew D. White, and Ben Blaiszik. 2023. 14 examples of how llms can transform materials science and chemistry: a reflection on a large language model hackathon. *Digital Discovery*, 2:1233–1250.

Shuo Jiang, Daniel Evans-Yamamoto, Dennis Bersenev, Sucheendra K. Palaniappan, and Ayako Yachie-Kinoshita. 2024. Protocode: Leveraging large language models (llms) for automated generation of machine-readable pcr protocols from scientific publications. *SLAS Technology*, 29:100134.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2023. Prometheus: Inducing fine-grained evaluation capability in language models.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 957–966, Lille, France. PMLR.

R. Kwok. 2010. Five hard truths for synthetic biology. *Nature*, 463:288–290.

Yanis Labrak, Mickael Rouvier, and Richard Dufour. 2024. A zero-shot and few-shot study of instruction-finetuned large language models applied to clinical and biomedical tasks.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6).

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models.

Odhran O'Donoghue, Aleksandar Shtedritski, John Ginger, Ralph Abboud, Ali Ghareeb, and Samuel Rodriques. 2023. BioPlanner: Automatic evaluation of LLMs on protocol planning in biology. In *Proceedings of the 2023 Conference on Empirical Methods* *in Natural Language Processing*, pages 2676–2694, Singapore. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets. In *Proceedings of the 2019 Workshop on Biomedical Natural Language Processing (BioNLP 2019)*, pages 58–65.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. Bleurt: Learning robust metrics for text generation.

P. Shetty, A.C. Rajan, C. Kuenneth, et al. 2023. A general-purpose material property data extraction pipeline from large polymer corpora using natural language processing. *npj Computational Materials*, 9:52.

Hoo-Chang Shin, Yang Zhang, Evelina Bakhturina, Raul Puri, Mostofa Patwary, Mohammad Shoeybi, and Raghav Mani. 2020. Biomegatron: Larger biomedical domain language model.

Leonid Teytelman, Alexei Stoliartchouk, Lori Kindler, and Bonnie L. Hurwitz. 2016. Protocols.io: Virtual communities for protocol development and discussion. *PLOS Biology*, 14(8):1–6.

Rian Touchent, Laurent Romary, and Eric de la Clergerie. 2024. Camembert-bio: Leveraging continual pre-training for cost-effective models on french biomedical data.

Alain C. Vaucher, Federico Zipoli, Jonas Geluykens, et al. 2020. Automated extraction of chemical synthesis actions from experimental procedures. *Nature Communications*, 11:3601.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Andrew D. White, Glen M. Hocky, Heta A. Gandhi, Mehrad Ansari, Sam Cox, Geemi P. Wellawatte, Subarna Sasmal, Ziyue Yang, Kangxin Liu, Yuvraj Singh, and Willmor J. Peña Ccoa. 2023. Assessment of chemistry knowledge in large language models that generate code. *Digital Discovery*, 2:368–376.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance.

Ruiyang Zhou, Lu Chen, and Kai Yu. 2024. Is LLM a reliable reviewer? a comprehensive evaluation of LLM on automatic paper reviewing tasks. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9340–9351, Torino, Italia. ELRA and ICCL.

Appendix

A BIOPROT 2.0

A.1 Data Curation

We used *protocols.io* (Teytelman et al., 2016) API for data collection. Protocols of $1 \leq score \leq 5$ and $3 \leq steps$ are collected. The collected data was in a *.json* format, every data point with slight differences in keys. Some protocols were present in the git repository but could not be found when retrieved using the API, and vice versa[11]. Also, even if the file ID in the git repository and the protocol ID retrieved using the API are the same, the dictionary key *number_of_steps* may differ[12]. Keywords[13] extracted from the *keywords.txt* file and the descriptions were converted to lowercase temporarily for comparison and scoring. As of May 2024, we collected a total of approximately 15k mirrored public protocols from *protocols.io*'s GitHub before refinement. Protocols were excluded if dictionary key *steps* is empty. Protocols were manually verified by experts in biology. The protocols were removed if they were multiple duplicated files for an identical protocol[14]. For the same title, we score the latest version of the protocol.

A.2 Metrics and Evaluation

Definitions of Evaluation Criteria

- `Consistency`: Consistency (1-5) - the factual alignment between the source and the target pseudocode. A factually consistent pseudocode contains only statements that are entailed by the source pseudocode. Annotators were also asked to penalize summaries that contained hallucinated facts.
- `Fluency`: Fluency (1-5): the quality of the pseudocode in terms of grammar, spelling, punctuation, word choice, and structure.
- `Relevance`: Relevance (1-5) - selection of important information from the source pseudocode. The target pseudocode should include only important information from the source document. Annotators were instructed to penalize summaries which contained redundancies and excess information.
- `Precision`: Precision (1-5) - the exactness and accuracy of the expressions and terminology used in the pseudocode. The target pseudocode should avoid vague or ambiguous terms and should use specific and appropriate terminology that accurately reflects the intended operations and logic.
- `Coverage`: Coverage (1-5) - the extent to which the target pseudocode addresses all aspects of the source pseudocode. The target pseudocode should comprehensively represent all the necessary steps, operations, and details present in the source pseudocode without omitting any critical information.

Note that above are criteria used for evaluation when GPT-generated pseudocode was a baseline. This was slightly modified when evaluating based on original protocol.

Example LLAM-EVAL Prompt Below is a prompt evaluating the generated pseudocode from a target LLM based on the criteria `Coherence` using the GPT-generated pseudocode as the ground truth. The GPT-generated pseudocode for each protocol is placed inside *{{Ground_truth_pseudocode}}*, and the target model-generated pseudocode is placed inside *{{Target_pseudocode}}*.

You will be given a source pseudocode as a ground truth. You will then be given a target pseudocode which is generated from an identical source of protocol.

Your task is to rate the target pseudocode on one metric. Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria: Coherence (1-5) - the overall quality of all lines in the pseudocode. The target

[11] The protocol with ID 3737 exists in *protocol.io* but doesn't exist in git repository.

[12] The *number_of_steps* for the protocol with ID 10489 is 3 in the git repository but 0 when retrieved using the API.

[13] The keywords are: Biology, Cell, DNA, Protein, Stem Cell, Molecular Biology, Molecular, Gene, Virus, E. coli, cDNA, Agarose, Agarose Gel, in vitro, PCR, NGS, Ethanol, Illumina, Cell Theory, Evolution, Genetics, Homeostasis, Cell Membrane, Mitochondria, Nucleus, Ribosomes, DNA Replication, Mutation, Chromosomes, Gene Expression, Natural Selection, Speciation, Adaptation, Phylogenetics, Ecosystems, Biodiversity, Conservation, Bacteria, Viruses, Fungi, Pathogens, Proteins, Enzymes, Metabolism, Photosynthesis, Gel Electrophoresis, Cloning, CRISPR-Cas9, Neurons, Brain, Synapses, Neurotransmitters, Antibodies, Vaccines, Immune Response, Autoimmunity, Embryogenesis, Stem Cells, Morphogenesis, Regeneration, Pollination, Growth Hormones, Tropisms, Coral Reefs, Oceanic Zones, Marine Conservation, Aquatic Ecosystems, Endangered Species, Habitat Destruction, Conservation Strategies, Rewilding, Genetic Engineering, Bioreactors, Bioinformatics, and Synthetic Biology.

[14] such as protocol ID: 9216

pseudocode should not be a rough overview but should provide a precise description of the ground truth pseudocode.

Evaluation Steps:

1. Read the Ground Truth Pseudocode: Carefully read and understand the source pseudocode provided as the ground truth. Ensure you comprehend the logic, flow, and details of the algorithm or protocol described.

2. Read the Target Pseudocode: Thoroughly read the target pseudocode that needs to be evaluated. Pay attention to the details, structure, and clarity of the pseudocode.

3. Compare Against Ground Truth: Compare each line and section of the target pseudocode with the corresponding parts of the ground truth pseudocode. Ensure that all critical steps, variables, and logic present in the ground truth are accurately reflected in the target pseudocode.

4. Assess Coherence: Evaluate the overall quality of the target pseudocode based on how well it translates the ground truth. Consider the following aspects: Clarity: Is the pseudocode easy to understand? Completeness: Does it cover all the steps and details present in the ground truth? Precision: Are the descriptions and instructions in the pseudocode precise and unambiguous? Consistency: Are there any contradictions or logical inconsistencies?

5. Assign a Coherence Rating (1-5):

1 (Poor): The target pseudocode is incomplete, confusing, and lacks most details from the ground truth. 2 (Fair): The target pseudocode is partially complete but has significant gaps and is often unclear. 3 (Good): The target pseudocode covers most details from the ground truth but has some minor inconsistencies or lacks clarity in parts. 4 (Very Good): The target pseudocode is mostly complete and clear, with very few minor issues. 5 (Excellent): The target pseudocode is complete, clear, precise, and fully coherent with the ground truth.

Source Pseudocode:

{{Ground_truth_pseudocode}}

Target Pseudocode:

{{Target_pseudocode}}

Evaluation Form (scores ONLY):

- Coherence:

A.3 Evaluator LLM Selection

Models without numerical responses include: Llama3-8b, Llama3-70b, Mixtral, and Gemma.

A.4 Evaluating LLMs on SPFT

Versions of LLMs

Model Name	Call Strings
GPT-4o	`gpt-4o`
GPT-4	`gpt-4`
GPT-3.5	`gpt-3.5-turbo-1106`
Llama3-8b	`llama3-8b-8192`
Llama3-70b	`llama3-70b-8192`
Mixtral	`mixtral-8x7b-32768`
Gemma-7b	`gemma-7b-it`
Cohere+	`command-r-plus`
Cohere	`command-r`
Gemini-1.0	`gemini-1.0-pro-001`
Gemini-1.5	`gemini-1.5-pro-001`
Gemini-2.0	`gemini-1.0-pro-002`

Table 6: **Versions of LLMs.** Exact API call strings for corresponding models.

A.5 Implementation Details

Except for `n` and `seed`, parameters were set to their default values. We used approximately $1000 for GPT API calls, $20 for Gemini, and other models were free of cost.

Counting Tokens We counted the tokens of the concatenated string of the title, original description, and steps, separated by "\n\n". The reason for this approach is to match the token count with that of the previous work.

Inconsistencies LLAM-EVAL Outputs To address this issue, we attempted the following methods: (1) Modified `max_token` = 5 to `max_token` = 1 : The scores became integers, but the model still generated sentences in addition to scores. (2) Use different versions of the model: Other model variations, such as `gpt-3.5-turbo-1106`, did not enhance the results.