Demo: Customizing Open-Source LLMs for Quantitative Medication Attribute Extraction across Heterogeneous EHR Systems

Zhe Fei[*,†] **Mehmet Yigit Turali**[*,‡] **Shreyas Rajesh**[*,‡]

Xinyang Dai[§] **Huyen Pham**[§] **Pavan Holur**[‡] **Yuhui Zhu**[§]

Larissa Mooney[§] **Yih-Ing Hser**[§] **Vwani Roychowdhury**[‡]

Abstract

Harmonizing medication data across Electronic Health Record (EHR) systems is a persistent barrier to monitoring medications for opioid use disorder (MOUD). In heterogeneous EHR systems, key prescription attributes are scattered across differently formatted fields and free-text notes. We present a practical framework that customizes open-source large language models (LLMs), including Llama, Qwen, Gemma, and MedGemma, to extract a unified set of MOUD prescription attributes (prescription date, drug name, duration, total quantity, daily quantity, and refills) from heterogeneous, site-specific data and compute a standardized metric of medication coverage, *MOUD days*, per patient. Our pipeline processes records directly in a fixed JSON schema, followed by lightweight normalization and cross-field consistency checks. We evaluate the system on prescription-level EHR data from five clinics in a national OUD study (25,605 records from 1,257 patients), using a previously annotated benchmark of 10,369 records (776 patients) as the ground truth. Performance is reported as coverage (share of records with a valid, matchable output) and record-level exact-match accuracy. Larger models perform best overall: Qwen2.5-32B achieves **93.4%** coverage with **93.0%** exact-match accuracy across clinics, and MedGemma-27B attains **93.1%/92.2%**. A brief error review highlights three common issues and fixes: imputing missing dosage fields using within-drug norms, handling monthly/weekly injectables (e.g., Vivitrol) by setting duration from the documented schedule, and adding unit checks to prevent mass units (e.g., "250 g") from being misread as daily counts. By removing brittle, site-specific ETL and supporting local, privacy-preserving deployment, this approach enables consistent cross-site analyses of MOUD exposure, adherence, and retention in real-world settings.

1 Introduction

The opioid crisis remains a major public health issue impacting communities across the United States, with over 105,000 overdose deaths recorded between December 2022 and January 2023

[*]These authors contributed equally.

[†]Department of Statistics, UC Riverside, Riverside, CA, 92521

[‡]Department of Electrical and Computer Engineering, UCLA, Los Angeles, CA, 90095

[§]Department of Psychiatry and Biobehavioral Sciences, UCLA, Los Angeles, CA 90024

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: The Second Workshop on GenAI for Health: Potential, Trust, and Policy Compliance.

[1]. While the opioid crisis has not been limited to a specific region, rural communities have been particularly hit hard [14]. Medications for opioid use disorder (MOUD) have been identified as an effective treatment approach in reducing opioid use [6, 18]. However, they tend to be significantly underutilized, especially in rural communities [3, 2, 16]. Individuals with OUD in rural communities face significant challenges in accessing MOUD treatment, largely due to geographical isolation, limited transportation infrastructure, and a shortage of providers [13, 4].

Effective OUD treatment monitoring and quality improvement require systematic analysis of medication patterns across healthcare systems. However, this analysis is severely hampered by heterogeneity across Electronic Health Record (EHR) systems, critical MOUD prescription data are scattered across differently formatted structured fields and unstructured clinical notes, making cross-clinic comparisons extremely tedious. Traditional ETL (Extract Transform Load) approaches require custom mappings for each EHR system and are highly brittle: even minor changes in field names, data formats, or clinical documentation patterns can break extraction pipelines, creating substantial maintenance burdens and limiting scalability. The emergence of large language models (LLMs)[15, 17, 21] offers a promising solution to this challenge, driven by their impressive instruction-following and natural language understanding capabilities. Recent advances have also enhanced their ability to reliably generate structured outputs, allowing them to return data that conforms to a predefined format such as JSON. Utilizing these dual strengths, we developed a framework that leverages open-source LLMs (Qwen, Llama) to extract and standardize MOUD prescription attributes—prescription date, drug name, duration, quantities, and refills—from five clinics' disparate EHR systems. The system computes standardized "MOUD days" (medication coverage duration) for each prescription, enabling consistent cross-site analyses of treatment patterns, retention, and adherence without site-specific ETL development.

We demonstrate our system's capabilities using EHR data from five rural clinics participating in a national OUD study (CTN-0102C supported by National Drug Abuse Treatment Clinical Trials Network). Section 2 details our system architecture and implementation, as well as the data used to evaluate our framework. Section 3 presents extraction performance across different open-source models and clinic sites. Section 4 discusses deployment considerations and limitations in future real-world clinical settings.

2 Methods

Our system is designed to extract and harmonize MOUD prescription information from heterogeneous EHR data sources. **Figure 1 provides a high-level overview of this framework.** The process begins by ingesting raw, heterogeneous EHR data from various clinics. At its core, an instruction-tuned LLM acts as a universal translator, converting this diverse data into a unified, structured JSON format. This standardized output then undergoes post-processing to calculate the final MOUD days metric. Finally, the results are evaluated against a manually annotated ground truth dataset to measure performance in terms of accuracy and coverage. The framework consists of three main stages, detailed below: (1) data preparation, (2) LLM-based unified output extraction, and (3) MOUD computation and evaluation.

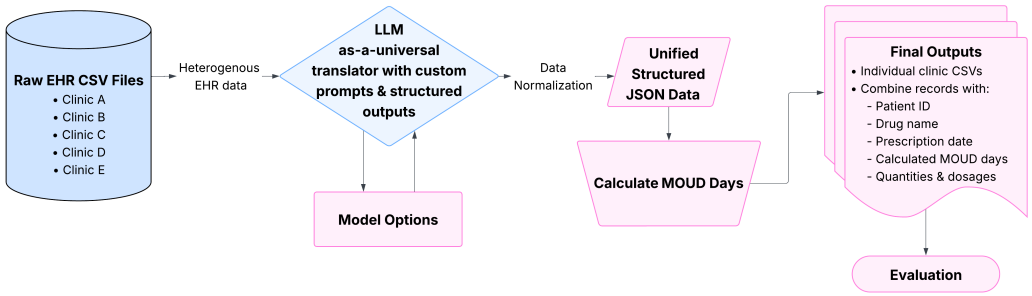

Figure 1: An overview of the framework for MOUD extraction and calculation. The system processes heterogeneous raw EHR data, uses an LLM to translate it into a unified JSON structure, calculates MOUD days, and evaluates the final output against a ground truth dataset.

2.1 Data Preparation

To evaluate our framework, we used prescription-level EHR data from five clinics participating in a multi-site national study on OUD, denoted as A – E. While the full dataset comprises **25,605 records** from **1,257 unique patients**, a benchmark ground truth set from the feasibility study period was established. This subset, consisting of **10,369 records** from **776 unique patients**, was manually annotated by UCLA medical professionals to extract the canonical values for MOUD days calculation. This annotated set serves as the gold standard for our quantitative evaluation, and its distribution is described in Table 1.

Across all sites, we targeted a common set of medication attributes to enable standardized calculation of "MOUD days": **clinic name, patient ID, prescription date, drug name, duration, total prescribed quantity, daily quantity, and number of refills**. These essential attributes were found in heterogeneous structured columns and, for some clinics, unstructured notes (Table 1).

Table 1: Ground truth data and example raw EHR fields by clinic (A–E).

Clinic	Annotated Records	Patients	Example raw EHR entries (name : value)
A	7	7	BRAND_NAME: BUPRENORPHINE-NALOXONE; GENERIC_NAME: buprenorphine HCl/naloxone HCl; PRESCRIPITION_DATE: 9/15/21; ROUTE_OF_ADMINISTRATION: SUBLINGUAL; UNIT_DOSE: 2 mg-0.5 mg; PRESCRIBED_QUANTITY: 1; DOSAGE_INSTRUCTIONS: place 2 tablet by sublingual route every day ... (2 tabs in am and 1 in pm); MEDICATION_INDICATION1: F11.11; ORIGINAL_REFILLS: 0; DATE_STOPPED: 1/27/22; RECORDID: 123835
B	5295	410	record_num: 322853; epic_medication_id: 120111686; epic_medication_name: BUPRENORPHINE HCL-NALOXONE HCL 8-2 MG SL SUBL; med_route: Sublingual; dose_unit: tablet; dose_instructions: Place 1 tablet under the tongue every 8 hours as needed for up to 28 days.; frequency: EVERY 8 HOURS PRN; quantity: 84 tablet; refill: 0; prescription_date: 10/29/19
C	4402	286	RecordID: 102724; DrugDescription: Suboxone 2-0.5 mg film; PrescribedDate: 6/17/21; SUMMARY: Suboxone 2-0.5 mg film; ROUTE: UNDER TONGUE; INSTRUCTIONS: Place 0.25 strip under tongue once a day For chronic OUD, XS2110928; Refills: 0; DOSE_UNIT: strip; PrescribedQuantity: 7 film; DoseQuantity: NULL
D	139	18	RecordNumber: 926195; Code: 657570300; Description: Vivitrol 380 mg suspension, extended rel recon; PrescriptionDate: 8/7/23; UnitDosage: ; DosageInstructions: INJECT 380 MG INTRAMUSCULARLY EVERY FOUR WEEKS ...; DoseQuantity: 1 each; NumberOfRefillsAuthorized: 2
E	426	27	RecordID: 114147; PrescribedDate: 6/22/21; DrugDescription: Suboxone 8-2 mg film; SUMMARY: pt missed appt today; ROUTE: UNDER TONGUE; Refills: 0; INSTRUCTIONS: Dissolve 1 film under tongue once a day; DOSE_UNIT: film; PrescribedQuantity: 28 film
Total	**10369**	**776**	

2.2 LLM-based Unified Output Extraction

Our extraction framework leverages instruction-tuned LLMs as a universal translator, converting heterogeneous EHR data into a common, unified schema. The core of our approach utilizes constrained generation: instead of parsing unstructured text output, we force the model's generation to directly conform to a predefined Pydantic-based JSON schema [5]. This method guarantees syntactically correct, structured data and has been shown to improve task performance by eliminating parsing errors [19]. The structured output pipeline enforces strict adherence to schemas containing required

fields (patient id, prescription date, drug name, etc.) crucial for computing MOUD days. We detail the prompts used across each clinic and the unified output schema in Appendix A.

To implement this strategy, we evaluated a comprehensive suite of open-source LLMs. Due to privacy regulations and HIPAA compliance requirements, only models that could be deployed locally were considered, excluding closed-source APIs such as GPT-4 [15] or Gemini [17]. The selected models, chosen to assess performance-efficiency trade-offs, included Qwen2.5 (32B) [20], Qwen3 (4B, 8B, 32B)[22], Gemma (4B, 7B, 27B), and MedGemma (4B, 27B). Initial experiments with sub-1B models revealed significant difficulties in generating reliable structured outputs, leading us to focus on these larger models.

For efficient and scalable inference, all models were deployed using the vLLM framework [12] on a single node with 4 NVIDIA A6000 GPUs (48GB VRAM each). We employed model quantization strategies [11, 8] to manage memory usage while maintaining inference quality. Tensor parallel processing was used across both GPUs to optimize throughput, with GPU memory utilization capped at 80% for stability. Detailed model parameters and inference settings are provided in Appendix B and C.

2.3 Post-Processing and MOUD Days Calculation

The JSON output from the LLM is parsed and subjected to a series of post-processing steps to ensure data quality and consistency. These include:

- **Type Normalization:** Casting extracted values to their correct data types (e.g., dates, integers) for ease of comparison with the ground truth.
- **Rule-Based Validation:** Applying rules to flag logical inconsistencies and remove duplicates. For example, a cross-field check ensures that *total quantity* is not less than *daily quantity*, and additional checks are implemented to filter out nonsensical and null values.
- **MOUD Days Calculation:** The primary outcome, "MOUD days", representing the total potential medication coverage from a prescription and its refills, is computed. A hierarchical logic is used which prioritizes the explicitly extracted duration. If duration is not available, it is derived from the quantity and dosage information. The total MOUD days are calculated for each record according to the following formula:

$$\text{MOUD_days} = (\text{number_of_refills} + 1) \times \begin{cases} \text{duration}, \text{if duration is provided;} \\ \frac{\text{total_quantity}}{\text{daily_quantity}}, \text{if duration is missing.} \end{cases}$$

2.4 Evaluation Pipeline

We evaluated our framework's performance and flexibility by integrating several leading open-source LLMs as its core information extraction engine. The output from each model was evaluated against the manually annotated ground truth data. To assess robustness against real-world data variations, the evaluation was conducted on a per-clinic basis, reflecting the distinct EHR data structures at each site. Performance was measured using the following quantitative metrics:

- **Coverage (%):** The percentage of ground truth records for which the model successfully generated a parsable output that could be matched using a composite key of **clinic name, patient ID, prescription date, and drug name**. This metric measures the system's ability to successfully process each input record before its accuracy is assessed.
- **Record-Level Exact Match Accuracy (%):** For the records successfully covered, this is the percentage where all five extracted attributes (as detailed in Subsection 2.1) and MOUD days perfectly match the ground truth values after normalization.

In addition to these metrics, we performed a qualitative error analysis to identify common failure modes, such as models struggling with ambiguous phrasing in clinical notes or complex dosage instructions. We further present these in Section 3

Table 2: **Performance of LLMs on MOUD Extraction**. Coverage (Cov.) and Exact Match Accuracy (Acc.) are in percentages (%). Clinic names are abbreviated. Full model details are in Appendix B.

	A		B		C		D		E		**Overall**	
Model	Cov.	Acc.	Cov.	Acc.	Cov.	Acc.	Cov.	Acc.	Cov.	Acc.	Cov.	Acc.
Qwen2.5 (32B)	100.00	100.00	95.98	93.51	96.32	93.04	97.12	80.74	52.35	89.69	93.4	93.0
Qwen3 (32B)	42.86	66.67	87.99	91.89	59.93	89.08	97.12	95.56	93.66	89.97	75.6	90.9
Qwen3 (8B)	100.00	100.00	89.54	92.77	51.61	69.24	27.34	94.74	42.96	95.63	69.4	85.2
Qwen3 (4B)	71.43	100.00	95.43	92.12	2.20	25.77	97.12	97.78	71.60	90.16	54.0	91.0
MedGemma (27B)	85.71	50.00	93.41	93.95	96.23	90.01	96.40	98.51	76.76	92.66	93.1	92.2
MedGemma (4B)	85.71	83.33	48.56	81.87	14.93	18.26	96.40	97.01	44.84	32.98	34.3	68.1
Gemma 3 (27B)	100.00	85.71	76.81	91.76	82.01	90.08	74.82	86.54	46.01	98.98	77.0	91.1
Gemma 3 (4B)	100.00	85.71	81.91	83.49	50.61	68.04	93.53	65.38	19.95	81.18	65.5	78.0

3 Results

The performance of the selected open-source LLMs on the MOUD extraction task is presented in Table 2. Our evaluation reveals two primary findings: 1) a significant variation in performance across different clinic datasets, highlighting the challenge of data heterogeneity, and 2) a consistent trend where models achieve high exact-match accuracy on the records they successfully process, but often struggle with overall record coverage.

A stark performance gap was observed between clinics. For **Clinic B and D**, most models achieved excellent coverage (often >95%) and high accuracy (>90%), suggesting that the data formats from these clinics are more amenable to automated extraction. In sharp contrast, a wide variance in performance was seen for **Clinic A and C**, with some models achieving perfect scores, while others failed completely. For example, **Qwen2.5 32B** achieved 100% on both metrics for Clinic A, while **Qwen3 32B** scored 0%. **Clinic E** represented a middle ground, where models like **Qwen3 32B** achieved high coverage (93.66%) while others, like **Gemma3 4B**, were less effective (19.95%).

When comparing model performance, the larger models generally yielded the best results. **Qwen2.5 32B** emerged as the most balanced model, achieving the highest overall coverage of **93.4%** while maintaining a strong overall accuracy of **93.0%**. **MedGemma (27B)** also performed robustly, with a comparable overall coverage of **93.1%** and accuracy of **92.2%**. While the smaller **Qwen3** models (4B and 8B) were competitive on accuracy, they exhibited lower and more inconsistent coverage, with the **Qwen3 8B** model's overall coverage dropping to **48.0%**.

Lastly, we reviewed LLM outputs with domain experts for recurring errors, and worked out corresponding fixes. First, Clinic B exhibited substantial missingness in dosage instructions (SIG) and daily quantity, as well as gaps in total prescribed quantity (about 20% records with key missing entries); to enable downstream calculations, we imputed daily quantity using the typical value observed for the same drug name and imputed total quantity with the median value. Second, while most medications are daily oral tablets/films with explicit instructions, about 5% prescriptions in Clinic C and E are administered monthly or weekly, because they are extended-interval injections (e.g., Vivitrol). LLMs frequently failed to translate these schedules into the structured output of daily and total quantities, so we manually set duration based on the documented dosing schedule for these cases. Third, we found unit-related outliers in the LLM outputs: whereas daily quantity is usually in tablets/films, occasional entries listed mass units (e.g., 250 g), which the models misread as a daily count (e.g., 250). We addressed this by adding unit normalization and plausibility checks to prevent misinterpretation and cap extreme values.

4 Concluding Remarks

In this work, we designed and evaluated a specialized framework to automate the extraction of structured medication data from heterogeneous EHR records, using this information to compute a standardized medication coverage duration (MOUD days). Our results demonstrate that a framework leveraging modern open-source LLMs with constrained JSON generation can serve as a robust and scalable alternative to traditional, brittle ETL pipelines. We found that models capable of running

on a single GPU, achieved high accuracy, successfully harmonizing prescription data across five disparate clinical systems without site-specific engineering.

This work provides a practical demonstration of how locally-deployable generative AI can address critical data exchange and integration problems in healthcare. By enabling the rapid, automated calculation of standardized metrics like MOUD days, our framework can significantly accelerate multi-site EHR research. This allows clinicians and policymakers to better understand treatment adherence and retention, particularly in the underserved rural communities highlighted in this study.

Limitations Despite these promising results, we acknowledge several limitations. First, while tested across five diverse clinics, the framework's generalizability to a wider range of EHR systems and documentation styles remains to be validated. We anticipate that newer reasoning models will handle heterogeneous EHRs with less prompt customization; evaluating this hypothesis is a priority for future work. Second, our current implementation focuses solely on MOUD prescriptions; extending it to other medication classes or clinical concepts would require further prompt engineering and evaluation. Finally, like all LLM-based systems, for high-stakes clinical applications, a human-in-the-loop review process would be essential to validate the extracted data.

Future Work Future work will address these limitations. We plan to expand our evaluation to a larger group of clinics and benchmark against reasoning-focused models [9, 23], which have shown strong performance on verifiable tasks like math and coding. We will also explore efficient fine-tuning techniques, such as LoRA [10, 7], to create smaller, specialized models. Finally, developing a user-friendly interface for expert review and error adjudication is a key priority to prepare the framework for clinical use.

5 Acknowledgment

Research reported in this publication was supported by the National Institute On Drug Abuse of the National Institutes of Health under Award Number UG1DA049435. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

[1] Farida B. Ahmad, Jennifer A. Cisewski, Lauren M. Rossen, and Patricia Sutton. Provisional drug overdose death counts. National Center for Health Statistics, Centers for Disease Control and Prevention, 2023. URL `https://www.cdc.gov/nchs/nvss/vsrr/provisional-drug-overdose.htm`. Accessed 2025-08-26.

[2] Solmaz Amiri, Katherine Hirchak, Michael G. McDonell, Justin T. Denney, Dedra Buchwald, and Ofer Amram. Access to medication-assisted treatment in the united states: Comparison of travel time to opioid treatment programs and office-based buprenorphine treatment. *Drug and Alcohol Dependence*, 224:108727, 2021. doi: 10.1016/j.drugalcdep.2021.108727.

[3] Solmaz Amiri, Michael G. McDonell, Justin T. Denney, Dedra Buchwald, and Ofer Amram. Disparities in access to opioid treatment programs and office-based buprenorphine treatment across the rural–urban and area deprivation continua: A u.s. nationwide small area analysis. *Value in Health*, 24(2):188–195, 2021. doi: 10.1016/j.jval.2020.08.2098.

[4] Tanner Bommersbach, Marissa Justen, Amanda M. Bunting, Melissa C. Funaro, Erin L. Winstanley, and Paul J. Joudrey. Multidimensional assessment of access to medications for opioid use disorder across urban and rural communities: A scoping review. *International Journal of Drug Policy*, 112:103931, 2023. doi: 10.1016/j.drugpo.2022.103931.

[5] Samuel Colvin, Eric Jolibois, Hasan Ramezani, Adrian Garcia Badaracco, Terrence Dorsey, David Montague, Serge Matveenko, Marcelo Trylesinski, Sydney Runkle, David Hewitt, Alex Hall, and Victorien Plot. Pydantic Validation, July 2025. URL `https://github.com/pydantic/pydantic`.

[6] Hilary S. Connery. Medication-assisted treatment of opioid use disorder: Review of the evidence and future directions. *Harvard Review of Psychiatry*, 23(2):63–75, 2015. doi: 10.1097/HRP.0000000000000075.

[7] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms, 2023.

[8] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

[9] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[10] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[11] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference, 2017. URL `https://arxiv.org/abs/1712.05877`.

[12] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[13] Jamey J. Lister, Addie Weaver, Jennifer D. Ellis, Joseph A. Himle, and David M. Ledgerwood. A systematic review of rural-specific barriers to medication treatment for opioid use disorder in the united states. *The American Journal of Drug and Alcohol Abuse*, 46(3):273–288, 2020. doi: 10.1080/00952990.2019.1694536.

[14] Katherine A. Mack, Christopher M. Jones, and Michael F. Ballesteros. Illicit drug use, illicit drug use disorders, and drug overdose deaths in metropolitan and nonmetropolitan areas—united states. *MMWR Surveillance Summaries*, 66(SS-19):1–12, 2017. doi: 10.15585/mmwr.ss6619a1.

[15] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin

Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report. 2024. URL https://arxiv.org/abs/2303.08774.

[16] Bernard Showers, Danielle Dicken, Jennifer S. Smith, and Aaron Hemlepp. Medication for opioid use disorder in rural america: A review of the literature. *Journal of Rural Mental Health*, 45(3), 2021. doi: 10.1037/rmh0000187.

[17] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[18] Nora D. Volkow, Thomas R. Frieden, Pamela S. Hyde, and Stephen S. Cha. Medication-assisted therapies—tackling the opioid-overdose epidemic. *New England Journal of Medicine*, 370(22):2063–2066, 2014. doi: 10.1056/NEJMp1402780.

[19] Brandon T Willard and Rémi Louf. Efficient guided generation for large language models. *arXiv preprint arXiv:2307.09702*, 2023.

[20] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[21] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[22] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[23] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Our technical appendix is structured as follows:

1. Appendix A: Prompts and Schemas for Prescription Information Extraction
2. Appendix B: Model Specifications and Parameters
3. Appendix C: Inference Configuration and Settings

A Prompts and Schemas for Prescription Information Extraction

This section details the system prompt, format instructions, and Pydantic output schemas used for extracting structured information from various prescription formats.

A.1 General System Prompt

All extraction tasks are guided by the following high-level instruction, which establishes the persona and objective for the LLM.

System Prompt

```
You are a medical expert, you are tasked with extracting
   useful information from a prescription. Before
   answering you should reason about the problem (using
   the "reasoning" field in the JSON response). You need
   to follow the format described below:
```

A.2 Shared Extraction Rules and Guidelines

Across all data formats, a common set of interpretation and post-processing rules are applied, as specified in the "Important Notes" section of each prompt.

Important Notes:

```
- If information is unavailable, set the field to None.
- Convert fractions with space (e.g., '3 1/2') to decimal
   values (e.g., '3.5').
- For medication frequency: interpret "X10" as 10 days,
   but only when X is followed by a reasonable number.
   Don't apply this rule if P follows X or if the number
   is unusually large.
- Always include detailed step-by-step calculations in the
    "reasoning" field, particularly for injections and
   complex dosing regimens.
- Watch for specialized dosing terms: "inject/injection,"
   "patch," "every 4 weeks," "monthly," "weekly," "once a
    week," "every 7 days," etc.
- For injection medications, carefully analyze the SIG
   field to determine proper administration schedule.
- Special medications like Vivitrol are injections
   administered monthly - always note this in your
   reasoning.
- When extracting daily quantities from dosage
   instructions, sum all individual doses (e.g., "one tab
    in morning, half tab at night" = 1.5).
- For duration calculations, extract explicit day counts
   or convert frequency information (weekly = 7 days,
   monthly = 30 days, etc.).
- Convert text numbers to numerals: "one" -> 1, "two" ->
   2, etc.
```

A.2.1 Hometown Prescription Format

Prompt Format:

```
{
    "reasoning": <reasoning about the answer>,
    "patient_id": <extract from 'RECORDID'>,
    "prescription_date": <extract from 'PRESCRIPTION_DATE
       '>,
    "drug_name": <extract from 'GENERIC_NAME', without
       usage info>,
    "drug_name_full": <extract from 'GENERIC_NAME'>,
    "total_quantity": <extract from 'PRESCRIBED_QUANTITY
       '>,
    "daily_quantity": <calculate from DOSAGE_INSTRUCTION>,
    "Refill": <extract from 'ORIGINAL_REFILLS'>,
    "drug_strength": <extract from 'UNIT_DOSE'>,
    "drug_form": <extract from 'ROUTE_OF_ADMINISTRATION'>,
    "SIG": <extract from 'DOSAGE_INSTRUCTIONS'>,
    "prescriber_id": <extract from 'PRESCRIBER_ID'>
}
```

Pydantic Schema: `HometownPrescription`

Field	`Type`	**Description**
reasoning	`Optional[str]`	Reasoning about the answer.
patient_id	`Optional[str]`	Extract from `RECORDID`.
prescription_date	`Optional[str]`	Extract from `PRESCRIPTION_DATE`.
drug_name	`Optional[str]`	Extract from `GENERIC_NAME`, without usage info.
drug_name_full	`Optional[str]`	Extract from `GENERIC_NAME`.
total_quantity	`Optional[float]`	Extract from `PRESCRIBED_QUANTITY`.
daily_quantity	`Optional[float]`	Calculate from `DOSAGE_INSTRUCTION` by summing quantities.
Refill	`Optional[float]`	Extract from `ORIGINAL_REFILLS`.
drug_strength	`Optional[str]`	Extract from `UNIT_DOSE`.
drug_form	`Optional[str]`	Extract from `ROUTE_OF_ADMINISTRATION`.
SIG	`Optional[str]`	Directions for use. Extract from `DOSAGE_INSTRUCTIONS`.
prescriber_id	`Optional[str]`	Extract from `PRESCRIBER_ID`.

A.2.2 Providence Prescription Format

Prompt Format:

```
{
    "reasoning": <reasoning about the answer>,
    "patient_id": <extract from 'record_num'>,
    "prescription_date": <extract from 'order_date'>,
    "drug_name": <extract from 'epic_medication_name',
       without usage info>,
    "drug_name_full": <extract from 'epic_medication_name
       '>,
    "total_quantity": <extract from 'dose_instructions'>,
    "daily_quantity": <extract from 'dose_instructions'>,
    "Refill": <number of refills>,
    "duration": <extract days from 'dose_instructions' or
       set NA>,
    "drug_strength": <extract from 'epic_medication_name
       '>,
    "drug_form": <extract from 'dose_unit'>,
    "SIG": <extract from 'dose_instructions'>,
    "prescriber_id": <extract from 'prescriber_id'>
}
```

Pydantic Schema: `ProvidencePrescription`

Field	`Type`	**Description**
reasoning	`Optional[str]`	Reasoning about the answer.
patient_id	`Optional[str]`	Extract from `record_num`.
prescription_date	`Optional[str]`	Extract from `order_date`.
drug_name	`Optional[str]`	Extract from `epic_medication_name`, without usage info.
drug_name_full	`Optional[str]`	Extract from `epic_medication_name`.
total_quantity	`Optional[float]`	Extract from `dose_instructions`.
daily_quantity	`Optional[float]`	Extract from `dose_instructions`.
Refill	`Optional[float]`	Number of refills prescribed.
duration	`Optional[float]`	Extract days from `dose_instructions` if present.
drug_strength	`Optional[str]`	Extract from `epic_medication_name`.
drug_form	`Optional[str]`	Extract from `dose_unit`.
SIG	`Optional[str]`	Directions for use. Extract from `dose_instructions`.
prescriber_id	`Optional[str]`	Extract from `prescriber_id`.

A.2.3 Seaport Prescription Format

Prompt Format:

```
{
    "reasoning": <reasoning about the answer>,
    "patient_id": <extract from 'Record_ID'>,
    "prescription_date": <extract from 'Prescription_Date
       '>,
    "drug_name": <extract from 'RX_Name', without usage
       info>,
    "drug_name_full": <extract from 'RX_Name'>,
    "total_quantity": <extract from 'Quantity'>,
    "daily_quantity": <extract from 'Unit_Dose' or 'SIG'>,
    "Refill": <number of refills>,
    "duration": <extract days from 'SIG' or set NA>,
    "drug_strength": <extract from 'RX_Name'>,
    "drug_form": <extract from 'Display_Dosage_Unit'>,
    "SIG": <extract from 'SIG'>,
    "prescriber_id": <extract from 'PRESCRIBER_ID'>
}
```

Pydantic Schema: `SeaportPrescription`

Field	Type	Description
reasoning	`Optional[str]`	Reasoning about the answer.
patient_id	`Optional[str]`	Extract from `Record_ID`.
prescription_date	`Optional[str]`	Extract from `Prescription_Date`.
drug_name	`Optional[str]`	Extract from `RX_Name`, without usage info.
drug_name_full	`Optional[str]`	Extract from `RX_Name`.
total_quantity	`Optional[float]`	Extract from `Quantity`.
daily_quantity	`Optional[float]`	Extract from `Unit_Dose` or `SIG`.
Refill	`Optional[float]`	Number of refills prescribed.
duration	`Optional[float]`	Extract days from `SIG` if present.
drug_strength	`Optional[str]`	Extract from `RX_Name`.
drug_form	`Optional[str]`	Extract from `Display_Dosage_Unit`.
SIG	`Optional[str]`	Directions for use. Extract from `SIG`.
prescriber_id	`Optional[str]`	Extract from `PRESCRIBER_ID`.

A.2.4 St. Mary's Prescription Format

Prompt Format:

```
{
    "reasoning": <reasoning about the answer>,
    "patient_id": <extract from 'RecordNumber'>,
    "prescription_date": <extract from 'PrescriptionDate
       '>,
    "drug_name": <extract from 'Description', without
       usage info>,
    "drug_name_full": <extract from 'Description'>,
    "total_quantity": <extract from 'DoseQuantity'>,
    "daily_quantity": <extract from 'UnitDosage' or '
       DosageInstructions'>,
    "Refill": <number of refills>,
    "duration": <extract days from 'DosageInstructions' or
        set NA>,
    "drug_strength": <extract from 'Description'>,
    "drug_form": <extract from 'Description'>,
    "SIG": <extract from 'DosageInstructions'>,
    "prescriber_id": <extract from 'PrescriberID'>
}
```

Pydantic Schema: `StMarysPrescription`

Field	Type	Description
reasoning	`Optional[str]`	Reasoning about the answer.
patient_id	`Optional[str]`	Extract from `RecordNumber`.
prescription_date	`Optional[str]`	Extract from `PrescriptionDate`.
drug_name	`Optional[str]`	Extract from `Description`, without usage info.
drug_name_full	`Optional[str]`	Extract from `Description`.
total_quantity	`Optional[float]`	Extract from `DoseQuantity`.
daily_quantity	`Optional[float]`	Extract from `UnitDosage` or `DosageInstructions`.
Refill	`Optional[float]`	Number of refills prescribed.
duration	`Optional[float]`	Extract days from `DosageInstructions` if present.
drug_strength	`Optional[str]`	Extract from `Description`.
drug_form	`Optional[str]`	Extract from `Description`.
SIG	`Optional[str]`	Directions for use. Extract from `DosageInstructions`.
prescriber_id	`Optional[str]`	Extract from `PrescriberID`.

A.2.5 Syringa Prescription Format

Prompt Format:

```
{
    "reasoning": <reasoning about the answer>,
    "patient_id": <extract from 'Record Number'>,
    "prescription_date": <extract from 'Order Dt/Tm'>,
    "drug_name": <extract from 'Order Mnemonic', without
       usage info>,
    "drug_name_full": <extract from 'Order Mnemonic'>,
    "total_quantity": <extract from 'Dispense Qty'>,
    "daily_quantity": <extract from 'Volume Dose'>,
    "Refill": <number of refills>,
    "Frequency": <extract from 'Frequency': daily=1, BID
       =2, TID=3>,
    "drug_form": <extract from 'Volume Dose Unit'>,
    "SIG": <extract from 'Frequency'>,
    "prescriber_id": <extract from 'Order Last Updt
       Provider Id'>
}
```

Pydantic Schema: `SyringaPrescription`

Field	`Type`	**Description**
reasoning	`Optional[str]`	Reasoning about the answer.
patient_id	`Optional[str]`	Extract from `Record Number`.
prescription_date	`Optional[str]`	Extract from `Order Dt/Tm`.
drug_name	`Optional[str]`	Extract from `Order Mnemonic`, without usage info.
drug_name_full	`Optional[str]`	Extract from `Order Mnemonic`.
total_quantity	`Optional[float]`	Extract from `Dispense Qty`.
daily_quantity	`Optional[float]`	Extract from `Volume Dose`.
Refill	`Optional[float]`	Number of refills prescribed.
Frequency	`Optional[float]`	Extract from `Frequency` (daily=1, BID=2, etc.).
drug_form	`Optional[str]`	Extract from `Volume Dose Unit`.
SIG	`Optional[str]`	Directions for use. Extract from `Frequency`.
prescriber_id	`Optional[str]`	Extract from `Order Last Updt Provider Id`.

A.2.6 Winterport Prescription Format

Prompt Format:

```
{
    "reasoning": <reasoning about the answer>,
    "patient_id": <extract from 'RecordID'>,
    "prescription_date": <extract from 'PrescribedDate'>,
    "drug_name": <extract from 'DrugDescription', without
       usage info>,
    "drug_name_full": <extract from 'DrugDescription'>,
    "total_quantity": <extract from 'PrescribedQuantity'>,
    "daily_quantity": <extract from 'DoseQuantity' or '
       INSTRUCTIONS'>,
    "Refill": <number of refills>,
    "duration": <extract days from 'INSTRUCTIONS' or set
       NA>,
    "drug_strength": <extract from 'DrugDescription'>,
    "drug_form": <extract from 'ROUTE'>,
    "SIG": <extract from 'INSTRUCTIONS'>,
    "prescriber_id": <extract from 'Prescriber'>
}
```

Pydantic Schema: `WinterportPrescription`

Field	Type	Description
reasoning	`Optional[str]`	Reasoning about the answer.
patient_id	`Optional[str]`	Extract from `RecordID`.
prescription_date	`Optional[str]`	Extract from `PrescribedDate`.
drug_name	`Optional[str]`	Extract from `DrugDescription`, without usage info.
drug_name_full	`Optional[str]`	Extract from `DrugDescription`.
total_quantity	`Optional[float]`	Extract from `PrescribedQuantity`.
daily_quantity	`Optional[float]`	Extract from `DoseQuantity` or `INSTRUCTIONS` by summing.
Refill	`Optional[float]`	Number of refills prescribed.
duration	`Optional[float]`	Extract days from `INSTRUCTIONS` if present.
drug_strength	`Optional[str]`	Extract from `DrugDescription`.
drug_form	`Optional[str]`	Extract from `ROUTE`.
SIG	`Optional[str]`	Directions for use. Extract from `INSTRUCTIONS`.
prescriber_id	`Optional[str]`	Extract from `Prescriber`.

B Model Specifications and Parameters

B.1 Qwen Model Family

Qwen2.5 (32B, GPTQ)

- **Architecture**: Transformer-based decoder-only architecture with RoPE positional embeddings, SwiGLU, RMSNorm, attention with QKV bias
- **Parameters**: 32.5 billion parameters
- **Quantization**: GPTQ 8-bit quantization
- **Context Length**: 131,072 tokens (native support), generation typically up to 8,000 tokens
- **Pre-training**: General domain corpus (no public confirmation of medical data)

Qwen3 (32B)

- **Architecture**: Enhanced transformer decoder with GQA (Grouped Query Attention), SwiGLU, RMSNorm, RoPE
- **Parameters**: 32.8 billion parameters
- **Quantization**: GGUF 4 bit.
- **Context Length**: 128,000 tokens (with YaRN scaling)

Qwen3 (8B)

- **Parameters**: 8.2 billion parameters (6.95B non-embedding)
- **Quantization**: No quantization, BF16 inference.
- **Context Length**: 32,768 tokens (extendable to 131,072 with YaRN)

Qwen3 (4B)

- **Parameters**: 4.8 billion parameters
- **Quantization**: No quantization, BF16 inference.
- **Context Length**: 32,768 tokens (YaRN extendable)

B.2 Gemma Model Family

Gemma 3 (27B)

- **Architecture**: Transformer-based decoder-only with grouped-query attention (GQA) and SigLIP vision encoder; multilingual (140+ languages), multimodal (text + image)
- **Parameters**: 27 billion parameters
- **Quantization**: No quantization, BF16 inference.
- **Context Length**: 128,000 tokens (long context support)

Gemma 3 (4B)

- **Parameters**: 4 billion parameters
- **Quantization**: No quantization, BF16 inference.
- **Context Length**: 128,000 tokens
- **Multimodal Capability**: Supports both text and image inputs
- **Pre-training**: Same distilled training approach as 27B

B.3 MedGemma Model Family

MedGemma (27B)

- **Parameters**: 27 billion parameters
- **Quantization**: No quantization, BF16 inference.
- **Context Length**: 128,000 tokens
- **Medical Pre-training**: Includes medical text (EHRs, question-answer pairs), FHIR-format clinical records, and medical images (e.g. chest X-ray, pathology, ophthalmology, dermatology)
- **Intended Use**: Research and development in healthcare AI; not clinical-grade; part of Health AI Developer Foundations

MedGemma (4B)

- **Parameters**: 4 billion parameters
- **Quantization**: No quantization, BF16 inference.
- **Context Length**: 128,000 tokens
- **Medical Pre-training**: Same data modalities as MedGemma 27B (medical text + images)
- **Intended Use**: Lightweight deployment and health-AI prototyping; not clinical use

C Inference Configuration and Settings

C.1 Hardware Configuration

- **Primary GPUs**: 4× NVIDIA A6000 (48GB VRAM each, 192GB total)
- **CPU**: Intel Xeon Gold 6448Y 64-Core Processor
- **System Memory**: 512GB RAM

C.2 vLLM Configuration

```
vllm_config:
  model_name: "model-specific"
  pipeline_parallel_size: 1
  tensor_parallel_size: 2
  gpu_memory_utilization: 0.95
  dtype: "auto"
  quantization: "gptq"  # for applicable models
  trust_remote_code: true
  seed: 42
```

C.2.1 Inference Parameters

- **Temperature**: 0 (for consistency)
- **Max New Tokens**: 4092
- **Batch Size**:
  - 32B models: 200
  - 8B models: 500
  - 4B models: 500
- **Output Format**: JSON with Pydantic validation

C.2.2 Optimization Techniques

- **Quantization Methods**:
  - GPTQ: 8-bit quantization
  - GGUF: 4-bit quantization
- **Memory Management**:
  - Gradient checkpointing (when applicable)
  - KV-cache optimization
- **Throughput Optimization**:
  - Continuous batching in vLLM
  - PagedAttention for memory efficiency
  - Tensor parallelism for large models