Search over the Visual World: Persistent Visual Memory, Layered Indexes, and Source-Grounded Evidence

Sankalp Nagaonkar Rohit Garg Ankit Raj Ashish Choithani Ashutosh Trivedi

VideoDB

{sankalp, rohit, ankit, ashish, ashu}@videodb.io

Technical Report · July 2026[1]

Thesis. Search over the visual world cannot be reduced to ranking video files. It requires infrastructure that turns archived and continuously arriving media into persistent, provenance-bearing understanding; organises that understanding through coexisting indexes; selects bounded, task-specific context; and returns evidence that remains playable at the original source. Search is the *context interface* between accumulated visual understanding and grounded action, and today it is a systems problem, not a training-time optimization.

Abstract. Most video-retrieval systems assume a bounded corpus and return ranked files or timestamps. Agents operating over cameras, screens, streams, and archives face a different systems problem: observations arrive continuously; different models interpret them at different temporal granularities; useful context must be selected without replaying the complete visual record; and a result must remain connected to inspectable source evidence. We argue that search over such a corpus is an infrastructure problem that cannot be reduced to ranking video files. We develop a conceptual and formal model of *search over the visual world* built on analyzer-defined *scenes*, persistent *understanding artifacts*, *visual memory* as coexisting scene spaces over shared source time, and capability-declared *indexes*; we draw a strict distinction between memory (everything retained), context (what is selected for a task), and evidence (the source intervals that ground it). The *VideoDB data format* (VDB) realizes this conceptual model in production, and a typed search surface exposes it: planned retrieval, a bounded stateful investigation mode, direct semantic, structured, and aggregate access, and grounded synthesis. We contrast this model-agnostic infrastructure (segmentation, sampling, model choice, embeddings, and ranking all exposed as system decisions, with live streams as first-class sources) with video-native foundation models offered as fixed APIs. In a complete-system semantic-retrieval comparison against a commercial video-native retrieval engine spanning 9,800+ natural-language queries over four public datasets, a pipeline of general-purpose components, none trained end-to-end for video retrieval, achieves higher macro-averaged Recall@1/@3/@10 (73.09/83.39/91.20 versus 65.75/77.13/89.10), while the baseline is higher at Recall@50 (96.42 versus 96.07). The results suggest that, today, retrieval quality over the visual world is governed more by system design than by video-specific pretraining, and that visual-memory infrastructure can deliver it while keeping source-grounded, playable evidence a first-class system object.

1 Introduction

The visual world unfolds continuously. Cameras watch loading docks, intersections, and operating rooms; screens record demonstrations, lectures, and incidents; meetings and broadcasts stream for hours; archives accumulate years of footage. AI systems are increasingly asked to act on this record: to notice, recall, verify, and decide. Yet the software through which they reach it still mostly exposes the visual world as

[1]Benchmark configurations and reproduction instructions are available at https://github.com/video-db/search-over-the-visual-world. The open-source *Deep Search* implementation of stateful retrieval discussed in §7 is available at https://github.com/video-db/deepsearch.

Property of the visual world	What it means	Systems obligation
R1 Temporal	Meaning lives in intervals, order, and change, not in atomic items	Represent intervals and query-specific boundaries, not only files
R2 Continuously changing	The corpus grows while being queried; a live stream has no final state	Understand and search live observations through the same abstraction as archives
R3 Distributed	Sources span cameras, screens, streams, and recordings	Join heterogeneous observations on stable identity and shared source time
R4 Larger than usable context	The corpus cannot be treated as one model context (Liu et al., 2024)	Select bounded, task-relevant context; never ship the corpus per request
R5 Understood in pieces	No single model produces all the observations a task needs	Preserve several analyzer-defined views; force no canonical representation
R6 Verifiable at the source	A conclusion an agent acts on must remain checkable against media	Retain provenance from every derived record down to a playable interval

Table 1. Properties of the visual world and the systems obligations they impose.

isolated files and transient model outputs. A multimodal model can analyze images and transcribe audio it is handed. An agent operating over months of recordings and days of live streams must do something harder: preserve what has already been observed, retrieve exactly the interval that matters, verify where it came from, and consume the answer as media rather than as a paraphrase. Recent long-video systems likewise route questions to selected temporal segments, maintain complementary memory representations, or incrementally construct retrievable state from streaming histories (Kim et al., 2025; Yeo et al., 2026; Liang et al., 2026; Xie et al., 2026).

Consider the questions such an agent must answer. *When did the forklift last enter bay 3, and was the dock door open? Find every moment in yesterday's launch stream where the demo failed. Show the second time the white car passes the intersection after 22:00.* Each question names an interval, not a file. Temporal-grounding work formalizes this distinction by localizing short moments within much longer sources (Soldan et al., 2022). Each question also depends on several kinds of understanding at once: objects, actions, speech, on-screen text. Each ranges over a corpus that was still growing when the question was asked. And each answer is useful only if it can be played and checked.

These questions expose six properties that jointly distinguish the visual world from a document corpus. Each property imposes a systems obligation that no model, however capable, discharges by itself (Table 1).

None of this is removed by more capable models. Models do not, by themselves, provide persistence, temporal alignment, indexing, retrieval, provenance, permissions, live access, context selection, evidence delivery, or action. These are obligations that arise before a model is invoked and after it returns. Large-scale live-video systems likewise expose scheduling, resource, quality, and processing-lag concerns as explicit systems obligations (Zhang et al., 2017). These obligations are the subject of this report.

Search over the visual world cannot be reduced to ranking video files. Its unit of retrieval is not the file but the scene*; its corpus is not media but* persistent visual memory*, the accumulated, provenance-bearing machine understanding of that media; and its output is not a hit list but* evidence*: source-grounded intervals that a person, a model, or an agent can inspect, play, and act on. Supporting it requires infrastructure that connects heterogeneous understanding, persistent memory, coexisting indexes, bounded context selection, and directly consumable evidence, uniformly over archived recordings and live streams.*

The argument proceeds as a chain of definitions with one feedback path (Figure 1). *Models produce understanding. A visual data format turns understanding into persistent memory. Indexes make memory searchable. Search selects the memory relevant to a task. The selected memory becomes context. Source-linked context becomes evidence.* In this account, search is the *context interface* of the visual world: the mechanism that translates accumulated visual understanding into a compact, task-conditioned representation while preserving the

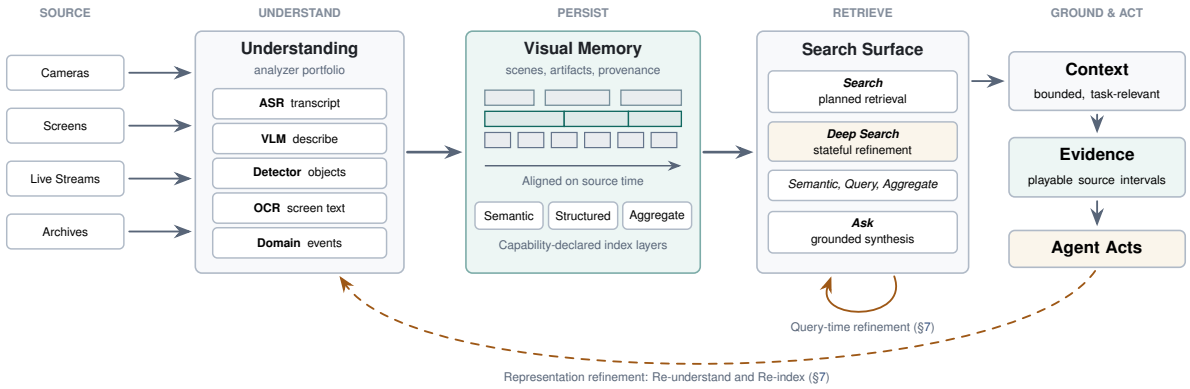


Figure 1. Search over the visual world. Sources feed a portfolio of analyzers; each row is a choice of model, segmentation, sampling, prompt, and schema, not a commitment fixed at training time. The VDB data format (§4) persists their outputs as visual memory: scenes and artifacts aligned on source time, carrying provenance, with capability-declared index layers over them. A typed search surface converts memory into bounded context and playable evidence for an agent. Two loops close the system: query-time refinement revises the plan within an investigation, and representation refinement directs re-understanding and re-indexing of memory itself.

route back to the original media. Search plays a second role as well: it is the probe that reveals whether the current representations of memory are adequate, and thereby drives their revision (§7).

We instantiate and evaluate these ideas in VideoDB, a production visual data infrastructure (VideoDB, 2026c). In the vocabulary of this report: search over the visual world is the scientific problem; visual data infrastructure is the systems category; the *VideoDB data format* (VDB) is the mechanism; persistent visual memory is the resulting capability; search is the interface through which an agent converts memory into task-specific context; streamable evidence is the source-grounded output; and VideoDB is the reference implementation and the evaluated system. The system is the vehicle of the paper, not its object.

Contributions This report makes four contributions.

1. A problem formulation and conceptual model of search over the visual world: analyzer-defined scenes, persistent understanding artifacts, visual memory as coexisting scene spaces over shared source time, and a strict memory–context–evidence distinction (§3).
2. The VideoDB data format (VDB): a logical data format binding source identity, source time, artifacts, provenance, and capability-declared indexes, with understanding separated from access so that representations can be re-derived without re-analysis, and tightly coupled to a streaming engine that realizes any logical selection, at millisecond granularity, as directly playable media (§4–§5).
3. A typed search surface over visual memory (planned retrieval, a bounded stateful investigation mode, direct semantic/structured/aggregate contracts, and grounded synthesis), together with the distinction between *indexed* and *resolved* scenes and a two-loop account of closed-loop visual search (§6–§7).
4. An architectural comparison of video-native model APIs with model-agnostic visual data infrastructure, and a controlled complete-system semantic-retrieval evaluation against a commercial video-native retrieval engine using 9,800+ natural-language queries from four public datasets, with explicitly scoped claims (§8–§9).

2 The Shape of the Problem

2.1 From corpus to world

Classical retrieval rests on assumptions that quietly fail here. A document corpus has a stable atomic unit (the document), one dominant representation (its text), an update model of insertion, and results a person can read directly. Video–text retrieval inherits the same shape: encode clip and query into one embedding

space and rank (Radford et al., 2021; Bain et al., 2021; Ni et al., 2022). This is effective for what it models, and we use such representations throughout. But as an account of search over the visual world it fails three ways at once. The *unit* is wrong: answers are intervals within and across sources, not files (R1). The *representation* is wrong: no single embedding space encodes counts, on-screen text, identities, procedures, and domain events at once, and different tasks need these at different fidelities and costs (R5). The *lifecycle* is wrong: the corpus grows while being queried (R2), model outputs evaporate unless persisted, and a ranked identifier is not something an agent can play or verify (R6).

Retrieval-augmented generation established the complementary lesson for text: models benefit from selective external context rather than from ingesting the corpus (Lewis et al., 2020), and long-context studies explain why bounded, well-chosen context beats indiscriminate stuffing (Liu et al., 2024). Recent work extends retrieval augmentation to video corpora (Jeong et al., 2025), and agent-memory systems demonstrate persistent stores mediated by retrieval (Park et al., 2023; Packer et al., 2023); but their memory objects are text records. What remains open is the question those lines converge on: *what is the retrieved unit of visual experience, and what infrastructure must exist for it to be produced, kept, found, bounded, and believed?*

2.2 The missing layer between models and media

Media containers and streaming protocols preserve encoded content and timing. Semantic observations, when they are kept at all, live in application-specific side stores. The result is a familiar production pathology: a second application repeats decoding and inference, writes a different schema, and implements its own path back to the source; observations produced for one pipeline cannot be reused by another; and nothing ties a derived row to the seconds of source video that justify it. The problem is not that a media file lacks a timeline. The problem is that stable source identity, source time, derived observations, retrieval capabilities, and evidence delivery never meet in one reusable machine interface.

Three database lessons shape the layer that is missing. First, derived results should be reusable across materializations rather than owned by one query path: the materialized-view principle (Gupta and Mumick, 1995). Second, every result should retain lineage to its source (Buneman et al., 2001); without lineage, verification (R6) degenerates into trust. Third, no single engine or representation serves all workloads (Stonebraker and Çetintemel, 2005); the consequence for visual data is not many databases but many coexisting access paths over one shared substrate. Ad hoc glue between models and media is also exactly the entanglement that accumulates as technical debt in production ML systems (Sculley et al., 2015).

2.3 What existing systems solve

Prior video systems address important stages of this lifecycle. VideoStorm manages resource, quality, and lag tradeoffs for live analytics (Zhang et al., 2017), while Scanner schedules raster computation over large video collections (Poms et al., 2018). Boggart builds query-independent representations for later heterogeneous analytics (Agarwal and Netravali, 2023), and EVA materializes expensive video-analysis outputs for reuse across subsequent queries (Xu et al., 2022). Rekall composes spatiotemporal labels into event queries (Fu et al., 2019). TASTI builds semantic indexes that accelerate ML-based queries over unstructured data (Kang et al., 2022a). VIVA supports interactive, relational video analytics with model selection (Kang et al., 2022b). V2V demonstrates that query results can themselves be synthesized as video (Winecki and Nandi, 2024). Together these systems cover computation, event algebra, reuse, index acceleration, analytics, and presentation. They primarily specialize in particular lifecycle stages rather than exposing as their central contract the connective layer studied here: heterogeneous, open-schema observations persisted with provenance; multiple capability-declared indexes coexisting over the same sources; and evidence that remains playable media. That connective layer is what this report formalizes and what VideoDB implements.

Term	Definition
UNDERSTANDING	Production of temporally grounded observations from visual media by a compatible analytical process.
ANALYZER	A concrete understanding process: a model with a segmentation policy and configuration (sampling, prompt, schema).
SCENE	An analyzer-defined, source-aligned temporal unit selected for understanding and indexing.
ARTIFACT	A persistent, model-derived observation attached to a scene, carrying provenance.
VISUAL MEMORY	The persistent collection of source-aligned scenes, artifacts, indexes, and provenance available for later retrieval.
INDEX	A materialized access path over selected artifacts, exposing declared capabilities.
SEARCH	Selection and ranking of relevant scenes from visual memory according to an intent and optional constraints.
CONTEXT	The bounded, task-relevant subset of visual memory selected for a model or agent.
EVIDENCE	The original, source-grounded interval supporting a result, conclusion, or action, consumable as media.
STREAM	The playable realization of evidence: any interval, or ordered composition of intervals across sources, generated on demand by the streaming engine directly from source-time coordinates.
INDEXED SCENE	An analyzer-defined temporal unit created to support understanding and retrieval.
RESOLVED SCENE	A query-specific evidence interval produced after reasoning over retrieved candidates.

Table 2. The vocabulary of search over the visual world. Memory is everything retained; context is what is selected for the present task; evidence is the source material that grounds it; a stream is evidence made playable, on demand.

3 A Conceptual Model of Search over the Visual World

We now define the objects of study. Table 2 summarizes the vocabulary; Figure 2 illustrates it over a small collection.

3.1 Sources and time

A *source* is an archived video or a live stream. Every source has two invariants: a stable identity, and a *source-time axis*: the clock of the media itself, starting at zero. For an archived recording the axis has a fixed extent; for a live stream it is still growing, and there is no final state (R2). Source time is the coordinate system in which every observation about a source is expressed. It is what lets observations produced by different processes, at different times, by teams that have never coordinated, meet again later on the same timeline.

3.2 Understanding, analyzers, and scenes

Definition 1 (Understanding)**.** *Understanding is the production of temporally grounded observations from visual media by a compatible analytical process.*

Understanding is deliberately plural (R5). It may be produced by classical computer-vision models; object detectors and trackers; OCR and speech systems; specialised domain models; small or medium open-weight vision–language models; hosted multimodal models; or frontier models, including video-native foundation models, which enter this framework as one member among many (§8). These processes differ by orders of magnitude in cost and latency and by task in reliability. Visual understanding is not the output of one large model; it is a portfolio, and the portfolio changes over time.

Definition 2 (Analyzer)**.** *An analyzer is a concrete understanding process: a model, a temporal segmentation policy, and a configuration: its frame-selection and sampling strategy, prompt, parameters, and output schema.*

Every element of an analyzer is a choice. Which model reads the frames, how the timeline is cut, how densely frames are sampled, what the model is asked, and what shape its answer takes are decisions made per analyzer, per workload, not properties fixed once for the whole system. Holding these choices open, rather than freezing them inside a trained artifact, is the design decision from which the rest of this report follows.

Definition 3 (Scene)**.** *A scene is an analyzer-defined, source-aligned temporal unit selected for understanding and indexing.*

We do not impose a universal scene segmentation. Online event-boundary research likewise treats segmentation for a live source as a causal operation over present and past frames rather than one that assumes access to the finished video (Jung et al., 2025). A speech system segments by utterance; a shot detector by visual cut; a sampling detector by fixed stride; a domain analyzer by domain events; a summarizer by narrative arc. Applying an analyzer to a source yields that analyzer's *scene space*: the set of scenes it produced, each recording four things: which source, which interval of source time, what was observed there, and the provenance of the observation. Distinct analyzers produce different, overlapping scene spaces over the same source: different boundaries, sampling frequencies, temporal resolutions, models, prompts, and schemas. Nothing requires the spaces to agree, and useful ones typically do not (Figure 2). What unifies them is not shared boundaries but the shared source-time axis.

Definition 4 (Understanding artifact)**.** *An understanding artifact is a persistent, model-derived observation associated with a scene and its provenance.*

Artifacts include descriptions, embeddings, transcript spans, detected objects and counts, tracks, OCR text, actions, locations, emotions, measurements, and domain-specific structures. The payload schema is open: these are examples, not an ontology. An artifact's provenance record states which analyzer, model, prompt, schema, and version produced the observation, and when. This follows database accounts in which lineage is retained and propagated through derived results (Buneman et al., 2001; Green et al., 2007). Provenance is what makes an artifact citable, comparable with artifacts from other analyzers, and safely replaceable. It is also what makes understanding *inspectable*: artifacts are data a user can read, export, and build on, not internal state hidden behind a retrieval endpoint.

3.3 Visual memory

Definition 5 (Visual memory)**.** *Visual memory is the persistent collection of source-aligned scenes, understanding artifacts, indexes, and provenance available for later retrieval.*

Visual memory is the union of every scene space produced over every source in a collection, together with the indexes built over them and the provenance that explains them. The term is operational, not an analogy to human memory: it denotes durable, source-aligned observations plus the access structures that let later requests recover bounded evidence. Model-level long-video work also uses complementary memory representations for query-dependent reasoning (Yeo et al., 2026); those inference-time memories are related motivation, not equivalents of the independently managed visual memory defined here. Two properties matter. Visual memory is the *complete retained superset*; it is never what is handed to a model. And it is *append-mostly and revisable*: new analyzers add coexisting views, and provenance allows old views to be replaced or retired deliberately rather than silently.

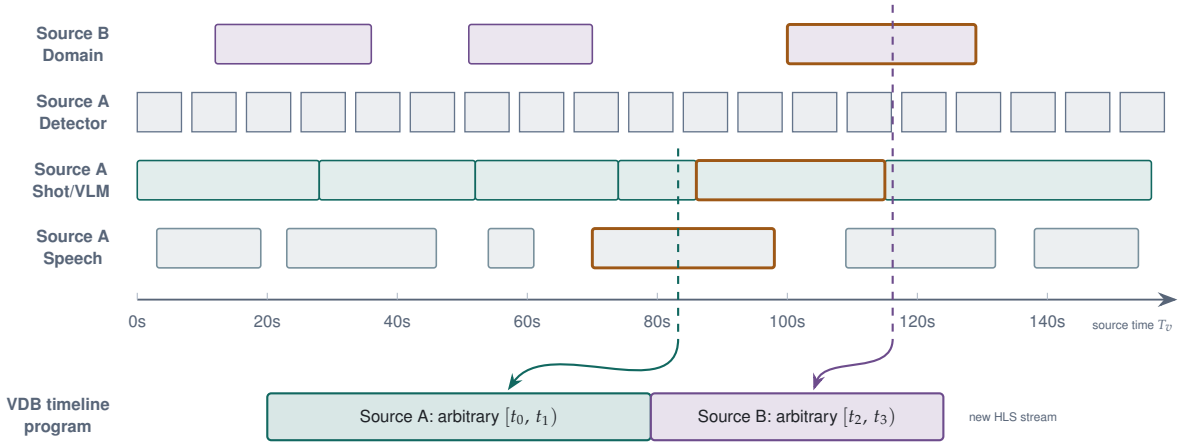


Figure 2. Coexisting scene spaces over a two-source collection, and the stream they resolve into. Analyzers segment each source differently: speech by utterance, a vision–language model by shot, a detector by fixed stride, a domain analyzer by events; every scene is anchored to shared source time. For a query, candidates (outlined) may come from several spaces, in several sources, with non-identical boundaries, and the resolved boundaries (dashed) are constructed to fit the question, coinciding with no stored boundary (§6.1). *Bottom:* a VDB timeline program concatenates arbitrary spans $[t_0, t_1)$ of source A and $[t_2, t_3)$ of source B, and the streaming engine delivers the program as one new HLS stream, generated on demand (§5). With fixed segments and fixed files off the read path, the whole visual world collapses into flexible start–end coordinates over a collection: a data system, not a file system.

3.4 Indexes

Definition 6 (Index)**.** *An index is a materialized access path over selected understanding artifacts, exposing declared capabilities.*

An index is derived from a chosen set of stored artifacts and declares what it can do: semantic retrieval over embedded fields, structured filters over typed fields, aggregation and metrics over grouped fields, computed properties, ranking, and domain-specific access. Prior video systems show that expensive analysis outputs can be materialized for reuse across later queries (Xu et al., 2022). Each index is a searchable *understanding layer* over the sources it covers, and several layers routinely coexist over the same source even when their scene boundaries differ. Because artifacts are exposed rather than hidden, indexing is something users do, not only something done to them: an application can read the stored understanding and index it its own way: its own embedding model, its own computed fields, its own layer composition. Two consequences follow from keeping stored understanding strictly separate from the indexes over it. A new index can be built from existing artifacts *without re-running the source-understanding model*: derivation reads artifacts; it never re-invokes the analyzer. And capability declaration lets a planner reason over the access paths that actually exist instead of inventing unsupported operations (§6). Related database systems optimize among alternative semantic operators under explicit quality, cost, and latency constraints (Russo et al., 2026), and combine vector traversal with relational filters and joins (Zhang et al., 2023b); these works motivate explicit capabilities without defining the visual data format used here. Computed properties make the point concrete: a detector artifact materialised as a numeric `person_count` lets "*scenes where more than two people are present*" evaluate as a typed filter over stored understanding, rather than as a fresh model pass over the corpus.

One further property distinguishes these indexes from conventional metadata search. Because every indexed scene carries source-time coordinates in the same format the streaming engine reads (§4), an index hit is not a pointer that some downstream export job must redeem: it terminates in an interval that is directly playable, and any set of hits, within or across sources, is directly composable into a single stream (§5). An index built over stored understanding is therefore also, automatically, an index into deliverable media.

3.5 Search, context, and evidence

Definition 7 (Search)**.** *Search is the selection and ranking of relevant scenes from visual memory according to an intent and optional constraints.*

Search takes an intent and optional constraints and proceeds in two stages. First, *candidate retrieval*: one or more index layers are consulted, and each contributes candidate scenes. Second, *reasoning over candidates*: the candidates are validated against the intent, ranked, and resolved into results. Candidates drawn from different layers need not share scene boundaries; they are combined on the source-time axis.

Definition 8 (Context)**.** *Context is the bounded, task-relevant subset of visual memory selected for a model or agent.*

Context is assembled, not found. Assembling it involves four decisions: *selection* (which intervals matter for this task), *hydration* (which stored fields ride along with them), *representation* (the cheapest faithful form: a transcript span, a description, a handful of frames, or the playable interval itself), and *budget* (how much the consuming model can use well Liu et al., 2024). The same visual memory yields different contexts for different tasks, and none of them changes the memory: context is disposable and rebuilt per task, while memory is durable. An agent does not carry the visual world with it; it carries the ability to retrieve exactly the part of the world the present task needs. We call the resulting property *context efficiency*: the system returns the representation an information need requires (scenes, rows, fields, or grounded text) rather than passing the visual record to a model. This report treats context efficiency as a design property of the interface; measuring it in model calls, visual tokens, and cost is future benchmark work (§10).

Definition 9 (Evidence)**.** *Evidence is the original, source-grounded visual interval supporting a search result, conclusion, or action, retained in directly consumable form.*

Evidence is not a generated answer, a similarity score, a file identifier, a textual citation, or a timestamp without an operational path to the media. An evidence item names its source and its interval on the source-time axis, carries the fields attached to it, and can be produced as media on demand (§5).

Produced as media is meant literally, and it adds a further term to the vocabulary. A STREAM is the playable realization of evidence: any interval, or ordered composition of intervals across sources, delivered on demand by the streaming engine coupled to the data format of §4. The timestamps this vocabulary traffics in are therefore not bookkeeping about media kept elsewhere; they are playable coordinates. Nothing in the delivery path requires a stored segment boundary or an intermediate file: the engine can begin at any point of any source, end at any other, and compose what lies between, together with overlays from other modalities, into media a person or an agent watches seconds after asking (§5). The distinction between *finding* a moment and *having* it as watchable media disappears. These distinctions anchor everything that follows:

Memory is everything retained. Context is what is selected for the present task. Evidence is the source material that grounds it. A stream is evidence made playable, on demand.

4 The VideoDB Data Format (VDB)

The conceptual model becomes infrastructure through a data format. The *VideoDB data format* (henceforth *VDB*, or the *VDB format*) is the durable representation in which the visual world is stored, indexed, and, just as essentially, *delivered*: once a source enters VDB, every later operation, from retrieval to composition to playback, is expressed over the format rather than over media files. For each source, VDB maintains one persistent cell that binds five things: the source's stable identity; its source-time axis; the scene spaces

and artifacts accumulated by every analyzer that has run over it; the indexes derived from those artifacts; and the provenance connecting each derived record to the analyzer, model, schema, and moment that produced it. Collections group cells and may carry collection-scoped indexes over many sources. Search contracts operate over cells and collections and return evidence views; results are never folded back into the cell as opaque state. The separation of logical video operations from physical representation has precedent in video storage systems (Haynes et al., 2021); VDB extends that separation to understanding artifacts, indexes, provenance, and evidence.

A logical data format VDB is a *logical* data format: it standardizes what the records of the visual world are and how they connect, while prescribing no codec, no container, and no byte layout. The distinction is the classical one between a logical model and its physical realization (Codd, 1970), and the nearest contemporary precedent is the open table format of analytic data systems (Armbrust et al., 2020): a logical specification layered over ordinary physical files, made operational by engines, through which files begin to behave like a database. VDB stands in the same relation to media. Sources remain ordinarily encoded media, and the format's contribution is the set of connections it makes durable: identity to time, time to scenes, scenes to artifacts, artifacts to provenance, artifacts to indexes, and indexes to evidence. What makes the format actionable rather than descriptive is its second, coupled half: a proprietary streaming *editing* engine that reads VDB directly. We characterize the engine by its observable contract only; the mechanism that realizes the contract is deliberately out of scope for this report. The contract is *temporal data independence*: an application addresses media purely in logical coordinates (this source or set of sources, this span of source time, from any timestamp to any timestamp, at millisecond granularity) and receives a conforming, immediately playable stream. No stored file boundary, segment boundary, or packaging decision is observable at the interface, and none constrains what can be addressed. Delivery arrives as a standard HLS (M3U8) manifest (Pantos and May, 2017), playable on commodity players, including Apple devices, with no special client support; overlays from other modalities (image, audio, text, captions) are declared over the same source-time axis and arrive composed into the delivered stream (VideoDB, 2026e). Generation is an interactive-latency operation, not a batch render (Appendix C, §5).

An alternative to file-based pipelines This coupling is what retires the media file as the working medium (VideoDB, 2026d). In file-based pipelines, every compilation, merge, edit, or overlay is a render job that decodes containers and encodes a new one; each derived MP4 is opaque to the next tool, and the archive fills with artifacts whose relationship to their sources is maintained by convention. Under VDB the same operations are declarations over logical coordinates, exposed to developers as a declarative *timeline* of clips and tracks over source time (VideoDB, 2026e) and evaluated at request time by the streaming engine; no intermediate file is created or touched, provenance survives because composition never leaves the format, and an MP4 or HLS rendition remains available as an *export* at the edge for interoperability, not as the substrate of work. Media becomes something the system computes over and responds with, the way a database responds to queries, rather than something it renders. Timestamps in VDB are playable coordinates, and §5 develops the consequence: search whose results are watchable the moment they are named.

P1: Understanding is separated from access Artifacts persist independently of the indexes over them (§3). This is the materialized-view principle applied to machine understanding (Gupta and Mumick, 1995; Xu et al., 2022): analysis is the slowly changing investment; access paths are derived and plural. A new index over existing artifacts requires no re-analysis; a replacement analyzer is useful exactly when it emits an artifact schema compatible with the intended access paths.

P2: Capabilities are declared Each index declares what it supports: semantic fields expose vector retrieval; queryable fields expose typed conditions; aggregatable fields expose grouping and metrics; stored fields can be hydrated into results. Declaration keeps the artifact model open-ended (any collection can advertise its own fields and types) while keeping planning honest: a planner selects among real access paths rather than assuming a universal one (Stonebraker and Çetintemel, 2005).

P3: Schemas are open Payloads and schemas are application-defined. Descriptions, transcripts, detections, OCR, measurements, and domain structures are examples, not a closed ontology; future analyzers add views without migrating old ones.

P4: Source time is the join key Every scene in every space is anchored to T_v. Temporal combination uses source and aligned interval keys, so observations from analyzers that have never heard of each other can be intersected, sequenced, and compared at retrieval time (R1, R3).

P5: Provenance is first-class Every derived record can answer: which analyzer, which model, which prompt and schema, over which interval, when (Buneman et al., 2001). Provenance is what allows layered memory to be audited, selectively rebuilt, and safely retired, and it survives into results so that evidence remains attributable.

P6: One format for archives and streams Nothing in the cell assumes a finished source. For a live stream, the source-time axis grows; analyzers with time-based segmentation append scenes as media arrives; indexes ingest them incrementally; identity and provenance are unchanged (R2). Archived and live media differ in the state of one axis, not in kind; this is why search, alerting, and evidence work over a stream that is still happening, and why an agent can be handed playable evidence of an event from moments ago (§5) (VideoDB, 2026c).

P7: Every selection is streamable Any temporal selection expressible in the format (a stored scene, a query-resolved interval, an ordered set of intervals spanning many sources of a collection) has a playable realization generated on demand by the streaming engine, together with any overlays declared over it. No pre-cut segment inventory and no per-request export job stand between a selection and its playback: the delivery path reads the format, not files (§5).

The concrete analyzer, segmentation, schema, indexing, and retrieval configuration used in the evaluation is reported in §9.2; the complete dataset-specific prompts appear in Appendix A.

5 Evidence as a Stream

The system object that closes the path from retrieval to action is the evidence stream, and its delivery properties are benchmarked directly. Every temporal result names its source and its interval; the streaming engine produces playback of exactly that interval, and an ordered set of resolved scenes, possibly spanning several sources of a collection, is composed into a single view delivered through standard manifest-based streaming (VideoDB, 2026c; Pantos and May, 2017), continuing prior work that treats video itself as a query result (Winecki and Nandi, 2024). Because the stream is constructed at request time from logical coordinates (P7), it is *boundary-free*: it can begin at any millisecond of any source, archived or still live, and end at any other, with no stored segmentation observable at the interface (Figure 2). It is also *immediate*. Appendix C reports observed median generation latency for playable evidence streams assembled from one source and from five sources, at requested durations from five seconds to ten minutes. Median generation remains below one second in every measured condition: 433–728 ms for single-source streams and 433–994 ms for five-source streams. We are not aware of another production video-search platform that documents an equivalent primitive for real-time cross-source, overlay-bearing composition. Verification of a retrieval result thereby changes from an act of trust into an act of playback.

6 Search over Visual Memory

A single search operation cannot serve every information need without collapsing distinctions that matter: planned retrieval, semantic similarity, exact selection, grouped analysis, iterative investigation, and grounded synthesis have different contracts. Related temporal-grounding work likewise distinguishes moment retrieval, highlight detection, and query-conditioned summarization as related but distinct tasks (Lin et al., 2023). The reference implementation therefore exposes one high-level *Search* with two modes, three direct-access contracts, and *Ask* (Figure 3). The contracts are complementary rather than successive

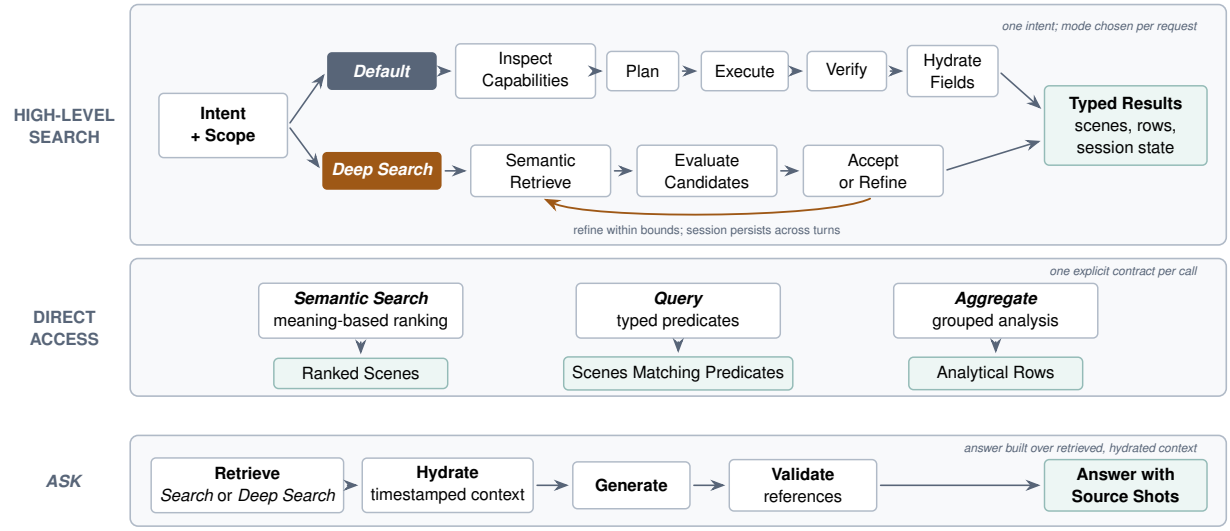


Figure 3. The search surface as staged flows. High-level *Search* takes one intent and runs it through the mode chosen for the request: default mode plans over declared capabilities, executes, optionally verifies with a precision-oriented filter, and hydrates stored fields; *Deep Search* mode iterates semantic retrieval within configured bounds and persists the session across turns. Direct access binds one explicit contract per call, each with its own typed output. *Ask* builds a validated answer over retrieved, hydrated context and returns the shots that ground it. Table 3 gives the corresponding contract mapping.

quality levels; the right choice depends on what the request needs to be true of its result. Table 3 states the mapping. Throughout the report, named contracts and modes are italicized, while quoted italics, as in "*label contains 'cell phone'*", mark literal request text. The contract chosen for a request determines what enters an agent's context: an analytical question should not return video, a verification task should not return only a count, and a synthesis should keep the scenes that ground it.

Information need	Contract	Returned context	Reference API
Meaning-based discovery	Semantic retrieval	Ranked, source-linked scenes	*Semantic Search*
Exact conditions	Structured query	Scenes matching typed predicates	*Query*
Counts and comparisons	Aggregation	Analytical rows, not intervals	*Aggregate*
Planned retrieval	Orchestrated search	Merged, verified, hydrated scenes	*Search* (default)
Iterative investigation	Stateful refinement	Results plus persisted session state	*Search* (*Deep Search*)
Grounded synthesis	Grounded answering	Answer plus its supporting scenes	*Ask*

Table 3. The search surface as a mapping from information needs to typed contracts. Reference-implementation names appear only to show one realization; the contracts, not the API names, are the claim.

High-level *Search* *Search* accepts intent and scope rather than an explicit index program. In *default mode* the system inspects available indexes, their schemas, and declared capabilities (P2); reasons jointly over the request and that catalog to select relevant indexes and direct methods; constructs and validates an execution plan; coordinates retrieval; merges compatible results in source time (P4); and hydrates requested stored fields into the returned shots (VideoDB, 2026c). The planner cannot assume an unavailable field or operation. Default search may invoke a verifier after retrieval: a precision-oriented filter that compares candidate shots with the original intent and removes nonmatching candidates. It cannot recover evidence omitted by candidate retrieval, an asymmetry that matters for how the loop in §7 is designed.

***Deep Search* mode** In *Deep Search* mode the system runs an agentic loop with verification and improvement around semantic retrieval (VideoDB, 2026b): it evaluates candidate acceptance, refines the retrieval strategy, repeats retrieval within configured limits, and persists session state so an investigation can continue across turns without discarding what has been learned. *Deep Search* currently reaches memory through its semantic-retrieval bridge only; it does not execute *Query* or *Aggregate*. Within that bridge, *Deep Search* can combine semantic retrieval with supported metadata filters derived from the request, then verify and improve the resulting candidates across iterations. Structured and analytical operations remain available through default *Search* and direct access.

Direct contracts *Semantic Search* is the explicit contract for meaning-based moment retrieval: the caller supplies text and may target one or more semantic indexes by name or ID. If neither index names nor index IDs are specified, VideoDB searches all indexes in scope that declare semantic capability. The caller may constrain eligibility with exact filters over fields those indexes expose; results are ranked shots that preserve source identity and intervals. *Query* executes typed conditions over a queryable index when the caller already knows field and operator semantics: exact categories, numeric thresholds, identifiers, conjunctions of declared predicates; matching temporal records return as shots, so exact selection still ends in source-linked evidence. *Aggregate* answers analytical questions (counts, groups, facets, metrics) and returns *rows*, not intervals. Keeping this type distinct prevents a grouped count from being mistaken for playable evidence; an application may use an aggregate to guide a later temporal request, but the row itself is not a shot. The typed contracts are literal where-clauses over declared fields: a *Query* request against an object layer of the form "*label contains 'cell phone', sorted by detection confidence*" returns exactly the matching shots, and an *Aggregate* request of the form "*detection confidence at least 0.8, grouped by label, counted*" returns one row per label. Declared operators (equality, containment, membership, numeric bounds, existence) compose with AND, OR, and NOT over any field an index exposes (VideoDB, 2026c). Neither request invokes a model at query time; both run entirely over stored understanding.

We also measured these direct stored-data contracts over five hours of indexed video. Across 100 measured calls per contract, median latency was 0.443 s for *Query* and 0.380 s for *Aggregate*. The TwelveLabs interface used in §9 does not expose corresponding direct contracts. Appendix C provides request examples and the measurement setup.

Ask *Ask* builds a synthesized response from retrieved visual memory: it retrieves through *Search* or *Deep Search*, hydrates timestamped artifact context, generates an answer over that bounded context, validates references to selected source entries, and optionally returns the corresponding shots. The answer and the evidence remain distinct typed outputs: text serves synthesis; shots remain inspectable source moments (VideoDB, 2026c; Gao et al., 2023).

6.1 Indexed scenes and resolved scenes

The scene spaces of §3 are built *before* any question is asked; queries arrive later and rarely respect stored boundaries. Temporal grounding research likewise evaluates query-specific interval localization (Soldan et al., 2022; Lin et al., 2023). This forces a distinction that retrieval systems usually leave implicit.

Definition 10 (Indexed scene)**.** *An indexed scene is an analyzer-defined temporal unit created to support understanding and retrieval.*

Definition 11 (Resolved scene)**.** *A resolved scene is a query-specific evidence interval produced after reasoning over retrieved candidates.*

Externally, the system retrieves candidate scenes from one or more layers, can inspect any relevant frame of the underlying media, and refines the final temporal boundary of what it returns. The granularity used to understand and index visual data therefore does not need to be the granularity returned as evidence: a question about a three-second gesture can be answered from half-minute indexed scenes, and a question about a full procedure can fuse many short ones (Figure 2). A resolved scene is an output,

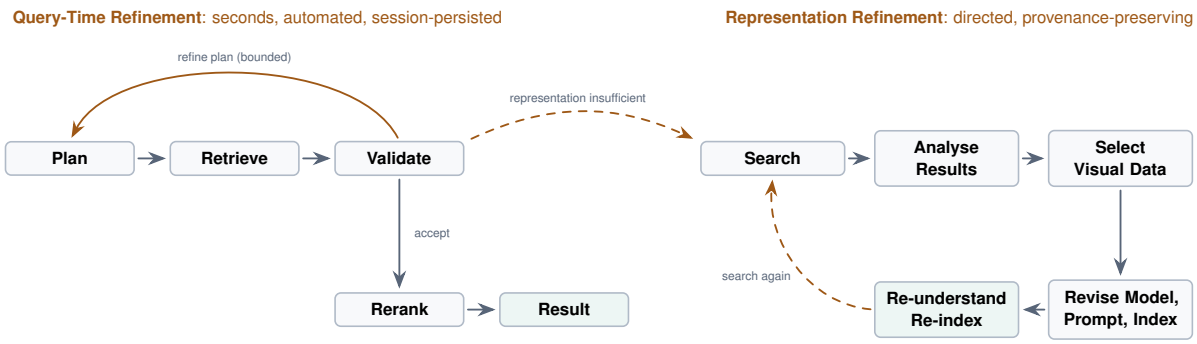


Figure 4. The two loops of closed-loop visual search. *Left:* in query-time refinement, the system plans, retrieves, and validates, then reranks into a result or refines the plan within configured bounds; session state persists across turns (*Deep Search*). *Right:* in representation refinement, analysis of search results leads a developer or an agent to select visual data; change the analyzer, prompt, scene strategy, computed property, or index; re-understand or re-index; and search again. The first loop is automated and demonstrated; the second is enabled by the separation of understanding, memory, and access (P1), and is not autonomous in the current system.

not an obligation: it need not be persisted, though a caller may store it as a new artifact if it has lasting value. And because the delivery path reads the format rather than stored segments (P7), a resolved scene is watchable the moment it is resolved: its query-specific boundary is served as-is by the streaming engine (§5), never rounded to a stored unit.

> *Search does not merely retrieve a previously stored scene. It can construct the scene appropriate to the question, and play it the moment it is constructed.*

7 Closed-Loop Visual Search

Search reads visual memory. It also *tests* visual memory: a query whose results are wrong or empty is information about the representations through which memory is being accessed: the analyzer, its prompt or sampling, the scene strategy, the computed properties, the index. We call the resulting discipline *closed-loop visual search* and separate it into two loops with different timescales, actors, and guarantees (Figure 4). Interactive video-database work provides a related precedent by synthesizing compositional event queries from limited user feedback (Zhang et al., 2023a).

7.1 Query-time refinement

The inner loop operates within a single investigation, and three operators describe it exactly. Retrieval first draws candidates from the index layers selected for the intent,

$$C_0 \;=\; \bigcup_k \text{Retrieve}(q,\, I_k),$$

where I_k ranges over the selected layers. A reasoning step then validates relevance, rejects false positives, inspects supporting frames, and refines the query within configured bounds,

$$q_{t+1} \;=\; \text{Refine}(q_t,\, C_t), \qquad C_{t+1} \;=\; \text{Retrieve}(q_{t+1},\, \mathcal{I}), \qquad t < T,$$

over the available layers $\mathcal{I}$, and finally resolves the scene or set of scenes appropriate to the question,

$$\hat{S} \;=\; \text{Resolve}(q,\, C_0, \ldots, C_T).$$

Resolve is where Definition 11 becomes operational: candidates arrive from layers with non-aligned boundaries, and the returned interval is constructed for the query rather than inherited from ingestion

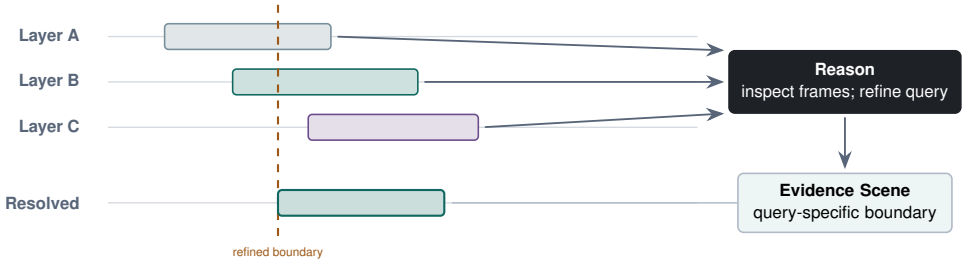


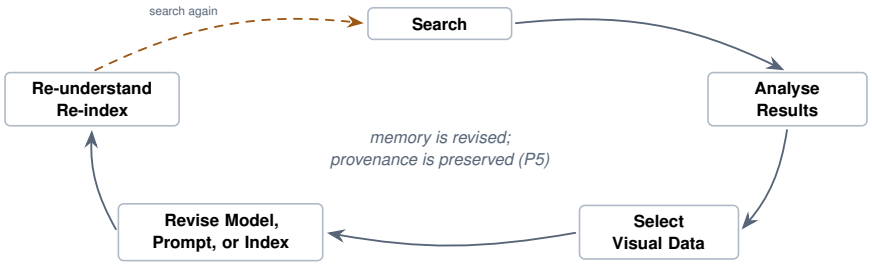
Figure 5. From retrieved candidates to resolved evidence. Candidates arrive from understanding layers with non-aligned boundaries; the reasoning step inspects relevant frames and refines the query within bounds; the returned evidence scene carries a query-specific boundary (dashed) that need not coincide with any stored scene (Definitions 10 and 11).

(Figure 5). In process terms the loop plans, retrieves, validates, then reranks into a result or refines the plan; it runs within configured limits and persists as session state.

This is the loop *Deep Search* mode implements, and the open-source *Deep Search* system built on VideoDB makes it concrete and inspectable (VideoDB, 2026b). There, a planner decomposes intent into subqueries, each targeting one or two named understanding layers (in the reference indexing pipeline: location, action, scene description, object description, topic, subplot summary, and whole-source summary layers, plus structured facets such as detected objects, emotion, and shot type); retrieval fans out with paraphrase variants and fuses candidates per subquery; a join plan combines subqueries in source time; a validator issues per-candidate verdicts (pass, ambiguous, fail) against the original intent and, when nothing passes, emits structured feedback with a bounded list of suggested plan operations; empty results trigger analysis and constraint relaxation along a declared fallback order rather than silent failure; an interpreter turns user steering (free text, choices on clarifying questions, "*more like this*" examples) into typed deltas on the plan; and the session, including plan history and verdicts, persists across turns. The loop composes retrieval, critique, and revision in the spirit of reflective retrieval-augmented systems (Asai et al., 2024; Yao et al., 2023), but its moves are typed plan edits over declared capabilities rather than free-form tool calls. Because the verifier is precision-oriented (§6), the loop's recovery mechanism for *missing* evidence is plan revision (new subqueries, relaxed constraints, different layers), not post-hoc filtering.

7.2 Representation refinement

The outer loop operates on memory itself:



A developer (or an agent workflow acting under a developer's direction) observes that a class of queries fails; localizes the failure to a representation (a prompt that never mentions dock doors; sampling too sparse to catch a gesture; a missing computed property; an absent domain analyzer); selects the affected sources or intervals; applies a different analyzer configuration or derives a new index from artifacts that already exist (P1); and re-runs the search. VOCAL-UDF demonstrates a related, more automated path in which a video-query system identifies missing analytic concepts and constructs or selects new UDFs (Zhang et al., 2025). Provenance (P5) makes the change inspectable: old and new views coexist, can be compared on the same queries, and can be retired explicitly. This loop is *enabled* by the infrastructure; it

is why understanding, memory, and access are kept separate. But we do not claim it is autonomous. In the current system, accepted results, rejected entries, edits, and downstream outcomes do not yet become durable learning signals that update future understanding and retrieval on their own (§10).

Search is therefore both the read interface to visual memory and the feedback mechanism for improving the representations through which memory is accessed. The two loops share one currency: evidence. The inner loop decides whether retrieved intervals answer the question; the outer loop decides whether the memory could have contained the answer at all.

8 Model or System: Two Ways to Build Video Search

Two design philosophies currently compete to define video search, and they disagree about where quality comes from.

The video-native model The first philosophy treats video as a first-class perceptual signal and trains an end-to-end foundation model for it. A representative commercial example is Twelve Labs: Marengo, a multimodal embedding model that maps video, audio, image, and text into one unified embedding space and powers its search and embedding APIs, and Pegasus, a video-first language model for generation (Twelve Labs Marengo Team, 2025). The interface this philosophy produces is an API around a trained artifact: as publicly documented, an application creates an index by naming a model version and enabling modalities, and the rest (how the timeline is segmented, how frames are sampled, how content is embedded, how results are ranked) is internal to the model and fixed at training time (Twelve Labs, 2026a). The representation is the product; improving the system means training the next model, and adopting it means re-indexing through the API.

The visual data infrastructure The second philosophy, the one this report develops, treats search quality as a property of the pipeline rather than of any single model. Every decision the video-native model internalizes is exposed here as a system choice: segmentation policy and scene granularity; frame rate and sampling per analyzer; which model produces each understanding layer, at which capability and cost tier; the prompt and output schema of each analyzer; the embedding model behind each semantic index; the ranker and fusion policy at retrieval; which computed properties and domain analyzers exist at all. The stored understanding is itself visible: artifacts are data the user can read, export, and re-index their own way (P1–P3). And because nothing in the data model assumes a finished source (P6), live streams are first-class: scenes append while the event is still happening, and search returns playable evidence of moments ago, a capability the uploaded-asset API model does not express. In the open-source *Deep Search* pipeline, these degrees of freedom are concrete configuration: per-stage model routing, per-layer index construction, detector on or off, prompts per enrichment stage (VideoDB, 2026b).

Table 4 summarizes the contrast. It should be read as a comparison of documented interfaces, not as an independent assessment of model quality. The two philosophies also compose rather than exclude one another. From the infrastructure's point of view a video-native model is simply one more analyzer in the portfolio; and this is not hypothetical: Twelve Labs' models run as a selectable analyzer inside VideoDB's real-time pipeline, a pairing both companies document (Twelve Labs, 2025). The architectural question is therefore not whether video-native models are good. It is where the remaining degrees of freedom should live: frozen inside a trained artifact, or exposed by the system that operates it.

8.1 Why search is a systems problem today

There is a structural reason to treat video search as a systems problem. Text models consume discrete, one-dimensional token sequences, although tokenizers differ across model families. Visual models must additionally discretize two spatial axes and time, and current systems make different choices. Image models patch space (Dosovitskiy et al., 2021); video models extend patches across time into spatio-temporal tokens (Arnab et al., 2021); embedding systems compress segments into one or several vectors (Bain et al., 2021; Ni et al., 2022; Twelve Labs Marengo Team, 2025); and long-video models adaptively reduce spatial and temporal tokens before reasoning (Shen et al., 2025). Each video-native model commits to an internal

	Video-native model API (Twelve Labs, as documented)	**Visual data infrastructure** (VideoDB, this report)
Primary artifact	A trained model behind an API	A logical data format (VDB) and a pipeline over many models
Understanding produced by	One proprietary model family (Marengo embeddings; Pegasus generation)	An open portfolio: ASR, detectors, OCR, VLMs of any tier, domain models; video-native models included
Segmentation & sampling	Internal to the model	Chosen per analyzer: shot- or time-based scenes, frame rate, frames per scene
Embedding space	Unified, fixed at training time; exportable to an external vector stack via a separate embed API	Chosen per semantic index; replaceable without redoing understanding
Prompts & output schemas	Not exposed at indexing	User-defined per analyzer; open schemas
Ranking	Internal	Selectable ranker, reranker, and fusion policy
Visibility of understanding	Embeddings and internal state opaque	Artifacts readable and exportable; users index their own way
New access path over old media	Re-index through the model	Derive a new index from stored artifacts; no re-analysis
Live streams	Managed search documented over uploaded, indexed assets; real-time workflows documented through the VideoDB integration	First-class sources; scenes append in real time; evidence from moments ago
Evidence output	Timestamped segments with playback metadata for indexed sources; cross-source composition left to the application	Query-resolved scenes; programmable composition (sequence, stack, overlay) into one playable evidence stream, archived or live
Improvement path	Train and migrate to the next model version	Recompose: swap an analyzer or index; provenance keeps old and new comparable

Table 4. Two architectures for search over video. The left column reflects the public product interface of a commercial video-native system as of July 2026 (Twelve Labs Marengo Team, 2025; Twelve Labs, 2026a); the right column is the infrastructure developed in this report. The columns compose: a video-native model can serve as one analyzer inside the infrastructure (Twelve Labs, 2025).

discretization and exposes only part of that choice through its interface. When that frozen choice is wrong for a workload (too coarse for a three-second gesture, too short-horizon for a forty-minute procedure, blind to a domain's structures), the application cannot fix it without a different model.

While temporal-visual representations remain workload-dependent, retrieval quality over the visual world can also be influenced by decisions a system can expose and revise (segmentation, sampling, model portfolio, index composition, ranking) and by the loops that revise them against evidence (§7). Search engine design over video is, today, a systems problem rather than a training-time optimization.

We take the image to be the native unit of stored visual information: a video is images ordered in time, joined by sound. The infrastructure is built on that position: understanding is anchored to frames and intervals on the source-time axis, and temporal structure is reconstructed by alignment across layers rather than baked irreversibly into any single representation.

9 Empirical Study

Section 8 locates specialization in two different places: a video-native system concentrates it inside a trained representation, whereas visual data infrastructure exposes segmentation, sampling, schemas, indexes, and ranking as choices that can be composed around stored understanding. That architectural contrast leaves an empirical question. Can a pipeline assembled from general-purpose components operate in the same retrieval-quality regime as a managed video-native search path?

We answer with a shared-corpus comparison between VideoDB and TwelveLabs. The unit of comparison is the retrieval path an application invokes, not an embedding model removed from its surrounding system. VideoDB uses collection-level semantic retrieval followed by reranking. TwelveLabs uses its managed Marengo 3.0 search path (Twelve Labs Marengo Team, 2025; Twelve Labs, 2026b). Each path is

configured through the controls its interface provides. The comparison evaluates the complete semantic-retrieval path exposed by each system. VideoDB's other search contracts are outside this experiment.

9.1 Corpus and Evaluation Design

The evaluation uses a shared corpus. Both systems indexed the same source videos and, for each request, received the same query text and dataset-defined relevant item. To keep the comparison from resting on one content regime, the corpus combines four public datasets and 9,800+ natural-language queries. MSVD contributes short open-domain clips paired with paraphrastic descriptions (Chen and Dolan, 2011). YouCook2 contributes temporally localized steps in instructional cooking videos (Zhou et al., 2018). VATEX contributes diverse web video with rich multilingual captions (Wang et al., 2019). MSR-VTT contributes broad-category web clips paired with multiple natural-language descriptions (Xu et al., 2016). The four regimes stress different retrieval behavior: short semantic correspondence, procedural action, multi-action description, and broad-category multimedia captioning.

Dataset	Videos	Hours	Queries
MSVD	213	7.3567	5,476
YouCook2	149	12.2989	1,152
VATEX	336	15.0000	3,000
MSR-VTT	187	12.9808	206
Total	**885**	**47.6364**	**9,834**

Table 5. Exact composition of the evaluated corpus.

9.2 Evaluated Retrieval Paths

With corpus and relevance judgments fixed, the remaining choices concern how each system represents, retrieves, and ranks the shared media. Table 6 summarizes those choices. To keep candidate pools dataset-specific, VideoDB uses one collection per dataset, while TwelveLabs uses one index per dataset. VideoDB makes scene construction, artifact shape, index composition, and reranking configurable. Marengo keeps the corresponding decisions provider-managed.

Term	VideoDB configuration	Marengo 3.0 configuration
SCENE	Five-second source intervals	Provider-managed
ANALYZER	English timed transcripts plus `ultra`-tier VLM analysis of six sampled frames per scene with aligned transcript context	Provider-managed visual and audio analysis
ARTIFACT	Strict-JSON scene records with fields configured for each use case (Table 7)	Internal representation without prompt or schema controls
INDEX	One collection per dataset, with a separate semantic index for each VLM-generated artifact field	One index per dataset; representation and index internals provider-managed
SEARCH	The unmodified caption retrieves 50 scene candidates through collection-level semantic search across the field-specific indexes in the dataset collection	The same caption retrieves 50 clip-grouped results from the dataset index using visual, audio, and transcript signals
RERANKING	BGE Gemma ranks candidates using concatenated scene fields	Provider-managed
EVIDENCE	Ranked, source-aligned shots with streamable evidence generated at a sub-second client-observed median	Ranked clips with timestamps

Table 6. Retrieval configurations compared in the shared-corpus experiment.

From source media to indexed representation VideoDB first constructs the representation that retrieval will search. Each dataset occupies a separate collection, and every source in that collection is divided into five-second scenes. The built-in `spoken_words` analyzer produces a timed transcript artifact and was configured here for English (VideoDB, 2026a). For each scene, a VLM analyzer at the `ultra` tier receives six uniformly sampled frames together with the aligned transcript and produces a strict-JSON `scene` artifact.

The infrastructure does not prescribe a single domain schema. Artifact fields can be configured for the information a use case needs (Table 7). In this evaluation, MSVD represents compact subject–action content, YouCook2 represents procedural state and ingredients, VATEX represents richer ordered activity, and MSR-VTT represents broad-category multimedia descriptions, including speech-oriented content. Each VLM-generated artifact field is indexed separately, so every dataset collection contains multiple semantic indexes. Collection-level search spans those indexes. Appendix A reproduces the exact prompts and schemas. This makes adaptation an explicit prompt-and-schema choice rather than requiring a new retrieval architecture or foundation model.

Dataset	Fields	Semantically indexed representation
MSVD	7	Scene and short captions, action, location, characters, objects, and on-screen text
YouCook2	7	Scene description, action, recipe step, ingredients, tools, food state, and on-screen text
VATEX	8	Scene and rich captions, ordered actions, location, characters, objects, on-screen text, and speech
MSR-VTT	8	Scene and short captions, action, location, characters, objects, on-screen text, and speech

Table 7. Use-case-specific VideoDB artifact schemas.

From candidates to ranked evidence Semantic retrieval determines which scenes enter the candidate set; reranking determines their order. For this experiment, semantic retrieval uses top_k $= 50$, so each query supplies 50 candidates to the reranker. This candidate depth is an evaluation setting, not a system limit. VideoDB makes both the ranking model and the construction of candidate text configurable. We evaluate four rerankers under two input modes: BGE Gemma (Beijing Academy of Artificial Intelligence, 2024), mxbai Rerank Base v2 and Large v2 (Li et al., 2025), and mxbai Edge ColBERT v0 32M (Takehi et al., 2025). *Match Exact* uses only the artifact field that produced the semantic match. *Concatenate* combines all VLM text fields for the scene, allowing the reranker to judge the candidate against the fuller stored representation.

Crossing the four rerankers with the two input modes yields the eight operating points in Table 8. The table reports four-dataset macro recall alongside median component latency; Appendix B reports every dataset-level result. BGE Gemma with Concatenate is highest at each cutoff affected by candidate ordering: 73.09 at R@1, 83.39 at R@3, and 91.20 at R@10. All configurations have the same 96.07 R@50 because they reorder the same 50 candidates. We therefore fix BGE Gemma with Concatenate for every VideoDB value in Table 9. Its median component latency is 0.716 s, while the alternatives range from 0.289 to 1.288 s. The selected point favors retrieval quality rather than minimum latency. Because reranking consumes fields already present in the artifact, a deployment can change this operating point without regenerating the stored understanding.

		Macro recall (%)				
Model	**Input mode**	**R@1**	**R@3**	**R@10**	**R@50**	**Median (s)**
BGE Gemma	Match Exact	67.66	80.74	90.04	96.07	1.288
BGE Gemma	**Concatenate**	**73.09**	**83.39**	**91.20**	96.07	0.716
mxbai Base v2	Match Exact	66.10	72.95	84.61	96.07	0.289
mxbai Base v2	Concatenate	69.44	78.91	87.98	96.07	1.117
mxbai Large v2	Match Exact	65.56	68.78	78.86	96.07	0.336
mxbai Large v2	Concatenate	67.20	71.91	82.37	96.07	0.457
mxbai Edge ColBERT	Match Exact	65.06	77.49	88.71	96.07	0.407
mxbai Edge ColBERT	Concatenate	68.85	80.28	89.14	96.07	0.512

Table 8. Macro recall and median component latency for the eight reranking operating points. Recall is the unweighted mean across the four datasets. Bold marks the selected configuration and its leading early-rank recall.

The managed video-native path The benchmark uses one TwelveLabs index per dataset. The four indexes apply the same provider-managed Marengo 3.0 configuration: visual and audio information is indexed, then search operates over visual, audio, and transcript signals. Transcript search combines lexical and semantic matching with OR. Results are grouped by clip, and the first 50 are returned (Twelve Labs, 2026b). Marengo does not expose dataset-specific prompts or artifact schemas, so the same configuration remains in place across datasets (Twelve Labs, 2026a). VideoDB makes use-case-specific representation a configurable system choice; Marengo keeps it provider-managed.

9.3 Retrieval Quality

We measure retrieval quality with Recall@k, the percentage of queries for which a relevant item appears among the first k results. The macro-average is the unweighted mean of the four dataset values. At query time, VideoDB applies collection-level semantic retrieval followed by the fixed BGE Gemma Concatenate reranking configuration. TwelveLabs applies the Marengo 3.0 path above. We refer to the complete fixed VideoDB path simply as *VideoDB*. Table 9 reports aggregate recall from the two runs.

	VideoDB				**TwelveLabs**			
Dataset	R@1	R@3	R@10	R@50	R@1	R@3	R@10	R@50
MSVD	**70.10**	**80.73**	89.17	96.39	67.86	78.22	**89.48**	**97.10**
YouCook2	**65.96**	**80.98**	**93.48**	**97.87**	47.07	65.56	84.18	96.41
VATEX	83.46	90.28	95.24	97.77	**85.43**	**92.40**	**97.30**	**99.43**
MSR-VTT	**72.82**	**81.55**	**86.89**	92.23	62.62	72.33	85.44	**92.72**
Macro-average	**73.09**	**83.39**	**91.20**	96.07	65.75	77.13	89.10	**96.42**

Table 9. Complete-system semantic-retrieval comparison using 9,800+ natural-language queries. Values are percentages; the macro row is the unweighted arithmetic mean of the four dataset rows. Bold marks the higher system for each dataset and cutoff.

The observed pattern Table 9 contains two results. First, on the unweighted macro-average, VideoDB is higher through the first ten returned results: 73.09 versus 65.75 at R@1, 83.39 versus 77.13 at R@3, and 91.20 versus 89.10 at R@10. The margin narrows as the cutoff grows, and TwelveLabs is higher at R@50, 96.42 versus 96.07.

Second, the ordering is dataset-dependent rather than universal. YouCook2 is the clearest separation in VideoDB's favor: it is higher at every reported cutoff, including by 18.89 points at R@1. MSVD divides by cutoff, with VideoDB higher at R@1 and R@3 and TwelveLabs higher at R@10 and R@50. MSR-VTT follows a similar early-rank pattern: VideoDB is higher through R@10, while TwelveLabs is higher at R@50. TwelveLabs is higher at every reported cutoff on VATEX.

What the comparison establishes The macro-average result answers the empirical question posed at the beginning of the section: a configurable pipeline of general-purpose components can operate in the same

quality regime as a managed video-native retrieval path and can be higher at the early ranks. The dataset-level variation is equally important. The result is not a universal ordering of the two systems; it shows that retrieval quality depends on how a representation and its ranking path fit the query distribution.

YouCook2 is consistent with the value of workload-specific representation: its prompt, schema, semantic index, and reranking input are all shaped around procedural cooking language. Because those choices move together, the study does not assign the advantage to any one of them. The supported architectural conclusion is compositional. Specialization can be assembled through exposed choices in understanding, indexing, and ranking; it need not reside exclusively inside one end-to-end video model. Conversely, a video-native model remains compatible with the same infrastructure as a replaceable analyzer (§8).

9.4 Configuring the Searchable Representation

The complete-system comparison fixes one VideoDB retrieval path so that every request encounters the same representation and ranking procedure. That fixed path is an evaluation choice, not a constraint of the infrastructure. We next vary two decisions made when the searchable representation is constructed: the analyzer that writes each scene artifact, and the temporal and visual sampling from which the artifact is written. In both studies, the dataset prompts and schemas, collection and index topology, semantic-search depth, and BGE Gemma Concatenate reranker remain fixed. Both studies use an evaluation subset drawn from MSVD, YouCook2, and VATEX, comprising 15 hours of video and more than 4,000 natural-language queries. In this subsection, video Recall@k counts a query as a hit when at least one of the first k returned scenes belongs to the dataset-defined source video; the reported macro is the unweighted mean across the three datasets.

9.4.1 Analyzer Choice

An analyzer determines what language is stored in each scene artifact and, therefore, what the semantic indexes can retrieve. The complete-system path in §9.2 divides each source into five-second scenes, samples six frames uniformly from each scene, and supplies the analyzer with those frames and the aligned transcript. This study preserves that scene input and substitutes one of six VLMs as the analyzer: Qwen 3.5 9B and 27B (Qwen Team, 2026b,a); GPT-5.4 Mini and GPT-5.5 (OpenAI, 2026a,b); and Gemini 3.1 Pro and Gemini 3.6 Flash (Google, 2026a,b). Each VLM receives the same frames, transcript context, dataset-specific prompt, and output schema; the resulting artifacts then pass through the same index topology, top_k = 50 semantic retrieval, and BGE Gemma Concatenate reranker. Table 10 reports the unweighted three-dataset macro video recall.

Analyzer	R@1	R@3	R@10	R@50
Qwen 3.5 9B FP8	32.87	43.69	54.61	62.28
Qwen 3.5 27B FP8	72.87	84.17	92.38	97.23
GPT-5.4 Mini	77.44	86.96	94.11	97.53
Gemini 3.1 Pro	79.52	88.17	94.91	98.15
GPT-5.5	80.33	88.84	95.10	98.40
Gemini 3.6 Flash	**80.59**	**89.28**	**95.55**	**98.53**

Table 10. Macro video recall for six analyzer choices under a fixed downstream retrieval path. Values are percentages and are the unweighted mean across the three datasets. Bold marks the highest observed value at each cutoff.

Table 10 contains two patterns. First, analyzer choice can change whether the relevant source enters the candidate set at all. Qwen 3.5 9B reaches 62.28 at R@50, while each of the other analyzers reaches at least 97.23. BGE Gemma only reorders the 50 candidates produced for a given representation, so it cannot recover a source that is absent from that set. The R@50 gap therefore locates this difference before reranking, in the artifacts and semantic candidate generation.

Second, the higher-performing analyzers occupy a much narrower quality band. Across GPT-5.4 Mini, Gemini 3.1 Pro, GPT-5.5, and Gemini 3.6 Flash, macro R@1 ranges from 77.44 to 80.59 and R@50 from 97.53 to 98.53. Gemini 3.6 Flash has the highest observed macro at every cutoff, but its margin over GPT-5.5

is at most 0.45 points through R@10 and 0.13 points at R@50. Differences of this size do not establish a universal model ordering, particularly when the same dataset-specific prompts need not be equally well matched to every model family. The result makes the role of model-agnostic infrastructure precise: the analyzer can be replaced without changing the surrounding retrieval path, while the retrieval value of the artifacts it produces remains visible and measurable.

9.4.2 Scene Duration and Frames per Scene

A scene artifact represents a bounded interval of source time using a finite sample of frames. Scene duration controls how much time the artifact covers; the frame count controls how densely that interval is shown to the analyzer. To examine these choices, we fix GPT-5.5 and build nine searchable representations by pairing scene durations of 2, 5, and 10 seconds with 2, 6, and 10 uniformly sampled frames per scene. For every pairing, GPT-5.5 receives the corresponding frames and aligned transcript, while the prompts, artifact schemas, index topology, top_k = 50 semantic retrieval, and BGE Gemma Concatenate reranker remain unchanged. Table 11 reports macro video recall for the nine representations.

Duration (s)	Frames	R@1	R@3	R@10	R@50
2	2	77.65	86.77	94.37	97.75
2	6	78.79	87.44	94.21	97.96
2	10	78.97	87.19	94.04	97.79
5	2	77.61	86.60	94.95	97.74
5	6	**80.33**	88.84	95.10	**98.40**
5	10	79.54	87.95	94.06	97.39
10	2	77.65	86.93	94.11	97.44
10	6	79.64	88.87	**95.59**	98.09
10	10	79.05	**89.26**	94.58	97.65

Table 11. Macro video recall across scene-duration and frames-per-scene choices with GPT-5.5 fixed. Values are percentages and are the unweighted mean across the three datasets. Bold marks the highest observed value at each cutoff.

For the source-video retrieval queries represented by these three datasets, the observed changes are modest. Across the nine configurations, macro recall spans 2.72 points at R@1, 2.66 at R@3, 1.55 at R@10, and 1.01 at R@50. No duration leads at every cutoff: five seconds and six frames is highest at R@1 and R@50, ten seconds and ten frames is highest at R@3, and ten seconds and six frames is highest at R@10. Averaged over the three durations, six frames has the highest observed recall at every cutoff, while increasing the sample to ten frames provides no consistent gain on this subset.

We therefore use five-second scenes and six frames as a balanced operating point for the complete-system comparison, not as a universal optimum. The narrow spread in this study does not imply that scene construction is generally inconsequential: duration and frame count still determine the temporal extent and visual detail stored in each artifact, as well as the amount of indexing work. Workloads with different event timescales or information requirements may favor a different point.

10 Future and a Research Agenda

Persistent, not adaptive The central limitation is that the infrastructure currently provides *persistent* visual memory, not memory that improves itself through use. Artifacts and indexes survive requests, and *Deep Search* persists investigation state; but accepted results, rejected entries, edits, and downstream outcomes do not yet become durable learning signals that update future understanding and retrieval. The end state we envision is a *self-improving* search system: one that treats every investigation as a training episode and closes the outer loop of §7 autonomously. Reinforcement-learning formulations fit that loop's structure naturally: retrieval plans, analyzer and prompt selection, segmentation policies, and index composition are actions; evidence accepted, corrected, or discarded downstream is the grounded reward;

and the provenance-preserving format supplies the replay substrate such learning requires, since old and new representations coexist and remain comparable on identical queries (P5). An adaptive system could collect explicit signals (relevance judgments, corrections, ratings) and implicit ones (inspected shots, playback duration, compiled clips, follow-up refinements, task completion). Implicit behavior must be interpreted cautiously, since interaction does not equal relevance (Joachims et al., 2005). Such signals could support preference learning (Christiano et al., 2017), personalized ranking through contextual bandits (Li et al., 2010), and carefully bounded reinforcement learning (Sutton and Barto, 2018); personalized reranking is one practical first step. Adaptation is also a memory-management policy: deciding when to run a new analyzer, consolidate repeated observations, revise schemas, rebuild indexes, or retire stale memory, while preserving source lineage and deletion semantics. That is the outer loop of §7 executed by the system itself, reinforcement learning over representation and retrieval policy rather than over model weights alone, under constraints that keep every change inspectable and reversible.

Toward native temporal representations Our position that the image is the native stored unit of visual information (§8.1) is a bet on the present, not a law. The present, however, is notable: on the evidence of §9, non-native solutions (general-purpose components composed by retrieval infrastructure) currently out-retrieve models pretrained natively for video retrieval at the early cutoffs where evidence work happens. Representations purpose-built for video have not yet earned back the flexibility they withdraw. How to encode temporal information in a learned representation therefore remains, in our view, among the most valuable open problems in the field. Joint-embedding predictive architectures learn video representations by predicting in latent rather than pixel space and now transfer to understanding, prediction, and planning tasks (Assran et al., 2025); optical context compression shows a single vision token can carry roughly an order of magnitude more information than a text token (Wei et al., 2025), a hint of the headroom visual tokenization retains. What would change our position is a standard tokenization of visual information across temporal boundaries: a three-dimensional analogue of the text tokenizer that let language retrieval move into training time. Research in that direction may find useful precedent in time-series foundation models, which confronted the same question for continuous signals and answered it with patch-based tokenizations learned at scale (Das et al., 2024; Ansari et al., 2024); visual streams add two spatial dimensions to that problem, and video transformers' spatio-temporal tokens are early, mutually incompatible answers (Arnab et al., 2021). If a standard emerges, part of what is orchestration today will migrate into training time. The VDB format is designed to survive that migration rather than be displaced by it: a native temporal representation enters as a new analyzer family, one more layer over the same sources, while memory, provenance, indexes, and evidence remain the durable substrate.

Governance Persistent collection-wide memory raises access-control, privacy, retention, and deletion requirements that span artifacts, indexes, cached context, answers, and evidence streams. Derived representations are not innocuous: dense embeddings can expose substantial information about source content (Morris et al., 2023). Deleting a source must therefore have a defined meaning for every derivative, while removing the source alone does not guarantee removal of information incorporated into a learned model (Guo et al., 2020). Video analytics systems also show that privacy constraints can be enforced at the query layer rather than delegated entirely to individual analyzers (Cangialosi et al., 2022). This report treats governance as a requirement of the category, not a solved problem of the implementation.

11 Conclusion

Search over the visual world is not document retrieval with larger files: its corpus grows while being queried, its unit of meaning is the interval, its understanding is a portfolio of models, and its answers must remain playable and attributable. We have given the infrastructure these conditions demand a precise shape: analyzers define scenes; scenes carry persistent artifacts; artifacts with provenance constitute visual memory; capability-declared indexes make memory searchable; search selects bounded context; and source-linked context becomes evidence that streams. The VDB format carries this shape in production, and it changes what a search result *is*: any span of any source, from any timestamp to any timestamp

across a collection, archived or still live, can be named, retrieved, composed, and watched within seconds, without a media file ever being touched. When fixed files and fixed segments leave the read path, video stops behaving like storage and starts behaving like data.

The empirical study locates today's leverage. Across 9,800+ queries, a pipeline of general-purpose components composed by infrastructure out-retrieved a commercial video-native foundation model at the strict cutoffs where evidence work happens. The lesson is architectural rather than adversarial: specialization is something a system can assemble from replaceable parts, not something that must live inside one indivisible model, and this infrastructure absorbs video-native models as analyzers the day the balance shifts. One challenge stays open: memory that improves itself through use without weakening provenance, control, or the right to be forgotten. The question this report closes is what a search over the visual world returns: evidence one can press play on.

References

Neil Agarwal and Ravi Netravali. Boggart: Towards general-purpose acceleration of retrospective video analytics. In *20th USENIX Symposium on Networked Systems Design and Implementation*, pages 933–951. USENIX Association, 2023. URL https://www.usenix.org/conference/nsdi23/presentation/agarwal-neil.

Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Türkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Bernie Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=gerNCVqqtR.

Michael Armbrust, Tathagata Das, Liwen Sun, Burak Yavuz, Shixiong Zhu, Mukul Murthy, Joseph Torres, Herman van Hovell, Adrian Ionescu, Alicja Łuszczak, Michał Świtakowski, Michał Szafrański, Xiao Li, Takuya Ueshin, Mostafa Mokhtar, Peter Boncz, Ali Ghodsi, Sameer Paranjpye, Pieter Senster, Reynold Xin, and Matei Zaharia. Delta Lake: High-performance ACID table storage over cloud object stores. *Proceedings of the VLDB Endowment*, 13(12):3411–3424, 2020.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6836–6846, 2021.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *Proceedings of the Twelfth International Conference on Learning Representations*, 2024.

Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning, 2025. arXiv preprint arXiv:2506.09985.

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in Time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1728–1738, 2021.

Beijing Academy of Artificial Intelligence. BAAI/bge-reranker-v2-gemma model card. https://huggingface.co/BAAI/bge-reranker-v2-gemma, 2024. Official model card. Accessed July 2026.

Peter Buneman, Sanjeev Khanna, and Wang-Chiew Tan. Why and Where: A characterization of data provenance. In *Database Theory: ICDT 2001*, volume 1973 of *Lecture Notes in Computer Science*, pages 316–330. Springer, 2001.

Frank Cangialosi, Neil Agarwal, Venkat Arun, Junchen Jiang, Srinivas Narayana, Anand Sarwate, and Ravi Netravali. Privid: Practical, privacy-preserving video analytics queries. In *19th USENIX Symposium on Networked Systems Design and Implementation*, pages 209–228. USENIX Association, 2022. URL https://www.usenix.org/conference/nsdi22/presentation/cangialosi.

David L. Chen and William B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 190–200, 2011.

Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Edgar F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 10148–10167. PMLR, 2024. URL https://proceedings.mlr.press/v235/das24c.html.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the Ninth International Conference on Learning Representations*, 2021.

Daniel Y. Fu, Will Crichton, James Hong, Xinwei Yao, Haotian Zhang, Anh Truong, Avanika Narayan, Maneesh Agrawala, Christopher Ré, and Kayvon Fatahalian. Rekall: Specifying video events using compositions of spatiotemporal labels. *arXiv preprint arXiv:1910.02993*, 2019.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, 2023.

Google. Gemini 3.1 Pro Preview. https://ai.google.dev/gemini-api/docs/models/gemini-3.1-pro-preview, 2026a. First-party model documentation. Accessed July 2026.

Google. Gemini 3.6 Flash. https://ai.google.dev/gemini-api/docs/models/gemini-3.6-flash, 2026b. First-party model documentation. Accessed July 2026.

Todd J. Green, Grigoris Karvounarakis, and Val Tannen. Provenance semirings. In *Proceedings of the Twenty-Sixth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 31–40. Association for Computing Machinery, 2007. doi: 10.1145/1265530.1265535. URL https://doi.org/10.1145/1265530.1265535.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3832–3842. PMLR, 2020. URL https://proceedings.mlr.press/v119/guo20c.html.

Ashish Gupta and Inderpal Singh Mumick. Maintenance of materialized views: Problems, techniques, and applications. *IEEE Data Engineering Bulletin*, 18(2):3–18, 1995.

Brandon Haynes, Maureen Daum, Dong He, Amrita Mazumdar, Magdalena Balazinska, Alvin Cheung, and Luis Ceze. VSS: A storage system for video analytics. In *Proceedings of the 2021 International Conference on Management of Data*, pages 685–696. Association for Computing Machinery, 2021. doi: 10.1145/3448016.3459242. URL https://doi.org/10.1145/3448016.3459242.

Soyeong Jeong, Kangsan Kim, Jinheon Baek, and Sung Ju Hwang. VideoRAG: Retrieval-augmented generation over video corpus. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21278–21298, 2025.

Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 154–161, 2005.

Hyungrok Jung, Daneul Kim, Seunggyun Lim, Jeany Son, and Jonghyun Choi. Online generic event boundary detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13741–13750, 2025. doi: 10.1109/ICCV51701.2025.01275. URL https://openaccess.thecvf.com/content/ICCV2025/html/Jung_Online_Generic_Event_Boundary_Detection_ICCV_2025_paper.html.

Daniel Kang, John Guibas, Peter Bailis, Tatsunori Hashimoto, and Matei Zaharia. TASTI: Semantic indexes for machine learning-based queries over unstructured data. In *Proceedings of the 2022 International Conference on Management of Data*, pages 1934–1947, 2022a.

Daniel Kang, Francisco Romero, Peter Bailis, Christos Kozyrakis, and Matei Zaharia. VIVA: An end-to-end system for interactive video analytics. In *12th Conference on Innovative Data Systems Research (CIDR)*, 2022b.

Junho Kim, Hyunjun Kim, Hosu Lee, and Yong Man Ro. SALOVA: Segment-augmented long video assistant for targeted retrieval and routing in long-form video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3352–3362, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Kim_SALOVA_Segment-Augmented_Long_Video_Assistant_for_Targeted_Retrieval_and_Routing_CVPR_2025_paper.html.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474, 2020.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pages 661–670, 2010.

Xianming Li, Aamir Shakir, Rui Huang, Tsz-fung Andrew Lee, Julius Lipp, Benjamin Clavié, and Jing Li. ProRank: Prompt warmup via reinforcement learning for small language models reranking. arXiv preprint arXiv:2506.03487, 2025. URL https://arxiv.org/abs/2506.03487.

Zhijia Liang, Jiaming Li, Weikai Chen, Yanhao Zhang, Haonan Lu, and Guanbin Li. OASIS: On-demand hierarchical event memory for streaming video reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2821–2831, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Liang_OASIS_On-Demand_Hierarchical_Event_Memory_for_Streaming_Video_Reasoning_CVPR_2026_paper.html.

Kevin Qinghong Lin, Pengchuan Zhang, Joya Chen, Shraman Pramanick, Difei Gao, Alex Jinpeng Wang, Rui Yan, and Mike Zheng Shou. UniVTG: Towards unified video-language temporal grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2794–2804, 2023. doi: 10.1109/ICCV51070.2023.00262. URL https://openaccess.thecvf.com/content/ICCV2023/html/Lin_UniVTG_Towards_Unified_Video-Language_Temporal_Grounding_ICCV_2023_paper.html.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the Middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024.

John Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander Rush. Text embeddings reveal (almost) as much as text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12448–12460. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.765. URL https://aclanthology.org/2023.emnlp-main.765/.

Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. X-CLIP: End-to-end multi-grained contrastive learning for video-text retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia*, 2022.

OpenAI. GPT-5.4 mini model. https://developers.openai.com/api/docs/models/gpt-5.4-mini, 2026a. First-party model documentation. Accessed July 2026.

OpenAI. GPT-5.5 model. https://developers.openai.com/api/docs/models/gpt-5.5, 2026b. First-party model documentation. Accessed July 2026.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems, 2023. arXiv preprint arXiv:2310.08560.

Roger Pantos and William May. HTTP Live Streaming. Technical Report RFC 8216, RFC Editor, 2017.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative Agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023.

Alex Poms, Will Crichton, Pat Hanrahan, and Kayvon Fatahalian. Scanner: Efficient video analysis at scale. *ACM Transactions on Graphics*, 37(4), 2018.

Qwen Team. Qwen3.5-27B model card. https://huggingface.co/Qwen/Qwen3.5-27B, 2026a. First-party model card. Accessed July 2026.

Qwen Team. Qwen3.5-9B model card. https://huggingface.co/Qwen/Qwen3.5-9B, 2026b. First-party model card. Accessed July 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763, 2021.

Matthew Russo, Chunwei Liu, Sivaprasad Sudhir, Gerardo Vitagliano, Michael Cafarella, Tim Kraska, and Samuel Madden. Abacus: A cost-based optimizer for semantic operator systems. *Proceedings of the VLDB Endowment*, 19(5):1060–1073, 2026. doi: 10.14778/3796195.3796215. URL https://doi.org/10.14778/3796195.3796215.

D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems*, volume 28, pages 2503–2511, 2015.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, Zhuang Liu, Hu Xu, Hyunwoo J. Kim, Bilge Soran, Raghuraman Krishnamoorthi, Mohamed Elhoseiny, and Vikas Chandra. LongVU: Spatiotemporal adaptive compression for long video-language understanding. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 54582–54599. PMLR, 2025. URL https://proceedings.mlr.press/v267/shen25j.html.

Mattia Soldan, Alejandro Pardo, Juan León Alcázar, Fabian Caba, Chen Zhao, Silvio Giancola, and Bernard Ghanem. MAD: A scalable dataset for language grounding in videos from movie audio descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5026–5035, 2022. doi: 10.1109/CVPR52688.2022.00497. URL https://openaccess.thecvf.com/content/CVPR2022/html/Soldan_MAD_A_Scalable_Dataset_for_Language_Grounding_in_Videos_From_CVPR_2022_paper.html.

Michael Stonebraker and Uğur Çetintemel. "One Size Fits All": An idea whose time has come and gone. In *Proceedings of the 21st International Conference on Data Engineering*, pages 2–11, 2005.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2 edition, 2018.

Rikiya Takehi, Benjamin Clavié, Sean Lee, and Aamir Shakir. Fantastic (small) retrievers and how to train them: mxbai-edge-colbert-v0 technical report, 2025. URL https://arxiv.org/abs/2510.14880.

Twelve Labs. Real-Time Video Understanding with VideoDB & Twelve Labs. https://www.twelvelabs.io/blog/twelve-labs-and-videodb, 2025. Partner integration announcement. Published August 13, 2025; accessed July 2026.

Twelve Labs. Twelve Labs Developer Documentation: Marengo and Index Creation. https://docs.twelvelabs.io/docs/concepts/models/marengo, 2026a. First-party product documentation. Accessed July 2026.

Twelve Labs. Search. https://docs.twelvelabs.io/docs/guides/search, 2026b. First-party product documentation. Accessed July 2026.

Twelve Labs Marengo Team. Marengo 3.0: More powerful, flexible and efficient video understanding. Twelve Labs technical blog, 2025. URL https://www.twelvelabs.io/blog/marengo-3-0. Company-authored technical material; not peer reviewed. Accessed July 2026.

VideoDB. Understanding Artifacts. https://docs.videodb.io/pages/understand/indexing-pipelines/understanding-artifacts, 2026a. First-party technical documentation. Accessed July 2026.

VideoDB. Deep Search: Stateful, multi-turn video retrieval with a production-grade indexing pipeline. https://github.com/video-db/deepsearch, 2026b.

VideoDB. VideoDB Python SDK and Technical Documentation: Search, Retrieval, and Programmable Media. https://github.com/video-db/videodb-python; https://www.videodb.io/programmable-media.md, 2026c. Software and first-party technical documentation. Accessed July 2026.

VideoDB. MP4 is the wrong primitive for AI. https://www.videodb.io/blogs/mp4-is-wrong-primitive, 2026d. Company-authored technical material; not peer reviewed. Accessed July 2026.

VideoDB. Timeline architecture: Programmable, code-first video editing. https://docs.videodb.io/pages/act/programmable-editing/timeline-architecture, 2026e. First-party product documentation. Accessed July 2026.

Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. VATEX: A large-scale, high-quality multilingual dataset for video-and-language research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4581–4591, 2019.

Haoran Wei, Yaofeng Sun, and Yukun Li. DeepSeek-OCR: Contexts optical compression, 2025. arXiv preprint arXiv:2510.18234.

Dominik Winecki and Arnab Nandi. V2V: Efficiently synthesizing video results for video queries. In *2024 IEEE 40th International Conference on Data Engineering*, pages 5614–5621, 2024.

Junlin Xie, Quanlong Zheng, Ruifei Zhang, Kuo Wang, Yanhao Zhang, Jinguo Luo, Haonan Lu, Xiang Wan, and Guanbin Li. StreamRAG: Enhancing real-time video understanding with retrieval augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 38870–38879, 2026. URL

https://openaccess.thecvf.com/content/CVPR2026/html/Xie_StreamRAG_Enhancing_Real-Time_Video_Understanding_with_Retrieval_Augmentation_CVPR_2026_paper.html.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5288–5296, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/Xu_MSR-VTT_A_Large_CVPR_2016_paper.html.

Zhuangdi Xu, Gaurav Tarlok Kakkar, Joy Arulraj, and Kishore Ramachandran. EVA: A symbolic approach to accelerating exploratory video analytics with materialized views. In *Proceedings of the 2022 International Conference on Management of Data*, pages 602–616. Association for Computing Machinery, 2022. doi: 10.1145/3514221.3526142. URL https://doi.org/10.1145/3514221.3526142.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *Proceedings of the Eleventh International Conference on Learning Representations*, 2023.

Woongyeong Yeo, Kangsan Kim, Jaehong Yoon, and Sung Ju Hwang. WorldMM: Dynamic multimodal memory agent for long video reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25599–25609, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Yeo_WorldMM_Dynamic_Multimodal_Memory_Agent_for_Long_Video_Reasoning_CVPR_2026_paper.html.

Enhao Zhang, Maureen Daum, Dong He, Brandon Haynes, Ranjay Krishna, and Magdalena Balazinska. EQUI-VOCAL: Synthesizing queries for compositional video events from limited user interactions. *Proceedings of the VLDB Endowment*, 16(11):2714–2727, 2023a. doi: 10.14778/3611479.3611482. URL https://doi.org/10.14778/3611479.3611482.

Enhao Zhang, Nicole Sullivan, Brandon Haynes, Ranjay Krishna, and Magdalena Balazinska. Self-enhancing video data management system for compositional events with large language models. *Proceedings of the ACM on Management of Data*, 3(3):1–29, 2025. doi: 10.1145/3725352. URL https://doi.org/10.1145/3725352.

Haoyu Zhang, Ganesh Ananthanarayanan, Peter Bodik, Matthai Philipose, Paramvir Bahl, and Michael J. Freedman. Live video analytics at scale with approximation and Delay-Tolerance. In *14th USENIX Symposium on Networked Systems Design and Implementation*, pages 377–392. USENIX Association, 2017. URL https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/zhang.

Qianxi Zhang, Shuotao Xu, Qi Chen, Guoxin Sui, Jiadong Xie, Zhizhen Cai, Yaoqi Chen, Yinxuan He, Yuqing Yang, Fan Yang, Mao Yang, and Lidong Zhou. VBASE: Unifying online vector similarity search and relational queries via relaxed monotonicity. In *17th USENIX Symposium on Operating Systems Design and Implementation*, pages 377–395. USENIX Association, 2023b. URL https://www.usenix.org/conference/osdi23/presentation/zhang-qianxi.

Luowei Zhou, Chenliang Xu, and Jason J. Corso. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Appendix

A. Exact Dataset-Specific Prompts

The following listings reproduce the complete VideoDB understanding prompts used in the benchmark, including their required output schemas. Every output field shown here was stored as text and included in the corresponding semantic index.

A.1 YouCook2

```
You are an expert multimedia-content analyst. For each cooking moment, integrate ordered visual frames,
     available
narration, on-screen text, and detected ingredients/tools. Output one unified JSON object using the schema
     below.

================================================================================
GLOBAL RULES
================================================================================
1. Multi-Modal Fusion: combine visible preparation with narration and text; state conflicts without inventing
      facts.
2. Evidence Discipline: report only supported ingredients, quantities, tools, techniques, and state changes.
3. Retrieval Style: queries are imperative recipe steps. Prioritize the exact cooking verb, ingredient,
   tool/container, order of operations, heat treatment, texture/shape change, and resulting food state.
4. Strict JSON Only: return exactly the JSON object. Each description must contain at most two concise
     sentences.

================================================================================
FIELD-BY-FIELD EXPECTATIONS
================================================================================
A. scene_description: unified visual description of the cooking action, ingredients, tools, and result.
B. action: exact preparation action using specific cooking verbs and operand ingredients.
C. recipe_step: Concise imperative instruction matching how a query is written.
D. ingredients: all visible or explicitly narrated ingredients used in this moment, including supported
     quantities.
E. tools: utensils, cookware, appliances, and containers directly used.
F. food_state: before-to-after condition, including cut shape, mixture consistency, doneness, temperature, or
      plating.
G. on_screen_text: exact legible ingredient names, quantities, temperatures, timers, or step labels.

================================================================================
OUTPUT JSON SCHEMA (RETURN EXACTLY THIS STRUCTURE)
================================================================================
{
  "scene_description": "A cook whisks eggs, flour, black pepper, salt, and cayenne powder together in a metal
      bowl until combined.",
  "action": "The cook rapidly whisks the eggs and dry seasonings in the bowl to form an even mixture.",
  "recipe_step": "Whisk the eggs, flour, black pepper, salt, and cayenne powder until combined.",
  "ingredients": "eggs, flour, black pepper, salt, cayenne powder",
  "tools": "metal mixing bowl, wire whisk",
  "food_state": "Separate eggs and dry ingredients become a smooth seasoned mixture.",
  "on_screen_text": "none observed"
}

Every text field is required. Use "none observed" when evidence is absent; never return null or an empty
     string.
```

A.2 MSVD

```
You are an expert multimedia-content analyst. For each clip, integrate ordered visual frames, available
     transcript,
on-screen text, and detected objects. Output one unified JSON object using the schema below.

================================================================================
GLOBAL RULES
================================================================================
1. Multi-Modal Fusion: use all available evidence together and report conflicts briefly without inventing
     facts.
2. Evidence Discipline: describe only observable subjects, actions, objects, and locations.
3. Retrieval Style: queries are very short subject-verb-object captions. Prioritize concrete everyday actions
     ,
```

```
   animals, cooking, instruments/music, sports, movement, and object manipulation using simple natural
     wording.
4. Strict JSON Only: return exactly the JSON object. Each description must contain at most two concise
     sentences.

================================================================================
FIELD-BY-FIELD EXPECTATIONS
================================================================================
A. scene_description: unified visual account of subject, action, object, location, activity, and content form.

B. caption: One natural subject-verb-object caption, ideally 3-10 words.
C. action: dominant action using precise present-tense verbs and relevant object interaction.
D. location: concise recognizable environment supported by the frames.
E. character_description: key people or animals with count, role, behavior, interaction, appearance, and
     colors.
F. object_description: salient manipulated or query-distinctive objects.
G. on_screen_text: exact legible text; never guess obscured text.

================================================================================
OUTPUT JSON SCHEMA (RETURN EXACTLY THIS STRUCTURE)
================================================================================
{
  "scene_description": "A brown squirrel sits on a stone ledge outdoors and eats a peanut held between its
     paws.",
  "caption": "a squirrel is eating a peanut",
  "action": "The squirrel holds the peanut with both front paws and bites through its shell.",
  "location": "Exterior park or garden with a stone ledge and greenery.",
  "character_description": "One small brown squirrel sits upright and focuses on the peanut.",
  "object_description": "A peanut in its shell is held close to the squirrel's mouth.",
  "on_screen_text": "none observed"
}

Every text field is required. Use "none observed" when evidence is absent; never return null or an empty
     string.
```

A.3 VATEX

```
You are an expert multimedia-content analyst. For each clip, integrate ordered visual frames, available
     transcript,
on-screen text, and detected objects. Output one unified JSON object using the schema below.

================================================================================
GLOBAL RULES
================================================================================
1. Multi-Modal Fusion: combine all available evidence and state conflicts briefly without inventing facts.
2. Evidence Discipline: include only supported subjects, actions, attributes, topics, and settings.
3. Retrieval Style: captions are comparatively rich and diverse. Capture multi-step actions, interactions,
   how-to activity, clothing/colors, tools, objects, environments, performance, sport, and visible outcomes.
4. Strict JSON Only: return exactly the JSON object. Each description must contain at most two concise
     sentences.

================================================================================
FIELD-BY-FIELD EXPECTATIONS
================================================================================
A. scene_description: rich but concise visual narrative joining subjects, actions, objects, setting, activity
     /topic,
   and content form.
B. caption: One natural caption, ideally 10-20 words, preserving multiple related actions.
C. action: principal and secondary actions in order, with specific verbs, interactions, and outcome.
D. location: supported environment, spatial layout, lighting, and setting cues.
E. character_description: key people/animals with count, role, behavior, interaction, appearance, clothing,
     and colors.
F. object_description: salient tools, materials, vehicles, instruments, food, or equipment and their use.
G. on_screen_text: exact legible titles, captions, signs, labels, or interface text.
H. speech_summary: concise spoken topic or instruction; use "none observed" if unavailable.

================================================================================
OUTPUT JSON SCHEMA (RETURN EXACTLY THIS STRUCTURE)
================================================================================
{
  "scene_description": "A young man stands at a table and displays shoe polish, water, a soft cloth, and a
     brush before beginning a shoe-care demonstration.",
  "caption": "a young man shows the supplies needed to polish a shoe",
  "action": "The man lifts each cleaning item toward the camera and arranges the supplies beside the shoe.",
```

```
  "location": "Interior instructional setting with a work table and neutral background.",
  "character_description": "One young man faces the camera and presents each item with deliberate hand
     gestures.",
  "object_description": "A container of shoe polish, water, an old soft cloth, a brush, and a shoe are
     arranged on the table.",
  "on_screen_text": "none observed",
  "speech_summary": "The presenter identifies the supplies required to polish a shoe."
}

Every text field is required. Use "none observed" when evidence is absent; never return null or an empty
     string.
```

A.4 MSR-VTT

```
You are an expert multimedia-content analyst.
For each video clip, integrate evidence from three sources simultaneously:

- Visual frames in logical order
- Full available transcript, including dialogue, sound effects, and spoken text
- On-screen text and detected objects when available

Your task is to output one unified JSON object using the schema at the end.

================================================================================
GLOBAL RULES
================================================================================
1. Multi-Modal Fusion
   Analyze every field using visual, transcript, on-screen-text, and object evidence together. If cues
     conflict,
   state the conflict briefly; do not invent a compromise.

2. Evidence Discipline
   Use only visible or audible evidence. Do not invent identities, titles, intent, locations, or events
     outside
   the clip. Include a detail only when it improves retrieval.

3. Retrieval Style
   Queries are usually short declarative captions. Prioritize subject + action + object + location and use
   natural query wording. Give special attention to speech/reviews/news, vehicles, music/dance, sports,
     gameplay,
   animation, cooking/how-to, animals, and everyday human actions.

4. Strict JSON Only
   Return exactly the JSON object and no prose or Markdown outside it. Every description must be concise and
     contain
   no more than two sentences.

================================================================================
FIELD-BY-FIELD EXPECTATIONS
================================================================================
A. scene_description: unified factual visual narrative covering the complete scene, thematic activity, and
     content
   form such as gameplay, animation, performance, news, tutorial, or movie/TV when evident.
B. caption
   Write one natural declarative caption, ideally 5-15 words, centered on the dominant subject and action.
C. action: dominant visible action with specific verbs, interaction, spatial relationship, pace, and outcome.
D. location: supported interior, exterior, or fantastical setting and distinctive environmental cues.
E. character_description: key people, groups, animals, or animated/game characters with count, appearance,
     and role.
F. object_description: salient manipulated or query-distinctive objects, vehicles, tools, food, or equipment.
G. on_screen_text
   Perform OCR and return exact legible titles, captions, labels, signs, chart text, or gameplay HUD text in
     reading
   order. Do not guess obscured text.
H. speech_summary
   Summarize the spoken topic and communicative purpose, including review, interview, news, explanation, or
   demonstration; if no meaningful speech is available, return "none observed".

================================================================================
OUTPUT JSON SCHEMA (RETURN EXACTLY THIS STRUCTURE)
================================================================================
{
  "scene_description": "A cook stands behind a kitchen counter beside a bowl of batter and a hot frying pan
     while demonstrating a pancake recipe.",
```

```
  "caption": "a man demonstrates how to cook pancakes",
  "action": "The cook pours batter into the pan, waits for it to set, and flips the pancake with a spatula.",
  "location": "Interior commercial kitchen with stainless-steel appliances and a preparation counter.",
  "character_description": "One adult man in a white chef coat faces the camera and cooks behind the counter
     .",
  "object_description": "A black frying pan, metal spatula, bowl of batter, and portable stovetop are used in
      the demonstration.",
  "on_screen_text": "PANCAKE BASICS",
  "speech_summary": "The cook explains when to pour the batter and how to tell when the pancake is ready to
     flip."
}

Every text field is required. When evidence is absent, return the literal string "none observed"; never
     return null
or an empty string.
```

TwelveLabs The Marengo 3.0 indexing interface used in this benchmark does not support dataset-specific prompts. TwelveLabs therefore has no corresponding prompt listing; the same provider-wide configuration described in §9.2 was used for every dataset.

B. Detailed Reranker Study

The reranker study holds each query's 50 semantic candidates fixed and varies only the reranker and its input mode. Tables 12– 15 report the resulting dataset-level recall. The *No reranking* row preserves the semantic-retrieval order. R@50 is unchanged within each dataset because every configuration reorders the same candidate set. The unweighted averages of these values and the corresponding median component latencies appear in Table 8.

Model	Input mode	R@1	R@3	R@10	R@50
No reranking	–	67.34	68.63	75.69	96.39
BGE Gemma	Match Exact	66.14	77.68	88.09	96.39
BGE Gemma	Concatenate	70.10	80.73	89.17	96.39
mxbai Base v2	Match Exact	66.86	71.50	82.90	96.39
mxbai Base v2	Concatenate	69.33	76.61	86.09	96.39
mxbai Large v2	Match Exact	67.45	70.21	79.31	96.39
mxbai Large v2	Concatenate	68.68	72.28	81.74	96.39
mxbai Edge ColBERT	Match Exact	64.77	76.21	87.13	96.39
mxbai Edge ColBERT	Concatenate	66.89	77.55	86.81	96.39

Table 12. Dataset-level reranker results on MSVD. Values are percentages.

Model	Input mode	R@1	R@3	R@10	R@50
No reranking	–	53.19	55.45	67.69	97.87
BGE Gemma	Match Exact	62.63	79.79	93.09	97.87
BGE Gemma	Concatenate	65.96	80.98	93.48	97.87
mxbai Base v2	Match Exact	56.78	69.55	86.70	97.87
mxbai Base v2	Concatenate	62.50	76.06	90.16	97.87
mxbai Large v2	Match Exact	53.72	57.85	74.47	97.87
mxbai Large v2	Concatenate	54.92	61.04	78.72	97.87
mxbai Edge ColBERT	Match Exact	60.90	76.60	90.29	97.87
mxbai Edge ColBERT	Concatenate	64.36	79.39	91.36	97.87

Table 13. Dataset-level reranker results on YouCook2. Values are percentages.

Model	Input mode	R@1	R@3	R@10	R@50
No reranking	–	76.16	76.91	81.67	97.77
BGE Gemma	Match Exact	78.26	87.81	94.50	97.77
BGE Gemma	Concatenate	83.46	90.28	95.24	97.77
mxbai Base v2	Match Exact	76.67	81.80	90.21	97.77
mxbai Base v2	Concatenate	78.93	85.79	92.67	97.77
mxbai Large v2	Match Exact	76.98	79.57	86.43	97.77
mxbai Large v2	Concatenate	78.19	81.50	88.45	97.77
mxbai Edge ColBERT	Match Exact	71.94	83.86	92.47	97.77
mxbai Edge ColBERT	Concatenate	78.12	87.00	93.42	97.77

Table 14. Dataset-level reranker results on VATEX. Values are percentages.

Model	Input mode	R@1	R@3	R@10	R@50
No reranking	–	63.11	64.08	68.45	92.23
BGE Gemma	Match Exact	63.59	77.67	84.47	92.23
BGE Gemma	Concatenate	72.82	81.55	86.89	92.23
mxbai Base v2	Match Exact	64.08	68.93	78.64	92.23
mxbai Base v2	Concatenate	66.99	77.18	83.01	92.23
mxbai Large v2	Match Exact	64.08	67.48	75.24	92.23
mxbai Large v2	Concatenate	66.99	72.82	80.58	92.23
mxbai Edge ColBERT	Match Exact	62.62	73.30	84.95	92.23
mxbai Edge ColBERT	Concatenate	66.02	77.18	84.95	92.23

Table 15. Dataset-level reranker results on MSR-VTT. Values are percentages.

C. Operational Measurement Details

The latency measurements in this appendix originated from the same client region. Requests from a different region may add approximately 100–200 ms.

C.1 Direct structured and aggregate access

We measured direct *Query* and *Aggregate* over five hours of indexed video. Both contracts operated collection-wide and applied the same eligibility filter. The following examples show representative request forms. The reported statistics contain 200 calls: 100 *Query* calls and 100 *Aggregate* calls.

```
eligibility_filter = [
    {
        "field": "outputs.attribute",
        "op": "exists",
        "value": True,
    }
]

query_results = collection.query(
    index_name="scene_query_aggregate_latency_v1",
    filter=eligibility_filter,
    limit=100,
    return_fields=["outputs.attribute"],
)

aggregate_results = collection.aggregate(
    index_name="scene_query_aggregate_latency_v1",
    filter=eligibility_filter,
    group_by="outputs.attribute",
    metric="count",
    limit=100,
)
```

Contract	Measured calls	Median latency (s)
Query	100	0.443
Aggregate	100	0.380

Table 16. Median latency for direct structured and aggregate access over five hours of indexed video.

C.2 Evidence-stream generation

Table 17 reports median generation latency for playable evidence streams assembled from one source and from five sources.

Duration	Single-source median	Five-source median
5 s	441 ms	469 ms
10 s	433 ms	433 ms
30 s	443 ms	495 ms
1 min	434 ms	447 ms
2 min	521 ms	595 ms
5 min	588 ms	600 ms
10 min	728 ms	994 ms

Table 17. Median generation latency for playable evidence streams assembled from one source and from five sources.

Requested stream duration increased by 120×, from five seconds to ten minutes, while median client-observed generation latency increased by only 1.65× for single-source streams (441 to 728 ms) and 2.12× for five-source streams (469 to 994 ms). Generation latency therefore grew much more slowly than requested stream duration. Across every measured duration and source count, the client-observed median remained below one second, giving a temporal search result a sub-second median path to playable evidence.