Semantic Tensor Memory

Applications, Failure Modes, and Implementation Strategy
for Applied Behavior Analysis

Joshua Farrow
December 2025

Abstract

This document presents a comprehensive analysis of Semantic Tensor Memory (STM) as applied to clinical documentation in Applied Behavior Analysis (ABA). STM is a memory architecture that preserves event-level embeddings across ordered steps, enabling measurement of semantic drift, inflection detection, and concept evolution over time. We examine high-impact clinical applications, systematically analyze failure modes with mitigation strategies, and provide detailed architectural requirements for production deployment. This work extends the theoretical STM framework into practical operational guidance for healthcare technology implementations.

Intended Use and Pilot Scope

This document focuses on retrospective quality assurance, documentation integrity, and audit readiness for ABA clinical operations. STM outputs are presented as evidence for human review, not automated clinical decisions.

STM is designed to:

- Surface documentation-protocol mismatches (ISP/BIP/etc. vs session notes) and procedural drift.
- Support supervision/QA/AI-generated summary validation.

STM is not designed to:

- Diagnose patients, recommend clinical interventions, or provide real-time safety predictions.
- Replace clinical judgment; it produces audit-ready evidence artifacts for review.

1. Introduction

Applied Behavior Analysis generates rich longitudinal documentation: session notes written by Registered Behavior Technicians (RBTs), supervision notes from Board Certified Behavior Analysts (BCBAs), progress reports, treatment plans, and caregiver communications. This documentation serves multiple purposes: clinical decision-making, insurance justification, regulatory compliance, and quality assurance.

Current approaches to analyzing this documentation rely on keyword searches, manual review, or simple sentiment analysis. These methods fail to capture the fundamental clinical signal: how

meaning changes over time. A patient's trajectory from "severe aggression with property destruction" to "verbal protest with redirection" represents profound clinical progress, but keyword-based systems see only isolated snapshots.

Semantic Tensor Memory addresses this gap by making temporal semantic structure explicit and queryable. Rather than compressing documents into point vectors for similarity search, STM preserves the internal structure of each document as an event embedding matrix, enabling drift measurement, trajectory analysis, and alignment-based comparisons across time.

2. QA, Documentation Integrity, and Operational Excellence

This section describes high-impact retrospective applications for documentation QA, audit readiness, and operational excellence. Outputs are evidence artifacts intended for human review and workflow prioritization.

2.1 2.1 Documentation Integrity & Audit Readiness: Protocol-to-Note Consistency

A primary audit and billing risk is drift between written protocols (ISP/BIP, treatment plans) and what is documented as delivered in session notes. STM operationalizes this by embedding protocol anchors and measuring semantic drift of ongoing documentation from those references. Procedural drift is a pervasive challenge in ABA service delivery. Staff gradually modify procedures over time, often without awareness, until implemented treatment diverges substantially from the written protocol.

STM can operationalize fidelity monitoring by creating "gold standard" embeddings from protocol descriptions and measuring ongoing drift of session documentation from these anchors. For example, a DRO procedure description can generate a reference embedding; notes that drift semantically away from this reference indicate procedural degradation requiring retraining.

This approach scales across entire organizations. Regional or site-level aggregation of protocol-to-note drift metrics can identify systemic training needs before they manifest in audit findings or operational performance issues.

2.2 Documentation Integrity and Individualization

Audits scrutinize authenticity, individualization, and documentation quality. STM detects low-variation or templated documentation patterns while differentiating legitimate clinical stability from copy-paste behavior using context and trajectory features.

STM's trajectory stagnation detection directly supports this use case. When semantic velocity approaches zero for extended periods, documentation can be surfaced for review. Legitimate clinical stability can be distinguished from documentation laziness by examining whether context vectors show variation even when narrative content is similar.

For maintenance-phase justification, STM can demonstrate that while the topic remains stable (low content drift), the context is changing (for example, clinic to school to community). This supports evidence of generalization, which is often a key justification for ongoing services.

2.3 AI-Generated Content Validation

As AI tools increasingly assist with documentation, quality assurance becomes critical. AI-generated summaries can omit important details, introduce errors, or drift from source content.

STM provides a principled approach to summary validation by treating raw session logs and AI summaries as parallel sources for the same session. The semantic distance between source and summary should remain low; elevated distance indicates the summary has drifted from the source and should be reviewed or regenerated.

3. Failure Mode Analysis

STM is a probabilistic system built on embedding models trained for general language understanding, not clinical documentation. Understanding its failure modes is essential for responsible deployment.

3.1 False Positives: Detecting Change That Isn't There

The Vocabulary Illusion

Scenario: A new RBT takes over a case. The previous technician wrote "client had a tantrum." The new technician writes "client engaged in emotional dysregulation."

The error: STM calculates high semantic velocity because "tantrum" and "dysregulation" occupy distant regions of embedding space, despite describing identical behavior.

Impact: The system reports "significant clinical change" when the only change was vocabulary preference.

Mitigation: Display token alignment heatmaps showing which specific words drove the drift signal. When users see "tantrum" aligning to "dysregulation," they can dismiss the alert. Additionally, domain-specific embedding fine-tuning can reduce distances between clinical synonyms.

The Jargon Trap

Scenario: An RBT describes an event plainly: "He got mad when I took the iPad." The BCBA describes it technically: "Tangible denial evoked an extinction burst."

The error: STM flags high inter-observer disagreement because general-purpose embedding models may not recognize these as equivalent descriptions.

Impact: Clinical directors investigate a "reality gap" that doesn't exist, creating unnecessary work and eroding trust in the system.

Mitigation: Train a domain-adapted embedding model or adapter layer that maps clinical jargon to plain-language equivalents. Alternatively, implement preprocessing that normalizes terminology before embedding.

Context Collapse in Anomaly Surfacing

Scenario: A patient begins "pacing" before sessions. Historically, pacing preceded aggressive episodes. Today, the patient is pacing because they're excited about a planned trip to the park.

The error: The model sees the "pacing" representation entering a historical incident-associated cluster and surfaces a high-deviance anomaly flag.

Impact: Alarm fatigue. Staff stop trusting QA signals after repeated false positives.

Mitigation: Context conditioning becomes critical, but manual tagging is often unreliable—the same documentation gaps that create the alert also make affect labels inconsistent. Where manual tagging is unreliable, STM should first leverage structured metadata already present in operational systems (time of day, location/site, service setting, activity type, staff role, and scheduled program block) to automatically condition embeddings. Optional clinician tags can serve as overrides or clarifiers, but the default path should be metadata-driven conditioning. Anomaly surfacing should also require multiple converging signals, not single-token triggers.

3.2 False Negatives: Missing Change That Occurred

The Negation Blind Spot

Scenario: A patient progresses from "Hit peer" to "Did not hit peer."

The error: Many embedding models (particularly smaller ones like MiniLM) place these sentences in close proximity because they share core concepts ("hit," "peer"). The negation does not shift the vector sufficiently.

Impact: STM reports low drift or stagnation, missing the most important clinical success.

Mitigation: Use larger embedding models (for example, MPNet or higher-quality API embeddings) with better negation handling. Implement explicit negation detection preprocessing that flags or transforms negated phrases. Consider sentence-level embeddings for progress detection when appropriate.

3.3 Systematic Vulnerabilities

The ChatGPT Evasion

Scenario: A clinician uses ChatGPT to "rewrite this note in 5 different styles" to create artificial variation.

The error: Vocabulary varies enough to create apparent semantic drift while underlying meaning is identical.

Impact: STM sees healthy variation and marks notes as unique/high integrity, failing to catch fraudulent behavior.

Mitigation: Avoid relying on third-party "AI detectors," which are often brittle and can produce false positives. Instead, treat this as a stylometric and distributional problem. The primary risk signal is the pattern "high vocabulary variance + low semantic drift" (unusual surface-level rewrites with stable underlying meaning). Supplement this with lightweight stylistic fingerprinting against the author's baseline (sentence-length distribution, function-word rates, template repetition), and optional perplexity/entropy checks using an internal language model. These signals should be used for prioritization and review, not as a definitive fraud label.

4. Implementation Architecture

4.1 Storage Strategy

Production STM deployment requires careful storage design that supports both the ragged tensor structure required for drift computation and efficient structured queries for filtering.

The Hybrid Data Lake Approach

Vector databases (Pinecone, Weaviate) are optimized for similarity search but poorly suited to STM's requirements. They cannot efficiently perform the joins required for multi-source comparison (for example, "get RBT vector AND BCBA vector for session X") and do not natively support ragged tensor structures.

The recommended architecture uses Parquet files as the primary storage format with optional LanceDB indexing for production scale. Parquet provides columnar storage with native support for nested and complex types, enabling efficient partial loads (for example, retrieve only behavior counts for 1,000 patients without loading embedding vectors).

Recommended schema per row (session):

- session_id (String): Unique identifier
- timestamp (DateTime): Session date/time

protocol_version_id (String): Identifier for the active protocol/BIP/ISP version in effect for this session (prevents false drift when protocols change).

- source_role (String): "RBT", "BCBA", "Parent", "AI_Summary"
- metrics_json (Struct): {duration, frequency, phase, setting}
- event_vectors (Binary/List): The Mt matrix as compressed array
- step_vector (List[Float]): The zt sequence embedding

Validation hooks

To make QA outputs defensible and continuously improvable, storage should also capture review and ground-truth metadata alongside embeddings. This enables expert calibration, retrospective cohort evaluation, and threshold tuning without reprocessing raw PHI exports.

- review_labels (Struct): human review outcomes (e.g., "true protocol mismatch", "vocabulary shift", "false positive").

- incident_tags (Optional): retrospective event tags (if available) for cohort analysis (e.g., restraint, injury, escalation).
- threshold_version / model_version: the exact configuration used to generate each artifact.
- artifact_manifest (JSON): pointers to rendered reports, alignment maps, and extracted evidence spans for traceability.

Scaling Considerations

For pilot deployments (hundreds to thousands of sessions), local PyTorch session files with a Parquet metadata index are sufficient. The architecture supports migration to LanceDB or cloud storage as scale increases.

Matryoshka embeddings (for example, OpenAI text-embedding-3-large) enable cost/quality tradeoffs: store full-dimension vectors for audit-critical data and reduced-dimension vectors for routine analysis.

4.2 Hybrid Query Engine

Clinical queries combine structured filters with semantic search. The query "show me the last episode longer than 10 minutes involving aggression toward peers" requires:

- Structured filter: WHERE duration_minutes > 10
- Semantic rank: CosineSimilarity(event_vectors, embed("aggression toward peers"))
- Temporal sort: ORDER BY timestamp DESC LIMIT 1

This filter-then-rank pattern leverages the Parquet schema for exact filtering before loading embedding vectors, minimizing memory and compute requirements.

Validation sampling

The same query layer can support validation and monitoring by generating stratified review samples (e.g., top drift events, low-velocity cases, high anomaly scores) and tracking reviewer dispositions over time.

- Weekly stratified sample: top-N protocol mismatch scores, top-N stagnation flags, random baseline sample.
- Bias audit slices: sample across sites, staff, settings, age groups, and payor types (where available).
- Drift-of-the-detector: monitor distribution shifts in the STM scores themselves across months.

4.3 Embedding Model Selection

Model choice significantly impacts performance on clinical documentation:

- MiniLM (all-MiniLM-L6-v2): Fast, local, but weak on negation and clinical terminology. Suitable for prototyping.
- MPNet (all-mpnet-base-v2): Better semantic coverage, still local. Recommended for pilot deployments.
- OpenAI/Voyage: Best semantic quality, Matryoshka support, but requires API calls and introduces data residency considerations.

- Domain-adapted models: Fine-tuning or adapter training on clinical documentation provides the best clinical synonym handling but requires investment in training infrastructure.

5. Governance, Privacy, and Safety

Token and event-level memory can expose sensitive clinical information. Healthcare deployments require comprehensive governance frameworks.

5.1 Data Minimization

Store only what analysis requires. Consider span-level events (extracted clinical entities) rather than raw token embeddings to reduce identifiability while preserving analytical capability.

5.2 Access Control

Implement per-session and per-field access policies. Clinical staff may access their assigned patients; supervisors access their supervisees; compliance officers access aggregate metrics without individual embeddings.

5.3 Retention and Deletion

Establish clear retention policies aligned with regulatory requirements (HIPAA and applicable state regulations). Deletion workflows must remove both raw text and derived embeddings; embeddings can contain recoverable information about source content.

5.4 Audit Logging

Log retrieval queries, retrieved evidence, and downstream outputs. Maintain audit trails showing what information informed which clinical or administrative decisions.

5.5 Bias Monitoring

Drift metrics can amplify spurious correlations present in embedding models. Evaluate metrics across demographic cohorts, clinical sites, and staff populations to identify systematic biases. Models trained on general text may encode harmful associations that surface in clinical contexts.

6. Validation Framework (Operationalized)

Validation is part of the implementation. STM metrics should be calibrated and monitored using stored review labels, model/threshold versions, and stratified sampling workflows built into the query layer (see Sections 4.1–4.2).

6.1 Expert Calibration

Clinical experts should blind-review documentation samples and rate quality and consistency. STM metrics should correlate with expert ratings. Systematic disagreements identify metric weaknesses.

Calibration success should be defined in buyer terms. For confident audit readiness, a high-recall state matters most: STM should surface ~95% of true issues (protocol mismatch, missing required

elements, templated/copy-forward patterns), even if that entails a manageable false-positive review load. Missing a violation (false negative) is typically far costlier than reviewing a benign note (false positive). Thresholds and UI messaging should be tuned to keep recall high while keeping review volume operationally tolerable.

- Use stratified samples: include high protocol-mismatch, high stagnation, and random controls.
- Require reviewers to choose a disposition category (e.g., "true mismatch", "terminology shift", "missing detail", "false flag").
- Tune thresholds and UI messaging based on dismissal patterns and agreement rates.

6.2 Retrospective Incident Review

For incident review applications, analyze whether STM trajectory features would have surfaced known incidents in retrospective cohorts. Calculate sensitivity/specificity across historical events, and ensure the outputs remain framed as QA evidence for review, not real-time prediction.

- Define cohorts using incident_tags where available (Section 4.1).
- Report performance as "surface rate" and reviewer usefulness, not clinical risk prediction.
- Monitor false-positive burden to avoid alarm fatigue.

6.3 A/B Testing

Before organization-wide deployment, pilot STM-informed workflows with randomized comparison groups. Measure whether STM guidance improves supervision efficiency or compliance metrics relative to standard practice.

- Primary metrics: reviewer time-to-triage, protocol mismatch resolution rate, reduction in cloned/templated note patterns.
- Secondary metrics: staff retraining completion, site-level drift reduction, audit finding rate (where measurable).
- Gate expansions on sustained improvements and stable false-positive rates.

7. Conclusion

Semantic Tensor Memory offers a principled approach to longitudinal clinical documentation analysis that addresses fundamental limitations of current vector store and keyword-based methods. By preserving temporal structure and event-level granularity, STM enables drift detection and inter-observer comparison capabilities unavailable in existing systems.

However, STM is not a panacea. Its reliance on embedding models introduces systematic failure modes that require mitigation through interface design, domain adaptation, and human oversight. The most critical insight is that STM should augment rather than replace clinical judgment, presenting evidence for human interpretation rather than automated decision-making.

For ABA specifically, the alignment between STM's capabilities and operational needs is strong. Documentation quality drives authorization and audit outcomes, while protocol consistency is a

persistent pain point. These are problems where semantic trajectory analysis can provide genuine operational value.

The path forward requires rigorous validation: calibrating thresholds through expert review, measuring usefulness in real QA workflows, and continuously monitoring for emergent failure modes. Organizations willing to invest in this validation work have the opportunity to pioneer a fundamentally new approach to documentation intelligence.

References

1. Farrow, J. (2025). Semantic Tensor Memory: Interpretable, Drift-Aware Memory for Tracking Meaning Across Time, Events, and Context. Working paper.
2. Lewis, P. et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. NeurIPS.
3. Devlin, J. et al. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805.
4. Reimers, N. & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv:1908.10084.
5. McInnes, L., Healy, J., & Melville, J. (2018). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv:1802.03426.
6. Kuhn, H. W. (1955). The Hungarian Method for the Assignment Problem. Naval Research Logistics Quarterly.

Appendix A: Glossary of Terms

ABA (Applied Behavior Analysis): A scientific discipline focused on applying behavioral principles to improve socially significant behaviors.

BCBA (Board Certified Behavior Analyst): A certified practitioner who designs and oversees ABA treatment programs.

RBT (Registered Behavior Technician): A paraprofessional who implements ABA treatment under BCBA supervision.

Semantic Drift: The change in meaning or linguistic representation over time, measured through embedding space distance.

Event Embedding: A vector representation of a token, phrase, or semantic unit within a document.

Ragged Tensor: A tensor structure where one or more dimensions have variable length across entries.

Centroid Drift: Change in the average embedding vector across time steps.

Semantic Velocity: Rate of change in semantic representation over time.

Semantic Acceleration: Rate of change in semantic velocity, indicating inflection points.

Hungarian Alignment: An algorithm for optimal one-to-one matching between event sets, used to compute assignment-based drift.

Context Conditioning: Prepending structured metadata to text before embedding to encode contextual information.

Temporal Batching: Aggregating multiple time steps into windows before computing drift metrics.