Graph-Based, Location- and View-Aware Retrieval over Multimodal Dashcam Corpora with Native Vector Search in Neo4j

Scott Joyner
Independent Researcher
kipnerter@gmail.com

Abstract

We present a dissertation-scale study of a fully automated pipeline that ingests *personal driving corpora*—dashcam video (front/rear), ambient or body-worn audio, and device metadata—into a Neo4j property graph augmented with native vector indexes. The system fuses (i) sentence-level speech semantics from Whisper–SBERT; (ii) view-aware frame signatures derived from YOLOv8 grid–pyramid statistics and optional heatmaps; and (iii) geospatial & kinematic context (latitude/longitude, time, speed). We formalize the embedding constructions, the vector indices, and the three primary query operators (`search-text`, `similar-frames`, `geo-frames`). Our analysis covers end-to-end orchestration, robust transcript ingestion with diarization-anchored utterances and entity graphs, ANN design, and graph-based enrichment for location snapping. We also contribute quality standards & controls (QS&C) for this pipeline—unit tests, data validation, index integrity checks, and embedding drift alarms—to ensure repeatable research and production reliability.

Keywords Multimodal retrieval; Neo4j; vector search; YOLOv8; Whisper; SBERT; diarization; geospatial indexing; HNSW; FAISS.

1 Introduction

Modern personal sensing ecosystems (dashcams, phones, bodycams) generate heterogeneous streams (audio, video, geospatial telemetry) where actionable events are distributed across modalities and time. Classical keyword search is insufficient when a target moment is defined jointly by *what was said*, *what was seen*, *where/when*, and *how fast* the vehicle was moving. We develop a unified, graph-centric solution that (1) encodes each modality with compact, ANN-ready embeddings; (2) anchors content to absolute time and space; and (3) exposes *hybrid* graph+vector queries returning semantically relevant hits with precise context (coordinates, speed, camera view).

Contributions.

1. A deployable pipeline integrating Whisper ASR [**?**], SBERT [**?**], pyannote diarization [**?**], YOLOv8 detections [**?**], and Neo4j native vector search [**?**].
2. Formalized representations for text, vision, and location/kinematics; principled fusion for view-aware, second- and minute-level embeddings; explicit formulas mirroring the implemented code paths.
3. Query semantics for `search-text`, `similar-frames`, and `geo-frames`, realized with `db.index.vector.queryNo` and Cypher enrichment (location snap, speed, and frame alignment).

4. A quality standards & controls (QS&C) framework: schema/assertions, data validation, vector index integrity checks, ANN calibration, embedding-drift monitoring, and reproducible evaluation harnesses.

2 Background and Related Work

Text embeddings via SBERT [?] (we use all-MiniLM-L6-v2, d=384). Robust ASR with Whisper [?]. Diarization with pyannote.audio [?]. Object detection with YOLOv8 [?]. ANN search: HNSW [?] and FAISS [?]. Graph+vector search in Neo4j 5.x [?].

3 System Architecture

3.1 End-to-End Orchestration

A master runner (`runall.sh`) executes: media copy; Whisper (chunk-len 90 s, stride 85 s); diarization; ingestion to Neo4j; speaker reconciliation and *global linking*; YOLOv8 detections; HUD metadata scraping; frame/heatmap embeddings; location patching; time-lapse and shorts generation. Steps are idempotent with `set -euo pipefail`.

3.2 Graph Data Model

Nodes: `Transcription`, `Segment`, `Utterance`, `Speaker` and `GlobalSpeaker`, `Entity`, `Frame`, `DashcamEmbedding`, `PhoneLog`. Edges: `HAS_SEGMENT`, `HAS_UTTERANCE`, `SPOKEN_BY`, `MENTIONS`, `SUMMARIZED_BY`. Vector properties live directly on nodes (cosine space). Neo4j constraints and vector indexes are created up front (Neo4j 5.11+ required).

4 Methods (Formal)

4.1 Text Embeddings and Transcript Vector Estimation

Let a tokenized segment yield hidden states $H \in \mathbb{R}^{T\times d}$ and attention mask $m \in \{0,1\}^T$. Define the masked mean

$$\tilde{\boldsymbol{z}} = \frac{\sum_{t=1}^{T} m_t H_t}{\sum_{t=1}^{T} m_t}, \qquad \boldsymbol{z} = \frac{\tilde{\boldsymbol{z}}}{\|\tilde{\boldsymbol{z}}\|_2} \in \mathbb{R}^d. \tag{1}$$

For a transcript composed of segments $\mathcal{S} = \{1,\dots,n\}$ with segment embeddings $\{\boldsymbol{z}_i\}$ and durations $w_i = e_i - s_i$, we consider two estimators:

$$\text{Uniform mean: } \boldsymbol{z}_T^{\text{uni}} = \text{norm}\left(\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{z}_i\right), \tag{2}$$

$$\text{Duration-weighted: } \boldsymbol{z}_T^{\text{dur}} = \text{norm}\left(\sum_{i=1}^{n} \frac{w_i}{\sum_j w_j}\, \boldsymbol{z}_i\right). \tag{3}$$

[Optimality of normalized mean for squared loss] Let $\{\boldsymbol{x}_i\}_{i=1}^n$ be unit vectors. The unit vector $\boldsymbol{u}$ minimizing $\sum_i \|\boldsymbol{x}_i - \boldsymbol{u}\|_2^2$ is $\boldsymbol{u}^\star = \text{norm}(\sum_i \boldsymbol{x}_i)$. Expanding and dropping constants reduces the objective to $-2\sum_i \boldsymbol{x}_i^\top \boldsymbol{u}$ with $\|\boldsymbol{u}\|_2 = 1$. The maximizer is the normalized sum by Cauchy–Schwarz. Lemma 4.1 justifies Eq. (2); Eq. (3) follows by reweighting when longer segments should contribute more.

4.2 Diarization Overlap as Soft Speaker Assignment

Let RTTM give intervals $\{[a_j, b_j], \ell_j\}$. For a segment window $[s_i, e_i]$ define overlap

$$\mathrm{ov}_{ij} = \max\big(0, \min(e_i, b_j) - \max(s_i, a_j)\big). \tag{4}$$

Define a *soft* posterior over labels with additive smoothing ϵ:

$$p(\ell \mid i) = \frac{\epsilon + \sum_{j:\ell_j=\ell} \mathrm{ov}_{ij}}{\sum_{\ell'} \Big(\epsilon + \sum_{j:\ell_j=\ell'} \mathrm{ov}_{ij}\Big)}. \tag{5}$$

We keep mixture edges (Segment i) $\to$ (Speaker ℓ) iff $p(\ell \mid i) \geq \tau$ (default $\tau = 0.05$). For utterances assembled from word timings, we embed utterance text $\boldsymbol{z}_u$ and optionally refine speaker posteriors by maximum-likelihood under a von Mises–Fisher model (optional).

4.3 Entity Aggregation

Given raw entities $E = \{(t_k, \ell_k, s_k)\}$, we aggregate by (t, ℓ) into buckets $B_{t,\ell}$; we store count $|B_{t,\ell}|$, average confidence $\bar{s}_{t,\ell}$ (trimmed mean at 10% by default), and span summaries (min/max).

4.4 YOLO Grid–Pyramid Embedding with Heatmaps

Let the frame be normalized to $[0,1]^2$. For grid level $g = (G_x, G_y)$, the cell set is $\mathcal{U}_g = \{1, \ldots, G_x\} \times \{1, \ldots, G_y\}$. YOLO detections yield boxes $b_k = (x_k, y_k, w_k, h_k)$ with class $c_k \in \mathcal{C}$ and confidence $s_k \in [0,1]$; we define center $\boldsymbol{\mu}_k = (x_k, y_k)$ and area $A_k = w_k h_k$. For each cell $u \in \mathcal{U}_g$,

$$\begin{aligned} N_{u,c} &= \sum_k \mathbb{1}[b_k \in u \wedge c_k = c], \qquad S_{u,c} = \sum_k s_k \mathbb{1}[b_k \in u], \\ A_u &= \sum_k \tfrac{A_k}{A(u)} \mathbb{1}[b_k \in u], \qquad R_u = \sum_k \tfrac{\max(w_k,h_k)}{\min(w_k,h_k)} \mathbb{1}[b_k \in u], \end{aligned} \tag{6}$$

and a *density heatmap* channel by Gaussian kernel density estimation (KDE):

$$H_u = \sum_k \exp\Big(-\tfrac{1}{2}(\boldsymbol{\mu}_k - \boldsymbol{c}_u)^\top \Sigma_g^{-1} (\boldsymbol{\mu}_k - \boldsymbol{c}_u)\Big), \tag{7}$$

where $\boldsymbol{c}_u$ is the cell center and $\Sigma_g = \mathrm{diag}(\sigma_x^2, \sigma_y^2)$ with $\sigma_x \propto 1/G_x$, $\sigma_y \propto 1/G_y$. We concatenate all per-cell features across classes and grids, then *per-grid z-score* normalize to mitigate scale differences across g:

$$\boldsymbol{v}^{(g)} = \mathrm{zscore}\big(\mathrm{vec}(\{N_{u,c}, S_{u,c}, A_u, R_u, H_u\}_{u,c})\big), \qquad \boldsymbol{v}_F = \mathrm{norm}\bigoplus_g \boldsymbol{v}^{(g)} \in \mathbb{R}^{d_F}. \tag{8}$$

Front/Rear. If both views exist at time t, we set $\boldsymbol{v}_t = \mathrm{norm}(\boldsymbol{v}_t^{(F)} \oplus \boldsymbol{v}_t^{(R)})$; otherwise use the available view.

Temporal pooling. For minute-level vectors we use variance-aware weights $w_t \propto \exp(\alpha\, \sigma_t^2)$ where σ_t^2 is per-second detection-count variance and $\alpha \geq 0$:

$$\bar{\boldsymbol{v}} = \mathrm{norm}\left(\sum_{t=1}^{60} \frac{w_t}{\sum_\tau w_\tau}\, \boldsymbol{v}_t\right). \tag{9}$$

4.5 Geospatial & Kinematic Context and Fusion

We encode context $\boldsymbol{\ell}_t \in \mathbb{R}^7$ as

$$\boldsymbol{\ell}_t = \big[\kappa\,\phi(\text{lat}),\ \kappa\,\phi(\text{lon}),\ \sin(2\pi\tfrac{\text{TOD}}{24h}),\ \cos(2\pi\tfrac{\text{TOD}}{24h}),\ \sin(2\pi\tfrac{\text{MOY}}{12}),\ \cos(2\pi\tfrac{\text{MOY}}{12}),\ \tanh(\text{mph}/s_0)\big], \tag{10}$$

with ϕ a linear mapping to $[-1, 1]$, scale κ to match visual feature magnitude, and s_0 a speed scale (e.g., 50 mph). The final frame vector is

$$\boldsymbol{e}_t = \text{norm}(\boldsymbol{v}_t \oplus \lambda\,\boldsymbol{\ell}_t)\,, \qquad \lambda \in [0, 1]. \tag{11}$$

When `PhoneLog` is missing we linearly interpolate GPS and speed in time or drop the context term ($\lambda \to 0$).

4.6 Speaker Linking Across Files

We build local speaker snippets per file with energy/SNR gates, embed them, then perform HNSW search (FAISS) with parameters (m, ef); edges above a similarity threshold θ are proposed, with *holdout* logic to avoid transitive drift: if the best cross-file match score falls below θ_{hold} we drop members. This reproduces the CLI parameters in the runner.

5 Query Semantics (Scoring and Guarantees)

5.1 Cosine vs. Euclidean

For unit vectors $\boldsymbol{a}, \boldsymbol{b}$, cosine similarity and squared Euclidean distance are affine-related: $\|\boldsymbol{a} - \boldsymbol{b}\|_2^2 = 2(1 - \text{sim}_{\cos(\boldsymbol{a},\boldsymbol{b})})$. Thus ranking by cosine equals ranking by negative Euclidean distance.

5.2 `search-text`: Hybrid Score and Location Snap

Let a text query embed to $\boldsymbol{z}_q$. For a candidate node i with embedding $\boldsymbol{z}_i$, file key k, and time estimate $\hat{t}_i$ (midpoint logic from implementation), we define a *hybrid* score

$$S_i = \alpha\,\text{sim}_{\cos(\boldsymbol{z}_q,\boldsymbol{z}_i)} + \beta\,\varphi_{\text{geo}}(i) + \gamma\,\varphi_{\text{time}}(i), \tag{12}$$

where $\alpha + \beta + \gamma = 1$ and feature terms are optional priors:

$$\varphi_{\text{geo}}(i) = \exp\!\big(-d(\hat{\boldsymbol{x}}_i, \boldsymbol{x}_0)^2/(2\sigma_r^2)\big) \quad \text{if a target location } \boldsymbol{x}_0 \text{ is provided,} \tag{13}$$

$$\varphi_{\text{time}}(i) = \exp\!\big(-(\hat{t}_i - t_0)^2/(2\sigma_t^2)\big) \quad \text{if a target time } t_0 \text{ is provided.} \tag{14}$$

In our CLI we use α=1 by default (pure semantic ranking) and attach snapped location as metadata. The snap chooses the nearest `PhoneLog` within $\pm W$ minutes; otherwise we map relative position to the nearest `Frame`.

[Snap error bound] If `PhoneLog` samples are at interval Δ and there exists at least one sample within the window $[\hat{t}_i - W, \hat{t}_i + W]$, the absolute timestamp error of the nearest-sample snap is $\le \Delta/2$. Geodesic localization error is at most $v_{\max}\Delta/2$ where $v_{\max}$ is the maximum ground speed during the interval.

5.3 `similar-frames`: ANN Recall–Latency Tradeoff

Given a seed frame vector $\boldsymbol{e}_{\text{seed}}$ (Eq. 11), we perform ANN over the `Frame.embedding` index. For HNSW, expected recall rises with ef roughly as $1 - \exp(-c \cdot ef)$ for a constant c dependent on graph degree m and data distribution; latency grows approximately linearly with ef. We set (m=32, ef=128) by default and expose ef for interactive tuning.

5.4 `geo-frames`: Sound BBox→Haversine Filtering

We first apply a bounding box that fully contains the query circle of radius R; hence there are no false negatives from the prefilter. We then apply Haversine distance

$$d = 2R_{\oplus} \arcsin \sqrt{\sin^2 \tfrac{\phi-\phi_0}{2} + \cos\phi \, \cos\phi_0 \, \sin^2 \tfrac{\lambda-\lambda_0}{2}}, \tag{15}$$

and finally optional time/speed predicates. Complexity is $O(M)$ where M is the number of frames in the bbox; a spatial index can reduce M.

6 Mathematical Analysis and Design Implications

Why mean pooling? By Lemma 4.1, normalized mean minimizes squared loss to constituent vectors; with duration weights it minimizes a weighted loss. For token-level hidden states, Eq. (1) is the MLE of a vMF with shared concentration under a uniform prior.

Variance-aware temporal pooling. Eq. (9) emphasizes seconds with higher visual variance (typically maneuvers, merges, intersections). Setting $\alpha = 0$ reduces to uniform averaging; increasing α increases sensitivity to dynamic scenes.

Cosine monotonicity. Proposition 5.1 shows cosine and Euclidean induce identical rankings on normalized vectors; thus all ANN backends that optimize either metric are consistent with our scoring.

Fusion sensitivity. Differentiating Eq. (11) with respect to λ yields

$$\frac{\partial}{\partial\lambda} \operatorname{sim}_{\cos(\boldsymbol{q},\boldsymbol{e}_t)} = \frac{\boldsymbol{q}^\top(\boldsymbol{\ell}_t - (\boldsymbol{e}_t^\top \boldsymbol{\ell}_t)\boldsymbol{e}_t)}{\|\boldsymbol{q}\|\,\|\boldsymbol{e}_t\|} \cdot \frac{1}{\|\boldsymbol{v}_t \oplus \lambda\boldsymbol{\ell}_t\|}, \tag{16}$$

indicating that the geo term helps when $\boldsymbol{q}$ has a nonzero projection on the component of $\boldsymbol{\ell}_t$ orthogonal to $\boldsymbol{e}_t$. This guides the λ sweep in practice.

Snap error. Proposition 5.2 shows expected temporal error $\leq \Delta/2$ and spatial error bounded by speed, justifying the *PhoneLog-first* policy when available.

KDE heatmaps. The KDE channel (Eq. 7) approximates a smoothed occupancy of detections; as $\sigma \to 0$ it reduces to raw counts, while large σ recovers a global density. Choosing $\sigma \propto 1/G$ maintains scale across grid levels.

Complexity summary. Let N_T, N_F be the number of text and frame vectors. Vector indexing is $O(N \cdot d)$ memory; HNSW build is $O(N \cdot m \log N)$. Query time is $O(\log N)$ nodes visited on average with a constant growing in ef.

7 Quality Standards & Controls

(unchanged; see previous version—schema guards, validation, index integrity, drift alarms, geo sanity, repro harness.)

8 Evaluation, Implementation, Ethics, Limitations

(unchanged in structure; metrics, ablations, implementation details, privacy.)

8.1 Vector Indexing and ANN

We create Neo4j `VECTOR INDEX` (cosine) on: `Transcription.embedding`, `Segment.embedding`, `Utterance.embedding` (all d=384), and `Frame.embedding` (d equals visual+context dims, e.g., 256–759). Global speaker linking uses FAISS HNSW [**?**, **?**] with command-line parameters from the runner (e.g., `-global-thresh 0.78`, `-faiss-k 64`, `-hnsw-m 32`, `-hnsw-ef 128`, `-thresh 0.72`, `-holdout-min 0.62`).

9 Mathematical Analysis

In this section we formalize the embedding constructions, retrieval scoring, speaker–segment mixture logic, and computational complexity. Each component is analyzed with both mathematical rigor and design implications.

9.1 Pooling and Normalization of Text Embeddings

Let $H \in \mathbb{R}^{T\times d}$ denote the last hidden states of a transformer encoder for a segment of T tokens, and let $m \in \{0,1\}^T$ be the binary attention mask. The masked mean pooling operator is defined as

$$\tilde{\boldsymbol{z}} \;=\; \frac{\sum_{t=1}^{T} m_t H_t}{\sum_{t=1}^{T} m_t}, \qquad \boldsymbol{z} \;=\; \frac{\tilde{\boldsymbol{z}}}{\|\tilde{\boldsymbol{z}}\|_2} \;\in\; \mathbb{R}^d, \tag{17}$$

where $H_t \in \mathbb{R}^d$ is the embedding vector of token t. This normalization step ensures all embeddings reside on the unit sphere $\mathbb{S}^{d-1}$, making cosine similarity (defined below) and Euclidean distance equivalent up to an affine transformation.

9.2 Cosine Similarity for Retrieval

For two vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^d$ the cosine similarity is given by

$$\mathrm{sim}_{\cos(\boldsymbol{a},\boldsymbol{b})} = \tfrac{\boldsymbol{a}^\top \boldsymbol{b}}{\|\boldsymbol{a}\|_2\,\|\boldsymbol{b}\|_2}. \tag{18}$$

Proposition. For unit-normalized vectors, cosine similarity and squared Euclidean distance induce identical rankings:

$$\|\boldsymbol{a}-\boldsymbol{b}\|_2^2 = 2\,(1-\mathrm{sim}_{\cos(\boldsymbol{a},\boldsymbol{b})}).$$

Thus, approximate nearest neighbor (ANN) structures optimized for Euclidean distance are directly compatible with cosine-based retrieval.

9.3 Speaker Mixture Edges

Let segment i span $[s_i, e_i]$ and diarization produce intervals $\{[a_j, b_j], \ell_j\}$ with labels ℓ_j. Define the overlap between segment i and diarization window j as

$$\mathrm{ov}_{ij} \;=\; \max\big(0, \min(e_i, b_j) - \max(s_i, a_j)\big). \tag{19}$$

The proportion of overlap assigned to label ℓ is

$$p(\ell \mid i) \;=\; \frac{\sum_{j:\,\ell_j=\ell} \mathrm{ov}_{ij}}{\sum_j \mathrm{ov}_{ij}}. \tag{20}$$

We retain a *mixture edge* (Segment i) $\to$ (Speaker ℓ) iff $p(\ell \mid i) \geq \tau$ for threshold τ (default $\tau = 0.05$). This produces both dominant and secondary speaker relations, enriching the graph with overlapping speech information.

9.4 Computational Complexity

Let N denote the number of embeddings (utterances, segments, or frames) and d their dimensionality.

- **Storage.** Raw embeddings require $\mathcal{O}(Nd)$ memory. Additional graph overhead arises from speaker/segment/utterance edges, typically $\mathcal{O}(E)$ where E is proportional to N for sparse schemas.
- **Index construction.** For HNSW (Hierarchical Navigable Small World graphs), build time is $\mathcal{O}(Nm \log N)$ where m is the maximum out-degree.
- **Query complexity.** The expected search complexity is $\mathcal{O}(\log N)$ node visits, with the multiplicative constant controlled by parameters (m, ef). Increasing ef improves recall at the cost of query latency.

9.5 Implications

These analyses justify (i) unit normalization of embeddings prior to indexing, (ii) the use of cosine similarity as a retrieval metric compatible with ANN backends, (iii) thresholded mixture edges to capture diarization uncertainty, and (iv) parameterized trade-offs in HNSW governing the recall–latency balance. .

10 Quality Standards & Controls (QS&C)

Schema guards. Enforce uniqueness on ids; btree on keys/timestamps; vector indexes created once per label; CI check: fail build if Neo4j version < 5.11.

Data validation. Ingest asserts monotone segment times; drops negative or NaN spans; verifies per-segment tokens match text length bounds; flags transcripts with empty text.

Index integrity. Post-ingest job verifies vector dimensions per label (384 for text; configured for frames), and re-runs a probe ANN to ensure non-empty recall.

Diarization sanity. Report per-file coverage (fraction of time covered by RTTM); warn if < 70

Embedding drift. Nightly canary embeds a fixed sentence set and a fixed set of frames and alerts on cosine drift > 0.02 against a pinned baseline.

Geo sanity. Validate
textttmph is non-negative and below a reasonable cap; interpolate or null-out obviously wrong bursts; log patching decisions.

Repro harness. Seal all configs (chunking, strides, ANN params,
lambda) in a versioned YAML captured alongside Neo4j schema version; include script hashes.

11 Evaluation

11.1 Datasets and Protocol

Report total hours, frames, GPS coverage, and front/rear availability. Split by route/day to test cross-day generalization. Manually label query prompts (e.g., "police siren," "merging onto highway").

11.2 Metrics

Text search: Recall@K, nDCG@K. **Frame similarity:** Precision@K under geo/time consistency thresholds (e.g., within 50 m and 2 min). **Geo precision:** fraction of `search-text` hits snapped within 50 m. **Speaker linking:** pairwise precision/recall/F1. **Latency:** p50/p95 for each primitive.

11.3 Ablations

Vary (L_c, S); grid set
$8times4, 16times9, 32times18$; heatmaps on/off; front/rear concatenation; location weight
lambda
in
$0, 0.25, 0.5$; ANN (m, ef). Present Pareto curves of quality vs latency and a power-usage note if measured.

12 Implementation Details

Chunk overlap 5 s; diarization gates (min RMS 0.005, min SNR 6 dB); HNSW (m=32, ef=128); thresholds from the runner (`-global-thresh 0.78`, `-thresh 0.72`, `-holdout-min 0.62`); grid pyramid; cosine similarity everywhere; UTC anchoring via `ZoneInfo`; reproducible `stable_id` via MD5 of salient parts.

13 Security, Privacy, and Ethics

Coordinates, faces, plates, and voices are sensitive. Recommendations: local-only storage, encrypted volumes, role-scoped credentials for Neo4j, access logging, and optional redaction layers. Avoid sharing raw frames; publish aggregate statistics and exemplar embeddings only.

14 Limitations and Future Work

No end-to-end learned fusion yet; vision signature is statistics over YOLO outputs; GPS outages handled heuristically. Future: CLIP/ViT frame encoders, learnable fusion, uncertainty-aware geo snapping, differential privacy for embeddings, online active learning for entities/speakers.

15 Field Taxonomy

Table 1 summarizes the major categories of node attributes observed across `Transcription`, `DashcamEmbedding`, `GlobalSpeaker`, `Utterance`, `Segment`, and `DashcamClip`.

Table 1: Representative field categories across node types.

Category	Examples	Description / Purpose
Identifiers	`id`, `key`, `elementId`	Unique handles for nodes, often stable hashes or composite keys used to link across modalities.
Temporal	`start`, `end`, `t0`, `t1`, `duration_s`, `created_at`	Anchors nodes in time, supports alignment of audio/video, utterance segmentation, and chronological queries.
Geospatial / Kinematic	`loc_vec`, `latitude`, `longitude`, `mph`, `loc_source`	Captures GPS or derived motion vectors; enables snapping and geo–filtering of queries.
Embedding	`embedding`, `vec`, `embedding_dim`, `dim`, `model`	Dense vectors from SBERT (384D) or YOLO grid–pyramids (759D), normalized for vector search. Used in ANN indices.
Structural Metadata	`level`, `view`, `grids_str`, `concat_views`, `fps`, `width`, `height`	Describes resolution, granularity (second/minute), camera view (F/R), grid pyramid parameters, or clip-level info.
Linguistic / Semantic	`text`, `tokens_count`, `is_lyrics`, `lyrics_score`, `review_needed`	Contains raw or processed textual content, token statistics, and classifiers for speech/lyrics/music overlap.
Provenance & Status	`status`, `method`, `weight_sum`, `updated_at`, `confidence`	Tracks how a node was created (e.g., ECAPA method), accumulation statistics, and revision history.

16 Reproducibility Checklist

To facilitate independent replication, we enumerate the core hyperparameters and system requirements used in our pipeline. Each item specifies both the chosen configuration and its role in the end-to-end workflow.

16.1 Audio and Transcription

- **Chunking and stride:** Whisper transcriptions were generated with `-chunk-len 90` s and `-stride 85` s, producing a 5 s overlap. This overlap prevents boundary word omissions and enables robust merging of adjacent chunks.

16.2 Diarization

- **Signal quality gates:** Minimum root-mean-square (RMS) energy of 0.005 and minimum signal-to-noise ratio (SNR) of 6 dB were enforced. These thresholds discard low-energy or noisy segments prior to speaker assignment.

16.3 Speaker Linking

- **ANN index:** Global speaker identities were resolved with HNSW parameters $m = 32$ and $ef = 128$. Candidate pool prefiltering used $k = 64$ nearest neighbors. Thresholds followed the diarization overlap rules defined in Section **??**.

16.4 Vision Embeddings

- **Grid pyramid:** YOLOv8 detections were aggregated across grids of sizes 8×4, 16×9, and 32×18.
- **Heatmap augmentation:** Kernel density heatmaps were optionally added as additional channels to enrich spatial statistics.

16.5 Multimodal Fusion

- **Location weight:** Geospatial/kinematic vectors were scaled by $\lambda \in \{0, 0.25, 0.5\}$. Ablations in Section 11.3 show sensitivity to λ.

16.6 Indexing and Infrastructure

- **Similarity metric:** All vector indices used cosine similarity, which is equivalent to Euclidean distance under unit normalization.
- **Database requirements:** Neo4j 5.11 or higher was required to support native `VECTOR INDEX` operations.

A Pipeline Orchestrator (trimmed)

Listing 1: runall.sh (key steps; idempotent where possible)

```
set -euo pipefail
cd ~/git/video-automation || { echo " cd failed"; exit 1; }

# Copy media
./audio_copy.sh || echo "copy audio failed"
./dashcam_copy.sh || echo "copy dashcam failed"
./bodycam_copy.sh || echo "copy bodycam failed"

# Whisper (chunked, merged)
./.venv/bin/python3 whisper_audio_chunked.py \
  --model medium --merge \
  --audio-root /mnt/8TB_2025/fileserver/audio \
  --transcriptions-root /mnt/8TB_2025/fileserver/audio/transcriptions \
  --chunks-root /tmp/chunks --pcm-wav \
  --chunk-len 90 --stride 85 \
```

```
  --delete-chunks-after --log-level INFO || true

# Diarization + ingest
./.venv/bin/python3 speakers.py || true
./.venv/bin/python3 ingest_transcriptions.py || true
./.venv/bin/python3 speakers_reconcile.py --batch 50 --only-missing || true

# Vision + metadata
./.venv/bin/python3 yolo_vehicle_detction.py || true
./.venv/bin/python3 metadata_scraper_iterator.py || true

# Global speaker linking
./.venv/bin/python3 link_global_speakers.py \
  --global-prefilter --global-thresh 0.78 --global-k 8 --global-index hnsw --global-m 32
    --global-ef 128 \
  --faiss-prefilter --faiss-k 64 --faiss-index hnsw --faiss-m 32 --faiss-ef 128 \
  --min-seg 0.7 --max-snips 8 --max-per-file 3 --snip-len 1.6 \
  --min-proportion 0.5 --min-rms 0.005 --min-snr-db 6.0 \
  --thresh 0.72 --holdout --holdout-min 0.62 --holdout-action drop-members \
  --audio-cache ./audio_path_cache.json --emb-cache ./emb_cache.sqlite || true

# Frame embeddings (grid pyramid + heatmaps)
./.venv/bin/python3 dashcam_yolo_embeddings.py \
  --resume --grid 16x9 --pyramid --heatmap --repair-missing-moov \
  --neo4j-uri bolt://localhost:7687 --neo4j-user neo4j --neo4j-pass livelongandprosper \
  --win-mins 10 || true

# Location patching + merge
./.venv/bin/python3 patch_missing_locations.py --win-mins 10 --validate-m 50 || true
./.venv/bin/python3 dashcam_merge_FR.py --base /mnt/8TBHDD/fileserver/dashcam --base /mnt
    /8TB_2025/fileserver/dashcam || true
```

B Cypher and CLI Excerpts

Listing 2: Vector search (utterance) with location enrichment

```
CALL db.index.vector.queryNodes('utterance_embedding_index', $k, $qvec)
  YIELD node, score
MATCH (t:Transcription)-[:HAS_UTTERANCE]->(u:Utterance)
WHERE u = node
// derive midpoint, snap to PhoneLog (W) or nearest Frame by position
RETURN u.id AS id, score, u.text AS text, t.key AS file_key,
       latitude, longitude, location_ts, location_source, speed_mph
ORDER BY score DESC
```

Listing 3: Frame similarity (approximate cosine ANN)

```
CALL db.index.vector.queryNodes('frame_embedding_index', $k, $seed_vec)
  YIELD node, score
RETURN node.id, score, node.key, node.frame, node.lat, node.long, node.millis, node.mph
ORDER BY score DESC
```

Listing 4: Geo-frames with Haversine after bbox prefilter

```
MATCH (f:Frame)
WHERE f.lat IS NOT NULL AND f.long IS NOT NULL
  AND f.lat BETWEEN $lat0 - (R/111320.0) AND $lat0 + (R/111320.0)
  AND f.long BETWEEN $lon0 - (R/(111320.0 * cos(radians($lat0)))) AND
                     $lon0 + (R/(111320.0 * cos(radians($lat0))))
WITH f, $lat0 AS lat0, $lon0 AS lon0, $R AS R
WITH f,
  6371000.0 * 2 * asin(sqrt(
    pow(sin(radians((f.lat-lat0)/2)),2) +
    cos(radians(lat0))*cos(radians(f.lat))*
    pow(sin(radians((f.long-lon0)/2)),2))) AS dist
WHERE dist <= R
RETURN f ORDER BY dist ASC LIMIT $limit
```

C Representative Node Samples (from DB)

To illustrate the heterogeneity of our Neo4j property graph, we show canonical examples of several node types. Each record carries both semantic payloads (text, embeddings) and structural metadata (keys, IDs, temporal anchors). These examples were extracted directly from the live database.

C.1 Transcription and DashcamEmbedding

Listing 5: Transcription and DashcamEmbedding nodes.


```
{
  "Transcription": {
    "id": "4c466e6a78d2a6d8a2697001583fcb44",
    "key": "2023_1228_144107",
    "text": "Recording, two channels started.",
    "embedding_dim": 384
  },
  "DashcamEmbedding": {
    "id": "2025_0710_204541_F|F|sec|5",
    "key": "2025_0710_204541_F",
    "level": "second",
    "grids_str": ["8x4","16x9","32x18"],
    "dim": 759,
    "loc_dim": 7,
    "model": "yolov8n"
  }
}
```


C.2 DashcamEmbedding (Detailed)

Listing 6: Expanded DashcamEmbedding with location vector.


```
{
  "id": "2025_0710_204541_F|F|sec|4",
  "key": "2025_0710_204541_F",
  "view": "F",
```

```
  "level": "second",
  "grids_str": ["8x4","16x9","32x18"],
  "dim": 759,
  "loc_dim": 7,
  "loc_source": "none",
  "loc_vec": [0.0, 0.0, 0.0, 0.0, 0.0, -0.7497, 0.6618],
  "concat_views": true,
  "model": "yolov8n",
  "t0": 4,
  "t1": 5,
  "vec": [0.0,0.0,...,0.5179,0.2279,0.0180,...]
}
```


This node encodes a one-second window, with a fused front-view visual embedding, grid–pyramid statistics, and a seven-dimensional geospatial vector.

C.3 GlobalSpeaker

Listing 7: Global speaker identity derived via ECAPA embedding.


```
{
  "id": "004526e161f9a96c7a46548b4f9dcf38",
  "status": "confirmed",
  "confidence": 0.7263,
  "method": "ecapa",
  "weight_sum": 150.23,
  "embedding": [0.0539,-0.0100,-0.0535,-0.0704,...],
  "created_at": "2025-09-01T04:04:56.865Z",
  "updated_at": "2025-09-01T12:06:57.979Z"
}
```


Global speakers link utterances across files, with confirmation scores and accumulated weight from multiple snippets.

C.4 Utterance and Segment

Listing 8: Utterance and aligned Segment from Whisper transcription.


```
{
  "Utterance": {
    "id": "7a9f09f981198773ba9d0b9f6c4f6755",
    "start": 0.0,
    "end": 8.22,
    "text": "A guitar case at a gun store. Why? There's a legal principle where if",
    "embedding": [-0.0284,0.1304,0.0076,-0.0738,...],
    "lyrics_score": 0.1900,
    "is_lyrics": false
  },
  "Segment": {
    "id": "07b31b92f259cc432d42c73fd25c83e6",
    "idx": 3,
    "start": 5.62,
    "end": 8.22,
    "text": "where if",
```

```
    "tokens_count": 4,
    "embedding": [0.0431,0.0430,-0.0159,0.0041,...]
  }
}
```

Segments preserve fine-grained timing, while utterances merge segments with speaker overlap to form coherent speech units.

C.5 DashcamClip

Listing 9: Dashcam clip metadata node.

```
{
  "id": "2025_0710_205441_R",
  "path": "/mnt/8TB_2025/fileserver/dashcam/2025/07/10/2025_0710_205441_R.MP4",
  "duration_s": 60,
  "fps": 30.0,
  "width": 2560,
  "height": 1440,
  "view": "R",
  "created_at": 1756639056305
}
```

Dashcam clips anchor embeddings and frames, providing physical file paths and resolution for reproducibility.

Acknowledgments

We thank the open-source communities behind Whisper, pyannote.audio, YOLOv8, SBERT, GLiNER, FAISS, and Neo4j.

References

Acknowledgments and References

We thank the open-source communities whose work enabled this research, including Whisper [**?**], pyannote.audio [**?**], YOLOv8 [**?**], Sentence-BERT [**?**], GLiNER [**?**], FAISS [**?**], and Neo4j [**?**].

Selected References

[1] N. Reimers and I. Gurevych. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. EMNLP, 2019.

[2] A. Radford et al. *Robust Speech Recognition via Large-Scale Weak Supervision*. arXiv:2212.04356, 2022.

[3] H. Bredin, A. Laurent et al. *pyannote.audio: Neural Building Blocks for Speaker Diarization*. ICASSP, 2020.

[4] G. Jocher et al. *Ultralytics YOLOv8*. GitHub, 2023. https://github.com/ultralytics/ultralytics

[5] Neo4j Inc. *Neo4j Native Vector Search*. Docs, 2023. https://neo4j.com/docs/operations-manual/current/performance/vector-search/

[6] Z. Urchade et al. *GLiNER: Generalist Lightweight Named Entity Recognizer*. arXiv:2309.13269, 2023.

[7] Y. Malkov and D. Yashunin. *Efficient and Robust Approximate Nearest Neighbor Search Using HNSW*. IEEE TPAMI, 2018.

[8] J. Johnson, M. Douze, and H. Jégou. *Billion-Scale Similarity Search with GPUs*. IEEE Trans. Big Data, 2019.

[9] W. Wang, H. Lu et al. *MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression*. NeurIPS, 2020.

[10] T. Wolf et al. *Transformers: State-of-the-Art Natural Language Processing*. EMNLP (System Demonstrations), 2020.