A Data-Centric Approach To Generate Faithful and High Quality Patient Summaries with Large Language Models - Reproduction

Raymond Macharia

University of Illinois Urbana-Champaign
rmm15@illinois.edu

Abstract

This project reproduces the paper A Data-Centric Approach To Generate Faithful and High Quality Patient Summaries with Large Language Models. The original paper investigates the potential of large language models (LLMS) to generate patient summaries based on doctors' notes and studies the effect of training data on the faithfulness and quality of the summaries generated. In reproducing the paper, we generate summaries using 4 LLM configurations. We then recreated the quantitative evaluation of the generated summaries and found our results to be close to those arrived at by the original paper. Finally, we extend the original paper's automated hallucination detection to using GPT-5.

Code —
https://github.com/machira/patient_summaries_with_llms

Datasets —
https://physionet.org/content/ann-pt-summ/1.0.1

Video — https://youtu.be/g0rQQegfUS0

Introduction

Many patients find it difficult to understand their hospitalization and discharge instructions, and anecdotal evidence suggests that some now turn to commercial LLMs for clarification (Astor). The fidelity of such model-generated medical summaries is therefore of growing academic and clinical importance. The paper by Hegselmann et al. studies whether LLMs can produce faithful, patient-facing discharge summaries from clinical notes and makes several contributions to the broader clinical NLP landscape. The authors (1) introduce a token-level annotation protocol for identifying hallucinations in medical summaries, (2) release two expert-labeled datasets of doctor-written and model-generated summaries, (3) demonstrate that fine-tuning on manually cleaned data can reduce hallucinations without sacrificing essential information, and (4) evaluate GPT-4 as an automatic hallucination detector. Together, these contributions provide an empirical foundation for measuring and improving factual consistency in LLM-generated clinical text.

Scope of Reproducibility

Our reproduction targets the paper's core modeling and evaluation components. We successfully reproduced the preprocessing pipeline, the summarization experiments using LED and Llama 2 7B, and the full scoring system (ROUGE, BERTScore, DeBERT, and SARI) used in the original study (Hegselmann et al. 2024a). We also replicated the summarization baselines across LED, Llama 2 7B, and GPT-4. Due to resource limitations, we did not reproduce the Llama 2 70B experiments, nor the human annotation pipeline or class-aware hallucination analysis. Within these constraints, we reproduced the primary quantitative findings and extended the original work by implementing an ablation examining whether LLMs can iteratively reduce hallucinations through automated, class-agnostic self-critique, following the authors' recommendation to use class-agnostic detection due to instability in class-aware hallucination labeling (Hegselmann et al. 2024a).

Methodology

Environment

For our reproduction, we used a hybrid environment. We largely reused the authors original code, but wrote some small snippets to orchestrate execution together. We had a local environment, and for the fine-tuned models, we ran on Google Colab. We had to pay for Google Colab+, so as to access the NVIDIA A100 GPUs, which enabled us to run these workloads. Here are the libraries that we needed across our environments:

- sentencepiece
- accelerate
- datasets
- trl
- py7zr
- scipy
- wandb
- evaluate
- rouge-score
- sacremoses
- sacrebleu
- seqeval
- bert_score
- swifter
- bioc
- medcat
- plotly
- nervaluate
- nbformat
- kaleido
- spacy
- fire

Data

The authors used the MIMIC-IV-Note dataset (Johnson et al. 2023; Goldberger et al. 2000) as the base dataset for their work. For their data processing, they selected as summaries

Dataset	Size	Description
MIMIC Discharge Instructions Datasets		
MIMIC-IV-Note-Ext-DI	100,175	Summarization dataset derived from MIMIC-IV-Note with the section Discharge Instructions as summary and all prior notes as context
MIMIC-IV-Note-Ext-DI-BHC	100,175	MIMIC-IV-Note-Ext-DI with the Brief Hospital Course as context
MIMIC-IV-Note-Ext-DI-BHC-Anno	26,178	Subset of MIMIC-IV-Note-Ext-DI-BHC with contexts $\leq$ 4,000 characters and summaries $\geq$ 600 characters to facilitate human annotation
Hallucination Datasets Annotated by Two Medical Experts		
Hallucinations-MIMIC-DI	100	Random examples from MIMIC-IV-Note-Ext-DI-BHC-Anno
Hallucinations-Generated-DI	100	20 random contexts from M.-IV-Note-Ext-DI-BHC-Anno and summaries generated with five models during hallucination-reduction experiments
Derived Datasets from Hallucinations-MIMIC-DI		
Original	100	Context-summary pairs from Hallucinations-MIMIC-DI
Cleaned	100	Original with labeled hallucinations manually removed or replaced
Cleaned & Improved	100	Cleaned with mistakes and artifacts removed or corrected

Table 1: Overview of all datasets used in this work. All datasets are publicly available on PhysioNet.

the Discharge Instructions (DI) sections of discharge notes. They initially considered all preceding note sections, but for their main dataset they narrow it to only the Brief Hospital Course (BHC) section — since BHC captures the hospital stay history and fits better into models' context windows. Because many DIs contained boilerplate text, templates, salutations or other irrelevant artifacts, they applied a cleaning/filtering pipeline to remove these; after filtering, they retained 100,175 out of 331,793 original discharge notes. From that cleaned set they additionally created a subset for human annotation (contexts $\leq$ 4,000 characters, summaries $\geq$ 600 characters), yielding 26,178 examples—this subset is used for labeling hallucinations. Finally, they produce several data variants: full-context (all sections pre-DI), BHC-context only, and the annotated subset.

We obtained their resulting dataset "Medical Expert Annotations of Unsupported Facts in Doctor-Written and LLM-Generated Patient Summaries" (Hegselmann et al. 2024b; Goldberger et al. 2000) dataset from PhysioNet after completing the requisite certification, training and signing the usage agreements. A summary of the files within that is provided in Table 1.

Models

The original paper uses several models pre-trained models. It is accompanied by a repository [1] to work with those models. Here is a description of those models and how we went about acquiring access to them.

Pretrained Models

- **LED**: The Longformer Encoder-Decoder was used as a baseline model (Beltagy, Peters, and Cohan 2020). LED weights start from Hugging Face[2] were further finetuned on an 87/13 training/validation data split (20931 vs 2608 examples). Each example feeds the full 4096-token encoder context and targets a 350-token decoder summary. Training ran end-to-end with sequence-to-sequence cross-entropy, batch size 1, no gradient accumulation, a single epoch, and a 5e-5 learning rate—enough for a lightweight domain adaptation pass while keeping the schedule consistent with the original paper's small-data setup. We log actual decoded summaries and metric scores at evaluation time. Progress and checkpoints were written every 50 steps. LED's vanilla architecture was retained (no adapters/LoRA), relying on the pretrained 16384-position Longformer attention to handle the long inputs; hyperparameters align with the best-performing settings documented in the paper for the high-truncation split.
- **Llama**: Llama 2 (Touvron et al. 2023) has shown promising performance on clinical text summarization (Van Veen et al. 2024), and we used the version with 7B.[3] As in the original paper, we always used 100 training examples for parameter-efficient fine-tuning with LoRA (Hu et al. 2021) and loaded the models in 8-bit to reduce the memory usage. The original paper used both the 7B and 70B parameter models. For our reproduction, we only used the 7B model. We did not have access to compute resources to run the 70B model. We had a batch_size of 1 and gradient accumulation set to 16 steps, the effective batch is 16 tokens, helping the small dataset without exceeding memory. Optimization follows a fixed 100-step schedule (saving/logging every 20 steps) with learning rate $2 \times e^{-5}$. These steps cover the entire 100-example training set roughly once, matching the paper's short fine-tune regime
- **GPT-4**: GPT-4 (OpenAI et al. 2024) represents the state of the art in clinical summarization (Van Veen et al. 2024). We accessed the model via the OpenAI API and made sure to opt-out of human review of the data to ensure data privacy. We tested the model with 5 in-context

[1] https://github.com/stefanhgm/patient_summaries_with_llms

[2] Huggingface models allenai/led-base-16384

[3] Huggingface models meta-llama/Llama-2-{7,70}b-hf

examples (5-shot) or no examples (0-shot). We used the GPT-4 model without additional fine-tuning.

Training

Since the models in the paper were pre-trained, here we shall focus on the fine tuning required, and the prompt engineering.

Parameter Tuning

Both LED and Llama rely on Hugging Face's seq2seq training loop, minimizing the standard cross-entropy (negative log-likelihood) between the generated tokens and the reference summaries (teacher forcing). GPT-4 is inference-only, so no loss function applies beyond the API's internal likelihood maximization.

- **LED-base (allenai/led-base-16384)**: We fine-tuned the pretrained LED encoder–decoder model on the MIMIC-IV discharge-instruction pairs. Inputs were truncated to 4096 tokens and targets to 350 tokens. Training used a batch size of 1 with no gradient accumulation, a single epoch over the selected training split, and the AdamW optimizer (`adamw_torch`) with a learning rate of 5×10^{-5} and a constant schedule. Evaluation employed `predict_with_generate`. The model (approximately 162M parameters) fits within the memory constraints of a single 16 GB GPU (T4 or A10).
- **Llama-2-7B LoRA**: We fine-tuned `meta-llama/Llama-2-7b-hf` using low-rank adapters (rank 8, $\alpha = 32$, dropout 0.1), updating only the adapter parameters. Training was performed with TRL's `SFTTrainer` on the 100-example subset using a batch size of 1, gradient accumulation of 16, between 5 and 100 steps, and a learning rate of 2×10^{-5} with mixed-precision execution (bf16 when available). Since only LoRA parameters are optimized, the procedure fits on a single 24–40 GB GPU.
- **GPT-4 inference**: We invoked OpenAIs temparetur through Guidance, with a temperature=0, max_new_tokens=600, n-shot=0 or 5 (number of in-context examples); no training loss since GPT-4 is only prompted.

Prompting

GPT-4 is not tuned; instead we prepared JSONL files with 0-shot and 5-shot in-context prompts (prompt id 3) and send them through Guidance (temperature 0, max 600 tokens). Calls are throttled to 5/min to respect API limits, and all computation happens on OpenAI's backend—we simply orchestrate the prompts locally and collect the generated summaries for scoring.

Hardware and Resources Required

Hardware: all fine-tuning runs were executed on single NVIDIA A100 GPUs (40GB VRAM) for both LED and Llama LoRA; GPT-4 runs leverage OpenAI's hosted infrastructure. **Runtime per epoch/run**:

- **LED** (full dataset) takes 60–90 minutes for one epoch on an A100; a 100-example smoke run took 15 minutes.
- **Llama LoRA** max_steps=100 completes in 30–40 minutes; the 5-step smoke run is 5 minutes. GPT-4 batches of 100 prompts take 20–30 minutes depending on API throttling. A total of 111.89 compute units were used on Google Colab.

Total trials / GPU hours / epochs: LED: 1 epoch per dataset variant ($\approx$ 1.5 GPU-hrs each). Llama: max_steps=5 (smoke) or 100 (full repro), repeated for each dataset ($\approx$0.5–1 GPU-hrs per run). **GPT-4**: two inference passes per experiment (0-shot and 5-shot) with 100 prompts each (no GPU time locally).

Evaluation

For every set of generated summaries (LED, Llama, and GPT-4) we compute the same quantitative suite: ROUGE-1/2/3/4/L F1, BERTScore (roberta-large and deberta-large variants), SARI (to capture simplification quality), and average output length. Those metrics are calculated by our compute_custom_metrics helper, which takes the reference discharge instructions and the system outputs and reports the scores on the held-out test split. This replicates the custom metrics used in the paper.

Results

Reproduced Results

Please see Table 2 for the results of our reproduction. The scores from our runs are presented alongside the original results from the paper.

Comparison with the Original Paper

Table 2 presents the results of our reproduction of the patient-summarization experiments on the MIMIC-IV-Note-Ext-DI-BHC dataset. We use the exact dataset and splits provided by the original authors, allowing us to focus on the reproducibility of model behavior rather than dataset construction. Across models, our reproduced metrics generally fall close to the originally reported values, and importantly, the relative ordering of models is preserved. As in the original work, LED-large achieves the strongest ROUGE and BERTScore performance (e.g., ROUGE-1 of 43.25 in our reproduction vs. 43.82 originally), while Llama 2 7B and GPT-4 exhibit slightly lower but still comparable scores. These results suggest that, at a high level, the summarization component of the study is reproducible under reasonable methodological fidelity.

Some model-specific discrepancies merit further discussion. Our GPT-4 5-shot results differ from the published values by approximately 2–3 ROUGE points across several metrics. We cannot definitively isolate the cause of these differences, but one plausible explanation is *model drift*: OpenAI periodically updates and improves its API-served models under fixed model names, and prior work has documented measurable variation in model behavior over time when using the same nominal model identifier (Chen, Zaharia, and Zou 2024). Because the original paper predates

Model	R-1↑	R-2↑	R-3↑	R-4↑	R-L↑	BERT↑	DeBERT↑	SARI↑	Words
MIMIC-IV-Note-Ext-DI-BHC (100,175 examples)									
LED-large (80,140 ex.)	43.25 (42.30)	15.95 (14.98)	7.97 (7.04)	4.89 (3.87)	27.57 (26.50)	86.84 (86.71)	60.90 (60.85)	45.68 (44.38)	117.38 (117.81)
Llama 2 7B (100)	36.95 (38.36)	11.68 (12.66)	4.64 (5.13)	2.08 (2.24)	22.55 (24.73)	84.86 (85.68)	57.59 (60.23)	41.95 (44.12)	106.00 (73.13)
Llama 2 70B (–)	– (40.58)	– (14.31)	– (6.09)	– (2.74)	– (26.19)	–(100) (86.30)	– (61.89)	– (45.16)	– (76.90)
GPT-4 5-shot (5)	41.14 (38.80)	12.93 (10.78)	5.10 (3.55)	2.29 (1.12)	24.03 (21.98)	87.06 (86.67)	62.47 (61.30)	42.51 (42.88)	133.98 (131.86)
GPT-4 0-shot (0)	38.66 (38.26)	11.40 (10.81)	3.90 (3.70)	1.51 (1.49)	21.65 (21.49)	86.59 (86.37)	61.28 (60.75)	42.18 (42.04)	177.59 (165.78)

Table 2: Quantitative metrics from our reproduction of patient summary generation on MIMIC-IV-Note-Ext-DI-BHC; main entries are reproduction scores, and parentheses list the originally reported ROUGE (R-n/R-L), BERTScore (BERT/DeBERT), SARI, and word-count values for comparison.

these updates, and because our evaluation could not access archived model snapshots, divergences of this magnitude are not unexpected in API-based reproduction settings. This observation echoes broader concerns raised in the literature about the reproducibility challenges posed by non-versioned LLM APIs (Biderman, Hallahan, and Gao 2023).

Overall, while numerical discrepancies arise—particularly for API-hosted models that evolve over time and generative models sensitive to decoding hyperparameters—the **qualitative conclusions of the original study are robust**. LED-large remains the strongest performer, Llama 2 7B and GPT-4 follow expected patterns, and the general scale and ranking of ROUGE and BERTScore metrics fall within the same operational range as the originally reported values. Our reproduction therefore supports the claim that the summarization component of the study is broadly reproducible, while also highlighting the practical challenges inherent in reproducing results involving cloud-hosted LLMs and complex generation pipelines.

Ablation: Automated Hallucination Reduction via Class-Agnostic LLM Self-Critique

To explore whether automated hallucination detection can meaningfully support iterative summary refinement, we conducted an ablation in which a large language model served as both generator and self-critic. Consistent with the methodological guidance of the original paper, we employed *class-agnostic* hallucination detection, rather than class-aware classification. The authors report that class-aware approaches—where a detector must not only identify unsupported spans but also assign them to specific error categories—exhibit substantially lower recall, making them unsuitable for reliable quantitative evaluation. In accordance with their annotation scheme, we additionally do not count general medical knowledge or benign patient-facing advice (e.g., "take your medications as prescribed" or "call your doctor if symptoms worsen") as hallucinations unless such statements contradict the clinical context. Only spans introducing unsupported clinical facts (e.g., medications, procedures, diagnoses, test results, or temporal events not grounded in the context) are treated as hallucinations in this experiment.

For each evaluation example, we first prompted an LLM to generate a *zero-shot* discharge-style summary (S_0) from the clinical context. This follows the original paper's finding that few-shot prompting does not meaningfully improve ROUGE, BERTScore, or related summarization metrics, and avoids introducing additional sources of variability. We next applied class-agnostic hallucination detection to S_0 using a structured prompt that extracts spans unsupported by the source context. These hallucinated spans, when present, were then provided back to the model along with the original summary to elicit a revised version (S_1) intended to remove or correct unsupported content. This yields a simple automated self-refinement loop (Figure 1) after which both summaries can be re-evaluated for hallucination content using the same detector.

Although the automatic detection experiments of the original paper rely on GPT-4, we deliberately substitute **GPT-5.1** as the engine for both detection and refinement. This substitution does not alter the conceptual design of the ablation—zero-shot generation, class-agnostic detection, and targeted correction—but instead reflects practical considerations of scalability and model capability. GPT-5.1 offers a substantially larger context window and lower inference cost, enabling efficient experimentation on long clinical inputs, and its improved reasoning ability provides a stronger test of whether iterative self-critique reliably reduces hallucinations. After generating both S_0 and S_1 for each example, we re-applied the same hallucination detector to quantify changes in unsupported content. We report in Table 3 the proportion of summaries that improved, remained unchanged, or worsened, along with mean hallucination counts and deltas. This ablation extends the original study by assessing whether SOTA LLMs can autonomously identify and reduce hallucinations through structured self-correction, without requiring human annotation or supervised fine-tuning.

Ablation Discussion

Although the gains observed in our ablation are modest, they nonetheless provide preliminary evidence that a simple self-critique loop can reduce hallucinations in model-generated summaries. As shown in Table 3, the revised summaries exhibit a lower average hallucination count (0.74 vs. 1.02), and roughly one-third of examples show measurable improvement. The fact that the majority of summaries remain

Metric	Value
Total examples	50
Improved ($\Delta < 0$)	16
Unchanged	26
Worsened ($\Delta > 0$)	8
Avg. hallucinations in S_0	1.02
Avg. hallucinations in S_1	0.74
Avg. delta ($H_1 - H_0$)	-0.28

Table 3: Ablation Summary for LLM Self-Refinement Experiment

unchanged, and a non-negligible portion worsen, highlights both the promise and the limitations of applying automated hallucination feedback without human supervision.

These results should be interpreted as an initial proof of concept rather than a definitive evaluation of self-refinement. Our implementation uses zero-shot prompting, a single-step refinement loop, and class-agnostic detection; more sophisticated prompting strategies, multi-step refinement procedures, or hybrid detection schemes could yield stronger improvements. Nonetheless, even a lightweight approach produces measurable reductions in unsupported content, suggesting that LLM-based self-critique is a viable direction for improving factual consistency in clinical summarization. Further research is encouraged to explore more robust refinement mechanisms, stronger hallucination detectors, and the conditions under which iterative correction is most effective.

Discussion

Impact and Reproducibility

This work addresses an increasingly important problem domain. Recent reporting and early empirical studies (Astor) suggest that patients are beginning to consult large language models for medical concerns and for help interpreting clinical documentation. As LLMs become more integrated into patient-facing workflows, the fidelity and factual reliability of automatically generated medical summaries take on heightened importance. In this context, the original paper makes several valuable contributions, and our reproduction and ablation studies underscore their relevance.

First, the paper provides clear evidence that commonly used lexical similarity metrics such as ROUGE and BERTScore are poor proxies for faithfulness in clinical summarization. High surface overlap does not reliably correspond to factual correctness, underscoring the need for evaluation frameworks that explicitly measure hallucinations rather than textual similarity alone. Second, the authors introduce a structured annotation protocol for identifying unsupported statements in clinical discharge summaries. Third, the paper is accompanied with a high-quality annotated dataset. This resource provides a foundation for systematic assessment of factual consistency in model-generated medical text. Finally, the paper demonstrates that large language models—and GPT-4 in particular—can serve as effective automatic hallucination detectors when prompted with the authors' class-agnostic annotation scheme. This finding suggests that self-critique pipelines may offer a scalable alternative to expert annotation for improving factual reliability.

Our ablation study extends this final point by exploring whether an LLM can not only detect hallucinations but also iteratively revise its own outputs to reduce them. While our results show only modest quantitative improvement, the experiment illustrates a promising research direction: automated self-refinement loops that enhance faithfulness without requiring additional supervision. As LLMs continue to evolve and as clinical deployment settings demand higher guarantees of factual correctness, we expect future work to build on these insights, integrating hallucination-aware training, evaluation, and inference-time feedback mechanisms.

Ease of Reproduction

What was easy? The paper was very well documented. The authors also made their post-processed data available as a dataset (Hegselmann et al. 2024b), allowing the reproduction to skip the data of processing the original MIMIC-IV Note dataset (Johnson et al. 2023; Goldberger et al. 2000). The authors also made their code available, allowing a deeper inspection of their methods, and a quicker set up of the ablation.

What was difficult Reproduction was severely encumbered by difficulties setting up the code as the original authors had it- the versions of libraries used were not pinned, so recreating their environment involved lots of trial and error. In our fork of their work, we have tried to remedy some of these issues. We were also hampered by difficulty getting a stable compute environment that allows iteration (editing, version control), can host the private data (consistent with the usage requirements of the datasets), and has the necessary compute resources (40+ GB of GPU RAM . Various alternatives were considered (running locally, Kaggle, Google Compute) but that process of iteration was expensive in its own right. Ultimately, we settled on Google Compute with sync handled variously by Google Drive (data) integration and Github (code). This janky set up was also its own source of new sync errors.

Recommendations for easing reproducibility

It is commendable that the authors made their code and datasets publicly available. To further enhance reproducibility authors should pin their software and dependency versions used. Use pyenv, docker or some other portable, precise environment description. A few entry level commands, easily available and documented will also make efforts at high fidelity reproduction much easier.

Author Contributions

This reproduction study was conducted by a single author, Raymond Macharia (`rmm15@illinois.edu`). The author was responsible for all components of the project, including: reviewing and interpreting the original paper; selecting and designing the experiments to reproduce; obtaining data access credentials and preparing the requisite

datasets; configuring computational environments and securing the necessary resources for model training and inference; implementing and executing the reproduction experiments; performing analysis of the resulting outputs; and compiling the findings into the present report. The author would be remiss not to express gratitude for the original papers' authors, CS 598 course staff and instructor.

References

Astor, M. ???? People are uploading their medical records to A.I. Chatbots - The New York Times.

Beltagy, I.; Peters, M. E.; and Cohan, A. 2020. Longformer: The Long-Document Transformer. arXiv:2004.05150.

Biderman, S.; Hallahan, E.; and Gao, L. 2023. Emergent Problems in Large Language Model Evaluation. In *Proceedings of the 40th International Conference on Machine Learning*.

Chen, L.; Zaharia, M.; and Zou, J. 2024. How Is ChatGPT's Behavior Changing Over Time? *arXiv preprint arXiv:2307.09009*.

Goldberger, A. L.; Amaral, L. A. N.; Glass, L.; Hausdorff, J. M.; Ivanov, P. C.; Mark, R. G.; Mietus, J. E.; Moody, G. B.; Peng, C.-K.; and Stanley, H. E. 2000. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation*, 101(23): e215–e220.

Hegselmann, S.; Shen, Z.; Gierse, F.; Agrawal, M.; Sontag, D.; and Jiang, X. 2024a. A Data-Centric Approach To Generate Faithful and High Quality Patient Summaries with Large Language Models. In *Proceedings of the fifth Conference on Health, Inference, and Learning*, volume 248 of *Proceedings of Machine Learning Research*, 339–379. PMLR.

Hegselmann, S.; Shen, Z.; Gierse, F.; Agrawal, M.; Sontag, D.; and Jiang, X. 2024b. Medical Expert Annotations of Unsupported Facts in Doctor-Written and LLM-Generated Patient Summaries.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685.

Johnson, A. E. W.; Stone, D. J.; Celi, L. A.; and Pollard, T. J. 2023. MIMIC-IV-Note, Version 2.2.

OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; Avila, R.; Babuschkin, I.; Balaji, S.; Balcom, V.; Baltescu, P.; Bao, H.; Bavarian, M.; Belgum, J.; Bello, I.; Berdine, J.; Bernadett-Shapiro, G.; Berner, C.; Bogdonoff, L.; Boiko, O.; Boyd, M.; Brakman, A.-L.; Brockman, G.; Brooks, T.; Brundage, M.; Button, K.; Cai, T.; Campbell, R.; Cann, A.; Carey, B.; Carlson, C.; Carmichael, R.; Chan, B.; Chang, C.; Chantzis, F.; Chen, D.; Chen, S.; Chen, R.; Chen, J.; Chen, M.; Chess, B.; Cho, C.; Chu, C.; Chung, H. W.; Cummings, D.; Currier, J.; Dai, Y.; Decareaux, C.; Degry, T.; Deutsch, N.; Deville, D.; Dhar, A.; Dohan, D.; Dowling, S.; Dunning, S.; Ecoffet, A.; Eleti, A.; Eloundou, T.; Farhi, D.; Fedus, L.; Felix, N.; Fishman, S. P.; Forte, J.; Fulford, I.; Gao, L.; Georges, E.; Gibson, C.; Goel, V.; Gogineni, T.; Goh, G.; Gontijo-Lopes, R.; Gordon, J.; Grafstein, M.; Gray, S.; Greene, R.; Gross, J.; Gu, S. S.; Guo, Y.; Hallacy, C.; Han, J.; Harris, J.; He, Y.; Heaton, M.; Heidecke, J.; Hesse, C.; Hickey, A.; Hickey, W.; Hoeschele, P.; Houghton, B.; Hsu, K.; Hu, S.; Hu, X.; Huizinga, J.; Jain, S.; Jain, S.; Jang, J.; Jiang, A.; Jiang, R.; Jin, H.; Jin, D.; Jomoto, S.; Jonn, B.; Jun, H.; Kaftan, T.; Łukasz Kaiser; Kamali, A.; Kanitscheider, I.; Keskar, N. S.; Khan, T.; Kilpatrick, L.; Kim, J. W.; Kim, C.; Kim, Y.; Kirchner, J. H.; Kiros, J.; Knight, M.; Kokotajlo, D.; Łukasz Kondraciuk; Kondrich, A.; Konstantinidis, A.; Kosic, K.; Krueger, G.; Kuo, V.; Lampe, M.; Lan, I.; Lee, T.; Leike, J.; Leung, J.; Levy, D.; Li, C. M.; Lim, R.; Lin, M.; Lin, S.; Litwin, M.; Lopez, T.; Lowe, R.; Lue, P.; Makanju, A.; Malfacini, K.; Manning, S.; Markov, T.; Markovski, Y.; Martin, B.; Mayer, K.; Mayne, A.; McGrew, B.; McKinney, S. M.; McLeavey, C.; McMillan, P.; McNeil, J.; Medina, D.; Mehta, A.; Menick, J.; Metz, L.; Mishchenko, A.; Mishkin, P.; Monaco, V.; Morikawa, E.; Mossing, D.; Mu, T.; Murati, M.; Murk, O.; Mély, D.; Nair, A.; Nakano, R.; Nayak, R.; Neelakantan, A.; Ngo, R.; Noh, H.; Ouyang, L.; O'Keefe, C.; Pachocki, J.; Paino, A.; Palermo, J.; Pantuliano, A.; Parascandolo, G.; Parish, J.; Parparita, E.; Passos, A.; Pavlov, M.; Peng, A.; Perelman, A.; de Avila Belbute Peres, F.; Petrov, M.; de Oliveira Pinto, H. P.; Michael; Pokorny; Pokrass, M.; Pong, V. H.; Powell, T.; Power, A.; Power, B.; Proehl, E.; Puri, R.; Radford, A.; Rae, J.; Ramesh, A.; Raymond, C.; Real, F.; Rimbach, K.; Ross, C.; Rotsted, B.; Roussez, H.; Ryder, N.; Saltarelli, M.; Sanders, T.; Santurkar, S.; Sastry, G.; Schmidt, H.; Schnurr, D.; Schulman, J.; Selsam, D.; Sheppard, K.; Sherbakov, T.; Shieh, J.; Shoker, S.; Shyam, P.; Sidor, S.; Sigler, E.; Simens, M.; Sitkin, J.; Slama, K.; Sohl, I.; Sokolowsky, B.; Song, Y.; Staudacher, N.; Such, F. P.; Summers, N.; Sutskever, I.; Tang, J.; Tezak, N.; Thompson, M. B.; Tillet, P.; Tootoonchian, A.; Tseng, E.; Tuggle, P.; Turley, N.; Tworek, J.; Uribe, J. F. C.; Vallone, A.; Vijayvergiya, A.; Voss, C.; Wainwright, C.; Wang, J. J.; Wang, A.; Wang, B.; Ward, J.; Wei, J.; Weinmann, C.; Welihinda, A.; Welinder, P.; Weng, J.; Weng, L.; Wiethoff, M.; Willner, D.; Winter, C.; Wolrich, S.; Wong, H.; Workman, L.; Wu, S.; Wu, J.; Wu, M.; Xiao, K.; Xu, T.; Yoo, S.; Yu, K.; Yuan, Q.; Zaremba, W.; Zellers, R.; Zhang, C.; Zhang, M.; Zhao, S.; Zheng, T.; Zhuang, J.; Zhuk, W.; and Zoph, B. 2024. GPT-4 Technical Report. arXiv:2303.08774.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971.

Van Veen, D.; Van Uden, C.; Blankemeier, L.; Delbrouck, J.-B.; Aali, A.; Bluethgen, C.; Pareek, A.; Polacin, M.; Reis, E. P.; Seehofnerová, A.; Rohatgi, N.; Hosamani, P.; Collins, W.; Ahuja, N.; Langlotz, C. P.; Hom, J.; Gatidis, S.; Pauly, J.; and Chaudhari, A. S. 2024. Adapted large language models can outperform medical experts in clinical text summarization. *Nature Medicine*, 30(4): 1134–1142.

Appendix

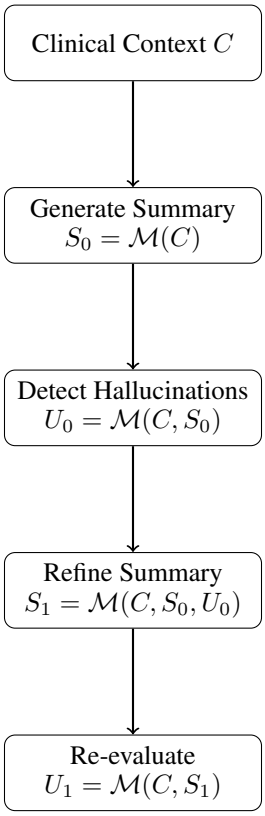

Figure 1: Self Critique workflow: a class-agnostic, single-step self-critique loop. The model generates an initial summary (S_0), identifies unsupported spans (U_0), produces a refined summary (S_1), and re-evaluates hallucinations.