engineering proceedings

MDPI

Proceeding Paper

Beyond Whole-Model Vectors: Graph-Aware Multi-Vector Representation for System Dynamics Models †

Pavel Kyurkchiev * and Anton Iliev *

Faculty of Mathematics and Informatics, University of Plovdiv "Paisii Hilendarski", 4000 Plovdiv, Bulgaria
* Correspondence: pkyurkchiev@uni-plovdiv.bg (P.K.); aii@uni-plovdiv.bg (A.I.)
† Presented at the International Conference on Electronics, Engineering Physics and Earth Science (EEPES2026), Bandirma, Turkey, 24–27 June 2026.

Abstract

Traditional Single-Vector embeddings for System Dynamics models suffer from an "Information Bottleneck", where the specific characteristics of granular subsystems are diluted into a single dense representation. To address this limitation, we propose a graph-aware Multi-Vector indexing architecture. Our approach decomposes simulation graphs into typed structural segments, primarily Stock-Flow Subsystems and Causal Paths. Each segment is then embedded independently via local ONNX inference. Evaluated at the representation level using segment-level cosine similarity on a corpus of 63 models across 33 domain-specific conceptual targets, the proposed architecture outperformed whole-model matching, improving semantic alignment in the latent space for 55.84% of the evaluated pairs. Standard information retrieval ranking metrics (Precision@k and Mean Reciprocal Rank) indicate that the architecture measurably enhances semantic resolution while maintaining highly competitive downstream discoverability, providing a high-fidelity vector representation of localized simulation behaviors without model-wide context dilution.

Keywords: system dynamics; semantic search; multi-vector indexing; graph decomposition; vector embeddings; information retrieval

1. Introduction

As established in foundational literature [1,2], System Dynamics (SD) modeling is an indispensable methodology for simulating and analyzing complex processes across social, economic, and physical domains. As modeling platforms have transitioned into distributed, cloud-based environments, they have evolved into large-scale repositories of simulation models, making the effective discoverability and reuse of these artifacts a prominent challenge. The development of robust representation models is crucial for decision support systems, drawing on established data modeling techniques such as OLAP cubes for structured analysis [3]. Traditional Information Retrieval (IR) methods, which rely heavily on lexical keyword matching, consistently suffer from the "Vocabulary Mismatch problem" [4] when applied to visual programming structures, failing to capture the causal logic and mathematical dependencies distributed across a model's graph topology. To address this, recent advancements have introduced semantic search architectures that serialize SD graphs into textual descriptions and generate dense vector embeddings using local Large Language Models (LLMs) via ONNX inference [5]. Building upon our recent findings [6] and established Single-Vector Embedding strategies [7,8], this approach maps

Check for updates

Academic Editors: Teodor Iliev, Ivaylo Stoyanov, Grigor Mihaylov and Selahattin Kosunalp

Published: 18 September 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

an entire model into a unified 768-dimensional space, successfully eliminating information noise and measurably outperforming standard full-text search engines.

Despite its high precision for broad conceptual queries, the Single-Vector representation introduces a critical structural limitation known as the "Information Bottleneck". When the comprehensive semantic context of a large-scale, multi-sectoral SD graph is aggregated and compressed into a single dense vector, the granularity of individual components is inevitably diluted. For instance, if an industrial model contains distinct financial, demographic, and ecological sectors, the averaged whole-model vector may obscure the specific mathematical characteristics of those isolated subsystems. Consequently, conceptual targets aimed at highly localized model behaviors (such as a specific resource allocation loop or a distinct causal path) often fail to strongly align with the generalized, model-wide embedding.

To overcome this limitation, this paper introduces a graph-aware Multi-Vector Indexing architecture designed specifically for System Dynamics repositories. Moving beyond monolithic representations, our approach programmatically decomposes the SD graph into granular, typed structural segments. Specifically, the algorithm isolates distinct Stock-Flow Subsystems, Causal Paths, and Feedback Loops. Each extracted segment is independently serialized and vectorized via local ONNX inference, generating a localized semantic signature. During the retrieval phase, the system computes segment-level cosine similarities, allowing the architecture to align domain-specific conceptual descriptions against the precise mathematical substructure rather than the entire model's header.

The primary objective of this study is to propose and empirically evaluate the architectural feasibility of Multi-Vector graph decomposition against the baseline Single-Vector approach. By evaluating a corpus of 63 diverse SD models against 33 domain-specific conceptual targets, we aim to demonstrate that segment-level embedding yields enhanced semantic resolution in the dense vector space and translates into consistent end-to-end search ranking quality. Furthermore, this paper analyzes the inherent computational trade-off, contrasting the documented gains in semantic precision against the expected latency overhead during the embedding phase, thereby establishing a scalable foundation for granular, next-generation model discovery.

2. Methodology: Graph-Aware Multi-Vector Decomposition

To overcome the Information Bottleneck inherent in whole-model embeddings, we propose a shift from monolithic document indexing to a granular, segment-based representation. The proposed methodology processes the raw System Dynamics (SD) graph (represented in JSON format) through three sequential phases: structural segmentation, independent vectorization, and segment-level retrieval via a MaxSim-based scoring approach.

2.1. Structural Segmentation

Unlike standard Graph Neural Networks (GNNs) [9,10] that embed graphs holistically, our approach leverages domain-specific knowledge of System Dynamics topology. The graph is programmatically partitioned into overlapping, semantically meaningful subgraphs, referred to as semantic segments. The raw JSON graph is programmatically decomposed into three types of semantic segments: Stock-Flow Subsystems, Causal Paths, and Feedback Loops. Each segment is independently serialized and vectorized via local ONNX inference, storing the model as a "constellation" of high-resolution vectors in the Qdrant database. This structural decomposition allows the system to match queries against specific simulation mechanisms rather than a diluted whole-model representation (Figure 1).

https://doi.org/10.3390/engproc2026154095

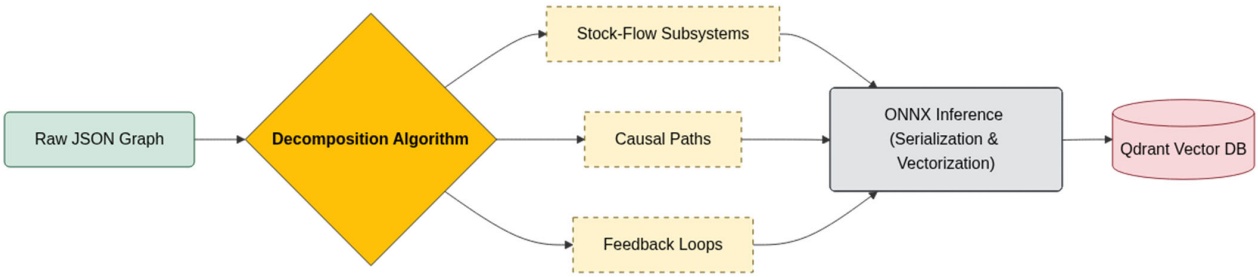


Figure 1. Architecture of the Multi-Vector indexing pipeline. Arrows indicate the direction of the data flow, from the raw JSON graph through the decomposition algorithm, which produces the three segment types (dashed boxes), to independent ONNX vectorization and storage in the Qdrant vector database. Colors are used for visual grouping only and carry no additional meaning.

The decomposition engine scans the graph for three primary topological constructs:

- Stock-Flow Subsystems: The fundamental physical or state accumulators in the model. The algorithm identifies state variables (Stocks) and their directly connected rate-of-change controllers (Flows/Valves).
- Causal Paths: Information and feedback transmission chains. The engine traverses sequences of auxiliary variables leading to a flow or stock, constrained to path lengths of 2 to 4 nodes to maintain semantic focus.
- Feedback Loops: Cyclic dependencies driving endogenous system behavior. The algorithm utilizes cycle detection logic (e.g., Depth-First Search with backtracking) to isolate closed loops within the graph matrix.

The foundational logic of this decomposition is formalized in Algorithm 1, which abstracts the extraction pipeline implemented in the core domain services. To ensure computational efficiency and semantic focus, the extraction engine employs strict deduplication and bounding rules. Overlapping segments are managed by a filtering logic based on node-set hashing. For instance, functionally redundant feedback loops containing identical node permutations are discarded. Furthermore, while Causal Paths are constrained to lengths of 2 to 4 nodes, Feedback Loop detection is bounded by a maximum traversal depth of 8 nodes to prevent computational overhead.

Algorithm 1. Graph-Aware Structural Decomposition

Input: Graph Model G = (V, E)
Output: List of Semantic Segments S
S ← ∅
// Phase 1: Extract Stock-Flow Subsystems
for each node ∈ V of type Stock do
    flows ← GetConnectedFlows(node, E)
    S ← S ∪ CreateSegment("StockFlowSubsystem", node ∪ flows)
end for
// Phase 2: Extract Causal Paths
for each path ∈ FindVariableChains(G) do
    if Length(path) ≥ 2 and Length(path) ≤ 4 then
        S ← S ∪ CreateSegment("CausalPath", path)
    end if
end for
// Phase 3: Extract Feedback Loops
cycles ← DetectCycles(G, max_depth=8)
for each cycle ∈ cycles do
    S ← S ∪ CreateSegment("FeedbackLoop", cycle)
end for
return S

https://doi.org/10.3390/engproc2026154095

On average, this focused decomposition yields 2.11 distinct valid segments per model.

2.2. Independent Vectorization

Once the graph is decomposed, each extracted semantic segment is processed independently. A serialization module translates the topological subset (nodes, connections, and metadata) into a cohesive natural language description.

These localized text representations are then passed to the local embedding service, built on Microsoft.ML.OnnxRuntime. Utilizing the nomic-embed-text-v1.5 transformer model [11], the system generates a distinct 768-dimensional dense vector for each segment. Consequently, a single complex simulation model is represented in the vector database (Qdrant) not as one averaged point, but as a constellation of vectors V_{S1}, V_{S2}, …, V_{Sn}, where each vector captures a high-resolution snapshot of a specific simulation mechanism.

2.3. Segment-Level Retrieval Strategy (MaxSim Approach)

During the retrieval phase, the Multi-Vector architecture requires an aggregation strategy to evaluate a model's overall relevance to a given user query Q. Averaging the segment vectors would reintroduce the Information Bottleneck. Therefore, we employ a Maximum Similarity (MaxSim) late-interaction approach, adapting the retrieval methodology introduced by Khattab and Zaharia [12].

The retrieval architecture embeds the user query into a query vector V_Q. It then calculates the cosine similarity between V_Q and every segment vector V_{Si} belonging to a model M. The final relevance score for the model is strictly defined by its highest-scoring segment:

$$\text{Score}(Q, M) = \max_{s \in \text{Segments}(M)} \cos(V_Q, V_s) \quad (1)$$

This ensures that if a user searches for a specific "inflation and money supply loop", the model is retrieved based solely on the strength of that specific FeedbackLoop segment, entirely unaffected by the noise of unrelated demographic or ecological subsystems within the same model file.

3. Experimental Setup

To empirically validate the effectiveness of the proposed graph-aware Multi-Vector architecture, a two-tiered evaluation strategy was established:

- Primary Comparative Baseline: The proposed Multi-Vector approach is primarily benchmarked against the traditional Single-Vector strategy, where the entire System Dynamics graph is serialized and compressed into a single monolithic dense embedding. This serves as the core evaluation to measure how well the architecture resolves the Information Bottleneck.
- Additional Structural Validation: To mathematically isolate the source of the architecture's performance, a Naive Chunking (NC) baseline was introduced as a secondary validation step. By blindly dividing the serialized graph into fixed-size text blocks without regard for graph topology, this validation tier explicitly tests whether potential retrieval gains stem genuinely from graph-aware semantic decomposition or merely from the processing of shorter, isolated text segments. Specifically, the serialized textual representation of the model was partitioned using a character-based sliding window, configured with a strict chunk size of 500 characters and a 50-character overlap.

3.1. Dataset and Query Configuration

The evaluation corpus consists of 63 diverse System Dynamics models, which were originally compiled and validated in our previous study [6]. These models were explicitly

selected to ensure a comprehensive multi-domain coverage, including population genetics, macroeconomic cycles, ecological sustainability, and industrial resource management. By utilizing this established dataset, we ensure that the Multi-Vector approach is tested against realistic, peer-reviewed graph topologies that vary significantly in complexity, providing a robust foundation for architectural comparison.

To simulate realistic knowledge-discovery scenarios, a test suite of 33 complex, domain-specific conceptual queries was formulated in collaboration with expert practitioners. These queries were designed to target localized feedback mechanisms and causal structures rather than broad model titles. Prior to the benchmark, the queries were manually categorized into two distinct cohorts: Specific/Structural (18 queries) and Broad/Conceptual (15 queries).

The classification followed systematic, predefined criteria: queries targeting explicit mathematical formulations or isolated causal paths (e.g., "Predator-prey dynamics using Rosenzweig MacArthur equations" or "Causal path from nutrient limitation to waste accumulation") were labeled as "Specific". Conversely, queries targeting macro-level policy outcomes or holistic system behaviors (e.g., "Integration of renewable energy and battery storage in power grids") were classified as "Broad". To ensure full methodological transparency and reproducibility, the complete list of all 33 queries, their exact classifications, and the 2079 unique query-to-model interactions are publicly accessible in the supplementary benchmark dataset repository.

3.2. Evaluation Metrics

The comparative analysis focused on both representation-level semantic alignment and ranking-level retrieval effectiveness, alongside computational efficiency. While the initial formulation focused heavily on latent space interactions, the evaluation was explicitly expanded to include standard ranking metrics to quantify practical top-k search performance. The following metrics were recorded for each query-model pair:

- Cosine Similarity Comparison: The system recorded the cosine similarity score of the traditional whole-model embedding (Single-Vector) and compared it against the maximum segment-level similarity (Multi-Vector Best) extracted from the decomposed graph.
- Improvement Ratio (Δ): The total percentage of interactions where the Multi-Vector MaxSim score strictly exceeded the Single-Vector score, indicating a stronger latent space similarity with the user's query intent.
- Standard Information Retrieval (IR) Metrics: To evaluate the end-to-end ranking quality and user-centric search effectiveness, Precision@1 (P@1), Precision@3 (P@3), and Mean Reciprocal Rank (MRR) were calculated across all evaluation baselines.
- Computational Latency and Overhead: To assess the architectural trade-off, the benchmark measured the average execution time (in milliseconds) for structural decomposition, Single-Vector embedding generation, naive text chunking, and exhaustive Multi-Vector embedding generation.
- Statistical Significance: A paired t-test ($\alpha = 0.05$) was applied to the resulting cosine similarity distributions to mathematically validate the consistency and reliability of the observed Multi-Vector improvements.

3.3. Technical Environment

To ensure the validity and reproducibility of the latency measurements, all vectorization processes were executed in an isolated, containerized environment. The specific hardware and software configurations of the experimental setup are detailed in Table 1.

Table 1. Experimental environment specifications.

Category	Component/Software	Specification
Hardware	CPU	AMD Ryzen PRO 7840U
	RAM	32 GB DDR5 6400 MT/s
Infrastructure	Container Engine	Docker Engine v29.1.3 (WSL2 Backend)
Data Storage	Vector DB	Qdrant v1.6.1 (Official Image)
Embedding Model	Framework	.NET 10.0 (C#)
	Inference Engine	Microsoft.ML.OnnxRuntime v1.23
	Embedding Model	nomic-embed-text-v1.5

4. Results and Discussion

To evaluate the efficacy of the proposed Multi-Vector architecture, the benchmark comprehensively analyzed 2079 query-to-model interactions, derived from evaluating 63 System Dynamics models against 33 complex conceptual queries. The results highlight a clear dichotomy between semantic granularity gains and the corresponding computational overhead.

4.1. Semantic Precision and Granularity Gains

The experimental data confirms that circumventing the Information Bottleneck via graph decomposition yields a measurable improvement in latent space similarity. When executing the MaxSim retrieval strategy over independently embedded segments, the Multi-Vector approach achieved an average cosine similarity of 0.5660. In contrast, the primary Single-Vector baseline recorded a lower average similarity of 0.5618, resulting in a statistically significant overall improvement ($\Delta = +0.0042$, $p < 0.0001$) across the 2079 interactions.

To verify that this improvement stems from the graph-aware decomposition rather than the use of shorter text segments, the results were compared against a Naive Chunking baseline. The naive approach recorded a substantially lower average cosine similarity of 0.4216. This decrease indicates that blind text segmentation disrupts the contextual and causal dependencies of System Dynamics models, reinforcing the premise that preserving graph topology is essential for accurate localized semantic representation.

From an information retrieval standpoint, this localized metric improvement is practically meaningful. In established late-interaction paradigms (such as the foundational ColBERT architecture), increasing the maximal segment-level similarity (MaxSim) is the critical mathematical prerequisite for improving downstream retrieval quality. By achieving closer semantic alignment for granular sub-graphs, the Multi-Vector indices generate a stronger localized signal for re-ranking engines to process.

More importantly, the decomposition algorithm successfully generated higher similarity scores for 55.84% of the evaluated pairs (1161 out of 2079 interactions). This indicates that in more than half of the search scenarios, a specific isolated sub-graph captured the user's intent more accurately than the monolithic whole-model vector. These results demonstrate that isolating structural segments measurably enhances semantic alignment in the latent space, particularly for specific/structural queries where the Multi-Vector approach outperformed the baseline in 60.85% of cases (Figure 2).

To quantitatively investigate the structural drivers behind this variance, the 33 conceptual queries used in the benchmark were explicitly categorized into two distinct cohorts: Specific/Structural queries (e.g., targeting explicit causal paths, isolated feedback loops, or strict mathematical formulations such as the Rosenzweig-MacArthur equations) and Broad/Conceptual queries (targeting macro-level model behaviors, general dynamics, and holistic resource management).

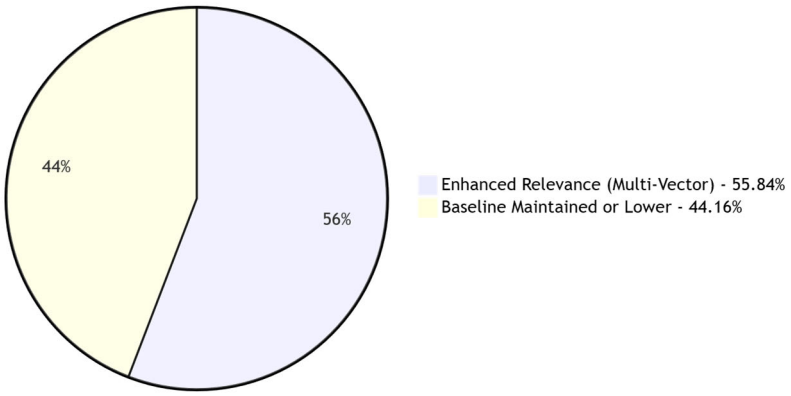


Figure 2. Semantic relevance improvement.

A granular breakdown of the dataset (Table 2) reveals a profound correlation between the semantic scope of the query and the optimal embedding strategy. For the Specific/Structural cohort (18 queries, 1134 interactions), the Multi-Vector MaxSim approach achieved a dominant improvement, outperforming the monolithic baseline in 60.85% (690 out of 1134) of the interactions. In these scenarios, isolating structural segments (primarily Stock-Flow Subsystems and Causal Paths) successfully captured the precise mathematical intent of the user without the semantic dilution caused by whole-model context.

Table 2. Retrieval performance by conceptual query scope.

Query Category	Query Count	Total Interactions	Multi-Vector Success ($\Delta > 0$)	Single-Vector Tie/Win ($\Delta \leq 0$)
Specific/Structural	18	1134	60.85% (690 pairs)	39.15% (444 pairs)
Broad/Conceptual	15	945	49.84% (471 pairs)	50.16% (474 pairs)
Global Baseline	33	2079	55.84% (1161 pairs)	44.16% (918 pairs)

Conversely, for the Broad/Conceptual cohort (15 queries, 945 interactions), the Multi-Vector pair-wise similarity success rate was 49.84%, with the traditional Single-Vector baseline retaining a higher or equal score in 50.16% of the cases (474 out of 945 interactions). This indicates that at the pure representation level, the holistic model context frequently maintains a strong generalized alignment for abstract intents. However, as demonstrated in the subsequent ranking analysis (Section 4.2), this marginal pair-wise density advantage does not translate into better retrieval order. The Multi-Vector MaxSim strategy ultimately compensates by filtering out macro-level noise, allowing localized signatures to drive more accurate end-to-end document ranking.

Analyzing the structural distribution reveals the specific mechanisms driving these matches. The decomposition algorithm generated an average of 2.11 valid segments per model. The retrieval engine disproportionately favored Stock-Flow Subsystem segments, which accounted for an average of 1.8 matching segments per model, followed by Causal Path segments at 0.3 per model. The data demonstrates that conceptual queries align exceptionally well with localized accumulator-flow interactions, which are otherwise semantically diluted when averaged with unrelated sectors in a single whole-model vector.

Illustrative Case Study

To practically illustrate how graph decomposition resolves the Information Bottleneck, Table 3 presents a direct query-to-model interaction from the benchmark dataset.

Table 3. Illustrative case study: resolving the Information Bottleneck via segment isolation.

Metric	Value
User Query	Herbivory influence on plant growth and water consumption
Target Model	Extended Model of Plant Growth with Light, Water, and Herbivory
Model Complexity	13 Nodes, 12 Links, 4 Semantic Segments
Baseline (Single-Vector) Score	0.7498
(Naive Chunking) Score	0.4630
MaxSim (Multi-Vector) Score	0.8279 (Improvement: +0.0781 over Single-Vector)
Matching Sub-graph	Segment Type: CausalPath Variables Isolated: Water → Herbivores

In this scenario, the target simulation model is complex, containing multiple dynamic sectors (light exposure, soil mechanics, water dynamics, and herbivory). When serialized holistically into a single document, the linguistic representation of the specific "water-to-herbivore" relationship is diluted by the surrounding environmental variables, resulting in a baseline similarity score of 0.7498. Similarly, when processed through the Naive Chunking baseline, the serialized graph is blindly segmented. This arbitrary division often severs the topological link between "water" and "herbivores" across different, disconnected text chunks, resulting in a degraded similarity score of 0.4630.

By evaluating the decomposed graph, the Multi-Vector architecture successfully isolated a specific Causal Path (Water → Herbivores). Because this extracted sub-graph aligned with the specific structural intent of the user's query, the localized dense embedding produced a stronger semantic signal (0.8279). This demonstrates how segment-level representations allow retrieval engines to match explicit mechanistic structures without being penalized by unrelated model context or falling victim to the structural fragmentation caused by naive text splitting.

4.2. Search-Result Ranking Evaluation

To comprehensively address the ranking-level effectiveness of the Multi-Vector graph decomposition, the retrieval performance was evaluated using standard Information Retrieval (IR) metrics: Precision at 1 (P@1), Precision at 3 (P@3), and Mean Reciprocal Rank (MRR). Table 4 summarizes the retrieval performance metrics across different query scopes, providing a granular breakdown of the ranking capabilities of each methodology.

Table 4. Evaluation of information retrieval (IR) metrics based on query scope.

Query Category	Retrieval Strategy	Precision@1	Precision@3	Mean Reciprocal Rank (MRR)
Specific/Structural	Single-Vector	0.9444	0.4074	0.9722
	Multi-Vector	0.9444	0.4259	0.9500
	Naive Chunking	0.0556	0.0556	0.1567
Broad/Conceptual	Single-Vector	0.8000	0.4667	0.8511
	Multi-Vector	0.8000	0.4889	0.8800
	Naive Chunking	0.1333	0.1111	0.2302
Overall Average	Single-Vector	0.8788	0.4343	0.9172
	Multi-Vector	0.8788	0.4545	0.9182
	Naive Chunking	0.0909	0.0808	0.1902

https://doi.org/10.3390/engproc2026154095

The empirical data demonstrates that the Multi-Vector approach yields highly competitive and consistently superior global retrieval signals compared to the traditional monolithic Single-Vector baseline.

On an overall level, both strategies achieved an identical top-1 retrieval accuracy (Precision@1 = 0.8788), indicating that segmenting the graph does not degrade the system's ability to position the most relevant model at the paramount rank. Furthermore, the proposed Multi-Vector architecture demonstrated superior overall ranking quality across the broader retrieval window, recording a higher Mean Reciprocal Rank (MRR = 0.9182 compared to Single-Vector 0.9172) and improved Precision@3 (0.4545 compared to 0.4343).

A granular analysis of the query cohorts reveals the underlying dynamics driving this performance shift:

- Specific/Structural queries: Both approaches maintained exceptionally high top-1 accuracy (Precision@1 = 0.9444). However, the Multi-Vector approach improved model discoverability within the top-3 window, increasing Precision@3 from 0.4074 to 0.4259. A marginal decrease in MRR (0.9500 vs. 0.9722) was observed, stemming from minor ranking permutations below the top rank that do not practically impair the retrieval of targeted artifacts.
- Broad/Conceptual queries: Contrary to the theoretical concern that subgraph fragmentation might hinder the processing of abstract, macro-level concepts, the Multi-Vector paradigm categorically outperformed the Single-Vector baseline across all IR metrics in this cohort. The Multi-Vector architecture matched the Single-Vector baseline in Precision@1 (0.8000 vs. 0.8000) and achieved an elevated MRR (0.8800 vs. 0.8511), along with a superior Precision@3 (0.4889 vs. 0.4667). This empirical evidence indicates that the Maximum Similarity (MaxSim) late-interaction strategy successfully leverages localized, high-fidelity semantic signatures (predominantly from isolated Stock-Flow Subsystems) to generate a cleaner relevance signal, effectively circumventing the semantic noise inherent to whole-model aggregation.

Finally, the evaluation clearly illustrates the failure of Naive Chunking, which yielded substantially lower results (an overall MRR of merely 0.1902 and a Precision@1 of 0.0909). This massive metric discrepancy mathematically proves that blind textual splitting destroys the causal and contextual dependencies of System Dynamics models. Strict adherence to graph topology during structural decomposition is an absolute architectural prerequisite for accurate, systemic model retrieval.

4.3. The Computational Trade-Off

The initial topological extraction involves reading the JSON matrix and isolating the subgraphs. This process is computationally trivial, with structural graph decomposition averaging merely 0.17 ms per model. However, the local ONNX inference utilized for generating the dense vector embeddings acts as the primary system bottleneck. Under the baseline Single-Vector strategy, parsing and embedding the entire model sequentially averaged 22,006.69 ms. Conversely, the Multi-Vector approach, which requires independent transformer inference passes for every extracted segment (averaging 2.11 segments per model), recorded an average embedding latency of 46,475.78 ms per model, maintaining a systemic overhead factor of 2.1×. Crucially, the alternative baseline, Naive Chunking, introduced a higher latency of 118,904.27 ms per model. This is driven by the fact that blind text splitting generates an average of 5.40 chunks per model, which reduces offline ingestion efficiency (Figure 3).

https://doi.org/10.3390/engproc2026154095

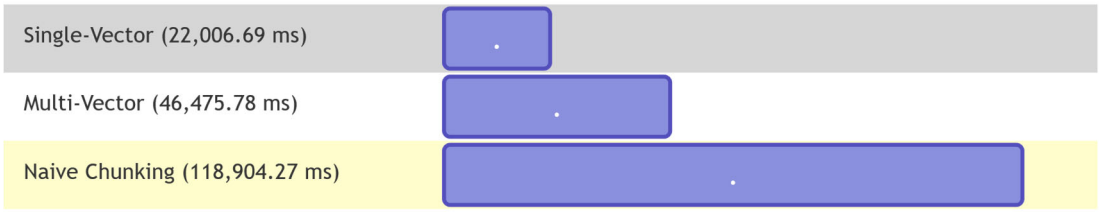


Figure 3. Embedding latency trade-off. Bar length is proportional to the average embedding latency per model. The values are given in the labels. Colors are used to distinguish the three strategies and carry no additional meaning.

From an architectural standpoint, this trade-off is justified. The computational penalty is incurred entirely during the asynchronous indexing phase. Because the Multi-Vector indices are pre-computed, the end-user query execution merely performs a MaxSim dot-product operation across the vector space. Regarding storage overhead, the decomposition algorithm generates an average of only 2.11 vectors per model. By comparison, the Naive Chunking approach generates an average of 5.40 vectors per model, more than doubling the storage requirements of the Multi-Vector architecture without delivering any retrieval benefits. In practical deployment, the vector database (Qdrant) increases in size by roughly a factor of 2.1× compared to the Single-Vector baseline. For modern vector databases, this storage footprint is negligible, and query-time execution latency remains virtually identical to the baseline. Thus, the system mitigates the Information Bottleneck, outperforming both monolithic and naively chunked representations, and improves semantic alignment for 55.84% of queries without degrading real-time search responsiveness.

5. Conclusions

In this study, we addressed the Information Bottleneck inherent to traditional Single-Vector representations of System Dynamics models. By introducing a graph-aware Multi-Vector indexing architecture, we successfully transitioned from monolithic document embeddings to localized, semantically cohesive representations. Our algorithmic approach programmatically decomposed complex simulation graphs into typed structural segments, predominantly isolating distinct Stock-Flow Subsystems and Causal Paths.

Empirical evaluation indicates that this granular approach offers a viable alternative to traditional representation methods. Utilizing a Maximum Similarity (MaxSim) retrieval strategy over independent segment embeddings generally outperformed both the monolithic whole-model baseline and the naive text chunking alternative in our test corpus. By matching user queries to localized mathematical substructures rather than broader global contexts, the Multi-Vector architecture showed measurable improvements in latent space alignment, particularly for targeted structural queries. Additionally, it appeared more resilient to semantic noise, maintaining competitive overall ranking quality (MRR) for broad, conceptual searches, while mitigating the fragmentation issues often associated with arbitrary text segmentation.

These potential gains in semantic precision come with an inherent computational trade-off. Generating independent transformer representations for multiple segments introduces a systemic latency overhead compared to a Single-Vector approach. However, because this computational cost is isolated to the asynchronous offline ingestion phase, and the resulting vector footprint remains relatively compact, real-time search responsiveness is largely preserved. Ultimately, the proposed graph-aware methodology provides a promising foundation for high-resolution retrieval, aiding in the discovery of localized mechanisms within complex simulation models while supporting macro-level discoverability.

https://doi.org/10.3390/engproc2026154095

While the Multi-Vector dense embedding approach measurably enhances conceptual retrieval and structural alignment, dense transformer models inherently prioritize semantic proximity over exact-match precision. Consequently, the proposed architecture may still exhibit limitations when processing conceptual targets that demand strict terminological constraints, such as specific variable nomenclature or explicit mathematical constants.

Furthermore, the minor ranking permutations observed in highly targeted queries (e.g., MRR slightly dropping from 0.9722 to 0.9500 for the Specific/Structural cohort) underscore a fundamental characteristic of purely dense retrieval approaches: they prioritize semantic proximity over exact topological bounds. To address this limitation and enforce strict terminological constraints, our objective is to develop a hybrid search architecture.

To achieve this, we plan to integrate sparse lexical scoring algorithms (such as BM25 [13]) applied directly at the segmented sub-graph level. We aim to fuse these complementary signals through Reciprocal Rank Fusion (RRF) [14], a rank-based fusion strategy widely adopted in hybrid retrieval systems. This fusion combines the structural abstraction of dense vectors with the exact-match accuracy of sparse lexical indices. Coupled with late-interaction re-ranking strategies, this Multi-Vector hybrid fusion aims to establish a comprehensive, end-to-end retrieval pipeline capable of robust, domain-specific discovery within large-scale System Dynamics repositories.

While the current empirical evaluation provides strong initial evidence for the foundational advantages of graph-aware decomposition using representation-level similarity and precision ranking metrics, future evaluations will expand to larger, web-scale model repositories. This will allow for the assessment of top-k retrieval quality under massive noise conditions and the incorporation of user-in-the-loop relevance judgments to further refine downstream re-ranking algorithms.

Author Contributions: Conceptualization, P.K.; methodology, P.K.; software, P.K.; validation, A.I.; formal analysis, P.K. and A.I.; investigation, P.K.; resources, P.K. and A.I.; data curation, A.I. and P.K.; writing—original draft preparation, P.K.; writing—review and editing, P.K. and A.I.; visualization, A.I. and P.K.; supervision, A.I.; project administration, P.K.; funding acquisition, A.I. All authors have read and agreed to the published version of the manuscript.

Funding: This study is financed by the European Union-NextGenerationEU, through the National Recovery and Resilience Plan of the Republic of Bulgaria, project No BG-RRP-2.004-0001-C01.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The System Dynamics model corpus utilized in this study was originally compiled in our previous work [6] and is openly available on Zenodo (https://doi.org/10.5281/zenodo.18047724). The raw experimental datasets and query configurations generated specifically for the Multi-Vector evaluation are available in a separate Zenodo repository (https://doi.org/10.5281/zenodo.19442824).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SD	System Dynamics
IR	Information Retrieval
P@1	Precision at 1
P@3	Precision at 3
MRR	Mean Reciprocal Rank
LLMs	Large Language Models

GNNs	Graph Neural Networks
MaxSim	Maximum Similarity
RRF	Reciprocal Rank Fusion
JSON	JavaScript Object Notation
ONNX	Open Neural Network Exchange

References

1. Sterman, J.D. *Business Dynamics: Systems Thinking and Modeling for a Complex World*; McGraw-Hill: New York, NY, USA, 2000.
2. Homer, J.B.; Hirsch, G.B. System dynamics modeling for public health: Background and opportunities. *Am. J. Public Health* **2006**, *96*, 452–458. [CrossRef] [PubMed]
3. Scerbakov, N.; Schukin, A.; Rezedinova, E. Usage of OLAP Cubes as a Data Model for DSS. *Eng. Proc.* **2025**, *104*, 4. [CrossRef]
4. Furnas, G.W.; Landauer, T.K.; Gomez, L.M.; Dumais, S.T. The vocabulary problem in human-system communication. *Commun. ACM* **1987**, *30*, 964–971. [CrossRef]
5. ONNX: Open Neural Network Exchange. Available online: https://github.com/onnx/onnx (accessed on 3 April 2026).
6. Kyurkchiev, P.; Iliev, A.; Kyurkchiev, N. Semantic Search for System Dynamics Models Using Vector Embeddings in a Cloud Microservices Environment. *Future Internet* **2026**, *18*, 86. [CrossRef]
7. Zhai, Q.; Zhu, W.; Zhang, X.; Liu, C. Contrastive Refinement for Dense Retrieval Inference in the Open-Domain Question Answering Task. *Future Internet* **2023**, *15*, 137. [CrossRef]
8. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), Hong Kong, China, 3–7 November 2019; pp. 3982–3992. [CrossRef]
9. Hamilton, W.L.; Ying, R.; Leskovec, J. Representation Learning on Graphs: Methods and Applications. *IEEE Bull. Tech. Comm. Data Eng.* **2017**, *40*, 52–74. [CrossRef]
10. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P.S. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 4–24. [CrossRef] [PubMed]
11. Nussbaum, Z.; Morris, J.; Duderstadt, B.; Mulyar, A. Nomic Embed: Training a Reproducible Long Context Text Embedder. *arXiv* **2024**, arXiv:2402.01613. [CrossRef]
12. Khattab, O.; Zaharia, M. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Xi'an, China, 25–30 July 2020; pp. 39–48. [CrossRef]
13. Robertson, S.; Zaragoza, H. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* **2009**, *3*, 333–389. [CrossRef]
14. Cormack, G.V.; Clarke, C.L.A.; Büttcher, S. Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods. In Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Boston, MA, USA, 19–23 July 2009; pp. 758–759. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.