Cognitive AI Decision Aid for Traceable Interaction with MBSE Behavior Models

Erfan Ashouri
Department of Systems Engineering and Operations Research, George Mason University, Fairfax, VA, USA
Email: eashouri@gmu.edu
ORCID: 0009-0000-0101-674X

Lance Sherry
Department of Systems Engineering and Operations Research, George Mason University, Fairfax, VA, USA
Center for Air Transportation Systems Research, George Mason University, Fairfax, VA, USA
Email: lsherry@gmu.edu
ORCID: 0000-0003-3630-5394

Corresponding author: Erfan Ashouri, eashouri@gmu.edu

Abstract

Large behavior models are central to Model-Based Systems Engineering (MBSE), but mission-level models can be difficult to inspect when engineers, trainers, and operators need to understand what a system will do, why it will do it, and when behavior will occur. This paper presents an SGB-aware retrieval framework for Retrieval-Augmented Generation (RAG) that supports traceable natural-language interaction with Situation-Goal-Behavior Models (SGBMs). The method treats each complete Situation-Goal-Behavior record, rather than an arbitrary text chunk, as the atomic retrieval unit. In a municipal-water emergency-response case study with a 150-question diagnostic benchmark, the balanced SGB-Hybrid configuration achieved 99.2% Any-HR@5, 91.7% All-HR@5, and 82.5% MRR@5. Sensitivity analysis showed that a semantic-heavy hybrid configuration further improved retrieval, while the balanced setting provides a transparent default. The results show that structure-preserving retrieval improves traceable access to behavior-model evidence and clarifies remaining challenges in confusable cases, multi-answer queries, and unsupported-query detection.

Keywords: behavior models; human-AI interaction; model-based systems engineering; retrieval-augmented generation; Situation-Goal-Behavior model; structured retrieval; traceable AI.

1. Introduction

The Situation-Goal-Behavior Model (SGBM) is a formal behavior-modeling method used to specify the behavior of complex, mission-level automation (Feary et al., 2002; Sherry et al., 2001). The SGBM represents system behavior through the binding of situations, goals, and behaviors. A Situation represents selected input states that describe what is happening in the system or operating environment. A Goal represents the purpose pursued under that situation. A Behavior represents the selected output functions or commanded response used to achieve the goal (Feary et al., 2002; Sherry et al., 2001). An SGBM is therefore a structured behavior model in which meaning is carried by the binding among input states, situations, goals, behaviors, output functions, and default assumptions.

SGBMs combine a formal behavior representation with a natural-language semantic layer. The formal representation supports completeness and consistency analysis, model checking, code generation, and test-case generation, while the semantic layer supports communication among engineering, operational, training, interface, and procedure-design stakeholders (Feary et al., 2002; Sherry et al., 2001; Sherry et al., 2021).

Although SGBMs support completeness and consistency checking, large infrastructure models can still be difficult for operators and system reviewers to inspect during design reviews, training, and operations. Designers and users may need to ask: What will the system do? Why will it do that? When will this behavior occur? What input states support this behavior? These questions are closely related to long-standing concerns in automation understanding and cognitive engineering (Wiener, 1988). More broadly, behavior-driven modeling research has shown that preserving dependencies among interacting behaviors is important for representing and analyzing complex safety-critical systems (Xing et al., 2021).

Recent applications of Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) offer a natural-language interface to query large repositories. However, many conventional RAG implementations partition source files into fixed-length text chunks. In structured system behavior models, this partitioning can break the relational integrity of the underlying data schema. A generic text chunk may separate a critical system behavior from the environmental input states that activate it, or decouple an operational goal from its default bounding assumptions. In safety-relevant systems, this structural fragmentation can increase the risk of ungrounded or incomplete model explanations. This paper addresses this structural mismatch by introducing an SGB-aware retrieval framework. The proposed method uses the complete, multi-field SGB record as the base unit of retrieval so that functional dependencies between system inputs and commanded behaviors remain preserved throughout the interaction pipeline.

This paper investigates SGB-aware retrieval as a structure-preserving interaction layer for large behavior models. The SGBM remains the authoritative model, while the language model generates explanations only from retrieved evidence. The methodological contribution is not a new retrieval algorithm; it is the

definition of the complete Situation-Goal-Behavior record as the atomic retrieval unit for structured behavior models. Unlike generic chunk-based RAG, which may split a behavior from the input states, goals, output functions, and assumptions that define it, the proposed method preserves the functional binding required for model-grounded explanation. This enables users to query an SGBM from multiple semantic directions — condition-to-behavior, behavior-to-condition, output-to-case, or goal-to-situation — while maintaining an explicit trace to the authoritative behavior model. Section 1.1 details the study's three contributions to industrial and systems engineering. Figure 1 summarizes the motivation for an LLM-supported retrieval interface that can interact with large MBSE behavior models while preserving model-grounded evidence.

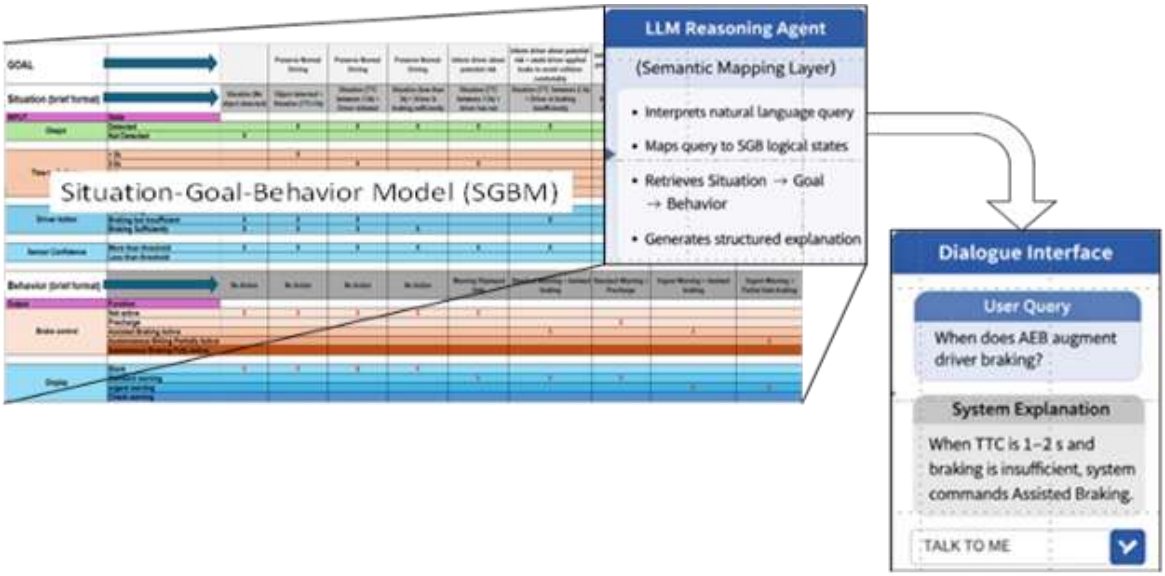


Figure 1. Overview of SGB-aware interaction with MBSE behavior models.

1.1 Contributions to Industrial and Systems Engineering

Within industrial and systems engineering, this study contributes to the design and evaluation of AI-assisted interfaces for structured engineering artifacts. The contributions are threefold:

Structure-preserving retrieval for AI interfaces: The proposed architecture replaces generic text chunks with deterministic SGB record units, preserving relationships among operational situations, system boundaries, and commanded behaviors.

Diagnostic benchmark for MBSE retrieval: The evaluation tests natural-language retrieval over structured behavior-model data using multidirectional entry points, paraphrased queries, unsupported queries, and highly confusable operational boundaries.

Operational retrieval and failure analysis: The municipal-water case study shows that SGB-aware hybrid retrieval improves top-5 evidence recovery relative to chunk-based baselines, while the top-1 error profile identifies confusable and threshold-adjacent cases that require stronger ranking safeguards before high-

consequence use. This positioning is consistent with engineering data-analysis research showing that integrating heterogeneous sources of information can provide a more complete understanding of complex systems than analyzing individual sources in isolation (Gaw et al., 2022).

2. Background

To contextualize structure-preserving retrieval, we establish the formal primitives of the SGB schema alongside conventional text-partitioning limitations.

2.1 Situation–Goal–Behavior Model (SGBM)

In the SGBM, a situation is specified by selected input states, a goal states the purpose pursued under that situation, and a behavior specifies the output functions or commanded responses used to achieve the goal (Feary et al., 2002; Sherry et al., 2001).

The SGBM structure matters because system behavior is not defined by a behavior label alone; it is defined by the relationship among the recognized situation, the intended goal, and the selected behavior. More broadly, recent behavior-driven modeling research has shown that preserving dependencies among interacting behaviors is fundamental to accurately representing and analyzing complex safety-critical systems (Xing et al., 2021). This same principle motivates preserving Situation–Goal–Behavior relationships during retrieval rather than fragmenting them into independent text chunks. This focus on relationships is broadly consistent with dependency-aware modeling approaches in other domains, such as time-varying network methods that represent directed conditional dependencies among variables explicitly rather than treating them as independent (Yang & Zhang, 2024), although that work addresses statistical change-point detection rather than behavior-model retrieval specifically.

While the dual-layer nature of SGBMs supports cross-stakeholder communication, scaling these models to complex industrial infrastructure introduces severe readability barriers during active operations. Figure 2 illustrates a basic Situation-Goal-Behavior structure and its tabular representation.

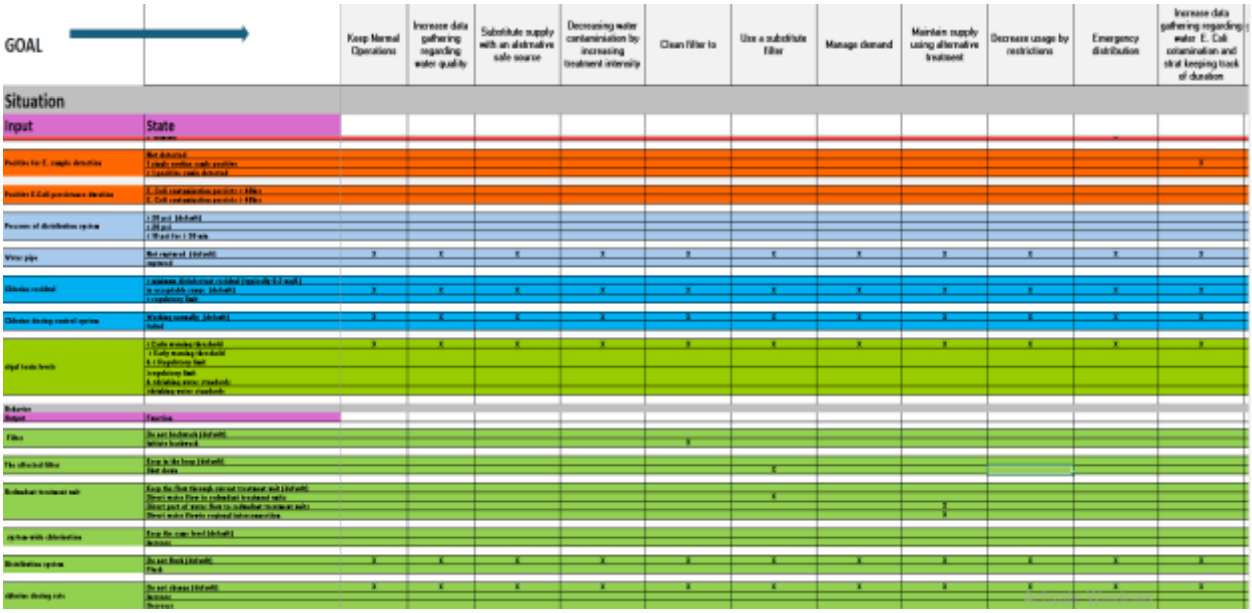


Figure 2. A simple example of Situation-Goal-Behavior structure and tabular SGB representation.

2.2 Retrieval-Augmented Generation (RAG)

Large Language Models (LLMs) provide a natural-language interface for asking questions, summarizing information, and generating explanations. Modern LLMs are commonly based on Transformer architectures, which use attention mechanisms to model relationships among tokens and generate fluent text (Vaswani et al., 2017). However, when used alone, an LLM does not have guaranteed access to the specific content of an external engineering model. It also does not inherently preserve the formal structure of that model.

Retrieval-Augmented Generation (RAG) addresses part of this limitation by combining a generative language model with retrieved external evidence. Lewis et al. (2020) introduced RAG as a framework that combines parametric memory in a pretrained model with non-parametric memory retrieved from an external corpus. In a typical RAG pipeline, a retriever first selects documents or passages relevant to a user query, and the selected evidence is then provided to the language model as context for answer generation.

Retrieval quality is therefore central to RAG performance. Dense Passage Retrieval demonstrated that dense vector representations can effectively retrieve relevant passages for open-domain question answering (Karpukhin et al., 2020). Subsequent retrieval-augmented language models showed that incorporating retrieved evidence can improve language-model performance without modifying the underlying model architecture (Ram et al., 2023). In parallel, recent industrial and systems engineering research has emphasized AI methods that support expert inspection and human-in-the-loop review of engineering artifacts (Shen & Kong, 2024). Collectively, these developments motivate retrieval-grounded interaction with structured engineering models.

Generic chunk-based RAG has an important limitation for structured engineering artifacts. A conventional RAG system divides source material into text chunks, retrieves chunks by lexical or semantic similarity, and passes the retrieved text to the LLM (Gao et al., 2023). Although this can improve factual grounding, it does not guarantee that the retrieved unit preserves the structure of the original source. In SGBMs, generic chunking may retrieve locally relevant text while separating the situation, goal, behavior, input-state, and output-function relationships needed for traceable explanation.

The SGB-aware RAG approach addresses this gap by defining the complete SGB record as the retrieval unit. Each retrieved unit preserves the case identifier, situation, goal, selected input states, behavior, selected output functions, default assumptions, and trace identifiers. The objective is to support natural-language questioning while maintaining traceability to the authoritative SGBM.

This paper differs from related retrieval paradigms primarily in the definition of the retrieved unit. Generic RAG retrieves passages (Lewis et al., 2020; Gao et al., 2023); GraphRAG retrieves graph-derived evidence (Edge et al., 2024); agentic and self-reflective RAG adapt retrieval and generation through planning, critique, or on-demand retrieval (Liang et al., 2025; Asai et al., 2024); and hierarchical methods such as RAPTOR retrieve across levels of abstraction using tree-organized summaries (Sarthi et al., 2024). In contrast, SGB-aware RAG starts from the formal structure of the engineering artifact: the retrieved unit is a complete SGB case.

3. Design

The engineering core of the SGB-aware framework relies on mapping deterministic tabular boundaries into isolated, semantic retrieval objects across design-time and query-time pipelines.

3.1 Concept of Operations

The SGB-aware RAG framework is an interaction layer over a structured SGBM. The process begins by transforming the SGB workbook into case-level records that preserve Case ID, Goal, Situation, selected input states, Behavior, selected output functions, default assumptions, and trace identifiers.

The central architectural decision is to treat the complete SGB record as the atomic retrieval unit, thereby preserving the relational structure required for traceable explanation. This structure also enables multi-directional interaction with the SGBM. Users may enter the model through input states, situations, goals, behaviors, output functions, or combinations of these elements. The retriever must therefore recover the relevant SGB record even when the query begins from a downstream element such as a goal, behavior, or output function.

The concept of operations has two phases. During design-time indexing, the workbook is validated, parsed, converted into structured records, embedded, and stored in a local retrieval index. During query-time

interaction, the system retrieves the most relevant SGB records, sends the retrieved evidence to the LLM, and returns both a natural-language answer and an explicit SGB trace.

3.2 Indexing Pipeline

Figure 3 shows the design-time indexing pipeline. The input is a two-sheet SGB Excel workbook: one sheet defines goal-situation relationships through input-state selections, and the other defines goal-behavior relationships through output-function selections.

The system validates the workbook structure, including expected headers, case identifiers, and alignment between the situation and behavior sheets. It then parses the workbook into a structured SGBM table representation and exports case-level JSON records.

Each structured SGB record is converted into a retrieval-ready text representation. In the evaluated implementation, one complete SGB record is represented as one retrieval unit. The system then uses the OpenAI text-embedding-3-small model to generate 1536-dimensional SGB record embeddings and stores them in a local retrieval index (OpenAI, n.d.-a).

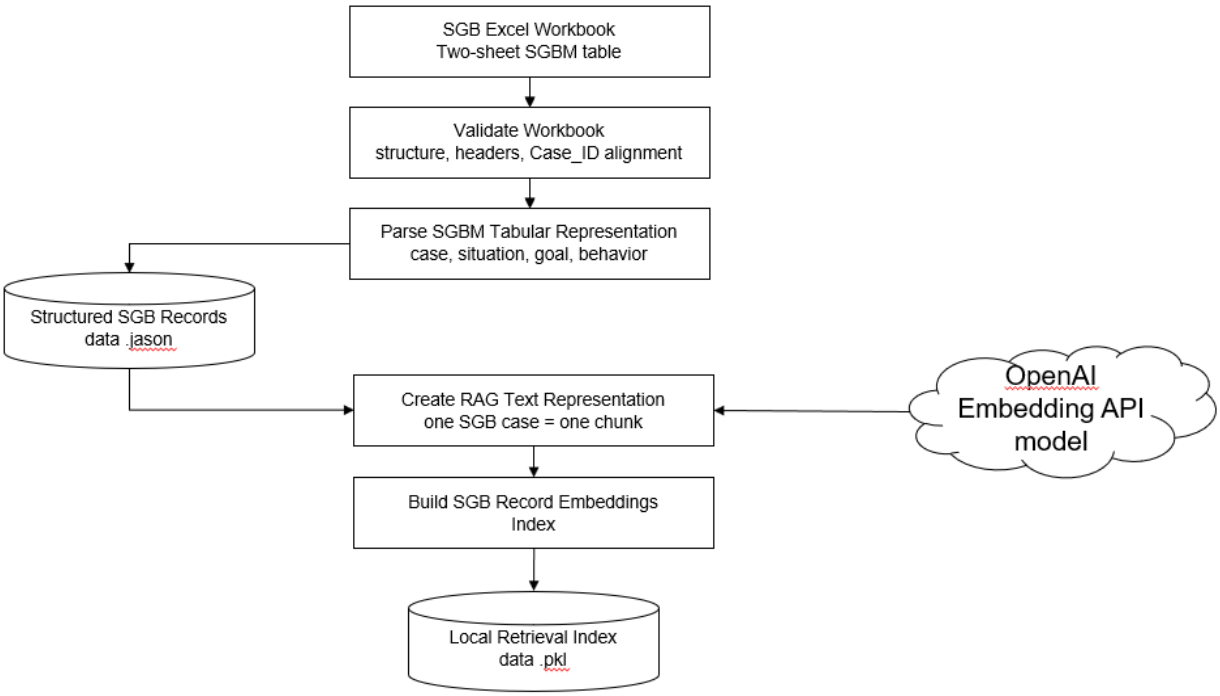


Figure 3. Design-time indexing pipeline.

3.3 Query-Time Retrieval and Answer Generation Pipeline

Figure 4 shows the query-time retrieval and answer-generation pipeline. For each user question, the system embeds the query and compares the query vector with stored SGB record vectors to compute semantic similarity scores.

In parallel, the system applies sparse lexical scoring using BM25. The query is normalized and compared with retrieval text constructed from important SGB fields, including situation terms, behavior names, goal descriptions, input states, and output functions. This lexical component is useful because engineering questions often contain specific terminology where exact terms, thresholds, and named behaviors matter.

The dense and sparse scores are normalized and combined into a final retrieval score. The system ranks the candidate evidence units, selects the top-5 records, and sends them to the answer-generation model with the user question and fixed grounding instructions. In the evaluated implementation, generation is executed through the OpenAI Chat Completions API (OpenAI, n.d.-b). The prompt instructs the model to answer only from retrieved SGB evidence and to return a direct answer, explanation, confidence statement, and SGB trace.

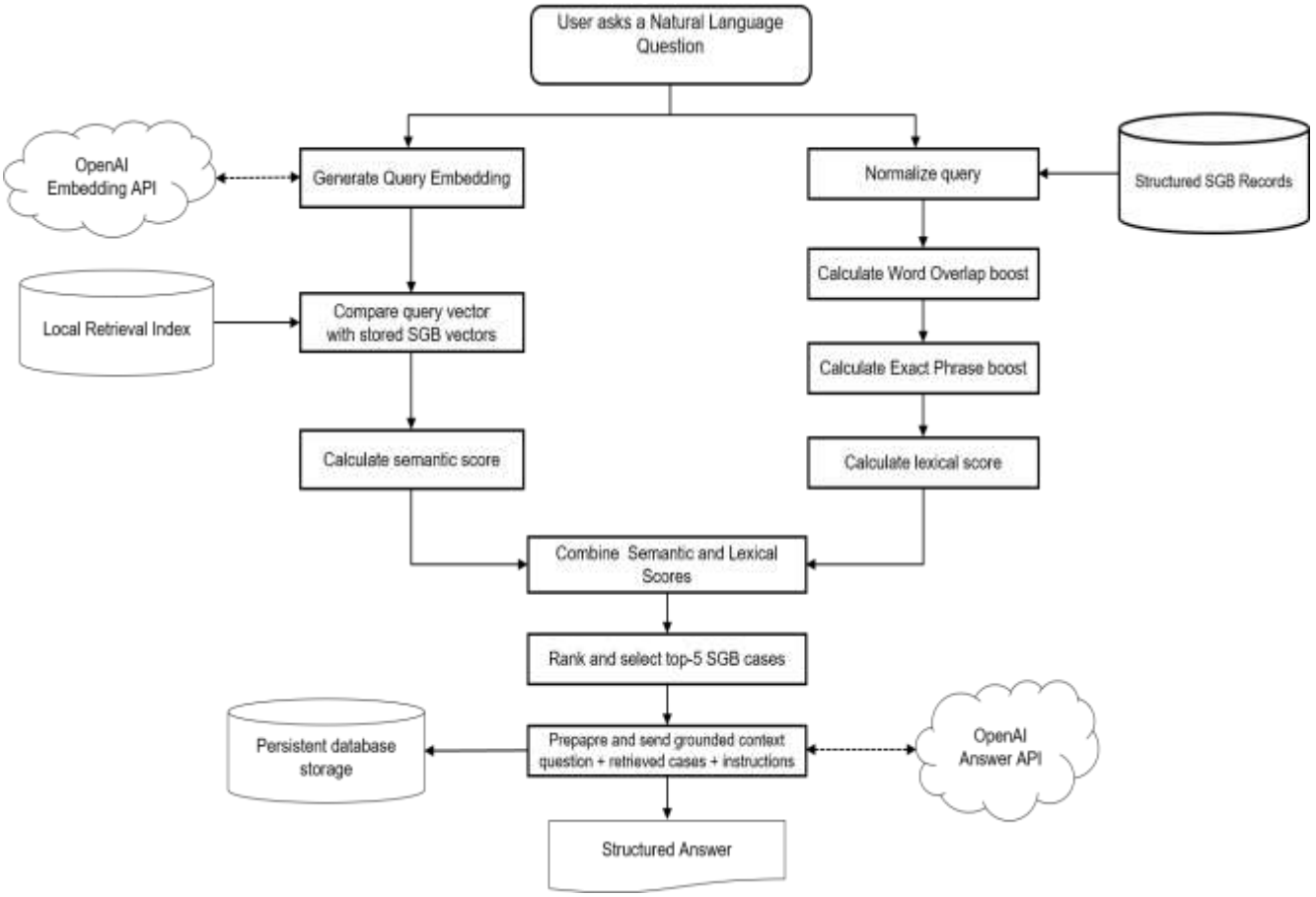


Figure 4. Query-time evidence retrieval and answer-generation pipeline.

3.4 Scoring and Ranking

The query-time retriever ranks candidate evidence units using three scoring families: dense semantic similarity, sparse BM25 lexical matching, and a convex hybrid of the two. For SGB-aware retrieval, the candidate unit u_i is a complete SGB record. For the generic chunk baselines, u_i is a text chunk created from the same source material. This design isolates the effect of the retrieval unit rather than the scoring algorithm alone.

Dense retrieval computes cosine similarity between the query embedding q and the candidate-unit embedding c_i:

$$S_i = \frac{\sum_{k=1}^{D} q_k c_{i,k}}{\sqrt{\sum_{k=1}^{D} q_k^2}\sqrt{\sum_{k=1}^{D} c_{i,k}^2}}. \tag{1}$$

where q is the embedding vector for the user query, c_i is the embedding vector for candidate unit u_i, and the score increases as the two vectors point in more similar directions in the embedding space.

Sparse retrieval uses a BM25-style lexical score over the retrieval text of u_i:

$$B_i = \sum_{t \in q \cap C_i} \mathrm{IDF}(t)\ \frac{f_{t,C_i}(k_1+1)}{f_{t,C_i}+k_1\left(1-b+b\frac{|C_i|}{\mathrm{avgdl}}\right)}. \tag{2}$$

where *t* denotes a query term, f_{t,C_i} is the frequency of term *t* in candidate unit u_i, $|u_i|$ is the length of the candidate text, *avgdl* is the average candidate length in the active corpus, and k_1 and b are the standard BM25 saturation and length-normalization parameters. The term weights are governed by an inverse document frequency formulation to dampen the effects of ubiquitous engineering vocabulary:

$$\mathrm{IDF}(t) = \ln\left(1 + \frac{|\mathcal{C}|-n_t+0.5}{n_t+0.5}\right). \tag{3}$$

In Equation (3), $|\mathcal{C}|$ is the number of candidate units in the index and n_t is the number of candidate units containing term *t*. This weighting reduces the influence of common engineering terms that occur in many records.

Because dense cosine values and BM25 scores are not bounded by equivalent scalar ranges, each scoring family is min–max normalized within the active candidate set before combination, preventing either dense semantic similarity or sparse lexical matching from dominating solely because of scale differences.

The final ranking objective balances semantic intent and precise keyword matching through the convex combination in Equation (4):

$$R_i(w) = w_s\ \frac{S_i-\min_{j\in\mathcal{C}} S_j}{\max_{j\in\mathcal{C}} S_j-\min_{j\in\mathcal{C}} S_j+\varepsilon} +\ w_b\ \frac{B_i-\min_{j\in\mathcal{C}} B_j}{\max_{j\in\mathcal{C}} B_j-\min_{j\in\mathcal{C}} B_j+\varepsilon}. \tag{4}$$

Here, w_s is the semantic weight and w_b is the BM25 weight. The hybrid score is a convex combination, so the two weights are nonnegative and sum to one. The primary SGB-Hybrid result uses $w_s = w_b = 0.5$ as a

balanced default that gives equal weight to dense semantic similarity and sparse lexical evidence. A separate sensitivity analysis evaluates whether the conclusion depends on the hybrid weighting choice.

To transition from candidate scoring to context assembly, Equation (5) isolates the highest-ranked subset of structural evidence used to populate the downstream prompt context window:

$$C_k(q) = \mathrm{TopK}_{C_i \in \mathcal{C}} R_i(w). \quad (5)$$

Here, k denotes the number of retrieved SGB records passed to the answer-generation model. Although the retrieval formulation supports arbitrary values of k, the present evaluation reports results for $k = 5$, consistent with the prototype configuration used throughout the experiments. The selected records are then provided to the large language model together with instructions to generate answers only from the retrieved SGB evidence and to produce an explicit SGB trace linking the response to the supporting records.

4. Case Study

We validate the proposed architecture using a representative operational emergency-response SGBM mapped from a municipal water distribution network. A municipal water system includes source-water intake, treatment processes, storage facilities, distribution mains, pressure-control infrastructure, and monitoring points. The system must maintain safe drinking-water quality while preserving adequate pressure and supply availability (U.S. Environmental Protection Agency, 2020; U.S. Environmental Protection Agency, 2007).

The emergency-response procedure covers seven categories of operational and public-health risk: source-water contamination, treatment process failure, microbial contamination, distribution-system pressure loss, chemical treatment dosage control, harmful algal blooms, and infrastructure failure or treatment-plant outage. These rules are suitable for SGB modeling because each rule maps a defined situation to an operational goal and a selected behavior.

The emergency response SGBM contains 13 inputs with 35 distinct input states. These inputs define a theoretical combinatorial decision space of 248,832 possible input-state combinations. This number represents the maximum potential decision space implied by the input-state structure, not the number of implemented table columns. The implemented SGB table encodes the feasible and operationally meaningful portion of this space as a bounded case-study model. The purpose of the case study is not to claim exhaustive

coverage of all municipal-water emergency scenarios, but to evaluate whether complete-record retrieval can preserve traceability across situation, goal, behavior, input-state, and output-function relationships.

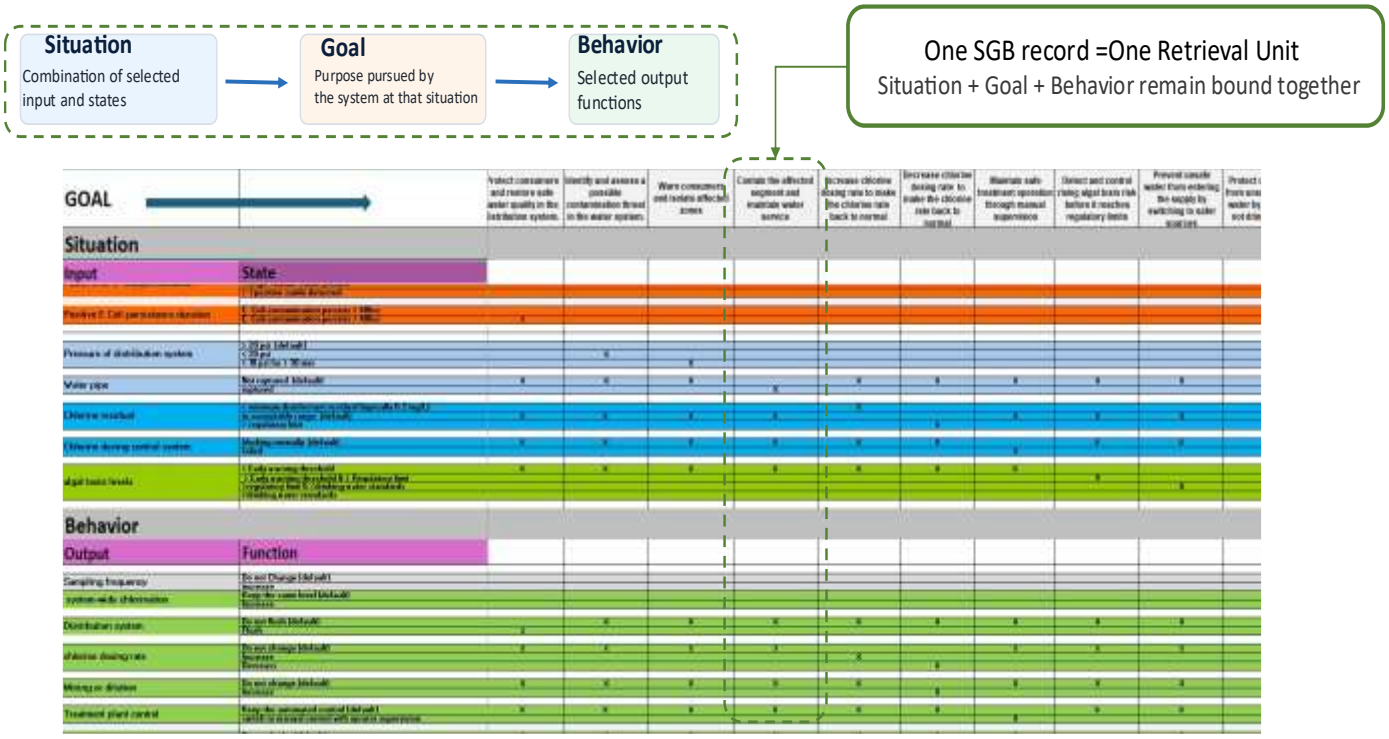


Figure 5. Example SGB table for the municipal water emergency-response case study.

The structural matrix detailed in Figure 5 establishes the precise deterministic boundaries used to ground the retrieval model's semantic context window. To evaluate how an operational systems engineer interacts with this structured logic, we deployed the formulation within a functional software application interface. Figure 6 illustrates the operational prototype interface designed to query the underlying SGB model. When

a user submits a natural-language query regarding system anomalies, the interface maps the output vector fields to return a strictly grounded answer enriched with verifiable trace metadata.

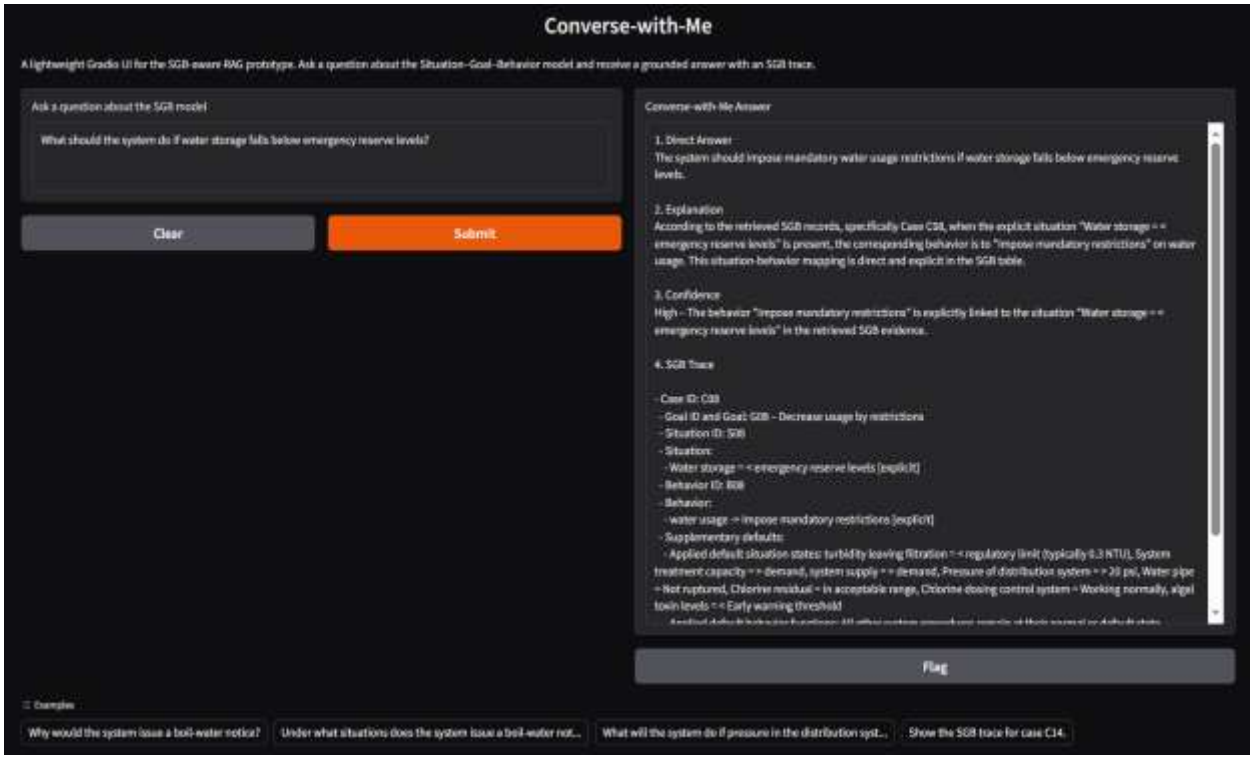


Figure 6. SGB-aware RAG interface demo for querying the SGB model.

4.1 Evaluation Design

The evaluation uses a 150-question diagnostic benchmark designed to test traceable interaction with the SGB model across both supported and unsupported queries. The benchmark contains 120 supported questions and 30 unsupported questions. Supported questions require retrieval of one or more expected SGB cases from the model. Unsupported questions ask for information outside the SGB model, such as procurement, live operational data, legal interpretation, repair estimates, or administrative decisions. This separation is important because an engineering assistant must not only retrieve correct behavior cases; it must also avoid presenting unsupported answers as if they were grounded in the behavior model.

The supported questions test multi-directional interaction with the SGBM. A query may begin from an observed condition, a behavior, an output function, a goal, a trace-like combination of fields, or a broad multi-answer request. The benchmark also includes paraphrased wording and confusable cases where several SGB records share overlapping water-emergency vocabulary. This design reflects realistic interaction with behavior models, because users do not always enter the model through the same field or with the exact terminology used in the SGB table.

Ground-truth mappings for the 150-question diagnostic benchmark were established manually from the authoritative SGB model before running the retrieval experiments. For each supported question, the expected mapping was defined as the SGB case identifier, or set of case identifiers, containing the situation, goal, behavior, input-state conditions, output-function selections, and trace information required to answer the question. Direct condition-to-case questions were mapped to the SGB case whose input-state configuration matched the query. Reverse questions that began from a behavior, output function, goal, or trace element were mapped by identifying the SGB case or cases in which that element appeared. For broad, multi-answer questions, the ground truth was defined as the complete set of relevant SGB cases rather than a single representative case.

Unsupported questions were mapped separately when the requested information fell outside the scope of the SGB model, including procurement information, live operational data, legal interpretation, cost estimates, and administrative decisions. A query was considered unsupported when no SGB case contained the evidence needed to answer it, even if the query shared surface vocabulary with water-system operations. After initial mapping, expected case identifiers were checked against the structured SGB records and their trace identifiers to confirm that each supported question was linked to the intended case-level evidence. This process produced 120 supported queries with one or more expected SGB cases and 30 unsupported queries requiring rejection rather than evidence retrieval. The ground-truth mappings were used only for evaluation and were not used by the retriever during indexing or ranking.

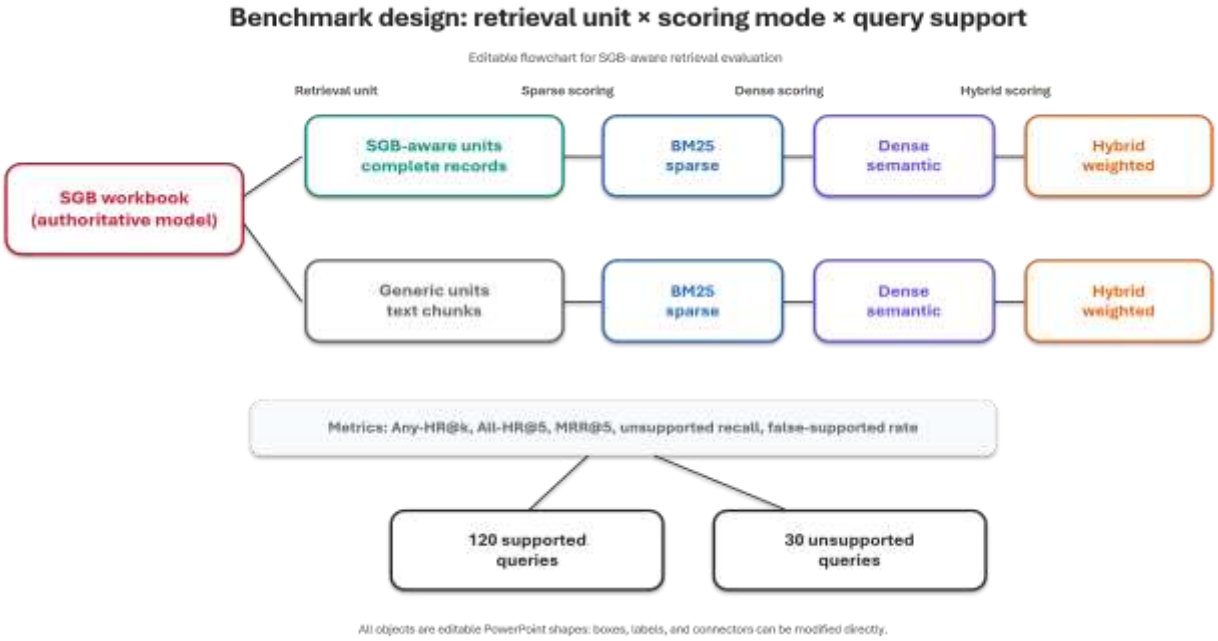


Figure 7. Retrieval benchmark design.

4.2 Retrieval Methods and Baselines

The evaluation compares SGB-aware retrieval against generic chunk-based retrieval under matched scoring conditions. In the SGB-aware setting, each complete Situation–Goal–Behavior record is treated as the atomic retrieval unit. This preserves the situation, goal, behavior, selected input states, output functions, assumptions, and trace identifiers within a single retrievable evidence unit. The purpose of this design is to test whether preserving the native structure of the behavior model improves retrieval performance compared with conventional text chunking.

For the generic chunk baselines, the same serialized SGB source text used in the SGB-aware conditions was partitioned into fixed-length word chunks using a 220-word window, producing 54 generic text chunks in the evaluated package; the final chunk contained 200 words because it represented the remaining source text. Unlike the SGB-aware conditions, these chunks did not preserve the complete situation–goal–behavior record as the retrieval unit and could contain partial fields from one case or text spanning adjacent cases. To keep the comparison controlled, the chunk baselines used the same BM25, dense-embedding, hybrid-scoring, min–max normalization, and top-5 retrieval procedure as the SGB-aware methods. Case identifiers associated with chunks were used only for evaluation of retrieval hits, not for ranking.

Six retrieval conditions were evaluated: SGB-BM25, SGB-Dense, SGB-Hybrid, Chunk-BM25, Chunk-Dense, and Chunk-Hybrid. The BM25 conditions use sparse lexical matching, the dense conditions use embedding similarity, and the hybrid conditions combine normalized BM25 and dense scores with equal weighting. This design separates the effect of retrieval-unit structure from the effect of scoring strategy: SGB-aware and chunk-based methods are compared under the same sparse, dense, and hybrid scoring families. Table 1 summarizes the evaluated retrieval conditions.

Table 1. Retrieval methods and baselines used in the benchmark.

Method	**Retrieval unit**	**Scoring mode**	**Purpose**
SGB-BM25	Complete SGB record	BM25 sparse lexical score	Tests whether structured records help even without embeddings
SGB-Dense	Complete SGB record	Dense embedding similarity	Tests semantic retrieval over preserved SGB records
SGB-Hybrid	Complete SGB record	0.5 dense + 0.5 BM25	Main proposed method; balanced hybrid
Chunk-BM25	Generic text chunk	BM25 sparse lexical score	Sparse lexical baseline without SGB record preservation
Chunk-Dense	Generic text chunk	Dense embedding similarity	Semantic baseline without SGB record preservation
Chunk-Hybrid	Generic text chunk	0.5 dense + 0.5 BM25	Closest chunk-based RAG baseline to SGB-Hybrid

4.3 Evaluation Metrics

To measure whether at least one relevant record is recovered, Equation (6) defines the Any-Hit Rate metric:

$$\text{Any-HR@}k = \frac{1}{N_s}\sum_{j=1}^{N_s} \mathbb{I}\left(E(q_j) \cap C_k(q_j) \neq \emptyset\right). \quad (6)$$

where N_s denotes the total number of supported evaluation queries, $I[\cdot]$ represents the indicator function, $E(q_j)$ defines the expected ground-truth SGB records for query j, and $C_k(q_j)$ represents the retrieved top-k candidate set.

To assess how highly a relevant record is ranked, Equation (7) defines the Mean Reciprocal Rank metric:

$$\text{MRR@5} = \frac{1}{N_s}\sum_{j=1}^{N_s} \frac{\mathbb{I}(r_j \leq 5)}{r_j}. \quad (7)$$

where r_j represents the explicit rank position of the first relevant retrieved record for query j. If no relevant record appears within the top-5 results, the reciprocal-rank contribution for that query is zero.

Some behavior-model queries require multiple relevant SGB records to achieve comprehensive diagnostic coverage. To capture this stricter requirement, Equation (8) defines the All-Hit Rate metric, which requires recovery of the complete set of relevant records:

$$\text{All-HR@}k = \frac{1}{N_s}\sum_{j=1}^{N_s} \mathbb{I}\left(E(q_j) \subseteq C_k(q_j)\right). \quad (8)$$

Here, N_s, $E(q_j)$, and $C_k(q_j)$ are defined as previously noted. This stricter formulation explicitly penalizes partial retrieval for multi-answer questions, where returning only a single subset row remains insufficient to achieve complete operational traceability.

For unsupported queries, Unsupported Recall measures whether the system correctly abstains from grounding an answer in an unrelated SGB record.

$$\text{Unsupported Recall} = \frac{1}{N_u}\sum_{j=1}^{N_u} \mathbb{I}\left(\hat{y}_j = \text{unsupported}\right). \quad (9)$$

where N_u denotes the total number of unsupported evaluation queries. The complementary failure mode is false-supported behavior, where an unsupported query is incorrectly grounded in unrelated evidence.

4.4 Main Results

The empirical evaluation shows that structure-preserving retrieval improves evidence recovery and ranking performance relative to generic text chunking across the supported-query retrieval metrics, while matching the Chunk-Hybrid baseline on Unsupported Recall under the balanced hybrid configuration. Under the balanced default configuration ($w_s = 0.5, w_b = 0.5$), the proposed SGB-Hybrid framework achieved 99.2% Any-HR@5, 91.7% All-HR@5, and 82.5% MRR@5. Relative to Chunk-Hybrid, this corresponds to improvements of 5.0 percentage points in Any-HR@5, 6.7 percentage points in All-HR@5, and 7.4 percentage points in MRR@5, as shown in Table 2 and Figure 8.

The ranking results also show that treating the complete SGB record as the atomic retrieval unit shifts relevant engineering evidence higher in the ranking queue. The system's total error footprint also decreased substantially: SGB-Hybrid committed only 33 Top-1 ranking errors, a 26.7% reduction in primary sorting failures compared to the 45 errors observed under Chunk-Hybrid. The sparse-only and dense-only ablation

variants further highlight the advantages of complete-record preservation: SGB-Dense outperformed Chunk-Dense by 9.2 percentage points in Any-HR@5 and 10.8 percentage points in All-HR@5, while SGB-BM25 outpaced Chunk-BM25 by 10.0 and 6.7 percentage points on the same metrics. Because these gains persist even when the scoring algorithm is held constant, they indicate that maintaining cross-field dependency constraints within the retrieval unit — not the choice of scoring function — drives the improvement in evidence recovery.

Table 2. Main retrieval results across SGB-aware and generic chunk baselines.

Method	**Family**	**Mode**	**Any-HR@1**	**Any-HR@5**	**All-HR@5**	**MRR@5**	**Unsup. Recall**	**Top-1 errors**
SGB-BM25	SGB-aware	BM25	68.3%	94.2%	84.2%	78.5%	66.7%	38
SGB-Dense	SGB-aware	Dense	70.0%	96.7%	90.0%	80.9%	76.7%	36
SGB-Hybrid	SGB-aware	Hybrid	72.5%	99.2%	91.7%	82.5%	66.7%	33
Chunk-BM25	Generic chunk	BM25	59.2%	84.2%	77.5%	68.3%	63.3%	49
Chunk-Dense	Generic chunk	Dense	60.8%	87.5%	79.2%	71.2%	70.0%	47
Chunk-Hybrid	Generic chunk	Hybrid	62.5%	94.2%	85.0%	75.1%	66.7%	45

Percentages for Any-HR@1, Any-HR@5, All-HR@5, and MRR@5 are computed over the 120 supported benchmark questions. Unsupported Recall is computed over the 30 unsupported benchmark questions. Top-1 errors are reported as counts over the supported benchmark questions.

A query-level paired bootstrap analysis was also conducted for the headline comparison between SGB-Hybrid and Chunk-Hybrid. The estimated differences were +10.0 percentage points for Any-HR@1, +5.0 percentage points for Any-HR@5, +6.7 percentage points for All-HR@5, and +7.4 percentage points for MRR@5, with 95% paired bootstrap confidence intervals of [0.8, 19.2], [0.8, 10.0], [0.0, 13.3], and [1.4, 13.5], respectively. Unsupported Recall was unchanged at 66.7% for both methods. Because the confidence

interval for All-HR@5 touches zero, that metric is interpreted as a directional improvement rather than a definitive paired difference.

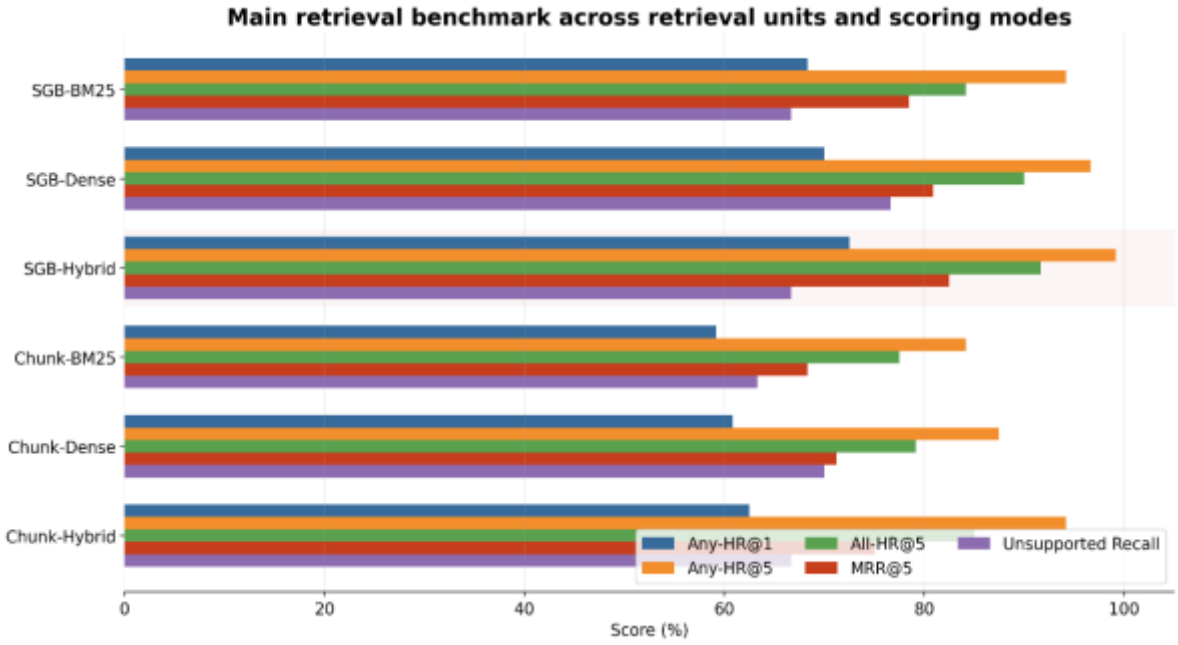


Figure 8. Retrieval performance across the six benchmark conditions.

4.5 Weight-Sensitivity Analysis

The balanced configuration ($w_s = 0.5, w_b = 0.5$) serves as a robust, non-tuned default that gives equal weight to both lexical and semantic retrieval vectors. As detailed in Table 3, the parametric sweep reveals that a semantic-heavy combination utilizing a (w_s = 0.75, w_b= 0.25) configuration achieved the highest overall performance across the benchmark. This optimized setting produced a perfect 100.0% Any-HR@5, a 95.0% All-HR@5, and a peak MRR@5 of 85.9%. These results establish that while basic keyword mapping is necessary, prioritizing dense vector alignments significantly accelerates the location of relevant engineering evidence.

Table 3. Sensitivity of SGB-Hybrid retrieval to semantic/BM25 weighting.

Semantic w	BM25 1−w	Any-HR@1	Any-HR@5	All-HR@5	MRR@5	Unsup. Recall
0.00	1.00	68.3%	94.2%	84.2%	78.5%	66.7%
0.25	0.75	70.0%	95.8%	86.7%	79.8%	66.7%
0.50	0.50	72.5%	99.2%	91.7%	82.5%	66.7%
0.75	0.25	76.7%	100.0%	95.0%	85.9%	73.3%
1.00	0.00	70.0%	97.5%	90.0%	81.0%	76.7%

Percentages are computed over the 120 supported benchmark questions, except Unsupported Recall, which is computed over the 30 unsupported benchmark questions.

This performance distribution indicates a critical operational trade-off within complex model interactions. Lexical matching through BM25 remains essential for pinning exact alphanumeric strings, specific

equipment codes, or distinct threshold tags common in municipal water infrastructure documents. However, because engineering queries often feature loose paraphrasing or highly localized phrasing variations, the retrieval system relies on strong dense semantic weights to bridge the gap between user intent and structured SGB database fields, as visually mapped in Figure 9.

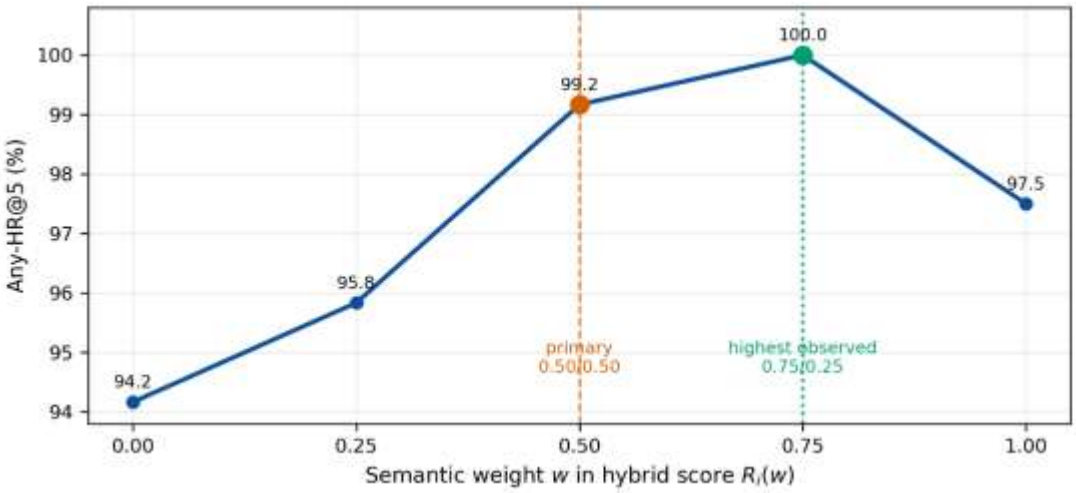


Figure 9. Sensitivity of retrieval performance to semantic/BM25 weighting.

4.6 Question-Type and Failure Analysis

Category-level performance metrics reveal that SGB-aware retrieval provides maximum operational utility when complete engineering trace integrity is paramount. The proposed SGB-Hybrid framework achieved a perfect 100.0% All-HR@5 score across combined-trace queries, paraphrased supported inputs, and reverse output-to-case conditions. Conversely, broad multi-answer retrieval emerged as the primary computational bottleneck, dropping to an All-HR@5 score of 62.5%. This localized performance drop stems from high semantic and lexical overlap across adjacent emergency-response scenarios, where multiple discrete system events share a highly homogeneous vocabulary, as visually itemized by category in Figure 10.

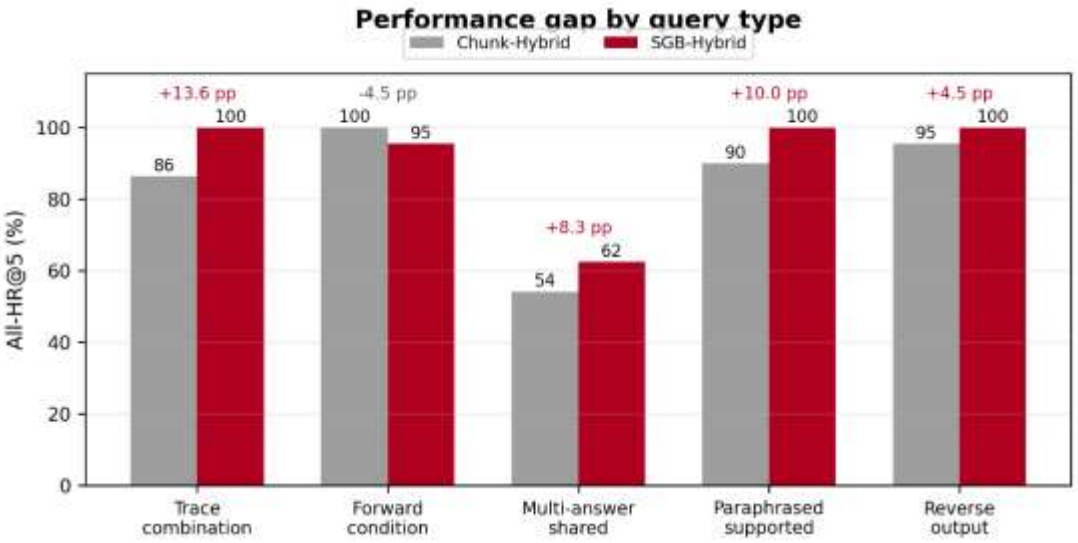


Figure 10. Retrieval performance by query type.

As detailed in Table 4, the system's top-1 error profile isolates these failure modes to specific boundary-evaluation challenges. For instance, biological-contamination escalation queries frequently retrieved a lower-tier, standard contamination event at rank 1 instead of the true, severe E. coli progression case. Because these operationally distinct records share ubiquitous domain keywords like "contamination," "water quality," and "dosage thresholds," uniform mathematical scoring fails to resolve differences in operational severity. In a municipal water decision-support environment, this ranking confusion can route an operator to an incorrect, low-severity procedure. Consequently, these systematic errors expose a clear methodological need for field-weighted scoring and threshold-aware matching layers capable of penalizing state-variable or numerical deviations during the ranking phase.

Table 4. Top-1 error profile for SGB-Hybrid and Chunk-Hybrid.

Method	Top-1 errors	Forward-condition	Paraphrased	Multi-answer	High-confusability	Other
SGB-Hybrid	33	9	8	7	9	0
Chunk-Hybrid	45	11	14	9	9	2

5. Discussion

Our evaluation reveals that the primary structural advantage of SGB-aware retrieval stems from maintaining cross-field dependency constraints rather than modifying the underlying scoring math.

5.1 Interpretation of Results

Our findings demonstrate that the main advantage of SGB-aware RAG comes from preserving behavior-model structure during retrieval. The improvement over chunk-based baselines is not explained by the scoring method alone, because BM25, dense retrieval, and hybrid scoring were also applied to the generic chunk baselines. The key difference is the retrieval unit: complete SGB records preserve relationships among input states, situations, goals, behaviors, output functions, assumptions, and trace identifiers.

The strongest gains occur in top-5 evidence recovery and complete-hit retrieval. Complete SGB records improve retrieval because each unit contains complementary fields: situation, goal, behavior, inputs, outputs, and trace information. A query that matches only one field can therefore still recover the full operational case. In contrast, chunk-based retrieval may isolate these fields into separate fragments, reducing evidence completeness and increasing ambiguity during ranking.

5.2 Comparison with Generic RAG, GraphRAG, and Agentic RAG

The proposed approach should not be interpreted as a replacement for general-purpose RAG, GraphRAG, or agentic RAG. Instead, it addresses a complementary problem: defining an appropriate retrieval unit for structured behavior models. Generic RAG is effective when evidence can be represented as independent passages, while GraphRAG explicitly models relationships among retrieved entities. More recent work has explored adaptive retrieval strategies, including one-shot retrieval under token-budget constraints and iterative agentic retrieval that progressively refines evidence through multiple retrieval rounds (Liang et al., 2025; Lin et al., 2025). These approaches improve retrieval strategies through planning, query reformulation, and adaptive evidence selection. In contrast, SGB-aware retrieval focuses on preserving the semantic integrity of engineering behavior models by treating each complete Situation-Goal-Behavior record as the atomic retrieval unit. These paradigms primarily improve retrieval strategy or evidence representation, but they do not inherently preserve the complete Situation-Goal-Behavior binding within the retrieved evidence.

SGB-aware RAG begins from the structure of the engineering artifact. The complete SGB case is treated as the atomic retrieval unit because the meaning of a behavior depends on its associated situation, goal, input states, output functions, assumptions, and trace identifiers.

5.3 Scope, Generalizability, and Future Directions

The proposed framework is intentionally designed for traceable interaction with structured behavior models, rather than as a general-purpose retrieval architecture for arbitrary document collections. Its primary design objective is to preserve the semantic relationships among situations, goals, behaviors and trace identifiers throughout the retrieval process. Accordingly, the evaluation focuses on whether complete-record retrieval improves evidence recovery for behavior-model interaction compared with conventional chunk-based retrieval.

The municipal-water emergency-response SGBM provides a controlled operational case study for evaluating this retrieval formulation. The purpose of the case study is not to demonstrate universal performance across all engineering domains, but to isolate the effect of the retrieval unit under realistic operational conditions. Because the retrieval objective is recovery of complete SGB records, the benchmark is intentionally aligned with this traceability objective. Consequently, the reported results should be interpreted as evaluating structure-preserving retrieval for behavior models, rather than as a universal

comparison against all document retrieval paradigms. Future studies should extend this evaluation to additional structured retrieval approaches, including table-aware retrieval, relational-query interfaces, graph-based retrieval, and larger industrial behavior models.

The present prototype also relies on a closed, third-party embedding and generation API to index SGB records and produce natural-language answers. For a research prototype this choice is reasonable, but an operational deployment over safety-relevant infrastructure models would need to address the data-governance, latency, and availability implications of routing SGB record content through an external service, particularly where municipal or critical-infrastructure data cannot leave a controlled environment. Testing local, self-hosted embedding and generation models, as noted below, is therefore an important direction for translating this prototype into an operational tool.

The proposed methodology is expected to generalize to engineering domains in which operational knowledge is represented as structured decision logic rather than independent narrative text. Examples include behavior models, operational procedures, emergency-response protocols, regulatory decision tables, financial decision frameworks, maintenance decision models, clinical pathways, maintenance troubleshooting guides, and safety-assurance artifacts. In these settings, explanation quality depends on preserving relationships among multiple fields rather than retrieving isolated passages. Conversely, for open-ended narrative corpora or knowledge collections where answers naturally reside in independent text passages, conventional RAG, GraphRAG, or hierarchical retrieval methods may remain more appropriate.

The evaluation deliberately emphasizes retrieval performance because reliable evidence recovery is a prerequisite for trustworthy language-model generation. Similarly, unsupported-query handling is treated as a safety-oriented retrieval requirement rather than an auxiliary evaluation metric. By explicitly measuring unsupported recall and false-supported behavior, the benchmark establishes a quantitative baseline for confidence-aware retrieval, abstention mechanisms, and uncertainty communication in future systems. These measurements provide actionable guidance for engineering retrieval systems intended for operational or safety-critical environments.

The error analysis further identifies opportunities for extending the retrieval formulation. Uniform lexical and semantic scoring occasionally confuses operationally adjacent cases that share similar terminology but differ in triggering conditions or severity thresholds. Rather than indicating a limitation of the structure-preserving retrieval concept, these results motivate the next generation of retrieval models that incorporate field-aware weighting, threshold-aware matching, structured re-ranking, and confidence calibration while preserving complete Situation–Goal–Behavior records as the fundamental retrieval unit.

6. Conclusions

By treating complete Situation–Goal–Behavior records as atomic units, this study addresses the structural fragmentation common to conventional chunk-based engineering retrievers. The approach treats the SGBM

as the authoritative behavior representation and uses RAG as an interaction layer. Its main design choice is to retrieve complete SGB records rather than arbitrary text chunks, preserving the relationships needed for model-grounded explanation and trace.

The study evaluates SGB-aware retrieval against generic chunk-based RAG baselines under BM25-only, dense-only, and hybrid retrieval modes. In a 150-question benchmark with 120 supported and 30 unsupported questions, SGB-Hybrid achieved 99.2% Any-HR@5, 91.7% All-HR@5, and 82.5% MRR@5 under the balanced 0.5/0.5 setting. Relative to Chunk-Hybrid, this corresponds to improvements of 5.0, 6.7, and 7.4 percentage points, respectively. These results support the central claim that SGB-aware retrieval units provide measurable value beyond generic chunking when the task requires traceable access to structured behavior-model evidence.

The weight sweep shows that the main conclusion is not dependent on a single weighting configuration. A semantic-heavy 0.75/0.25 hybrid achieved the highest observed retrieval performance, while the balanced 0.5/0.5 configuration provides a transparent primary reference setting. Remaining failures are concentrated in confusable, threshold-adjacent, and multi-answer cases, suggesting that future work should focus on field-aware ranking, threshold-aware matching, structured multi-case aggregation, and improved unsupported-query detection.

Future work will extend the benchmark to user-generated operational questions, evaluate answer-generation correctness after retrieval, compare against structured baselines such as relational and knowledge-graph retrieval, test local embedding and LLM models for restricted engineering environments, and examine how SGB-aware retrieval can support assurance-oriented review of behavior models before deployment. A particularly important extension is field-aware and threshold-aware ranking for confusable cases, consistent with prior work on SGB-based model analysis and reliability assurance for autonomous systems (Zheng et al., 2026).

Acknowledgments

The authors thank Dr. John Shortle, Shahab Aref, Fahimeh Ghorbani, Ashim Thapa, and Andrew Kirk for their review, comments, and suggestions.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data availability statement

The benchmark questions, ground-truth mappings, retrieval outputs, and evaluation scripts used in this study are available from the corresponding author upon reasonable request, subject to applicable institutional, security, confidentiality, and project-specific restrictions. The full SGB model is not publicly available in full, but may be shared in anonymized or reduced form where appropriate. Consistent with the

accompanying IISE Transactions reproducibility report, the cleaned Python evaluation scripts will be made publicly available through a stable repository such as GitHub or Zenodo upon acceptance.

Declaration of interest statement

The authors report there are no competing interests to declare.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this manuscript, the authors used OpenAI ChatGPT, GPT-5.5, to support language editing and manuscript formatting review. The tool was only used to improve clarity and readability. It was not used to generate the study design, experimental results, ground-truth mappings, evaluation data, analysis, or conclusions. The authors reviewed, revised, and verified all content and take full responsibility for the final manuscript.

References

Asai, A., Wu, Z., Wang, Y., Sil, A., & Hajishirzi, H. (2024, May 7–11). Self-RAG: Learning to retrieve, generate, and critique through self-reflection [Paper presentation]. International Conference on Learning Representations, Vienna, Austria. https://openreview.net/forum?id=hSyW5go0v8

Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Ness, R. O., & Larson, J. (2024). From local to global: A Graph RAG approach to query-focused summarization. *arXiv*. https://doi.org/10.48550/arXiv.2404.16130

Feary, M., Sherry, L., Palmer, E., & Polson, P. G. (2002). *A formal method for integrated system design: How to incorporate cognitive engineering principles in a systems engineering method*. NASA Ames Research Center.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2023). Retrieval-augmented generation for large language models: A survey. *arXiv*. https://doi.org/10.48550/arXiv.2312.10997

Gaw, N., Yousefi, S., & Reisi Gahrooei, M. (2022). Multimodal data fusion for systems improvement: A review. *IISE Transactions, 54*(11), 1098–1116. https://doi.org/10.1080/24725854.2021.1987593

Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W.-T. (2020). Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (pp. 6769–6781). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.550

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-T., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33* (pp. 9459–9474). Curran Associates.

Liang, J., Su, G., Lin, H., Wu, Y., Zhao, R., & Li, Z. (2025). Reasoning RAG via System 1 or System 2: A

survey on reasoning agentic retrieval-augmented generation for industry challenges. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics* (pp. 1954–1966). The Asian Federation of Natural Language Processing and The Association for Computational Linguistics. https://doi.org/10.18653/v1/2025.findings-ijcnlp.122

Lin, H., Su, G., Liang, J., Wu, Y., Zhao, R., & Li, Z. (2025). Fishing for answers: Exploring one-shot vs. iterative retrieval strategies for retrieval augmented generation. *arXiv*. https://doi.org/10.48550/arXiv.2509.04820

OpenAI. (n.d.-a). *Vector embeddings*. https://platform.openai.com/docs/guides/embeddings

OpenAI. (n.d.-b). *Create chat completion*. https://platform.openai.com/docs/api-reference/chat

Ram, O., Levine, Y., Dalmedigos, I., Muhlgay, D., Shashua, A., Leyton-Brown, K., & Shoham, Y. (2023). In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics, 11*, 1316–1331. https://doi.org/10.1162/tacl_a_00605

Sarthi, P., Abdullah, S., Tuli, A., Khanna, S., Goldie, A., & Manning, C. D. (2024, May 7–11). RAPTOR: Recursive abstractive processing for tree-organized retrieval [Paper presentation]. International Conference on Learning Representations, Vienna, Austria. https://openreview.net/forum?id=GN921JHCRw

Shen, B., & Kong, Z. (2024). Active defect discovery: A human-in-the-loop learning method. *IISE Transactions, 56*(6), 638–651. https://doi.org/10.1080/24725854.2023.2224854

Sherry, L., Baldo, J., & Berlin, B. (2021). Design of flight guidance and control systems using explainable AI. In *Proceedings of the 2021 Integrated Communications Navigation and Surveillance Conference* (pp. 1–10). IEEE. https://doi.org/10.1109/ICNS52807.2021.9441577

Sherry, L., Feary, M., Polson, P., Mumaw, R., & Palmer, E. (2001). *A cognitive engineering analysis of the vertical navigation (VNAV) function* (NASA/TM-2001-210915). NASA Ames Research Center.

U.S. Environmental Protection Agency. (2007). *The effectiveness of disinfectant residuals in the distribution system*. Office of Water.

U.S. Environmental Protection Agency. (2020). *Surface water treatment rule turbidity guidance manual* (EPA 815-R-20-004). Office of Water.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems 30*. Curran Associates.

Wiener, E. L. (1988). Cockpit automation. In E. L. Wiener & D. C. Nagel (Eds.), *Human factors in aviation* (pp. 433–461). Academic Press.

Xing, L., Zhao, G., Xiang, Y., & Liu, Q. (2021). A behavior-driven reliability modeling method for complex

smart systems. *Quality and Reliability Engineering International, 37*(5), 2065–2084. https://doi.org/10.1002/qre.2845

Yang, X., & Zhang, C. (2024). Online directed-structural change-point detection: A segment-wise time-varying dynamic Bayesian network approach. *IISE Transactions, 56*(5), 527–540. https://doi.org/10.1080/24725854.2023.2207252

Zheng, S., Lu, L., Hong, Y., & Liu, J. (2026). Planning reliability assurance tests for autonomous vehicles based on disengagement events data. *IISE Transactions, 58*(2), 131–146. https://doi.org/10.1080/24725854.2025.2475324