

If Only My CGM Could Speak: A Privacy-Preserving Agent for Question Answering over Continuous Glucose Data

Yanjun Cui¹, Ali Emami², Temiloluwa Prioleau^{2*}, Nikhil Singh^{1*}

¹Dartmouth College ²Emory University

yanjun.cui.gr@dartmouth.edu, ali.emami@emory.edu,
temiloluwa.prioleau@emory.edu, nikhil.u.singh@dartmouth.edu

Abstract

Continuous glucose monitors (CGMs) used in diabetes care collect rich personal health data that could improve day-to-day self-management. However, current patient platforms only offer static summaries which do not support inquisitive user queries. Large language models (LLMs) could enable free-form inquiries about continuous glucose data, but deploying them over sensitive health records raises privacy and accuracy concerns. In this paper, we present **CGM-Agent**, a privacy-preserving framework for question answering over personal glucose data. In our design, the LLM serves purely as a reasoning engine that selects analytical functions. All computation occurs locally, and personal health data never leaves the user’s device. For evaluation, we construct a benchmark of 4,180 questions combining parameterized question templates with real user queries and ground truth derived from deterministic program execution. Evaluating 6 leading LLMs, we find that top models achieve 94% value accuracy on synthetic queries and 88% on ambiguous real-world queries. Errors stem primarily from intent and temporal ambiguity rather than computational failures. Additionally, lightweight models achieve competitive performance in our agent design, suggesting opportunities for low-cost deployment. We release our code and benchmark to support future work on trustworthy health agents.¹

1 Introduction

Modern health sensors generate dense longitudinal records that, in principle, support fine-grained understanding of personal physiology and behavior. In diabetes care, continuous glucose monitors (CGMs) collect data for close monitoring of glucose patterns roughly every five minutes (American

Diabetes Association, 2025a; Dexcom, 2025; Abbott Diabetes Care, 2025). Over months of CGM use, these devices accumulate detailed records of glucose fluctuations, encoding behavioral and physiological signals that could inform lifestyle decisions, medication adjustments, and clinical conversations. However, current user-interaction platforms like Dexcom Clarity and LibreView offer only static, pre-defined summaries (Dexcom, Inc., 2025; Abbott Diabetes Care, 2024), thereby limiting the ability for a user to explore and learn hidden insights from their own data. For example, when a user has targeted questions like, “How does my glucose control on weekdays compare to weekends?”, they have to download their data and manually compute the needed metrics to find answers to their questions. The gap between the *richness* of CGM data and *rigidity* of existing interfaces leaves users without personalized, on-demand insights they need to maximally benefit from their personal health data.

LLMs offer a promising solution for a conversational interface through which users can ask targeted questions in plain language. However, deploying LLMs over sensitive health data raises serious concerns. Cloud-based models would require transmitting personal health data to external servers which introduces significant safety and privacy risks. Moreover, LLMs are prone to numerical errors when processing time-series data, making them less reliable for direct use (Healey and Kohane, 2024; Healey et al., 2025).

In this paper, we present **CGM-Agent**, a modular framework that enables free-form question answering (QA) on personal glucose data while ensuring that personal health data never leaves the user’s device (Figure 1). Our key approach is to decouple *reasoning* from *computation* so that the LLM serves *only* as a reasoning engine that interprets user intent and selects appropriate analytical functions, while all data processing occurs in a lo-

*Co-corresponding authors.

¹Code available at: <https://github.com/yanjunCC/cgm-agent-release>

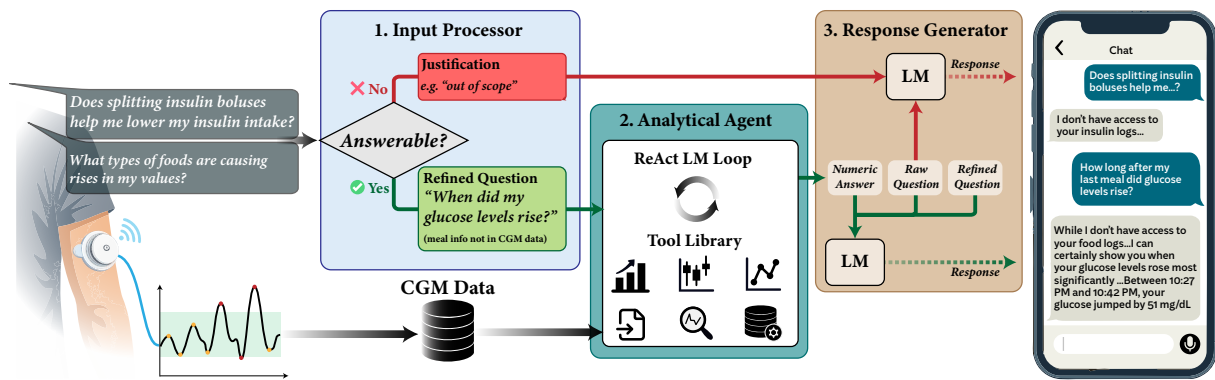


Figure 1: Overview of **CGM-Agent**. The pipeline consists of three layers: (1) the *Input Processor* resolves ambiguous queries, (2) the *Analytical Agent* generates tool calls executed in a local sandbox, and (3) the *Response Generator* synthesizes a natural language answer. Raw CGM data (shown in the bottom sandbox) never crosses the privacy boundary; only function calls and aggregated metrics are exchanged with the LLM.

cal execution sandbox. In our implementation, the LLM receives only aggregated CGM metrics, never raw time-series, eliminating the need to transmit sensitive personal health data to the cloud.

Our main contributions are:

A Grounded QA Benchmark. We construct a dataset of 4,180 questions spanning both synthetic templates grounded in clinical guidelines (Battelino et al., 2019; American Diabetes Association, 2025b) and real user queries sourced via interviews with CGM users. All ground-truth answers are derived from deterministic execution of a custom-designed Python toolkit developed in this work, ensuring reproducibility (§3).

A Privacy-Preserving Agent Architecture. We propose a three-layer pipeline that separates LLM reasoning from data access. Confining raw CGM data to a local sandbox and exposing only function calls to the model supports accurate, personalized responses without compromising user privacy (§4).

Empirical Analysis of Tool-Augmented LLMs. We benchmark 6 leading proprietary and open-weight models through a layer-wise evaluation. Our results show that lightweight models achieve competitive accuracy on well-structured queries, while real-world, often ambiguous user questions expose significant challenges in intent disambiguation and temporal grounding (§5).

2 Related Work

Tool-Augmented LLM Reasoning. Large language models are prone to errors in mathematical reasoning and factual recall, motivating work on tool augmentation (Schick et al., 2023; Yao et al.,

2022). Toolformer (Schick et al., 2023) showed that LLMs can learn to invoke external APIs via self-supervised learning; ReAct (Yao et al., 2022) established interleaving reasoning traces with executable actions for multi-step problem solving. These frameworks have been extended to domain-specific settings: SciAgent (Ma et al., 2024) and ToolHop (Ye et al., 2025) advance tool-augmented reasoning and evaluation in scientific domains, while MMedAgent (Li et al., 2024a) and MeNTi (Zhu et al., 2025) adapt tool calling to clinical decision support. However, these systems typically operate over structured databases or static documents rather than personal longitudinal data, and do not address the privacy constraints inherent in patient-facing applications.

LLMs for Diabetes Management. Prior work has applied LLMs to diabetes education, coaching, and forecasting, yet evaluations have exposed critical limitations (Abbasian et al., 2024; Mitchell et al., 2025; Mamun et al., 2025; Li et al., 2025). For example, Healey and Kohane (2024); Healey et al. (2025) showed that GPT-4 frequently misapplies diabetes-specific calculations and fails at CGM pattern recognition. DM-Bench (Cardei et al., 2025) benchmarks LLMs on patient-facing diabetes decision-making tasks across 15,000 individuals with multimodal behavioral logs. Furthermore, commercial systems such as ChatCGM (ChatCGM, 2024) and Diabetes Cockpit (Cockpit, 2025), which reflect growing market demand, offer LLM-powered glucose interfaces but remain closed-source with undisclosed evaluation procedures. Our work complements these efforts by providing an open benchmark with deterministic,

reproducible ground truth, and a privacy-preserving architecture that keeps raw CGM data local.

Privacy in Personal Health AI. Cloud-based LLMs require transmitting sensitive records to external servers, raising concerns about data security and regulatory compliance (Winslow et al., 2025). CGM data carries significant privacy risks, thus many existing platforms operate under regulatory gaps that leave patients with little to no control over how their data is collected, stored, or shared (Downey et al., 2025; Randine et al., 2025). Some commercial systems mitigate this by sending only statistical summaries to the LLM, however this approach limits the richness of analysis and is proprietary in most cases (Cockpit, 2025). Prior work on privacy-preserving LLM inference formalizes the *Honest-but-Curious* (HbC) threat model for cloud providers (Li et al., 2024b): the provider executes instructions faithfully but may log and exploit transmitted data. Locally deployed LLMs have been proposed as one response to this threat in healthcare (Wiest et al., 2024; Montagna et al., 2025), but typically at the cost of model capability. In contrast, our work proposes an open, modular architecture in which raw CGM data never leaves the local device; the LLM receives only aggregated metrics, whose low temporal resolution provides protection against re-identification (U.S. Department of Health and Human Services, 2012; El Emam et al., 2011).

3 Dataset

We construct a QA benchmark for CGM data comprising 4,180 queries. The dataset combines synthetic templates grounded in clinical guidelines with real user questions collected through formative interviews with current CGM users. All ground-truth answers are derived from deterministic execution of a validated Python toolkit, ensuring mathematical precision and reproducibility.

3.1 CGM Data Source

For an empirical evaluation, we curate CGM records from 19 individuals drawn from two public datasets, covering both Type 1 Diabetes (T1D) and Type 2 Diabetes (T2D). We prioritized subjects with the longest recording duration, thereby selecting 11 T1D subjects from AZT1D (Khamesian et al., 2025) (CGM duration: 45–49 days, 5-minute sampling interval), and 8 T2D subjects from ShanghaiT2DM (Zhao et al., 2023) (CGM duration:

Characteristic	AZT1D	ShanghaiT2DM
Diabetes Type	T1D	T2D
Subjects	11	8
Sampling Interval	5 min	15 min
Duration (days)	45–49	24–179
Missing Days	0	0–153
Missing Samples (%)	1.2–2.4	4.0–86.2
TIR Range (%)	63.2–92.7	41.5–98.7

Table 1: CGM dataset used in this study ($N = 19$).

24–179 days, 15-minute sampling interval). The mapping between anonymous subject identifiers and original dataset identifiers is provided in Appendix J.

This combination provides a realistic testbed as AZT1D offers high-frequency, continuous recordings ideal for verifying the functional correctness, while ShanghaiT2DM introduces challenges of sparser sampling and significant missing data segments (up to 153 missing days for some subjects). Together, the combined dataset reflects the heterogeneity of real-world CGM use, spanning a wide range of Time in Range (TIR, 70–180 mg/dL) values (41.5–98.7%) and missing data rates (Table 1).

3.2 Analytical Toolkit

To ensure clinically meaningful and reproducible evaluation, we developed a Python toolkit that serves two purposes: (1) generating ground-truth answers for benchmark queries, and (2) providing executable functions for the agent at inference time. We organize the toolkit into two tiers.

Standard Glycemic Metrics. Informed by the international consensus recommendation report for CGM data interpretation (Battelino et al., 2019) and the ADA Standards of Care (American Diabetes Association, 2025b), this tier implements standard CGM metrics such as mean glucose, Time in Range (TIR, 70–180 mg/dL), Time Below Range (TBR, <70 mg/dL), and Time Above Range (TAR, >180 mg/dL), while enabling statistical aggregation and comparative analysis across user-defined date ranges. For each specified time window, the system calculates these metrics alongside the CGM wear-time during the associated date range. Following recommended guidelines, results are flagged when CGM wear-time falls below 70% (Battelino et al., 2019), enabling the agent to recognize when a query cannot be reliably answered due to insufficient glucose data.

User-Driven Analytics. Based on findings from our formative CGM user interviews (described below), we extended the toolkit to address user-expressed needs beyond standard metrics. This tier includes: (1) *glucose excursion detection*, identifying rapid glucose rises or falls exceeding a threshold rate (e.g., >2 mg/dL/min), and (2) *trend visualization*, generating plots of glucose trajectories over specified periods. Full function signatures are provided in Appendix A.

3.3 Question Collection

We constructed questions through two complementary approaches: synthetic generation from templates and collection from real users.

Synthetic Questions. We developed a question generation factory to systematically create diverse analytical queries: single-metric queries (“*What was my TIR on [Date A]?*”), and temporal comparisons (“*Compare my average glucose between [Time Period A] and [Time Period B]*”). By programmatically instantiating templates with randomized time windows (e.g., single dates, duration, intraday) and injecting both valid CGM metrics and out-of-scope features (e.g., insulin, sleep), this engine produced 2,470 synthetic QA pairs using CGM data from 19 subjects as described in Section 3.1; thereby ensuring robust test coverage.

Real User Questions. To capture authentic information needs, we conducted semi-structured interviews with three individuals who have diabetes and actively use CGM devices (IRB-approved, Appendix I). Participants were asked to pose questions they would want to ask about their own glucose data, yielding 30 seed questions. While modest in scale, this formative study identified important gaps between clinical metrics and patient priorities. Users expressed strong interest in understanding glucose *fluctuations* and *trends*, not just aggregate statistics. For example, questions like “*What are my typical 24-hour glucose patterns?*” require trend analysis rather than single-value metrics. These findings directly informed our Tier 2 toolkit extensions.

After de-identification and normalization of seed questions, we instantiated those questions through varying dates, time windows, and thresholds using CGM data from all 19 subjects described in Section 3.1 to generate 1,710 user-derived questions.

Handling Unanswerable and Underspecified Questions. Real user questions often reference external context unavailable in CGM data alone (e.g., meals, exercise, insulin doses). We categorize these into two types:

- **Unanswerable:** Questions requiring modalities outside CGM scope are labeled as unanswerable (e.g., “*Does splitting boluses reduce my insulin intake?*”). In this case, the agent should recognize and refuse these gracefully.
- **Proxy Questions:** Questions referencing unobserved behaviors that can be approximated via time windows (windows which could, for example, be obtained from the user via a dialog). For example, “*How does exercise affect my TIR?*” cannot be answered without activity logs, but can be instantiated as “*Compare TIR between 5–7 PM and other times today*” if the user typically exercises in that window. We manually mapped such questions to proxy instantiations, acknowledging that the agent provides *approximate* analyses that may prompt user reflection rather than definitive answers.

The final benchmark dataset includes 513 unanswerable questions and 399 proxy questions to support evaluation of the agent’s robustness to realistic edge cases. Detailed user questions and instantiation mappings are provided in Appendix F. Table 2 summarizes the final dataset composition.

Source	Category	Count
Synthetic	Template-generated	2,470
User-derived	Directly answerable	798
	Proxy (instantiated)	399
	Unanswerable	513
Total		4,180

Table 2: Composition of the CGM-QA benchmark. Proxy questions are user queries referencing unobserved behaviors, mapped to answerable approximations.

3.4 Ground Truth Construction

A distinguishing feature of our benchmark is that ground truth is defined as the *output of deterministic program execution*, not human annotation. For each question, we specify the corresponding reference procedure of function call(s) and their associated parameters. The ground-truth answer is the numerical result of executing these functions on the subject’s CGM data.

This design ensures: (1) **mathematical preci-**

sion—answers are computed, not estimated; (2) **reproducibility**—any system can verify answers by re-running the toolkit; and (3) **scalability**—similar new questions can be added with minimal manual annotation effort. This methodology establishes a rigorous standard for evaluating tool use in healthcare, where numerical accuracy is paramount.

4 Agent Architecture

Figure 1 illustrates the overall architecture of CGM-Agent. In this section, we describe each component in detail. We begin by stating our core design principles, then walk through the three layers of the pipeline: Input Processor, Analytical Agent, and Response Generator.

4.1 Design Principles

Our architecture is guided by two principles:

Separation of Reasoning and Computation. The LLM serves purely as a reasoning engine: it interprets user intent, resolves ambiguities, selects appropriate analytical functions, and synthesizes natural language responses. All numerical computation over CGM data occurs in a local execution sandbox. To maximally preserve user privacy, the LLM never receives raw glucose readings in its context window, it only aggregates the results from function calls.

Modularity for Evaluation. Each layer performs a distinct function. This separation enables targeted evaluation: we can measure whether the Input Processor correctly resolves temporal references, whether the Analytical Agent selects appropriate tools, and whether the Response Generator produces accurate and readable output. Thus, errors can be localized to specific components.

4.2 Pipeline Overview

Given a user query q , the pipeline proceeds as follows. The Input Processor \mathcal{P} validates feasibility, resolves temporal references, and maps vague terms to precise system features, producing a refined query q' . The Analytical Agent \mathcal{T} decomposes q' into function calls and executes them locally with user’s private CGM data D , producing numerical results R . The Response Generator \mathcal{G} synthesizes a natural language response y :

$$q \xrightarrow{\mathcal{P}} q' \xrightarrow[\text{Tools}(D)]{\mathcal{T}} R \xrightarrow{\mathcal{G}(q,q',R)} y \quad (1)$$

If the Input Processor determines that q is unanswerable (e.g., requires insulin logs), it generates an explanation e and the pipeline bypasses the Analytical Agent:

$$q \xrightarrow{\mathcal{P}} e \xrightarrow{\mathcal{G}(q,e)} y \quad (2)$$

Appendix B illustrates these two paths with representative examples.

4.3 Layer 1: Input Processor

The Input Processor serves as a gatekeeper, performing two functions before any data access.

Feasibility Classification. As mentioned previously, the LLM evaluates whether the query can be answered given available modalities (CGM only).

Temporal and Feature Instantiation. For answerable queries, the LLM grounds user query q into a fully specified format q' . This process involves resolving natural language time expressions (e.g., “last weekend”) into precise datetime ranges, and normalizes vague descriptors into fully specified variables, transforming inputs like “how’s my sugar” into precise requests for “mean glucose” and “data sufficiency” (CGM wear-time) compatible with the analytical toolkit.

Example.

Input: “What’s my TIR last weekend?”
Context: Current date is 2024-01-10 (Wednesday)
Output: is_answerable: True,
 refined_question: What’s my TIR and CGM wear-time on 2024-01-06 and 2024-01-07?

4.4 Layer 2: Analytical Agent

The Analytical Agent translates the refined query q' into executable function calls and retrieves numerical results. Crucially, the LLM interacts with CGM data only through predefined tool interfaces, never directly accessing raw glucose readings.

Planning and Decomposition. Some queries require multiple analytical steps. For example, “What’s my TIR over this week and last week?” involves computing TIR separately for two time windows. The LLM decomposes such queries into a sequence of atomic sub-tasks, each mapped to a single function call.

Tool Execution. For each sub-task, the LLM generates a function call specifying the operation and parameters (metric type, date range,

reference thresholds). This call is dispatched to the local execution sandbox, where the Python toolkit (Section 3.2) processes the user’s CGM data and returns aggregated results. The LLM receives only these aggregated outputs (e.g., {avg_TIR_sufficient_wear_time: 72%, days_sufficient_wear_time: 4}), never the underlying time series.

Modularity and Extensibility. Given that the Analytical Agent relies entirely on deterministic tool calls rather than end-to-end parametric modeling, new data modalities can be integrated by registering a corresponding local executor, without modifying the overarching framework.

Example. The following example illustrates the decomposition and execution process. Note that the toolkit computes TIR on a per-day basis and then aggregates across days, reporting separate statistics for days with sufficient CGM wear time ($\geq 70\%$) and for all recorded days.

Input: q' = What’s my TIR over this week and last week?

Decomposition:

Task A: What’s my TIR and CGM wear time on (dates=this_week)?

Task B: What’s my TIR and CGM wear time on (dates=last_week)?

Execution Results:

$R_A = \{days_sufficient_wear_time = 4,$
 $avg_TIR_sufficient_wear_time: 72\%,$
 $days_all = 6, avg_TIR_all: 76\%$

$R_B = \{days_sufficient_wear_time = 5,$
 $avg_TIR_sufficient_wear_time: 78\%,$
 $days_all = 7, avg_TIR_all: 86\%$

4.5 Layer 3: Response Generator

The final layer synthesizes the numerical results into a natural language response for the user.

Answerable Queries. For queries with valid results, the generator produces a response that: (1) directly answers the user’s question with specific numbers, (2) explicitly states the time period analyzed and metrics instantiated to verify alignment with user intent, and (3) provides brief clinical context where appropriate (e.g., comparing TIR to recommended targets).

Unanswerable Queries. For queries flagged by Layer 1, the generator produces a response that: (1) acknowledges the user’s question, (2) explains specifically why it cannot be answered with CGM data alone, and (3) suggests alternative approaches (e.g., consulting a healthcare provider or integrating additional data sources).

Example (Answerable).

Input: $q =$ “How long after eating do my glucose levels rise?”

Results: $R = "(2021-08-29\ 09:37:00,$
 $2021-08-29\ 09:52:00)":$

"magnitude": 30.6, "speed": 2.04

Response: “Based on your data from August 29th, a significant rise began at 9:37 AM, where your levels increased by 30.6 mg/dL in 15 minutes. Typically, glucose rises within 15–60 minutes after eating, depending on the food. Your levels were rising at about 2.0 mg/dL per minute. Tracking these patterns can help you understand how different meals affect you!”

Example (Unanswerable).

Input: “Does splitting insulin boluses help reduce my total daily insulin dose?”

Response: “I can’t determine if splitting boluses reduces your total daily insulin, as I don’t have access to your insulin logs. To answer this, we’d need to compare total units across different periods. I recommend reviewing your insulin records with your healthcare provider.”

All system prompts are provided in Appendix G.1.

5 Experiments

We evaluate CGM-Agent along two dimensions: (1) tool-use accuracy on synthetic queries with fully specified intent, and (2) end-to-end performance on ambiguous real-world user queries.

5.1 Experimental Setup

Models. We evaluate a total of 6 proprietary and open-weight LLMs as backbones for our agent:

- Proprietary Models (Temp=1.0): GPT-5.2, GPT-5-Mini, Gemini 3.0 Pro and Flash.
- Open-Weight Models (Temp=0.6): Llama-4 Maverick (17B), Nemotron-Nano (9B).

All models are accessed via their respective APIs. We use the same system prompts across all models for consistency.

Evaluation Metrics. For the Analytical Agent (Layer 2), we report:

- **Precision / Recall / F1:** Whether the agent selects the correct analytical function(s) and parameters. Precision measures the fraction of predicted function calls that match ground truth; Recall measures the fraction of ground-truth calls that were predicted.
- **Value Accuracy:** Whether the final numerical output matches the ground-truth value within a tolerance of $\pm 1\%$.

For the Input Processor (Layer 1), we report classification accuracy, precision, recall, and F1 for the feasibility decision (answerable vs. unanswerable).

Automated Evaluation. Given the scale of evaluation (4,180 queries), we use Gemini 3.0 Pro as an automated judge to assess function-call correctness and value matching. To validate this approach, we manually reviewed 40 randomly selected samples. The evaluation showed high alignment with human judgment, yielding a Mean Absolute Error (MAE) of ≈ 0.05 for precision and ≈ 0.04 for recall. Evaluation prompts are provided in Appendix G.2.

5.2 Results on Synthetic Queries

The synthetic dataset ($N = 2,470$) serves as a controlled evaluation of the Analytical Agent (Layer 2), where temporal references and user intent are fully specified. This isolates tool-use capability from the ambiguity challenges present in real-world queries.

Table 3 presents the results. Key findings:

- **Proprietary models achieve strong tool-use accuracy.** All proprietary models achieved 0.8+ F1 and three models have 0.94 Value Accuracy, demonstrating reliable function selection and parameter extraction.
- **Smaller models are competitive.** GPT-5-Mini and Gemini 3.0 Flash achieve Value Accuracy comparable to their larger counterparts (≈ 0.94), suggesting that tool-call does not require the largest-scale frontier models. This is encouraging for lower-cost deployments.
- **Open-weight models lag but remain viable.** Llama-4-17B achieves 0.75 Value Accuracy, trailing proprietary models but demonstrating that local-only deployments may be feasible with some performance trade-off.

Model	Prec	Rec	F1	Val Acc
<i>Proprietary</i>				
GPT-5.2	0.84	0.89	0.86	0.81
GPT-5-Mini	0.75	0.87	0.80	0.94
Gemini 3.0 Pro	0.92	0.71	0.80	0.94
Gemini 3.0 Flash	0.89	0.74	0.81	0.94
<i>Open-weight</i>				
Llama-4-17B	0.73	0.74	0.73	0.75
Nemotron-Nano-9B	0.45	0.61	0.52	0.67

Table 3: Layer 2 (Analytical Agent) performance on synthetic queries ($N = 2,470$). Precision, Recall, and F1 measure function-call correctness; Value Accuracy measures numerical output correctness ($\pm 1\%$ tolerance). Results shown are micro-averaged.

5.3 Results on Real-World Queries

Real-world queries ($N = 1,710$) introduce natural language ambiguity and require the full three-layer pipeline. We evaluate each layer separately to localize performance bottlenecks.

Layer 1: Input Processor. Table 4 reports classification performance on the feasibility decision. All models achieve high recall (> 0.95), indicating they rarely reject valid queries. Gemini 3.0 Pro achieves the best overall F1 (0.97), making it an effective gatekeeper that minimizes both false rejections and false acceptances.

Model	Acc	Prec	Rec	F1
GPT-5.2	0.92	0.90	1.00	0.94
GPT-5-Mini	0.91	0.90	0.97	0.94
Gemini 3.0 Pro	0.96	0.95	1.00	0.97
Gemini 3.0 Flash	0.95	0.93	1.00	0.96
Llama-4-17B	0.86	0.87	0.95	0.91

Table 4: Layer 1 (Input Processor) performance on feasibility classification ($N = 1,710$). Positive class is “answerable.” Best results in **bold**.

Layer 2: Analytical Agent. To isolate analytical reasoning from upstream errors, we evaluate Layer 2 using only the queries correctly classified as answerable by the best Layer 1 model (Gemini 3.0 Pro), yielding $N = 1,197$ samples.

Table 5 presents the results. Performance drops substantially compared to synthetic queries: F1 falls from ≈ 0.80 to ≈ 0.65 , and Precision drops from ≈ 0.90 to ≈ 0.65 . This gap reflects the challenge of mapping natural language to precise function calls when user intent is ambiguous.

However, Value Accuracy remains relatively high (> 0.82 for top models), indicating that when the agent correctly identifies the intended analysis, execution is reliable. This suggests the primary bottleneck is intent disambiguation, not computation.

Model	Prec	Rec	F1	Val Acc
GPT-5.2	0.65	0.76	0.70	0.82
GPT-5-Mini	0.56	0.68	0.62	0.86
Gemini 3.0 Pro	0.65	0.62	0.64	0.88
Gemini 3.0 Flash	0.65	0.58	0.61	0.86
Llama-4-17B	0.44	0.66	0.53	0.44

Table 5: Layer 2 (Analytical Agent) performance on real-world queries ($N = 1,197$). Evaluation uses answerable queries from the best Layer 1 model. Results shown are micro-averaged.

Metric	Value	Interpretation
Avg. Length	108 words	Concise paragraph
Flesch Reading Ease	60.3	Standard English
Flesch-Kincaid Grade	9.7	10th grade level

Table 6: Readability analysis of generated responses ($N = 1,710$). Scores indicate accessible, non-technical language appropriate for most CGM users.

Framework Extensibility. To validate that the Analytical Agent generalizes beyond CGM data, we conducted a pilot study integrating insulin and carbohydrate logs as additional modalities (Appendix D). Evaluated on 132 new queries with the AZT1D dataset spanning single-feature extraction and multi-feature interactions using Gemini 3.0 Flash, the agent achieves F1 of 0.987 and Value Accuracy of 0.998, demonstrating that the tool-call interface supports new data modalities with minimal overhead.

5.4 Response Quality

Beyond numerical accuracy, we assess the readability of responses generated by Gemini 3.0 Flash given the relatively lower complexity of this task (Layer 3).

Qualitative Examples. Representative outputs for both answerable and unanswerable queries in §4.5 demonstrate the agent’s ability to (1) transform indirect questions into data-driven insights, and (2) provide informative refusals that explain limitations and suggest alternatives.

Readability Analysis. We assessed linguistic quality of all generated responses ($N = 1,710$) using standard readability metrics (Kincaid et al., 1975). Table 6 summarizes the results.

The Flesch-Kincaid Grade of 9.7 corresponds to a 10th-grade reading level. Although this exceeds the general AMA recommendation of 6th-8th grade (Eltorai et al., 2014), the score is artificially inflated by essential multi-syllabic terms (e.g., *hypoglycemia*), while the concise length (avg. 108 words) ensures readability on mobile interfaces.

5.5 Error Analysis

Quantitative metrics on real-world queries are lower than on synthetic ones, but manual inspection reveals that standard metrics may *underestimate* the agent’s true utility due to inherent ambiguity. We identify two primary sources of valid divergence.

Intent Ambiguity. Real-world queries often underspecify the desired analysis. A question like “*How was my blood glucose yesterday?*” can be validly answered in multiple ways:

- **Aggregate metrics:** Report TIR, mean glucose, or variability for the day.
- **Event detection:** Identify specific excursions (e.g., “You had a sharp drop of 45 mg/dL at 3 PM”).
- **Visualization:** Generate a trend plot showing the full day’s trajectory.

Our ground truth captures only one interpretation per question, established through consensus discussions among the authors, including diabetes domain experts, to standardize ambiguous intents. If the agent produces a valid alternative (e.g., event detection instead of aggregate metrics), it is penalized despite providing a potentially useful answer for some users depending on their (unobserved) true informational needs. This inflates Precision/Recall errors.

Temporal Ambiguity. Implicit time references like “*this afternoon*” lack precise boundaries. If the ground truth defines “afternoon” as 12:00–17:00 but the agent interprets it as 13:00–18:00, numerical outputs will differ even though both interpretations are reasonable. This inflates Value Accuracy errors despite correct reasoning.

Robustness to Glucose Variability. We also examine whether performance differences across subjects are driven by underlying glucose variability rather than query difficulty. To investigate this, we compute Pearson correlations between each subject’s TIR and their F1 and Value Accuracy scores under Gemini 3.0 Pro. Neither correlation is statistically significant (see Appendix E), suggesting that measured errors may primarily reflect linguistic ambiguity rather than systematic sensitivity to clinical data characteristics.

Implications. These findings motivate Layer 3’s response explicitly stating the time period analyzed (e.g., “*Based on your data from 12 PM to 5 PM...*”). This transparency allows users to verify whether the agent’s interpretation matches their intent, mitigating the practical impact of such ambiguities.

5.6 Ablation Study

To quantify the contribution of the Layer 1 Input Processor, we conducted an ablation study by removing this module.

Setup. We constructed an experiment where raw user queries were fed directly into the Analytical Agent (Layer 2) alongside the timestamp context based on the CGM data used (e.g., “*Today is 2024-03-15. User Question: ...*”). We evaluated this baseline on the same subset of answerable questions ($N \approx 1197$) used in the main experiment to ensure a fair comparison.

Results. Table 7 isolates the contribution of the *Input Processor* (Layer 1) using Gemini 3.0 Pro. The results demonstrate that the full pipeline substantially outperforms the direct-to-agent baseline across all metrics.

- **Impact on Feature Instantiation (F1 Score):** The significant decline in F1 ($\Delta = 0.19$) reveals that the primary bottleneck for raw LLMs is correctly **identifying and aligning** the required feature parameters. Without the Input Processor, the model frequently omits necessary arguments or hallucinates invalid feature keys.
- **Latent Tool-Use Capability (Value Acc.):** Value Accuracy remains relatively high (0.82). This suggests that the failure mode is parsing question intent, not reasoning. Once the correct features are identified, the agent is highly effective at deriving the correct answer. This validates the practical utility of the tool-use paradigm, provided that the input arguments are rigorously grounded.

These findings underscore our design choice: decoupling semantic refinement from computational execution is essential for handling real-world ambiguity. As shown in Appendix C, this design naturally extends to **interactive clarification** on Input Processor, yielding further performance gains on ambiguous queries.

Configuration	F1 Score	Value Acc.
Ours (Full Pipeline)	0.64	0.88
w/o Input Processor	0.45	0.82
Performance Drop (Δ)	-0.19	-0.06

Table 7: Ablation results on the answerable user query subset ($N = 1,197$) using Gemini 3.0 Pro. Removing the Input Processor leads to significant degradation, particularly in F1 Score, highlighting the necessity of explicit query instantiation.

6 Conclusion

Our results in this work suggest that privacy and utility in patient-facing health agents need not be inherently in conflict. CGM-Agent serves as a real-world case study in diabetes care, showing that a carefully designed interface between natural language queries and clinically grounded computation can support meaningful patient interactions without exposing raw data. Across both our scalably parameterized questions and ambiguous user-grounded questions, existing LLMs perform reliably when operating this interface. We release CGM-Agent and our benchmark as a foundation for future work, and invite the community to join in building systems to better support patient-facing health AI.

Limitations

While our modular framework supports integration of multiple data modalities, real-world diabetes management involves a broader range of signals not yet validated in our system, such as physical activity and sleep. Extending the framework to incorporate these modalities remains an important direction for future work, and will depend on the availability of large-scale, naturalistic multimodal datasets.

The benchmark ground truth reflects our best assessment of the most valid reference interpretation per question, yet some natural language queries admit multiple defensible analyses given the same data. Our error analysis suggests this ambiguity accounts for a meaningful portion of measured errors, and developing useful frameworks for rewarding semantically defensible alternatives could be a fruitful pathway to metrics that account for this.

Given the specialized population (CGM users) and challenge of obtaining realistic questions in-situ, we opted to conduct in-depth interviews with a modest number of participants rather than quick surveys with a broader set. We chose this approach to surface qualitatively rich, ecologically grounded queries, which we believe is a strength of this work. The interview protocol is lightweight and can be repeated to expand the dataset in future work.

Ethical Considerations

Clinical Safety and Scope. Our system focuses on retrospective analysis and reflection rather than prospective recommendations. Given the clinical sensitivity of diabetes management and associated

ethical considerations, we deliberately avoid generating behavioral suggestions (e.g., insulin dosing adjustments) that could pose safety risks without professional oversight. Extending the framework to support safe, clinically validated recommendations remains an important direction for future work.

Data Privacy and Participant Protection. Our user study protocol prioritized data minimization and was approved by Committee for Protection of Human Subjects at Dartmouth College. We did not collect raw sensor data from participants; data collection was restricted exclusively to written, free-form textual questions. To mitigate the risk of disclosing Personally Identifiable Information (PII), all collected queries were manually reviewed and de-identified prior to inclusion in the dataset. All data was stored on secure, HIPAA-compliant servers in accordance with our approved protocol.

References

- Mahyar Abbasian, Zhongqi Yang, Elahe Khatibi, Pengfei Zhang, Nitish Nagesh, Iman Azimi, Ramesh Jain, and Amir M Rahmani. 2024. Knowledge-infused llm-powered conversational health agent: A case study for diabetes patients. In *2024 46th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1–4. IEEE.
- Abbott Diabetes Care. 2024. Libreview freestyle blood glucose monitoring report set overview. https://www.libreview.com/files/documents/en-US/ReportTour_2024-08-19.pdf. Accessed: 2025-11-09.
- Abbott Diabetes Care. 2025. Freestyle libre 3 system. <https://www.freestyleprovider.abbott/us-en/freestyle-libre-3.html>. Accessed: 2025-11-09.
- American Diabetes Association. 2025a. 2. diagnosis and classification of diabetes: standards of care in diabetes—2025. *Diabetes Care*, 48(Supplement_1):S27–S49.
- American Diabetes Association. 2025b. 6. glycemic goals and hypoglycemia: Standards of care in diabetes—2025. *Diabetes Care*, 48(Supplement_1):S128–S145.
- Tadej Battelino, Thomas Danne, Richard M Bergental, Stephanie A Amiel, Roy Beck, Torben Biester, Emanuele Bosi, Bruce A Buckingham, William T Cefalu, Kelly L Close, and 1 others. 2019. Clinical targets for continuous glucose monitoring data interpretation: recommendations from the international consensus on time in range. *Diabetes care*, 42(8):1593–1603.
- Maria Ana Cardei, Josephine Lamp, Mark Derdzinski, and Karan Bhatia. 2025. Dm-bench: Benchmarking llms for personalized decision making in diabetes management. *arXiv preprint arXiv:2510.00038*.
- ChatCGM. 2024. Privacy policy. <https://www.chatcgm.com/privacy/>. Last updated 15 May 2024; accessed 22 Nov 2025.
- Cockpit. 2025. Cockpit — diabetes cockpit app. <https://cockpit-app.com/>. Accessed: 2025-11-22.
- Inc. Dexcom. 2025. Dexcom g6 continuous glucose monitoring (cgm) system — how it works. <https://www.dexcom.com/en-us/g6/how-it-works>. Accessed: 2025-11-09. Sends glucose numbers every 5 minutes.
- Dexcom, Inc. 2025. [Dexcom official website](https://www.dexcom.com). Accessed: 2025-11-09.
- Laura Downey, Hanne Ballhausen, Louise Hatherall, Holly Hayes, and Shane O’Donnell. 2025. The urgent need for a diabetes data rights charter. *The Lancet Diabetes & Endocrinology*, 13(12):999.
- Khaled El Emam, Elizabeth Jonker, Luk Arbuckle, and Bradley Malin. 2011. A systematic review of re-identification attacks on health data. *PloS one*, 6(12):e28071.
- Adam EM Eltorai, Soha Ghanian, Charles A Adams Jr, Christopher T Born, and Alan H Daniels. 2014. Readability of patient education materials on the american association for surgery of trauma website. *Archives of trauma research*, 3(2):e18161.
- Elizabeth Healey and Isaac Kohane. 2024. Llm-cgm: A benchmark for large language model-enabled querying of continuous glucose monitoring data for conversational diabetes management. In *Biocomputing 2025: Proceedings of the Pacific Symposium*, pages 82–93. World Scientific.
- Elizabeth Healey, Amelia Li Min Tan, Kristen L Flint, Jessica L Ruiz, and Isaac Kohane. 2025. A case study on using a large language model to analyze continuous glucose monitoring data. *Scientific Reports*, 15(1):1143.
- Saman Khamesian, Asiful Arefeen, Maria Adela Grando, Bithika M Thompson, and Hassan Ghasemzadeh. 2025. Type 1 diabetes management using glimmer: Glucose level indicator model with modified error rate. *arXiv preprint arXiv:2502.14183*.
- J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report.
- Binxu Li, Tiankai Yan, Yuanting Pan, Jie Luo, Ruiyang Ji, Jiayuan Ding, Zhe Xu, Shilong Liu, Haoyu Dong,

- Zihao Lin, and 1 others. 2024a. Mmedagent: Learning to use medical tools with multi-modal agent. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8745–8760.
- Caihua Li, In Gim, and Lin Zhong. 2024b. Confidential prompting: Privacy-preserving llm inference on cloud. *arXiv preprint arXiv:2409.19134*.
- Qingrui Li, Kapileshwor Ray Amat, and Juan Li. 2025. Llm-powered personalized glucose prediction in type 1 diabetes. *Computational and Structural Biotechnology Reports*, page 100068.
- Yubo Ma, Zhibin Gou, Junheng Hao, Ruochen Xu, Shuohang Wang, Liangming Pan, Yujiu Yang, Yixin Cao, and Aixin Sun. 2024. Sciagent: Tool-augmented language models for scientific reasoning. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 15701–15736.
- Abdullah Mamun, Asiful Arefeen, Susan B Racette, Dorothy D Sears, Corrie M Whisner, Matthew P Buman, and Hassan Ghasemzadeh. 2025. Llm-powered prediction of hyperglycemia and discovery of behavioral treatment pathways from wearables and diet. *Sensors*, 25(17):5372.
- Elliot G Mitchell, Pooja Desai, Arlene Smaldone, Andrea Cassells, Jonathan N Tobin, David Albers, Matthew Levine, and Lena Mamykina. 2025. T2 coach: A qualitative study of an automated health coach for diabetes self-management. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–17.
- Sara Montagna, Stefano Ferretti, Lorenz Cuno Klopfenstein, Michelangelo Ungolo, Martino Francesco Pengo, Gianluca Aguzzi, and Matteo Magnini. 2025. Privacy-preserving llm-based chatbots for hypertensive patient self-management. *Smart Health*, 36:100552.
- Pietro Randine, Matthias Pocs, John Graham Cooper, Dimitrios Tsolovos, Miroslav Muzny, Rouven Besters, and Eirik Årsand. 2025. Privacy concerns related to data sharing for european diabetes devices. *Journal of Diabetes Science and Technology*, 19(3):611–619.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551.
- U.S. Department of Health and Human Services. 2012. Guidance regarding methods for de-identification of protected health information in accordance with the health insurance portability and accountability act (hipaa) privacy rule. *Human Health Services: Washington, DC, USA*.
- Isabella Catharina Wiest, Dyke Ferber, Jiefu Zhu, Marko van Treeck, Sonja K Meyer, Radhika Juglan, Zunamys I Carrero, Daniel Paech, Jens Kleesiek, Matthias P Ebert, and 1 others. 2024. Privacy-preserving large language models for structured medical information retrieval. *NPJ Digital Medicine*, 7(1):257.
- Brent Winslow, Jacqueline Shreibati, Javier Perez, Hao-Wei Su, Nichole Young-Lin, Nova Hammerquist, Daniel McDuff, Jason Guss, Jenny Vafeiadou, Nick Cain, and 1 others. 2025. A principle-based framework for the development and evaluation of large language models for health and wellness. *arXiv preprint arXiv:2512.08936*.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.
- Junjie Ye, Zhengyin Du, Xuesong Yao, Weijian Lin, Yufei Xu, Zehui Chen, Zaiyuan Wang, Sining Zhu, Zhiheng Xi, Siyu Yuan, and 1 others. 2025. Toolhop: A query-driven benchmark for evaluating large language models in multi-hop tool use. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2995–3021.
- Qinpei Zhao, Jinhao Zhu, Xuan Shen, Chuwen Lin, Yinjia Zhang, Yuxiang Liang, Baige Cao, Jiangfeng Li, Xiang Liu, Weixiong Rao, and 1 others. 2023. Chinese diabetes datasets for data-driven machine learning. *Scientific Data*, 10(1):35.
- Yakun Zhu, Shaohang Wei, Xu Wang, Kui Xue, Shaoting Zhang, and Xiaofan Zhang. 2025. Menti: Bridging medical calculator and llm agent with nested tool calling. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5097–5116.

A Appendix: CGM Analysis Toolkit

Table 8 lists the complete set of deterministic functions implemented in our toolkit. These functions are categorized into three groups:

- **Data Processing & CGM wear-time:** Handles data ingestion, filtering, and quality checks.
- **Daily Metrics Extraction:** Calculates clinical metrics for individual days (e.g., TIR, GMI).
- **Long-term Aggregation & Analysis:** Performs statistical operations across multiple days (e.g., averages, trends, comparisons).

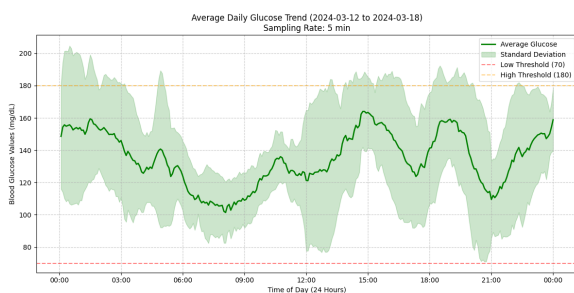


Figure 2: **Average Daily Glucose Profile.** Generated by the `plot_daily_trends` function, this visualization aggregates 7 days of data to display the mean glucose trajectory (solid green line) and glucose standard deviation (shaded area, ± 1 SD) relative to standard clinical target boundaries (70–180 mg/dL).

B Appendix: Control flow for two representative query types.

Figure 3 shows that a query that can be approximately answered via proxy instantiation, and an unanswerable query requiring graceful refusal.

C Interactive Clarification Study

C.1 Setup

The Input Processor (Layer 1) can ask a clarification question upon detecting ambiguity, rather than defaulting to a fixed interpretation. This requires no architectural changes—it is a natural extension of the existing query refinement logic in Layer 1.

We evaluated this *Interactive* clarify-then-answer setting against the *Original* single-turn setting on 63 ambiguous real-user questions per subject across four subjects (2 T1D, 2 T2D) using Gemini 3 Pro. A simulated user agent, with access to the underlying data templates, provides grounded responses to clarification questions.

C.2 Results

A single clarification round improves performance for 3 out of 4 subjects in F1, and all 4 subjects in Value Accuracy, yielding average absolute gains of +5.5% F1 and +4.3% Value Accuracy. For P12, the F1 score increased significantly from 0.60 to 0.75, which suggests that the agent effectively synthesizes user-provided constraints to refine its analysis. For P11, clarification improved accuracy but led to more conservative tool selection (higher precision, lower recall), suggesting the clarification sometimes narrowed scope excessively.

C.3 Clarification Example

The following exchange illustrates how the Input Processor resolves temporal ambiguity through clarification:

User: “What is the standard deviation of my blood glucose around dawn?”

Agent: “Please specify the time range you consider ‘dawn’ and the dates you would like analyzed.”

User: “4 AM to 6 AM on 2024-02-29.”

Refined query with interaction: “What are my standard deviation of blood glucose and CGM adherence for the time range 2024-02-29 04:00 to 2024-02-29 06:00?”

Refined query without interaction: “What is the standard deviation of my blood glucose restricted to the time range 04:00 to 07:00 (dawn) over the following dates: 2024-01-01 to 2024-02-29? Also provide the CGM adherence for these dates.”

D Appendix: Multi-modal Extensibility Pilot

To validate the modular design of the Analytical Agent, we integrated insulin and carbohydrate logs as additional data sources alongside CGM data, using the T1D cohort (AZT1D dataset). New local executors were registered for each modality without modifying the core framework.

We evaluated the agent on 132 new queries in two categories:

- **Single-Feature Extraction** (66 queries): Calculating specific daily totals for insulin or carbohydrate intake.

Table 8: Complete definition of Python tools available to the Agentic Framework.

Category	Function Name	Description & Clinical Utility
Data Processing & CGM Weartime	filter_cgm_csv	Filters raw CGM data based on a user-specified date list or time window (e.g., "last 3 days" or "6AM-12PM"). Handles chronological sorting.
	estimate_cgm_sampling_rate	Infers the device sampling rate (e.g., 5 min vs. 15 min) to accurately calculate expected readings for CGM wear-time checks.
	find_adherence	Calculates the percentage of active wear time for full days or specific time windows to determine data sufficiency.
Daily Clinical Metrics	find_BG_time_range	Computes percentage and duration of Time in Range (TIR), Time below Range (TBR), and Time above Range (TAR).
	find_avg_std_gv_BG	Calculates Mean blood glucose, Standard Deviation, Glycemic Variability (CV), estimates A1c and Glucose Management Indicator (GMI) based on mean glucose.
	find_BG_min_max	Identifies the absolute minimum and maximum glucose values for specified dates to detect extreme outliers.
	find_hypo_hyper_events	Counts discrete events of Hypoglycemia (< 70 mg/dL for 15+ min) and Hyperglycemia (> 180 mg/dL for 15+ min).
	extract_features_json	Pipeline wrapper that executes all daily metric functions and aggregates results into a structured JSON for downstream analysis.
Long-term Aggregation & Analysis	get_average	Computes the average of a feature (e.g., TIR) across multiple days. Crucially, it returns two values: one for "All Data" and one specifically for "Good Wear-time" ($\geq 70\%$) days.
	count_satisfied_condition	Counts how many days meet a specific criterion (e.g., "Days with no hypoglycemia events").
	feature_range	Finds the global minimum and maximum of a specific feature (e.g., "Which day had the lowest TIR?") across a long period.
	compute_difference_ratio	Compares two different time periods (e.g., "Last week vs. This week") and calculates the absolute difference and ratio for any feature.
	calculate_blood_glucose_excursion	Detects rapid glycemic excursions (spikes/drops).
	plot_daily_trends	Generates and saves a 24-hour aggregate plot (Average Daily Profile) to visualize daily trends and patterns. see Figure 2)

- **Multi-Feature Interaction** (66 queries): Analyzing glucose fluctuations within 1–3 hours following recent insulin doses or carbohydrate intake.

The system achieves near-ceiling performance on both query types, confirming that the tool-call interface generalizes cleanly to new modalities. Full integration of physical activity, sleep, and meal logs remains an important direction for future work, and will depend on the availability of large-scale naturalistic multimodal datasets.

E Appendix: Robustness to Glucose Variability

To assess whether agent performance is confounded by underlying glucose variability, we compute Pearson correlations between each subject's TIR and two performance metrics (F1 and Value Accuracy) using Gemini 3.0 Pro results. Results are reported overall and by diabetes type in Table 11.

No correlation reaches statistical significance across any group. The moderate coefficient in the T2D subgroup ($r = 0.622$) does not survive the significance threshold ($p = 0.100 > 0.05$), likely reflecting the small subgroup size ($N = 8$) rather than a genuine performance dependency. These

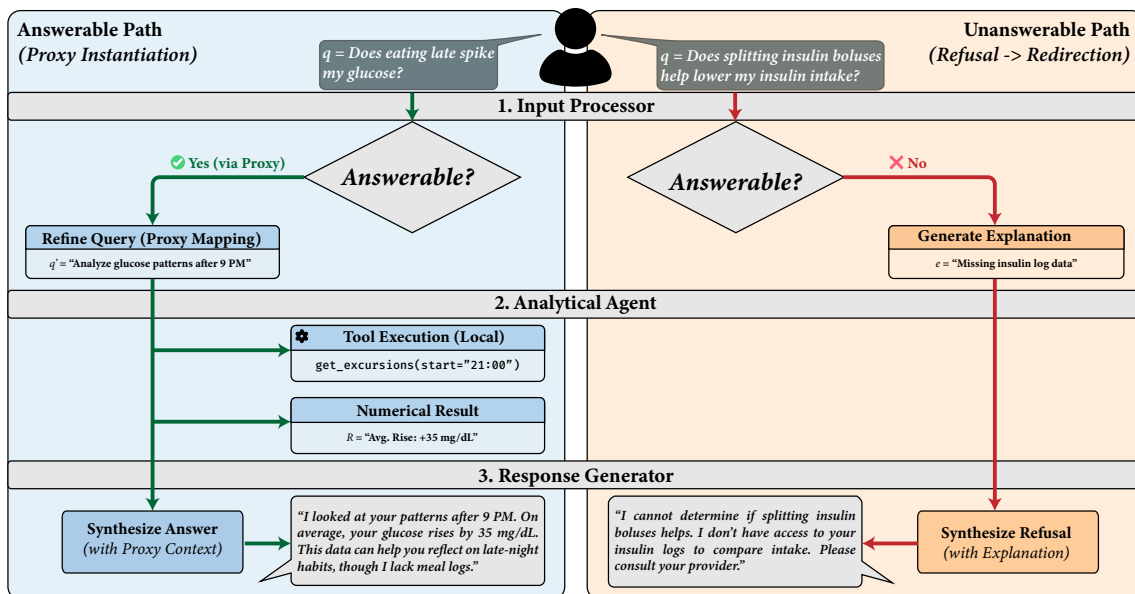


Figure 3: Control flow for two representative query types. **Left (blue)**: A query referencing unobserved behavior (late-night eating) is mapped to a proxy instantiation and answered with a reflection-prompting response. **Right (orange)**: A query requiring external data (insulin logs) is classified as unanswerable; Layer 2 is bypassed and the system generates an explanation.

Subject	Setting	Prec	Rec	F1	Val Acc
P1	Original	0.65	0.59	0.62	0.91
	Interactive	0.67	0.77	0.71	0.93
P11	Original	0.60	0.57	0.58	0.79
	Interactive	0.64	0.42	0.51	0.87
P12	Original	0.67	0.54	0.60	0.82
	Interactive	0.73	0.77	0.75	0.86
P19	Original	0.63	0.59	0.61	0.88
	Interactive	0.63	0.69	0.66	0.91
Avg	Original	0.64	0.57	0.60	0.85
	Interactive	0.67	0.66	0.66	0.89

Table 9: Interactive clarification results on 63 ambiguous real-user questions per subject (Gemini 3 Pro). **Bold** marks the higher-performing setting within each subject pair.

results suggest that the agent’s accuracy is not systematically driven by the glucose management in the underlying data.

F Appendix: User Question Analysis and Instantiation

As discussed in §3, real-world user queries may be ambiguous and beyond simple statistical retrieval. To handle this, our Input Processor (Layer 1) classifies questions based on feasibility and maps subjective intent to computable proxies.

Model	Prec	Rec	F1	Val Acc
Gemini 3.0 Flash	0.996	0.978	0.987	0.998

Table 10: Agent performance on multi-modal queries integrating insulin and carbohydrate logs ($N = 132$). Results confirm that adding new data modalities requires only registering a corresponding local executor.

Group	Metric Pair	r	p -value
Overall	TIR vs. F1	0.385	0.104
	TIR vs. Val Acc	0.095	0.699
T1D	TIR vs. F1	0.358	0.280
	TIR vs. Val Acc	-0.044	0.898
T2D	TIR vs. F1	0.622	0.100
	TIR vs. Val Acc	0.181	0.667

Table 11: Pearson correlation between subject-level TIR and agent performance metrics (Gemini 3.0 Pro).

F.1 Unanswerable Questions

Table 12 lists examples of queries classified as Unanswerable. These typically fall into categories such as:

- **Missing Modality**: Requests requiring external data logs not present in standard CGM files (e.g., insulin type, food ingredients, activity and sleep logs).
- **Predictive/Causal Inference**: Requests asking for future predictions or definitive causal

links (e.g., "Did X cause Y?") which cannot be clinically proven solely from retrospective glucose data.

- **Medication Consultation:** Queries seeking specific medical advice, drug pharmacodynamics, or long-term prognosis, which require professional clinical judgment beyond the scope of a data analysis agent.

F.2 Proxy Instantiations

For questions that imply a causal link (e.g., impact of lifestyle) but lack explicit logs, we employ Proxy Instantiation. As shown in Table 13, we map these intent-driven questions into time-based comparative analyses. This allows the agent to provide *data-driven correlations* (e.g., "Glucose was lower during your workout window") while acknowledging the lack of definitive causation.

G Appendix: System Prompts

We employ distinct system instructions for each layer of the CGM-Agent framework. We present the core prompts used for the Input Processor, Analytical Agent, Response Generator, and the Automated Judgment.

G.1 Agent Prompts

Layer 1: Input Processor (Query Refiner)

Role: You are a Question Refiner that processes raw user questions about CGM data into a standardized, answerable format for a data analysis agent.

Goal: Determine if the question can be answered using only CGM glucose timestamp and value data, and if so, rephrase it into a standard query.

Supported Features: 1. Time in Range (TIR), Time below Range (TBR), Time above Range (TAR), ideal blood glucose control
2. Average blood glucose, Standard Deviation, Glycemic Variability (CV), estimated A1c, estimated Glucose Management Indicator (eGMI)
3. Min/Max blood glucose, Hypoglycemia and hyperglycemia events
4. Glucose excursions and glucose trends

Answerability Logic: 1. **Direct Data (YES):** Questions about past glucose data.
2. **Behavioral (Indirect YES):** Questions about "food/exercise/sleep" ARE answerable IF convertible to glucose trends during a specific time.
3. **Medical/External (NO):** General medical knowledge, future predictions, or questions strictly requiring insulin/food logs (e.g.,

"What is my insulin sensitivity?").

Refinement Guidelines (Standardized Formats):

While extracting features, consider if user needs to know CGM weartime to determine if the calculation includes enough data points to show that feature is reliable. Some standardized formats to make your refined questions be clear with specific information:

- **Basic Retrieval:** "What are my {features} and CGM weartime over the following dates: {dates_str}?"
- **Conditional Statistics:** "What are my average {features} over {dates_str}? Consider two conditions: 1. Days with any CGM records. 2. Days with good weartime (>70%)."
- **Cohort Comparison:** "Compare {features} between groups of dates: {dates_str}. Calculate: 1. Average value per group. 2. Absolute difference. 3. Which group is higher."
- **Event Analysis:** "Analyze glucose excursions for {dates_str}. Find significant rapid changes and details on timing, magnitude, and speed."
- **Visualization:** "Plot my typical daily CGM blood glucose trends for {dates_str}. Output the mean values used to generate the plot."

Input Fields:

- **user_question:** The original raw question.
- **reference_date:** To interpret "yesterday", "last week".
- **reference_datetime:** To interpret "over last 4 hours".

Output Fields:

- **is_answerable:** Boolean.
- **refined_question:** The standardized query or specific missing modality explanation.
- **rationale:** Reasoning for how agent process the question.

Layer 2a: Analytical Router (Task Decomposer)

Role: You are a Router Agent that acts as the entry point for the analytical pipeline.

Goal: Analyze the user's request to determine if it constitutes a **Single Task** or **Multiple Separate Tasks**, and delegate accordingly.

Routing Logic:

1. Single Task (Batch/Comparison):

- Pattern: Requests involving a specific list of dates (e.g., "Dec 29, Dec 31") or

User Query (q)	Reason for Rejection / Missing Modality
<i>"What type of foods are generally safe to consume and do not need massive insulin boluses?"</i>	Missing Dietary Log: CGM data contains glucose levels but lacks specific food intake records required to correlate food types with insulin response.
<i>"Does splitting insulin boluses help me lower my insulin intake?"</i>	Missing Medication Log: Requires detailed insulin dosing logs (timing, amount, split vs. single) which are external to the CGM sensor data.
<i>"Given my CGM data, can you find moments that were likely incorrect blood glucose values?"</i>	Sensor Hardware Limitation: Detecting sensor errors (compression lows, calibration drifts) requires raw electrical signal data or external calibration values, not just the processed glucose stream.
<i>"Does the stability of my blood glucose over the last three days predict more stability in the coming days?"</i>	Forecasting/Prediction: The current agent is designed for retrospective analysis and insight generation, not predictive modeling of future physiological states.
<i>"How does Mounjaro impact insulin intake, and what are the long-term effects on insulin-to-carb ratios? Will these ratios return to what they were...?"</i>	Medication Consultation: Questions regarding specific drug mechanisms (e.g., GLP-1 agonists) and long-term medical prognosis constitute clinical advice, which is strictly out of scope for a retrospective data analysis agent.

Table 12: Examples of user questions labeled as **Unanswerable** in our dataset ($N = 513$). The Input Processor is trained to generate an explanation e (right column) instead of attempting a hallucinated calculation.

Ambiguous User Query (q)	Instantiated Computable Proxy (q')
<i>"Given my CGM data, can you infer when I ate my meals?"</i>	Trend Analysis: "Given my CGM data over [Date], can you infer when my glucose level rises fast?"
<i>"What is my time in range during workouts vs. after exercise?"</i>	Temporal Windowing: "What is my time in range between [Datetime A] and [Datetime B]?"
<i>"How long after ingesting food do my glucose levels rise?"</i>	Event Detection: "When my glucose levels rise over today?"
<i>"What patterns do I see around my menstrual cycle?"</i>	Blood Glucose Trends: "Plot my typical daily CGM blood glucose trends for the following dates: [Date A] to [Date B]."

Table 13: Examples of **Proxy Instantiations** ($N = 399$). The vague user intent (left) is transformed into a precise, computable query (q') targeting specific time windows or statistical features (right), enabling the Analytical Agent to execute code.

<p>explicit comparisons (keywords: "compare", "difference", "vs").</p> <ul style="list-style-type: none"> - Action: Treat as ONE self-contained task. Do NOT split. - Example: "Compare TIR between Group A dates and Group B dates." → Send as 1 query. <p>2. Multiple Tasks (Decomposition):</p> <ul style="list-style-type: none"> - Pattern: Requests containing keywords like "separately", "each week", or distinct disjoint time ranges (e.g., "Week 1 AND Week 2"). - Action: Split into a list of focused sub-questions to be executed independently. - Example: "What was my average glucose for Week 1 and Week 2 separately?" → Split into ["Average glucose Week 1?", "Average glucose Week 2?"]. 	<p>Input: user_request: Refined question from Input Processor.</p> <p>Output:</p> <ul style="list-style-type: none"> • date_list: List of date strings/ranges. • question_list: List of decomposed queries to be sent to the Executor.
Layer 2b: Analytical Executor	
<p>Role: You are a Healthcare Scientist and the primary worker agent.</p> <p>Workflow: You MUST strictly follow this three-step pipeline: filter_cgm_csv → extract_features_json → computation_tool.</p> <p>Execution Constraints:</p> <ul style="list-style-type: none"> • Internal Data Only: Process features 	

derivable strictly from CGM timestamps and values.

- **Missing Modalities:** If a feature requires external logs (insulin, food, sleep), set value to -1.
- **Boolean Logic:** For Yes/No questions, return 1 (Yes) or 0 (No).
- **No Hallucination:** Do not invent features not present in the data.

Output Format Schema: Return a nested dictionary: {Date_Key: {Feature_Name: Value}}.

- **Single Date:** "2025-09-01": {"TIR": 70.5}
- **Date Range:** "(2025-09-01, 2025-09-07)": {"mean_glucose": 120}
- **Date List:** "['2025-01-01', '2025-01-03']": {"days_with_good_wear": 2}

Input: A single specific query from the Router.

Output: result (The computed numerical dictionary).

Layer 3: Response Generator

Goal: Generates a clear, concise, and empathetic final response to a user about their CGM (Continuous Glucose Monitor) data.

Response Guidelines:

1. **Handling Refusals:** Since `include_rationale` is `True`, if `is_answerable` is `False`, you **MUST** explain why based on the rationale (e.g., "I cannot analyze this because I lack food logs").
2. **Answer Structure:**
 - **Direct Answer:** Start with the key finding from `execution_result`.
 - **Contextual Bridge:** Explain how the data relates to the user's intent (e.g., "To answer your question about glucose during exercise, I looked at your glucose from 3 - 5 PM").
 - **Data Evidence:** Cite specific numbers/trends from `execution_result` to support the claim.

Dynamic Input Fields (Per-Query):

- `raw_question`: The user's original raw input.
- `is_answerable`: Boolean flag indicating feasibility.
- `rationale`: Reasoning for why the question was accepted or rejected.

- `execution_result`: The raw JSON output containing numerical data.
- `tone`: The tone of the response.
- `complexity_level`: The target audience complexity level (e.g., middle school knowledge).

Output: `final_response` (Natural language text).

G.2 Evaluation Prompts of Analytical Agent

Automated Evaluator (Gemini 3.0 Pro)

Task: rigorous comparison between the Agent's numerical output and Ground Truth (GT).

Feature Matching Guidelines:

- **Semantic Mapping:** Match features by concept, not exact naming (e.g., "avg bg" \equiv "mean blood glucose").
- **Missing Data Handling:**
 - If GT is -1 (No Data) and Agent result is missing that feature \rightarrow **Match** (Both correctly identified no data).
 - If GT has a valid value but Agent is missing it \rightarrow **Mismatch** (False Negative).

Value Comparison Logic:

- **Numerical Tolerance:** Values match if within $\pm 1\%$.
- **Boolean Logic:** `1.0` = "Yes", `0.0` = "No".
- **Special Case (-1 vs 0):**
 - CGM Weartime/Usage: -1 and 0 are considered **Equivalent** (both imply "No Data").
 - Event Counts/TIR: -1 (Missing Data) \neq 0 (Zero occurrences). These represent different clinical states.

Filtering Logic: (Primarily for Real-world User Queries)

- If `required_features` is provided: **ONLY** evaluate features in this list. Ignore extras.
- If `required_features` is empty: Evaluate **ALL** features present in GT.

Input Fields:

- `question`: The context query.
- `required_features`: List of priority metrics to verify.
- `gt_res`: Ground truth dictionary.
- `agent_result`: Agent prediction dictionary.

Output Schema: comparison dictionary containing:

- Counts: num_gt_features, num_agent_features, num_overlap.
- Sets: features_in_gt_not_in_agent (FN), features_in_agent_not_in_gt (FP).
- Value matches: feature_value_comparison.

H Appendix: Latency and Resource Footprint

Table 14 reports end-to-end latency for the Analytical Agent (Layer 2) across 130 queries for a randomly selected subject (P2).

Model	Mean	Median	P95	LLM Calls
Gemini 3 Flash	82.8	74.7	153.7	8.4
Gemini 3 Pro	100.4	98.4	148.8	8.0

Table 14: Analytical Agent (Layer 2) latency over 130 queries (P2). Mean, Median, and P95 are reported in seconds. LLM Calls reports the mean number of API calls per query.

Latency is driven primarily by sequential LLM API calls (~8 per query) and repeated file I/O for per-request metric computation. In production, pre-computing and caching data structures would substantially reduce overhead, and independent sub-tasks could be parallelized. For latency-sensitive deployments, lightweight SLMs could replace the backbone for lower-complexity sub-tasks without modifying the overarching framework.

I Appendix: Participant Recruitment and Instruction Material

Below is the condensed text used for participant recruitment and instructions, anonymized for review purposes.

Ethics and Protocol This study was approved by the Institutional Review Board (IRB) at Dartmouth College. Participation was voluntary. All participants were required to complete a consent form indicating their willingness to participate before proceeding with the study tasks.

Participant Requirements and Recruitment We recruited adults (aged 18 and older) who actively use a continuous glucose monitor (CGM).

Compensation Participants received a \$10 USD gift card as compensation for their time upon successfully submitting 10 or more questions.

Task Instructions Participants were informed that they did not need to share their actual raw glucose data. Instead, they were asked to write free-form questions reflecting their curiosity when reviewing their CGM data. The specific instructions provided to participants is as follows:

- Questions can relate to prior glucose data (e.g., from previous hours, days, weeks, or months).
- Questions can seek to uncover insights regarding personal glucose management and trends.
- The focus should be on retrospective review (looking back at data to reflect on progress) rather than real-time monitoring (e.g., “Is my blood glucose high or low right now?”).
- Questions can be short or detailed; participants are encouraged to be clear and concrete.

J Appendix: Subject Selection

Table 15 lists the mapping between the anonymous subject identifiers used throughout this paper and the original identifiers in their respective public datasets. All subjects were selected by prioritizing the longest available CGM recording duration within each dataset, as described in §3.1.

Subject ID	Original ID	Dataset
P1	Subject 15	AZT1D
P2	Subject 23	AZT1D
P3	Subject 21	AZT1D
P4	Subject 20	AZT1D
P5	Subject 7	AZT1D
P6	Subject 19	AZT1D
P7	Subject 5	AZT1D
P8	Subject 13	AZT1D
P9	Subject 6	AZT1D
P10	Subject 11	AZT1D
P11	Subject 4	AZT1D
P12	2069	ShanghaiT2DM
P13	2014	ShanghaiT2DM
P14	2017	ShanghaiT2DM
P15	2015	ShanghaiT2DM
P16	2078	ShanghaiT2DM
P17	2001	ShanghaiT2DM
P18	2055	ShanghaiT2DM
P19	2074	ShanghaiT2DM

Table 15: Mapping of anonymous subject identifiers to original dataset identifiers. P1–P11 are drawn from AZT1D (Khamesian et al., 2025) (T1D, 5-min sampling), and P12–P19 from ShanghaiT2DM (Zhao et al., 2023) (T2D, 15-min sampling).