ISVS: A Proposed Architecture and Research Roadmap for Non-Linear Media Navigation

Syed Muhammad Aqdas Rizvi

Date: April, 2026

Abstract

This specification introduces Interactive Semantic Video Seeking (ISVS), an agentic non-linear navigation framework that treats media retrieval as probabilistic search over a semantic graph rather than scalar movement on a chronological timeline. The system combines low-latency client-side bitstream analysis, edge-side lightweight inference, and asynchronous server-side multimodal indexing to decouple semantic navigation from strict chronological constraints. ISVS integrates robust normalization, gated multimodal fusion, sequential state estimation, and feedback-driven traversal updates to improve intent alignment during seeking. The architecture also addresses practical deployment constraints including variable frame rates, live-stream topology growth, resource arbitration, spoiler-safe traversal, and privacy-preserving interaction logging.

List of Abbreviations

DAG: Directed Acyclic Graph
ESS: Effective Sample Size
HGNN: Hyperbolic Graph Neural Network
LDP: Local Differential Privacy
MAD: Median Absolute Deviation
MCTS: Monte Carlo Tree Search
MEC: Mobile Edge Computing
MMS: Mean-Max Similarity
MV: Motion Vector
NAL: Network Abstraction Layer
NSD: Necessary Sampling Density
POMDP: Partially Observable Markov Decision Process
PTS: Presentation Timestamp
QP: Quantization Parameter
SVLM: Small Vision-Language Model
TTC: Time-to-Content
VFR: Variable Frame Rate

1. Research Objectives

To convert the full ISVS architecture into a thesis-feasible agenda, this work is structured around three testable research questions:

- **RQ1 (Systems/Networking):** How can cloud compute overhead be reduced by proxying deep semantic video features through zero-decode bitstream analysis and MEC-based sub-4-bit SVLM quantization?

- **RQ2 (AI/ML):** How does embedding hierarchical video relationships in a Riemannian hyperbolic manifold reduce distortion relative to Euclidean multimodal retrieval baselines?
- **RQ3 (HCI/Algorithms):** Can Time-to-Content (TTC) in unstructured media be reduced by modeling seeking as a POMDP and optimizing interaction via MCTS-driven active suggestions?

These objectives align with current regional priorities including cloud-edge-device collaboration, agentic systems, Green AI efficiency, and 5G/6G edge intelligence deployment.

2. Introduction

Standard video navigation interfaces rely on a linear scalar model, mapping spatial displacement on a timeline directly to temporal displacement in the media ($t \to t+\Delta$). This assumes that the semantic utility of video data correlates linearly with chronological order. However, video data is semantically semi-structured, containing local monotonic sequences (narrative continuity) and global non-monotonic clusters (thematic recurrence).

This document proposes an agentic framework that redefines video navigation as a probabilistic vector search over a topological graph. By combining client-side bitstream analysis with server-side semantic indexing, the system decouples chronological time from semantic relevance. The architecture employs robust statistical normalization, sequential state estimation, and active relevance feedback to allow users to navigate large-scale media archives based on intent rather than linear temporal scanning [1, 2, 3].

In summary, the primary contributions of this proposal are:

- **Tri-Tiered Edge-Cloud Collaborative Architecture:** A bandwidth-aware pipeline that shifts compute-heavy pixel decoding to low-latency bitstream analysis at the client edge.
- **Riemannian Semantic Topology:** Replacing Euclidean retrieval with Hyperbolic Graph Neural Networks (HGNNs) to preserve hierarchical video structure with lower geometric distortion.
- **POMDP-Driven Active Navigation:** The formulation of video seeking as a Partially Observable Markov Decision Process, utilizing MCTS to actively minimize user "Time-to-Content".

3. System Architecture: Hierarchical Processing

The architecture is partitioned into hierarchical processing levels to satisfy latency and compute constraints across client edge and server backend environments.

3.1 Level 1: Client-Side Bitstream Analysis

This layer operates within the client environment (browser or native application) with a strict latency constraint (< 50ms). It avoids computationally expensive pixel reconstruction by operating directly on the encoded bitstream.

- **Mechanism:** The system parses the Network Abstraction Layer (NAL) units of the video container (e.g., H.264, AV1).
- **Feature Extraction:**
  - **Kinetic Energy (E_k):** Derived from the magnitude of Motion Vectors (MVs) found in predicted frames (P-frames and B-frames). High aggregate magnitude correlates with high visual activity.

- **Visual Entropy (H_v):** Approximated via the ratio of Bitrate to the Quantization Parameter (QP). High bitrate allocation at a fixed QP indicates high textural complexity.
- **Shot Segmentation:** Detected via identifying instantaneous spikes in I-Frame insertion frequency, denoting hard cuts or scene transitions.
- **DRM Fallback via Manifest and BIF Parsing:** Client-side pixel extraction is blocked by Encrypted Media Extensions (EME). For DRM-protected streams, Level 1 bypasses the media pipeline entirely. It extracts visual entropy proxies by fetching the streaming provider's unencrypted **Base Index Frame (BIF)** tracks or sprite sheets (commonly used for UI seek-scrubbing). Kinetic energy (E_k) is approximated by parsing the byte-size variability of the encrypted video segments listed in the DASH/HLS manifest, under the assumption that Variable Bitrate (VBR) encoding allocates larger segment sizes to high-motion sequences.

3.2 Level 2: Edge-Based Inference

To minimize 5G/6G network backhaul and leverage Mobile Edge Computing (MEC) capabilities, this layer executes quantized inference directly within the client's constrained hardware environment (e.g., Browser Sandbox or Mobile OS Container), bridging the gap between raw signal extraction and deep indexing.

- **Latency constraint:** $100\text{ms} - 500\text{ms}$.
- **Mechanism:** Execution of Small Vision-Language Models (SVLMs) on keyframes triggered by Level 1 variance thresholds.
- **Models & Quantization:** This tier implements open-weight Small Vision-Language Models (SVLMs, e.g., Qwen3.5 2B) [4]. To satisfy edge memory budgets, models MUST undergo *Salience-Driven Mixed-Precision Quantization* at an average precision below 4 bits (nominally 2 bits) [5, 6]. This constrains memory footprint while retaining threshold accuracy for intent estimation and interactive routing.

3.3 Level 3: Server-Side Semantic Indexing

This layer handles high-dimensional, long-context embedding and temporal modeling. It is triggered asynchronously based on access-frequency thresholds (Zipfian distribution logic).

- **Mechanism:** Asynchronous batch processing on accelerator clusters.
- **Models & The Two-Stage Pipeline:**
  - **Stage 1: Coarse Graph Construction (Dual-Tower):** To construct the Semantic Graph (G) across millions of nodes, ISVS uses a Dual-Tower multimodal embedding model (e.g., Qwen3-VL-Embedding [7]). This projects audio, video, and text into a unified representation space, enabling low-latency Approximate Nearest Neighbor (ANN) indexing using Faiss to establish the Semantic Edges (E_s).
  - **Stage 2: Fine-Grained Observation (Single-Tower/Late Interaction):** For real-time user queries and Particle Filter observation updates, ISVS switches to a Contextualized Late-Interaction architecture (e.g., Video-ColBERT). Instead of relying on a single pooled vector, this preserves token-level granularity across both spatial and temporal dimensions, calculating precise semantic alignment only for the subgraph currently being traversed.
  - **Text Refinement and Alignment (TRA):** Temporal action localization is point-supervised. System-generated textual descriptions are iteratively refined against visual signals, integrating query semantics directly into the boundary scoring function.

4. Data Structure: The Topological Semantic Graph

The linear array of video frames V is mapped to a Directed Acyclic Graph (DAG) $G = (N, E)$.

4.1 Node Definition

Nodes N represent micro-segments of media (2s−5s). To account for Variable Frame Rate (VFR) sources, nodes are indexed strictly by **Presentation Timestamp (PTS)** in the nanosecond domain, rather than frame count.

4.2 Edge Definition: Hyperbolic Topology

Video data intrinsically exhibits hierarchical, scale-free structures (e.g., Scene → Event → Sub-event → Frame). Embedding such relationships in Euclidean space induces severe geometric distortion. ISVS resolves this by mapping the graph onto a Riemannian manifold with constant negative curvature [8].

- **Temporal Edges (E_t):** Directed edges $n_t \to n_{t+1}$ representing the deterministic flow of playback.
- **Semantic Edges (E_s):** Undirected edges connecting disjoint nodes n_i, n_j evaluated within a **Poincaré ball** model ($\mathbb{B}^d$). The exponential volume growth of hyperbolic space natively accommodates the branching factor of hierarchical video semantics, yielding higher-fidelity relational structures at lower dimensionalities.

4.3 Dynamic Topology for Unbounded Streams

For live media (where duration $T \to \infty$):

- **Ring Buffering:** Level 1 maintains a sliding window of raw signal data in volatile memory.
- **Micro-Batch Indexing:** Level 3 processes incoming stream chunks in fixed-time batches, appending nodes to the DAG in real-time.

5. Mathematical Framework

To ensure system reliability across diverse media types, heuristic thresholds are replaced with robust statistical estimators.

5.1 Robust Feature Normalization

Video feature distributions are non-stationary and heavy-tailed. Standard normalization (Z-score) is susceptible to outliers. The system employs the **Median Absolute Deviation (MAD)** over a sliding window W:

$$\text{MAD}_W = \text{median}(|X_i - \text{median}(W)|) \tag{1}$$

The robust Z-score calculation is:

$$Z_{\text{robust}}(t) = \frac{X_t - \text{median}(W)}{k \cdot \text{MAD}_W} \tag{2}$$

The constant $k \approx 1.4826$ is the reciprocal of the third quartile of the standard normal distribution $(1/\Phi^{-1}(0.75))$. It scales the estimator to be asymptotically consistent with the standard deviation of a normal distribution, allowing for the detection of local saliency relative to the immediate context.

5.2 Adaptive Compute Allocation via Necessary Sampling Density (NSD)

Historically, systems fused modalities using heuristic entropy gates. ISVS formalizes multimodal compute allocation using the **Necessary Sampling Density (NSD)** metric, which determines the minimum temporal sampling rate ρ required to resolve a semantic query [9].

$$\text{NSD}(q) = \min_{\rho} \rho \quad \text{s.t.} \quad \mathcal{A}(q, \rho) \geq \tau \tag{3}$$

where $\mathcal{A}$ is the task-specific accuracy threshold τ. The client predicts the relative NSD for the visual and acoustic streams based on initial Level 1 heuristics. If the visual NSD is predicted to be high (e.g., a highly dynamic action sequence), compute tokens are aggressively allocated to visual frame sampling. If visual NSD is minimal (e.g., a static presentation), resources are dynamically routed to acoustic processing, enabling task-driven, content-adaptive feature extraction.

5.3 Sequential State Estimation

User intent is modeled as a hidden state within a dynamic system, estimated using a **Particle Filter (Sequential Monte Carlo)** [2].

- **State:** A set of N particles $\{x^{(i)}\}$ representing hypothetical target timestamps on the timeline.
- **Update:** Upon an observation (Query or Seek action), particles are re-weighted based on the likelihood of the observation given the content at the particle's location:

$$w_t^{(i)} \propto P(\text{Query} \mid \text{Content}(x_t^{(i)})) \tag{4}$$

- **Resampling:** Particles concentrate in high-probability regions. This allows the system to maintain multiple simultaneous hypotheses regarding the user's target until sufficient evidence prompts convergence.

6. Algorithmic Logic

6.1 Graph Traversal: Active Planning via MCTS

Rather than passively projecting relevance, ISVS operates as an active agent solving a **Partially Observable Markov Decision Process (POMDP)**. The system must plan its navigational suggestions to minimize the user's "Time-to-Content."

To navigate the hyperbolic semantic graph, ISVS employs **Monte Carlo Tree Search (MCTS)**.

- **State Representation:** The belief state of the user's intent, maintained by the Particle Filter.
- **Action Space:** The set of candidate nodes (timestamps) presented to the user via the UI.
- **Reward Function:** Maximizing Information Gain (reducing the entropy of the belief state) while penalizing traversal costs. Crucially, transitions that violate narrative causality (see Appendix D) carry a reward of $-\infty$, strictly preventing MCTS from exploring spoiler pathways.

This transforms the seek bar into a proactive exploration system, deliberately surfacing "Wildcard" nodes that optimally bisect the hypothesis space.

Node selection during the MCTS expansion phase is governed by a modified **Hyperbolic Upper Confidence Bound applied to Trees (UCT)** formula. The algorithm selects the candidate node j that maximizes:

$$\text{UCT}(j) = \frac{R_j}{N_j} + c\sqrt{\frac{\ln N_p}{N_j}} - \lambda \cdot d_{\mathbb{B}^d}(n_{\text{curr}}, n_j) \tag{5}$$

Where R_j is the accumulated reward, N_j is the visit count of node j, N_p is the parent visit count, c is the exploration constant, and $d_{\mathbb{B}^d}$ is the hyperbolic geodesic distance (Equation 11). During the MCTS simulation rollout, the simulated reward is defined as the expected reduction in the Shannon entropy of the Particle Filter belief state:

$$R_j = H(\text{State}_t) - \mathbb{E}[H(\text{State}_{t+1} \mid \text{Action} = j)] \tag{6}$$

To ensure real-time responsiveness ($<$ 50ms action selection), the MCTS rollout phase is strictly bounded to a temporal depth $d_{\max}$, beyond which the remaining hypothesis entropy is approximated via a lightweight heuristic value function.

This forces the tree search to discover "Wildcard" nodes that optimally bisect the remaining hypothesis space. For any unvisited candidate node where $N_j = 0$, the UCT value is defined as ∞, ensuring that the traversal algorithm exhaustively samples the immediate action space before deepening the search tree. The penalty term λ ensures the system prefers semantically proximal suggestions unless the exploration bound strictly dictates a "Wildcard" jump.

6.2 Active Relevance Feedback: Differential Backtracking

When a user rejects a suggested segment (False Positive), the system utilizes **Rocchio's Algorithm on the Stage 1 pooled macro-embeddings** to escape the local semantic neighborhood [3]. The coarse query vector $\boldsymbol{Q}$ is updated, while the Stage 2 late-interaction query tokens remain unmodified for micro-level scoring.

$$\boldsymbol{Q}_{\text{new}} = \alpha \boldsymbol{Q}_0 - \beta \frac{\boldsymbol{D}_{\text{rej}}}{\|\boldsymbol{D}_{\text{rej}}\|} \tag{7}$$

where $\boldsymbol{D}_{\text{rej}}$ is the vector of the rejected segment, α is the original query retention weight (typically $\alpha = 1$), and β is the negative feedback penalty.

- **Orthogonal Constraint:** To avoid suggesting segments that are distinct in vector space but compositionally identical, the system filters candidates using the **Jaccard Index** of their semantic labels.

$$\text{Valid}(C) \iff \frac{\left|\text{Tags}_{\text{rej}} \cap \text{Tags}_{\text{cand}}\right|}{\left|\text{Tags}_{\text{rej}} \cup \text{Tags}_{\text{cand}}\right|} < \tau_{\text{Jaccard}} \tag{8}$$

  where τ_{Jaccard} is the semantic-overlap rejection threshold.

6.3 Topology Constraints: Dependency Sorting

For narrative content, the system identifies causal dependencies (e.g., Event A must precede Event B). A **Dependency Graph** is constructed. The traversal algorithm enforces a topological sort, disabling semantic edges that would violate narrative causality (spoiler prevention) unless the antecedent node has been visited.

7. Operational Workflow

7.1 Initialization & Visualization

Upon media load, Level 1 processing generates a **Density Map** overlaid on the timeline. This visualization encodes the dominant modality (via color) and saliency intensity (via height), providing the user with an immediate topology of the media's structure.

7.2 Projection & Prediction

User input (text query or hover interaction) is projected into the joint embedding space. The Particle Filter estimates the target distribution [2]. A visual indicator ("Ghost Cursor") displays the Maximum A Posteriori (MAP) estimate on the timeline, bridging the gulf of evaluation between system state and user expectation.

7.3 Interaction Physics

Upon seeking:

- **High Coherence Regions:** The playback head aligns to the nearest shot boundary or audio zero-crossing to ensure clean entry.
- **Low Coherence Regions:** The playback head aligns to the point of maximum feature magnitude (peak saliency).

7.4 Correction Loop

If the user abandons a segment within a short dwell time threshold (< 2s):

1. The system registers a negative label for the cluster.
2. The Residual Exclusion Mask (Differential Backtracking) is applied to the graph [3].
3. The interface highlights an alternative candidate from the Stratified Beam Search (the "Wildcard").

8. Systems Engineering & Resilience

8.1 Network Partition Handling (Dead Reckoning)

In the event of server connection failure, the client falls back to **Dead Reckoning**. The system extrapolates search relevance using only Level 1 metadata (Motion Vectors and Audio Energy). The interface visually desaturates to indicate the degradation in confidence.

8.2 Variable Frame Rate Synchronization

To accommodate VFR sources, all data fusion and graph indexing occur strictly in the nanosecond time domain using Presentation Timestamps (PTS). Feature vectors from audio and video are aligned via nearest-neighbor interpolation (or Spherical Linear Interpolation for dense latent features) on the PTS timeline, consistent with long-video sampling considerations [9].

9. Human-Computer Interaction & Accessibility

9.1 Sonification

For non-visual navigation, the system maps data variance to audio.

- **Pulse Rate:** Mapped to Local Coherence (feature density).
- **Pitch:** Mapped to Semantic Relevance (query similarity).
- **Haptic Feedback:** Actuation intensity is proportional to the Robust Z-Score (Z_{robust}) of the current timestamp, allowing users to "feel" the content topology.

9.2 Dynamic Level of Detail (LOD)

To manage cognitive load, the visual representation of search results adapts based on zoom level and query-conditioned semantic granularity [1].

- **Macro View:** Displays global cluster centers.
- **Meso View:** Displays section subdivisions.
- **Micro View:** Displays discrete atomic events.

10. Security & Privacy

10.1 Local Differential Privacy

To protect user intent during heatmap generation, seek timestamps are perturbed via the Laplace Mechanism before transmission. This ensures that aggregate usage patterns remain statistically significant while individual search trajectories are mathematically deniable. The precise sensitivity bounds and privacy budget formulations are detailed in Appendix F.1.

10.2 Adversarial Input Filtering

- **Retention Gating:** Interaction data is weighted by post-seek retention duration. Short-duration interactions are treated as noise or adversarial input and are filtered from the aggregate prior.
- **Query Sanitization:** Textual inputs are immediately projected into vector space, preventing prompt injection attacks against backend inference models.

11. Proposed Evaluation Methodology

To rigorously validate the ISVS protocol, future empirical evaluations will utilize a heterogeneous hardware-software testbed encompassing diverse mobile edge devices (e.g., ARM-based NPUs, Snapdragon Hexagon) and cloud-scale accelerators (e.g., Nvidia H100, Huawei Ascend 910B) via hardware-agnostic runtimes.

While the ISVS architecture defines a global cloud-edge system, the thesis implementation scope is a scaled prototype. Initial phases isolate Level 1 and Level 1.5 components (Bitstream Analysis and Edge Inference) to validate bandwidth reduction and energy metrics on LSDBench, then progressively expand to hyperbolic graph traversal and active planning logic.

- **Datasets:** Benchmarking will be conducted on long-form, high-NSD datasets such as **LSDBench** (Long Semantic Video) and **Ego4D** (unstructured egocentric video).

- **Systems & Energy Metrics:** Edge-to-Cloud latency (ms), client-side memory footprint (MB), network bandwidth reduction (%), and Edge NPU energy consumption (Joules/query) to quantify the battery preservation of Level 1 bitstream proxying.
- **Evaluation Baselines:** ISVS performance will be benchmarked against standard linear chronological scrubbing, zero-shot bi-encoder retrieval (baseline CLIP), and static point-supervised temporal action localization frameworks (ActionFormer standalone [10]).
- **HCI & Retrieval Metrics:** Mean "Time-to-Content" (TTC) in seconds, normalized trajectory length (number of seek actions), and Frustration Index (ratio of abandoned queries).

12. Conclusion

The ISVS proposal defines a comprehensive protocol for non-linear media navigation. By acknowledging the probabilistic nature of semantic relevance and implementing a robust, tiered architecture, the system enables efficient retrieval within massive, semi-structured data streams. It transitions the seek bar from a passive temporal tool into an active, intent-driven interface [1, 9, 2, 3].

References

[1] Arun Reddy et al. *Video-ColBERT: Contextualized Late Interaction for Text-to-Video Retrieval*. 2025. arXiv: 2503.19009 [cs.CV]. URL: https://arxiv.org/abs/2503.19009.

[2] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2005. ISBN: 9780262201629.

[3] J. J. Rocchio. "Relevance Feedback in Information Retrieval". In: *The SMART Retrieval System: Experiments in Automatic Document Processing*. Ed. by Gerard Salton. Prentice-Hall, 1971, pp. 313–323.

[4] Qwen Team. *Qwen3.5: Towards Native Multimodal Agents*. Qwen Blog. 2026. URL: https://qwen.ai/blog?id=qwen3.5.

[5] Wei Huang et al. *SliM-LLM: Salience-Driven Mixed-Precision Quantization for Large Language Models*. 2025. arXiv: 2405.14917 [cs.LG]. URL: https://arxiv.org/abs/2405.14917.

[6] Xianglong Yan et al. *D^2Quant: Accurate Low-bit Post-Training Weight Quantization for LLMs*. 2026. arXiv: 2602.02546 [cs.LG]. URL: https://arxiv.org/abs/2602.02546.

[7] Mingxin Li et al. *Qwen3-VL-Embedding and Qwen3-VL-Reranker: A Unified Framework for State-of-the-Art Multimodal Retrieval and Ranking*. 2026. arXiv: 2601.04720 [cs.CL]. URL: https://arxiv.org/abs/2601.04720.

[8] Ines Chami et al. "Hyperbolic Graph Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/0415740eaa4d9decbc8da001d3fd805f-Paper.pdf.

[9] Tianyuan Qu et al. *Does Your Vision-Language Model Get Lost in the Long Video Sampling Dilemma?* 2025. arXiv: 2503.12496 [cs.CV]. URL: https://arxiv.org/abs/2503.12496.

[10] Chenlin Zhang, Jianxin Wu, and Yin Li. *ActionFormer: Localizing Moments of Actions with Transformers*. 2022. arXiv: 2202.07925 [cs.CV]. URL: https://arxiv.org/abs/2202.07925.

[11] Ao Wang et al. *YOLOv10: Real-Time End-to-End Object Detection*. 2024. arXiv: 2405.14458 [cs.CV]. URL: https://arxiv.org/abs/2405.14458.

A. Edge Inference & Resource Arbitration

Scope: Client-side model execution and hardware abstraction.

A.1 Execution Runtime

Inference is managed via **ONNX Runtime Web** utilizing the **WebGPU** backend for parallel execution. The system employs a **Sharded Weight Architecture**:

- **Backbone (Feature Extractor):** Loaded immediately ($\sim$ 15MB).
- **Task Heads (Classifiers):** Lazy-loaded based on Level 1 signal entropy (e.g., high audio entropy triggers the download of the Audio Classification Head).

A.2 Concurrency Control (Token Bucket)

To prevent UI freezing, the Level 2 worker thread operates under a **Token Bucket** rate limiter.

- **Bucket Capacity (B):** Max burst inference operations (e.g., 5 frames).
- **Refill Rate (R):** Tokens added per second, dynamically scaled by the device's thermal state and battery level.
- **Logic:** Let inference task T_i (e.g., Object Detection [11] vs. ActionFormer [10] temporal pass) require a specific computation cost of c_i tokens. If $c_i >$ Tokens available in the bucket, task T_i is dropped and the frame is skipped. Indexing latency is actively sacrificed to maintain a strict 60Hz UI rendering loop.

A.3 Storage Quota

Model weights and the computed Semantic Graph are stored in **IndexedDB**. An **LRU (Least Recently Used)** eviction policy manages the storage quota (typically clamped at 10% of available disk space or 500MB).

A.4 Salience-Driven Mixed-Precision Quantization

To execute Large Vision-Language Models (e.g., Qwen3.5 SVLMs) within the browser memory quota, ISVS implements a Salience-Weighted Quantizer [4, 5, 6].

Weights are grouped g and assigned varying bit-widths $b_g \in \{2, 4, 8\}$ based on a salience metric $S(g)$ (e.g., activation sensitivity or gradient magnitude). Uniform quantization within a group is defined as $\hat{w}_g = \text{round}(w_g/\Delta_g^*)\Delta_g^*$, where the optimal scaling factor Δ_g^* is derived via salience-weighted calibration:

$$\Delta_g^* = \arg\min_{\Delta} \sum_{w \in g} S(w) \left| w - Q(w; \Delta) \right|^2 \tag{9}$$

This ensures that parameters critical to semantic representation retain higher precision. In practice, this yields about 6$\times$ memory compression with negligible degradation in intent-estimation accuracy.

B. Adaptive Graph Topology

Scope: Derivation of dynamic connectivity thresholds.

B.1 Distribution Modeling

A fixed similarity threshold θ is invalid across diverse media. Upon indexing, the system computes a reservoir sample of pairwise Cosine Similarities to model the asset's specific distribution $D_{\text{sim}} \sim \mathcal{N}(\mu_{\text{sim}}, \sigma_{\text{sim}})$.

B.2 Adaptive Thresholding

The connection threshold θ is dynamic:

$$\theta = \mu_{\text{sim}} + k \cdot \sigma_{\text{sim}} \tag{10}$$

where k is a sparsity hyperparameter (typically $k = 2.5$). This ensures that a video with low variance (e.g., a lecture) does not degenerate into a fully connected graph.

B.3 Hyperbolic Edge Weighting and Distance

Standard vector addition and Euclidean distance metrics are invalid in hyperbolic geometry. Nodes are embedded in the Poincaré ball $\mathbb{B}^d = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\| < 1\}$. The geodesic distance between two semantic nodes $\boldsymbol{x}, \boldsymbol{y}$ is defined as:

$$d_{\mathbb{B}^d}(\boldsymbol{x}, \boldsymbol{y}) = \operatorname{arcosh}\left(1 + 2\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{(1 - \|\boldsymbol{x}\|^2)(1 - \|\boldsymbol{y}\|^2)}\right) \tag{11}$$

Semantic edges $E_s(\boldsymbol{x}, \boldsymbol{y})$ are established when $d_{\mathbb{B}^d}(\boldsymbol{x}, \boldsymbol{y})$ falls below the adaptive threshold θ. Graph message passing and vector interpolation require projecting node features into the Euclidean tangent space via the logarithmic map $\log_{\boldsymbol{x}}^c(\cdot)$, performing the operation, and projecting back to the hyperbolic manifold via the exponential map $\exp_{\boldsymbol{x}}^c(\cdot)$ [8].

C. Stochastic State Estimation (Particle Filter Mechanics)

Scope: The mathematical definition of the user intent tracking system.

C.1 Motion Model (Prediction Step)

In the absence of interaction, the user's target x drifts forward. We model this as a Constant Velocity model with diffusion noise [2].

$$x_t^{(i)} = x_{t-1}^{(i)} + v_{\text{playback}} \cdot \Delta t + \eta_t \tag{12}$$

$$\eta_t \sim \mathcal{N}(0, \sigma_{\text{diffusion}}^2) \tag{13}$$

The diffusion term $\sigma_{\text{diffusion}}^2$ increases linearly with dwell time, widening the uncertainty window the longer the user watches without seeking. Here, $v_{\text{playback}} \equiv \frac{dt}{d\tau}$ is the temporal derivative of media time with respect to wall-clock time τ, which makes the motion model explicit for arbitrary playback rates.

C.2 Observation Model (Correction Step)

When an observation z_t (Query) occurs, particle weights are updated based on the Contextualized Late Interaction score S_{obs} derived in Appendix C.4 (Equation 19):

$$w_t^{(i)} \propto \exp\left(-\frac{(1 - S_{\text{obs}}(\boldsymbol{Q}, V(x_t^{(i)})))^2}{2\sigma_{\text{obs}}^2}\right) \tag{14}$$

Weights are subsequently normalized such that $\sum_{i=1}^{N} \tilde{w}_t^{(i)} = 1$.

C.3 Resampling Criterion

To avoid particle degeneracy, resampling (Sequential Importance Resampling) is triggered only when the **Effective Sample Size (ESS)** drops below $N/2$:

$$\text{ESS} = \frac{1}{\sum_{i=1}^{N} (\tilde{w}_t^{(i)})^2} \tag{15}$$

C.4 Multimodal Contextualized Late Interaction

The observation likelihood $P(z_t \mid x_t^{(i)})$ relies on the interaction between the query embedding $\boldsymbol{Q}$ and the media sequence V. ISVS utilizes a **Multimodal Mean-Max Similarity (MMS)** operator to evaluate fine-grained alignment across text, spatial video, and acoustic tokens.

Let $\boldsymbol{q}_j$ be the embedding of the j-th query token, $\boldsymbol{f}_i$ be the i-th frame-level visual token, and $\boldsymbol{a}_k$ be the k-th acoustic token from the Audio Spectrogram Transformer (AST). The spatial visual interaction is defined as:

$$\text{MMS}_V(\boldsymbol{Q}, V) = \frac{1}{M} \sum_{j=1}^{M} \max_i \left(\boldsymbol{q}_j \cdot \boldsymbol{f}_i \right) \tag{16}$$

Similarly, the acoustic interaction is defined as:

$$\text{MMS}_A(\boldsymbol{Q}, V) = \frac{1}{M} \sum_{j=1}^{M} \max_k \left(\boldsymbol{q}_j \cdot \boldsymbol{a}_k \right) \tag{17}$$

The final observation score dynamically weights these late-interaction streams based on the relative Necessary Sampling Density (NSD) of the modalities (as defined in Equation 3). Let NSD_V and NSD_A denote the predicted sampling densities for the visual and acoustic streams, respectively. The multimodal fusion gate g is defined as:

$$g(\text{NSD}) = \frac{\text{NSD}_V}{\text{NSD}_V + \text{NSD}_A + \epsilon} \tag{18}$$

The unified observation score is:

$$S_{\text{obs}}(\boldsymbol{Q}, V) = g(\text{NSD}) \cdot \text{MMS}_V(\boldsymbol{Q}, V) + (1 - g(\text{NSD})) \cdot \text{MMS}_A(\boldsymbol{Q}, V) \tag{19}$$

D. Narrative Integrity & Spoiler Guard Protocol

Scope: Handling causality, narrative dependency, and preventing non-linear jumps that violate information structure.

D.1 Definition of a Spoiler

Mathematically, a spoiler is a traversal from Node A to Node B where:

1. $t(B) > t(A)$ (Forward jump).
2. The **Information Gain** $IG(B|A)$ is maximal (resolves uncertainty).
3. There exists a path $A \to ... \to C \to ... \to B$ where Node C contains a **Causal Precondition** that has not been visited.

D.2 Entity State Tracking (EST)

The Level 3 indexer identifies persistent entities E and assigns them a discrete state vector S_E based on visual features (e.g., Car: [Clean] vs [Damaged]).

- **Irreversible Transitions:** A transition $S_{\text{Clean}} \rightarrow S_{\text{Damaged}}$ is marked as irreversible (Entropy increases).
- **Dependency Edge:** If Node C depicts the transition event (The Crash), then any Node B showing S_{Damaged} has a hard dependency on C.

D.3 The Spoiler Guard Logic

When the user attempts to traverse a Semantic Edge $E_s(A, B)$:

$$\text{Blocked}(A, B) = \exists C \in \text{Dependencies}(B) \text{ s.t. } \text{Visited}(C) = \text{False} \tag{20}$$

The state Visited(C) evaluates to True if and only if the playback playhead resides within the temporal bounds of node C at $1\times$ playback speed for a cumulative duration exceeding the Minimum Dwell Threshold ($t_{\text{dwell}} > 2\text{s}$).

- **UI Manifestation:** If Blocked, the "Ghost Cursor" turns grey, and the system prompts: "`This segment depends on unwatched events at [Timestamp C]. Jump there first?`"

D.4 Narrative Arc Detection

For content without clear object states, we analyze the **Audio Sentiment Arc**. A sudden sentiment inversion (Positive $\rightarrow$ Negative) often denotes a plot twist. High-magnitude sentiment shifts are flagged as **Critical Nodes** requiring sequential access.

E. Memory Arbitration & Threading Model

Scope: Managing client resources for heavy media data.

E.1 Ring Buffer Logic (Live Streams)

The Level 1 signal buffer operates as a **Priority-Eviction Queue**.

- **Retention Policy:** Keyframes (I-frames) and High-Saliency segments (High Z_{robust}) are pinned in RAM.
- **Eviction Policy:** Low-entropy segments (static scenes) and B-frames are evicted first. This maintains a "Semantic Skeleton" of the stream history while discarding redundant data.

E.2 Spatial Locality Caching

The Semantic Graph is too large to hold in memory for long videos.

- **Fetch Strategy:** The client loads the subgraph centered at t_{playhead} with radius R (e.g., ± 10 minutes semantic distance).
- **Prefetching:** As the particle filter distribution shifts, subgraphs in high-probability regions are pre-fetched.

E.3 Isolation

- **Worker A:** Signal Processing (Level 1).
- **Worker B:** Inference (Level 2) + WebGPU context.
- **Main Thread:** UI Rendering + Density Map Visualization.
- **Communication:** Zero-copy transfer via `SharedArrayBuffer`, strictly requiring the host application to enforce Cross-Origin Isolation (COOP/COEP headers) to mitigate Spectre-class side-channel attacks.

F. Differential Privacy & Security

Scope: Protecting user intent data.

F.1 Local Differential Privacy (LDP)

Seek timestamps are never reported raw. We use the **Laplace Mechanism**.

$$t_{\text{reported}} = t_{\text{true}} + \mathcal{L}\left(0, \frac{\Delta T}{\epsilon}\right) \tag{21}$$

where ΔT is the sensitivity (maximum time jump) and ϵ is the privacy budget.

F.2 Randomized Response for Query Logs

For textual queries, the client employs **Randomized Response**:

- Flip a coin.
- **Heads:** Send true query embedding.
- **Tails:** Send a random noise embedding.
- The server aggregates vectors to find global trends but cannot attribute specific intent to a specific user with confidence $> 50\%$.

F.3 Prompt Injection Defense

The system uses **Vector-Only Interfaces**. The text query is converted to an embedding on the client (Tier 1.5). The server accepts *only* numeric vectors, making string-based injection attacks against server-side logic impossible.