# Driving Video Retrieval for Complex Queries with Structured Grounding

**Manyi Yao**[†], **Sparsh Garg**[‡], **Christian Shelton**[†],
**Amit Roy-Chowdhury**[†], **Abhishek Aich**[‡*]
[‡]NEC Laboratories, America, [†]University of California, Riverside

## Abstract

Video retrieval at scale is central to data curation and safety validation in autonomous driving, where users want to find not only scenes but also dynamic events such as cut-ins and hard braking. Existing vision-language and keyword-based retrieval methods often miss these events because the relevant motion may not be explicitly described in text or captured by lexical overlap. Rule-based retrieval can encode such events more directly, but it is brittle: generated or hand-written rules often fail when their assumptions do not match real driving data. We propose STRIVE-D, a data-calibrated retrieval framework for driving videos. It uses weakly labeled in-domain videos to estimate when a query rule is reliable, adapt rules that mismatch observed data, and fuse calibrated rule scores with vision-language and keyword-based retrieval signals. Across three driving benchmarks, including newly released human-annotated event data on DrivingDojo, STRIVE-D delivers up to $84\%$ relative improvement in top-1 accuracy over state-of-the-art method, and remains effective under simulated degraded perception. Annotations are available at `https://github.com/nec-labs-ma/strive-d-data-annotation`.

## 1 Introduction

The deployment of advanced driver-assistance and autonomous systems has put fleet-scale dashcam and sensor logs at the center of safety engineering, with a single instrumented vehicle generating tens of terabytes of footage per day [1–3]. Within these archives, the events that matter most for safety, such as rare interactions, near-misses, and abnormal driving behaviors, are sparsely distributed and difficult to surface manually. Natural-language video retrieval is the principal tool engineers use to query these archives by intent.

The dominant approach uses dense vision–language embeddings [4, 5] to score query–video similarity. These embeddings pool fine-grained kinematic patterns, such as a hard brake or a lane cut-in, into a single representation that mixes them with surrounding scene content; queries that hinge on specific motion patterns are often not reliably answered by embedding similarity alone. We refer to this failure mode as **dilution** (Figure 1).

Structured-grounding approaches [6–9] address dilution by translating queries into executable rules over per-frame perception output and scoring videos against the rule rather than against a single embedding. Rule-based scoring depends on a rule's numerical constraints: thresholds at which gates fire, magnitudes that normalize graded scores, window lengths, and combination weights. A large language model (LLM) at synthesis time has text-level priors over what events "mean," but those priors do not reflect the empirical distribution of signals as measured by the deployed perception pipeline; the numerical constraints an LLM proposes are therefore typically structurally plausible but miscalibrated in scale. We refer to this as **miscalibration**. Prior structured-retrieval methods

*Corresponding author: `aaich001@ucr.edu`.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

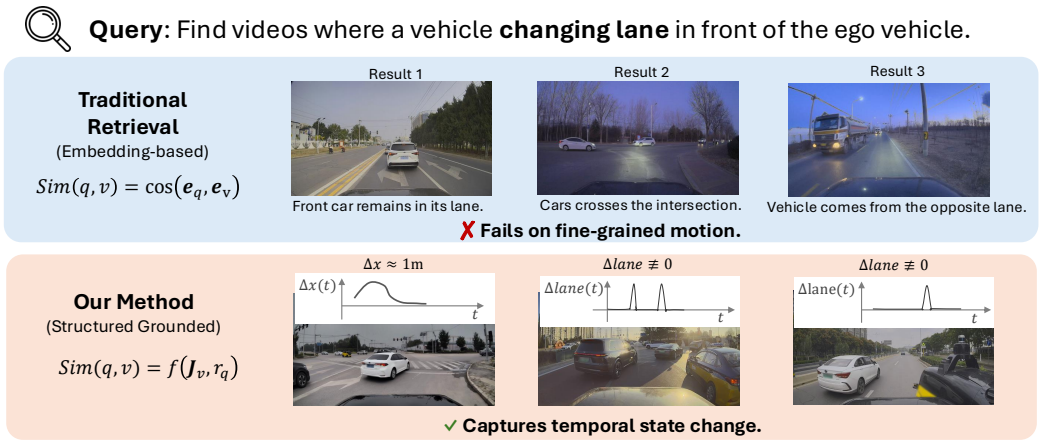



Figure 1: **Advantages of our method.** Embedding-based retrieval captures global scene similarity but often ignores short-term motion dynamics, such as lane changing and hard braking. Our structured grounding method explicitly models these motion patterns, enabling accurate retrieval of such events.

either hand-author rules with hand-set constraints or translate queries via LLM without empirically calibrating them against the operating distribution; closing this calibration gap is our central technical contribution. We reframe constraint calibration as learning-to-rank under distant supervision [10, 11] (using weak supervision from auto-generated captions of unlabeled auxiliary videos), with an LLM proposing constraint assignments inside an iterative optimization loop [12, 13].

We introduce STRIVE-D (**STR**uctured and semant**I**c **V**ideo r**E**trieval, **D**riving), that instantiates structured-grounding retrieval around three components. First, a calibrated rule library handles queries that match library events directly; each library entry is an (event, rule) pair whose numerical constraints are fitted on auxiliary footage via the calibration loop above. Second, because the library covers a finite set of events while user queries may name events outside that inventory (a **coverage** gap), a confidence-aware adaptation policy decides per query whether to reuse a calibrated rule or synthesize a new rule conditioned on the closest calibrated rule as an in-context exemplar. Third, the symbolic score is combined with dense and lexical retrieval through query-conditioned routing, complementing the symbolic path on descriptive content and rare entities.

The primary contributions of this work are summarized as follows:

- **Failure-mode reframe.** We characterize three distinct failure modes of structured-grounding retrieval over driving footage: **dilution** of motion signals in dense embeddings, **miscalibration** of LLM-proposed numerical constraints against the deployed perception pipeline, and **coverage** gaps in any fixed rule library. We design a method whose three components each address one.
- **Numerical-constraint calibration via learning-to-rank under distant supervision.** An LLM-as-optimizer loop jointly fits a rule's thresholds, magnitudes, and window lengths against auxiliary captions; no labels are required in the target benchmark.
- **Confidence-aware adaptation policy.** For queries outside the calibrated library, the policy bridges calibrated reuse and exemplar-conditioned synthesis, preserving the calibrated parameter scale while allowing structural specialization.
- **Empirical receipts and benchmark release.** On three driving benchmarks, STRIVE-D matches or exceeds the strongest dense and structured baselines on every metric (*e.g.*, $83.6\%$ relative Acc@1 gain on DrivingDojo [14], $\sim 1{,}570\times$ lower per-query latency than NSVS-TL [6]); we release a human-annotated event-grounding set on DrivingDojo.

## 2 Related Works

**Vision-Language and Lexical Video Retrieval.** Dense vision–language models [5, 4, 15] retrieve videos by similarity in a shared video–text embedding space, with retrieval-focused variants adding

temporally aware aggregation [16–18]. Recent video LLMs and retrieval-augmented systems further integrate generation into the same representation [19–21]. Sparse lexical retrieval [22, 23] complements these by matching query terms against indexed captions on rare-entity and long-tail queries. Recent benchmarks [24–26] document a persistent shortfall of both paradigms on queries that require fine-grained spatiotemporal compositionality. Our work addresses this shortfall for motion-event retrieval, complementing dense and lexical retrieval with a structured rule-based path (Section 3).

**Neuro-Symbolic Video Understanding.** Neuro-symbolic approaches to video understanding ground language queries in structured representations of the scene, executing rules or programs to verify physical conditions. Temporal-logic methods reason about long-form events by composing per-frame vision-language model (VLM) scores under a temporal logic specification [6, 7]; and scene-graph reasoning composes spatial and temporal predicates over parsed entities [9]. In the driving domain, related work translates dashcam content into structured representations for downstream language reasoning and analytics [27–31]. Across these methods, the numerical constraints that the underlying LLM emits (e.g., thresholds on distance changes or deceleration) are typically used as-is, without an explicit calibration step against the operating distribution. Our work introduces such a calibration step, leaving the symbolic translation to the LLM and grounding the numerical constraints in auxiliary in-domain data.

**Weak Supervision, Ranking, and LLM-as-Optimizer.** Our calibration procedure draws on three established lines of work. Learning to rank under binary relevance [10] provides the standard formulation for optimizing average precision over positive/negative pairs. Distant supervision [11] obtains training pairs from heuristic alignments rather than human labels, a paradigm widely adopted in natural-language processing and information retrieval. Recent work casts large language models as discrete optimizers, using their proposal distribution to search structured solution spaces under black-box objective evaluations [12, 13]. Our method combines these three traditions in the setting of symbolic video retrieval: positive/negative video pairs derived from auxiliary captions provide weakly supervised rankings, and an LLM-driven proposal loop searches over rule parameters under the average-precision objective.

# 3 Method

## 3.1 Problem Formulation

We address natural-language video retrieval. Given a query $q$ and a corpus of $N$ videos $\mathcal{V} = \{v_1, \ldots, v_N\}$, the task is to learn a ranking function $\mathcal{R}$ that returns the top-$K$ most relevant videos. We do *not* assume access to query–video relevance labels on the target corpus. The only labels available during training are weak signals derived from a separate auxiliary set of in-domain footage; their construction is described in Section 3.4.

## 3.2 Three Failure Modes

Motion-event retrieval over driving footage exhibits three failure modes that, in our experiments, no single retrieval mechanism resolves on its own. **(F1) Dilution.** Dense vision–language embeddings pool fine-grained kinematic patterns (e.g., a hard brake, a lane cut-in) into a single representation that mixes them with surrounding scene content; queries that hinge on specific motion patterns are therefore often not reliably answered by embedding similarity alone. **(F2) Miscalibration.** Symbolic rules over per-frame perception output mitigate dilution by probing physical signals directly, but their *numerical thresholds* (the values at which a rule fires) depend on the empirical distribution of those signals, to which an LLM at synthesis time has no access. The thresholds an LLM produces therefore tend to reflect text-level priors rather than the operating distribution, leaving rules miscalibrated in scale. **(F3) Coverage.** Even a fully calibrated library of rules is necessarily incomplete: query-time intents are not restricted to the library's event inventory, and any fixed library will miss matches.

STRIVE-D answers these failure modes with three components, illustrated in Figure 2. A *symbolic path* (Section 3.3) translates queries into executable symbolic rules over per-frame structured representation and ranks videos by satisfaction, addressing **F1**. A *calibrated library* (Section 3.4), constructed once on auxiliary in-domain footage, fits the numerical constraints of each rule via learning-to-rank (L2R) under distant supervision, addressing **F2**. A *confidence-aware adaptation policy* operates

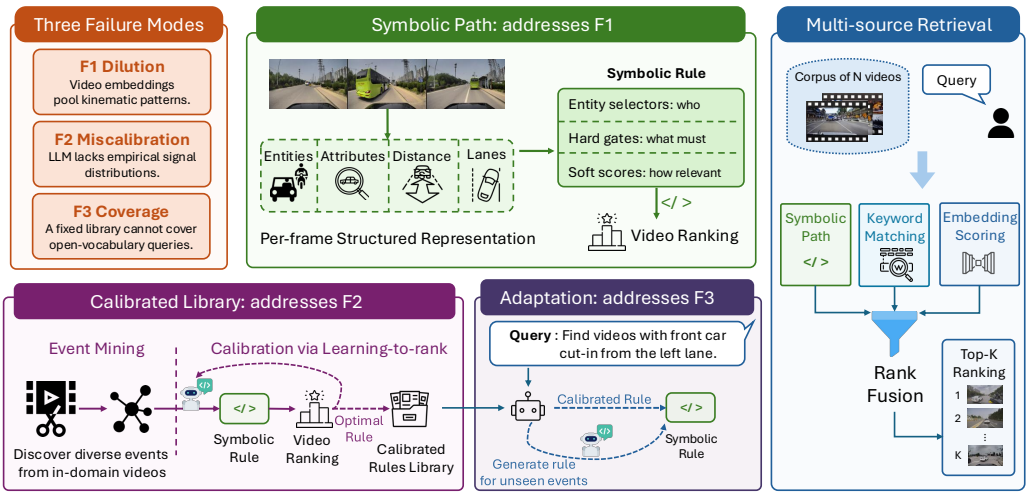



Figure 2: **Overview of STRIVE-D.** We identify three failure modes of existing query-to-video retrieval in driving (*top-left*): F1 *Dilution* of kinematic patterns by dense embeddings, F2 *Miscalibration* of LLM-generated rules against empirical signal distributions, and F3 limited *Coverage* of any fixed rule library for queries beyond the library's event inventory. Three components address them in turn: the *Symbolic Path* (*top-middle*, F1) scores videos by matching a per-frame structured representation (entities, attributes, distance, lane) against a symbolic rule composed of entity selectors, hard gates, and soft scores; the *Calibrated Library* (*bottom-left*, F2) mines diverse events from in-domain videos and calibrates rule thresholds via learning-to-rank, yielding a library of empirically grounded rules; *Adaptation* (*bottom-middle*, F3) handles unseen queries at test time by generating new rules from the calibrated library. *Multi-source Retrieval* (*right*) combines the symbolic path with keyword matching and embedding scoring through rank fusion to produce the final top-$K$ ranking.

within the symbolic path: at retrieval time, it reuses a calibrated rule when the query semantically matches a library event, and otherwise synthesizes a new rule conditioned on the closest calibrated rule as an in-context exemplar, switching between calibrated reuse and exemplar-conditioned synthesis to address **F3**. Finally, *multi-source fusion* (Section 3.5) combines the symbolic path with dense and sparse retrieval signals (defined in Section 3.5), complementing the symbolic rules on descriptive and rare-entity content that they are not designed to express.

### 3.3 Symbolic Path: Rule Design

**Per-frame structured representation.** Each video $v$ in the target corpus is represented as a per-frame structured record $\mathbf{j}_v$ produced by an upstream perception pipeline operating on raw video. Each frame contains a set of tracked objects with associated fields such as object class, distance from the ego vehicle, lane assignment, and lateral offset. We use the structured representation as input to scoring. The specific perception components and field schema used in our experiments are described in Section 4 and Appendix A.

**Rule composition.** A motion-event description naturally decomposes into three questions: *who* is involved, *what* qualitative conditions must hold, and *how strongly* the relevant fields change within a short window. A symbolic rule captures these as three components: **entity selectors**, **hard gates**, and **soft scores**, respectively. A rule's overall score combines them: hard gates determine admissibility (a video scores zero if any single precondition is unsatisfied), soft scores provide the graded ranking magnitude (a calibrated weighted average across multiple constraints), and the entity selector aggregates per-object scores to a video-level score (*any*-mode for events localized to a single agent, *consensus*-mode for group behaviors). Two design properties make the decomposition useful in practice. The *Boolean-plus-continuous* split between hard gates and soft scores mirrors the

satisfaction-and-robustness semantics of signal temporal logic [32, 33], and yields an interpretable composite (a video's relevance is its temporal-pattern strength conditioned on the event's preconditions being met) rather than a single opaque scalar. The split also keeps the per-event constraint set small and flat: a handful of thresholds, magnitudes, window lengths, and weights, so the calibration loop in Section 3.4 can search the constraint space jointly per event. Listing 1 shows a calibrated rule for the event *cut-in*; a per-component ablation isolating the contribution of each piece is in Section 4.3.

Listing 1: A calibrated symbolic rule for the event *cut-in*. The schema and operators are fixed; the numerical values `40.0`, `4.0`, and `14.0` are fit by L2R against an auxiliary set (Section 3.4).

```
{
  "entity":      { "class_filter": ["car","truck"], "agg_mode": "any" },     // who
  "hard_gates":  [ { "field": "distance", "op": "<", "value": 40.0 } ],     // what must
  "soft_scores": [ { "field": "loc_x", "type": "max_change",
                     "window_seconds": 4.0, "magnitude": 14.0 } ]           // how strongly
}
```

### 3.4 Calibrated Library: Construction and Adaptation

**Event mining.** For each video $v'$ in a separate auxiliary set of in-domain footage (disjoint from the target corpus and unlabeled), a video captioner generates a description $\mathcal{C}'$ targeted at kinematic content rather than generic scene description. An LLM then extracts candidate motion events from each $\mathcal{C}'$ as (event type, object) pairs, and consolidates them across videos into a deduplicated set $\mathcal{E} = \{e_1, \ldots, e_M\}$. For each event $e_m$, an auxiliary video is treated as positive if the LLM extracts $e_m$ from its caption, and as negative otherwise; the resulting sets $\mathcal{V}_m^+, \mathcal{V}_m^-$ serve as weak supervision. We rely on captions only to label which auxiliary videos belong to $\mathcal{V}_m^+$ versus $\mathcal{V}_m^-$; candidate rules are scored against the auxiliary videos' per-frame structured representation, not their captions.

**Calibration via learning-to-rank.** For each event $e_m$, an LLM constructs a rule within a shared, manually defined schema, conditioned on the event label and auxiliary-set field statistics. We hold the resulting structure fixed while optimizing its numerical constraints [12, 13]. The loop alternates between three steps: (i) the LLM proposes a candidate constraint assignment for the rule; (ii) the assignment is scored by the average precision (AP) of the ranking it induces over $\mathcal{V}_m^+ \cup \mathcal{V}_m^-$, computed from the auxiliary videos' per-frame structured representation; (iii) the LLM is shown the AP it just achieved together with the best-scoring assignment from prior iterations and asked to propose a revision. The loop terminates when AP exceeds a target threshold or after $\mathcal{T}$ iterations, and the best assignment populates the calibrated library $\mathcal{L}$. AP under distant supervision [10, 11] is the sole selection criterion: captions are a noisy motion-event signal (captioners often emphasize scene description over kinematic detail), but AP averages over many positive and negative candidates per event, mitigating this noise. Direct caption-based retrieval offers no comparable averaging, since each query depends on a single per-video score, so motion events not verbalized in the caption tend to be missed.

**Adaptation at retrieval time.** The calibrated library is, by construction, incomplete: queries may not match any library event. We use a confidence-aware policy that bridges between calibrated reuse and on-the-fly synthesis. For a query $q$, an LLM identifies the closest event $e^* \in \mathcal{E}$ and produces a confidence score $s_{\text{conf}} \in [0, 1]$. When $s_{\text{conf}} \geq \gamma_h$, the calibrated rule is reused. Otherwise, the LLM adapts the closest calibrated rule to the new query, treating it as an in-context exemplar; this preserves the calibrated threshold scale of the related event while allowing structural specialization.

### 3.5 Multi-Source Fusion

The symbolic path is designed for physically grounded motion events. Descriptive content and rare entities are better served by two complementary retrieval signals: *dense retrieval*, which scores videos by similarity in a shared video–text embedding space, and *sparse retrieval*, which matches query terms against per-video captions via lexical overlap. The three sources are complementary rather than substitutes for each other. To merge their per-source rankings into a single ordering, we use a query-conditioned weighted reciprocal rank fusion [34] in which the per-source weights are produced

by an LLM that we call the *fusion router*: given the query and a short description of each source's strengths, the fusion router outputs a weight per source, and these weights scale the rank-based contribution of each source in the fusion.

# 4 Experiments

## 4.1 Settings, Datasets, and Metrics

**Perception pipeline and library construction.** Per-frame structured representations (Section 3.3) are produced by iFinder's perception pipeline [27], with fields including class, distance-to-ego, lane assignment, and lateral offset. All perception modules use their released configurations without dataset-specific tuning. The auxiliary set is the Nexar corpus [35]; from $1{,}147$ Nexar videos disjoint from any evaluation benchmark, we mine and calibrate a rule library of $M = 21$ events. No queries, labels, or videos from evaluation benchmarks enter library construction.

**Models and hyperparameters.** All language model components (rule synthesis, the library matcher used by the confidence-aware adaptation policy, and multi-source fusion weighting) use GPT-4.1 [36] unless otherwise specified, with full prompts in Appendix D. Dense retrieval uses Qwen3-VL-Embedding-8B [5]; sparse retrieval uses BM25 [22] over auxiliary captions. The reranking pool size is fixed at $K' = 20$ for all methods that include a reranker stage; this value is selected from the per-method pool sweep in Appendix B.1, where it is at or near the optimum for every evaluated model. The library matcher confidence threshold is $\gamma_h = 0.6$, the L2R proposal loop runs up to $\mathcal{T} = 20$ iterations per event, and the RRF smoothing constant $k_{\mathrm{RRF}} = 60$ (following [34]). All experiments run on a single NVIDIA RTX A6000 (48 GB).

**Datasets and queries.** We evaluate on three dashcam corpora that exercise complementary query types. *DrivingDojo* [14] contains 583 videos; we release human annotations of 41 distinct motion events tied to specific object classes (*e.g.*, "*truck cut in*," "*ego car change lane*"), supporting fine-grained event-grounded retrieval; 236 videos are relevant to at least one query (Appendix A.6). *CarCrashDataset* [37] provides "reason of accident" annotations across 45 categories (*e.g.*, "*slipping in the snow*"); we evaluate on the 110 videos carrying these annotations, targeting causal/explanatory queries. *MM-AU* [38] contains $1{,}953$ test videos with 49 accident-type annotations (*e.g.*, "*ego car hits a crossing pedestrian*"), targeting accident-description queries. The three datasets together span queries from compositional motion (DrivingDojo), through causal explanation (CarCrash), to accident description (MM-AU).

**Baselines.** We compare against three families: (i) *embedding-based* dual encoders, including Qwen3-VL-Embedding (2B/8B) [5], SigLIP [39], and SigLIP2 [4] (Qwen3-VL is reported paired with its reranker; without-reranker variants in Appendix B); (ii) *retrieval-augmented* VideoRAG [21]; and (iii) *neuro-symbolic* NSVS-TL [6]. All baselines are used in their released form; Section 4.3 substitutes NSVS-TL into our pipeline with identical dense, sparse, router, and reranker components.

**Evaluation metrics.** We report top-$k$ accuracy ($k \in \{1, 3, 5, 10\}$), the fraction of queries with at least one ground-truth video in the top-$k$; mean reciprocal rank (MRR); and mean average precision (mAP). We do not report recall because the number of relevant videos per query varies widely across our annotations, making recall not directly comparable across queries.

## 4.2 Main Results

Table 1 reports retrieval performance on all three benchmarks under a unified reranking pool size $K' = 20$. STRIVE-D is highest on every metric on every dataset; the Acc@1 advantage over the neuro-symbolic baseline NSVS-TL [6] ranges from $+4.8$ percentage points (pp) on DrivingDojo [14] to $+20.4$ pp on MM-AU [38].

Gains concentrate at top-of-list precision. On DrivingDojo [14], STRIVE-D delivers $26.8\%$ Acc@1 versus $14.6\%$ for the strongest dense baseline (an $83.6\%$ relative improvement), and $8.6\%$ mAP versus $4.7\%$ ($+82.9\%$ relative). The pattern holds elsewhere: MM-AU [38] Acc@1 improves $20.4 \rightarrow 26.5$ ($+29.9\%$); on CarCrashDataset [37], Acc@1 ties the best baseline at $28.9\%$ while every other metric dominates. Broad-recall metrics (Acc@10) close more because reranker-equipped baselines partially

Table 1: **Quantitative comparison on three benchmarks (%).** All methods are evaluated under a unified reranking pool size $K' = 20$. STRIVE-D achieves the highest score on every metric on every dataset. **Bold**: best per column. Underline: second-best.

| Type | Method (Year) | MRR(↑) | mAP(↑) | Acc@1 | Acc@3 | Acc@5 | Acc@10 |
|---|---|---|---|---|---|---|---|
| | | | ***DrivingDojo***[14] | | | | |
| *Embedding* | QWen3-vl-2B[5](2026) | 12.2 | 1.4 | 4.9 | 26.8 | 46.3 | 65.9 |
| | QWen3-vl-8B[5](2026) | 20.1 | 2.2 | 14.6 | 31.7 | 51.2 | 68.3 |
| | SigLip[39](2023) | 23.7 | 4.7 | 14.6 | 22.0 | 31.7 | 58.5 |
| | SigLip2[4](2025) | 25.0 | 1.9 | 14.6 | 31.7 | 34.2 | 48.8 |
| *RAG* | VideoRAG[21](2025) | 21.6 | 3.5 | 12.2 | 29.3 | 29.3 | 39.0 |
| *Neuro-sym.* | NSVS-TL[6](2024) | 31.8 | 4.9 | 22.0 | 34.2 | 43.9 | 53.7 |
| | STRIVE-D (Ours) | **38.7** | **8.6** | **26.8** | **48.8** | **53.7** | **73.2** |
| | | | ***MM-AU*** [38] | | | | |
| *Embedding* | QWen3-vl-2B[5](2026) | 26.9 | 6.0 | 18.4 | 30.6 | 42.9 | 49.0 |
| | QWen3-vl-8B[5](2026) | 32.7 | 6.3 | 20.4 | 38.8 | 55.1 | 61.2 |
| | SigLip[39](2023) | 16.1 | 4.3 | 6.1 | 24.5 | 30.6 | 38.8 |
| | SigLip2[4](2025) | 17.8 | 6.1 | 8.2 | 22.5 | 26.5 | 40.8 |
| *RAG* | VideoRAG[21](2025) | 22.9 | 6.0 | 18.4 | 24.5 | 24.5 | 36.7 |
| *Neuro-sym.* | NSVS-TL[6](2024) | 18.0 | 2.8 | 6.1 | 20.4 | 36.7 | 53.1 |
| | STRIVE-D (Ours) | **40.5** | **9.5** | **26.5** | **53.1** | **59.2** | **65.3** |
| | | | ***CarCrashDataset*** [37] | | | | |
| *Embedding* | QWen3-vl-2B[5](2026) | 38.3 | 32.1 | 26.7 | 44.4 | 55.6 | 64.4 |
| | QWen3-vl-8B[5](2026) | 40.1 | 33.5 | **28.9** | 46.7 | 62.2 | 66.7 |
| | SigLip[39](2023) | 30.0 | 22.0 | 17.8 | 35.6 | 51.1 | 60.0 |
| | SigLip2[4](2025) | 28.9 | 23.4 | 15.6 | 33.3 | 51.1 | 62.2 |
| *RAG* | VideoRAG[21](2025) | 32.1 | 25.6 | 20.0 | 40.0 | 46.7 | 55.6 |
| *Neuro-sym.* | NSVS-TL[6](2024) | 23.1 | 19.3 | 11.1 | 24.4 | 37.8 | 53.3 |
| | STRIVE-D (Ours) | **45.0** | **37.7** | **28.9** | **60.0** | **73.3** | **75.6** |

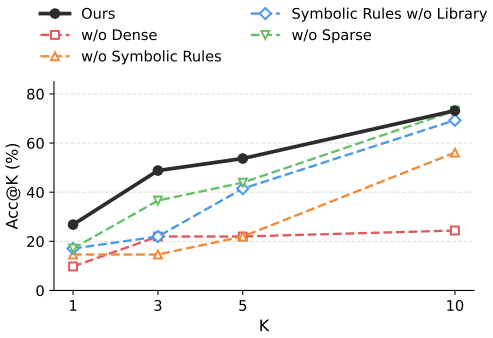



Figure 3: **Component ablation on DrivingDojo.** Removing the dense path causes the most severe degradation; removing the calibrated library, symbolic-rule path, or sparse retrieval each isolates a distinct failure mode.

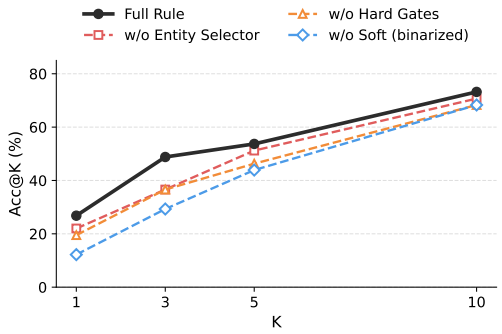



Figure 4: **Rule composition ablation on DrivingDojo.** All three pieces are load-bearing. Soft scores carry the most weight at top-of-list precision; hard gates contribute substantially; entity selectors contribute incrementally.

recover at wider candidate pools, a behavior dense alone cannot deliver at top-of-list precision. Across five full-pipeline runs, the Acc@1 advantage over the strongest dense baseline persists on DrivingDojo and MM-AU ( Appendix B.4).

### 4.3 Ablations and Analysis

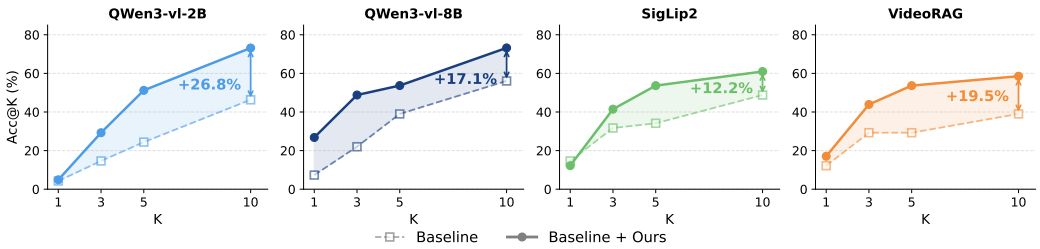



Figure 5: **Plug-in generalizability evaluated on DrivingDojo.** Our method consistently improves the full retrieval curve when integrated into 4 off-the-shelf retrievers spanning different architectures, from lightweight VLM embeddings (QWen3-vl[5]) to video retrieval systems (VideoRAG[21]) and vision encoders (SigLip2[4]). Numbers on the right indicate the absolute gain in top-10 accuracy.

**Component ablation.** Figure 3 removes each component of STRIVE-D in turn. **Removing the dense path** causes the largest drop, Top-1 from $26.8\%$ to $9.76\%$: sparse and symbolic rules alone do not fully recover the descriptive content carried by dense embeddings, and dense alone does not reach the same top-of-list precision (Table 1); the two pathways are complementary. **Removing the calibrated library** forces the LLM to synthesize rules from scratch at query time, using the same schema (entity selectors, hard gates, soft scores) constrained by few-shot exemplars but without L2R threshold calibration; Top-3 drops sharply with partial recovery at Top-10. Because the rule *form* is held constant, this gap isolates threshold calibration from any structural variation. **Removing the symbolic path entirely** also drops Top-3 sharply, indicating that the symbolic path contributes precision that dense+sparse alone do not provide in this setting. **Removing sparse retrieval** leaves Top-10 essentially unchanged but slightly hurts top-rank precision, consistent with sparse's rare-entity recovery role rather than primary recall.

**Rule composition: entity selectors, hard gates, soft scores.** Each symbolic rule (Section 3.3) is composed of three pieces (entity selector, hard gates, soft scores) mapping to *who*, *what conditions*, and *how strongly*. We disable each piece in turn while holding calibrated thresholds, fusion, and the reranker fixed (no L2R re-run). Figure 4 shows all three pieces are load-bearing, with a clear ordering at top-of-list precision. **Graded soft scores carry the most weight**: binarizing them at the calibrated threshold drops Acc@1 from $26.8\%$ to $12.2\%$ ($-14.6$ pp), as the rule loses the ability to distinguish strong from marginal matches. **Hard gates contribute substantially** ($-7.3$ pp Acc@1): the binary admissibility check filters out videos otherwise falsely admitted by soft scores alone. **The entity selector contributes incrementally** ($-4.8$ pp Acc@1), the smallest drop, as dense and sparse fusion partially compensate for off-target object scoring. All variants converge at Acc@10 (within 5 pp), reflecting the dense+sparse recall floor; differentiation lives at top-of-list precision where the calibrated symbolic rule is most discriminative.

**Plug-in generalizability.** We attach our symbolic-rule pathway to four off-the-shelf retrievers spanning architectures: Qwen3-VL-2B and -8B [5], VideoRAG [21], and SigLIP2 [4] (Figure 5). Top-10 accuracy improves on every backbone, with gains ranging from 12.2 pp on SigLIP2 to 26.8 pp on Qwen3-VL-2B. The 17.1 pp lift on Qwen3-VL-8B is particularly informative: it shows that the improvement is not solely a result of compensation for a weak backbone.

**Efficiency.** Table 2 reports mean per-query online latency on DrivingDojo [14]. STRIVE-D averages 3.10 s per query. Pure dense retrievers run in $\sim$130 ms (Qwen3-VL-Embedding-8B [5]) but do not perform structured grounding and underperform on the precision metrics in Table 1. Within the structured-retrieval paradigm, the meaningful comparison is to NSVS-TL [6]: at 4,868.05 s per query, NSVS-TL is $\sim$1,500$\times$ slower than STRIVE-D. Two LLM stages, rule synthesis (invoked only when no library rule matches the query) and multi-source fusion, account for $\sim$87% of the per-query cost, and both are amenable to caching across queries sharing an event category.

**VLM binary classification baseline.** A natural alternative to structured retrieval is to ask a general-purpose VLM directly: *"Does this video show q?"* for each candidate, and rank by the predicted *Yes* probability. Table 3 reports this baseline on DrivingDojo using Qwen3-VL-8B-Instruct [40], with an

Table 2: **Online per-query latency on DrivingDojo (mean).** Embedding and retrieval-augmented methods run in milliseconds; structured-grounding methods run in seconds.

| Type | Method | Latency |
|---|---|---|
| *Embedding* | SigLIP2 [4] | 26.80 ms |
| | Qwen3-VL-Emb-8B [5] | 128.54 ms |
| *RAG* | VideoRAG [21] | 37.18 ms |
| *Neuro-sym.* | STRIVE-D (Ours) | 3.10 s |
| | NSVS-TL [6] | 4,868.05 s |

Table 3: **VLM binary classification baseline on DrivingDojo (%).** Qwen3-VL-8B-Instruct [40] is prompted with *"Does this video show {q}?"* per candidate; the +Reranker variant additionally re-ranks the predicted-*Yes* subset with Qwen3-VL-Reranker-8B [5].

| Method | Acc@1 | Acc@5 | Acc@10 |
|---|---|---|---|
| VLM Binary | 2.4 | 29.3 | 41.5 |
| + Reranker | 7.3 | 34.2 | 48.8 |
| STRIVE-D | **26.8** | **53.7** | **73.2** |

Table 4: **Calibration proposer comparison on DrivingDojo (Acc@10, %)**. Calibrated variants share rule structure, AP objective, and budget (20 proposals per rule). $\pm$: std over 3 runs.

| Proposer | Acc@10 |
|---|---|
| Uncalibrated (no search) | 65.9 |
| Random search | $69.9 \pm 3.7$ |
| Bayesian search | $69.9 \pm 2.8$ |
| Grid search | 70.7 |
| LLM (ours) | **73.2** |

Table 5: **Library size vs. adaptation.** The calibrated library is reduced to random subsets of the 21 events (5 seeds per size). Adaptation rate: fraction of DrivingDojo queries routed to rule synthesis.

| # library events | Adapt. rate (%) | Acc@10 (%) |
|---|---|---|
| 0 (no library) | 100 | $69.3 \pm 0.0$ |
| 5 | 85.4 | $70.2 \pm 2.0$ |
| 10 | 36.6 | $71.7 \pm 1.3$ |
| 21 (full) | 12.2 | $\mathbf{73.2 \pm 0.0}$ |

optional Qwen3-VL-Reranker-8B [5] second stage. The scorer alone trails every baseline in Table 1, and even with the reranker it remains well below the strongest embedding-based baselines (Acc@1 7.3 vs. 14.6). Per-video binary classification is not a competitive retrieval strategy on this benchmark.

**Calibration proposer.** To assess the effect of the calibration proposer, we compare LLM and non-LLM search with fixed rule forms, auxiliary-set AP objective, and budget of 20 proposals per rule (Table 4). The LLM conditions proposals on per-field statistics and best-so-far feedback. All calibrated variants outperform the uncalibrated baseline (65.9% Acc@10). The LLM achieves 73.2%, compared with 70.7% for the best non-LLM search, and also synthesizes query-specific rules when no library rule matches with sufficient confidence (Section 3.4).

**Library size and query-time adaptation.** To assess the benefit of calibrated exemplars, we progressively reduce the 21-event library (Table 5). As the library shrinks from 21 events to zero, the fraction of queries routed to rule synthesis rises from 12.2% to 100%, while Acc@10 decreases from 73.2% to 69.3%. With all rules synthesized at query time, accuracy remains above the strongest dense baseline (Qwen3-VL-8B [5], 68.3%). Calibrated exemplars improve accuracy, while query-time synthesis supports retrieval beyond the retained event inventory.

**Robustness to perception errors.** To assess sensitivity to perception errors, we perturb precomputed records one module at a time, with the calibrated library and other retrieval paths fixed. Independent detection removal and track fragmentation reduce Acc@10 by at most 3.4 pp, remaining above the strongest dense baseline (68.3%; Table 6). Sustained missed detections and 20-frame track interruptions reduce accuracy to 62.6% and 63.4%, respectively. Full protocols are in Appendix B.6.

**Structured grounding choice.** To compare structured scoring methods, we substitute NSVS-TL for symbolic rules within the same retrieval pipeline. With the other components fixed, our rules improve Acc@1 by 14.6 pp and mAP by 4.7 pp (Table 7).

### 4.4 Qualitative Results

Figure 6 illustrates representative retrieval results. For the "*bus cut-in*" query, baseline models retrieve videos containing buses but fail to identify the cut-in behavior, returning scenes where

Table 6: **Perception robustness** (DrivingDojo; five-seed mean ± std). Rates are affected fractions; for false detections, added/original count ratios. Unperturbed Acc@10: 73.2%.

| Perturbation | Acc@10 (%) |
|---|---|
| *Independent (rate: 50%)* | |
| Detection removal | $70.2 \pm 2.0$ |
| Track fragmentation | $69.8 \pm 1.3$ |
| Vehicle depth ($\times\mathcal{N}(1, 0.5^2)$) | $72.2 \pm 3.3$ |
| *Temporally correlated (rate: 75%)* | |
| False detections | $69.1 \pm 3.0$ |
| Consecutive misses (per object) | $62.6 \pm 7.5$ |
| Track interruption (20 frames) | $63.4 \pm 3.1$ |

Table 7: **Structured grounding choice** (DrivingDojo, %). NSVS-TL + pipeline replaces symbolic rules with NSVS-TL, with dense retrieval, sparse retrieval, routing, and reranking fixed.

| Variant | Acc@1 | mAP | Acc@10 |
|---|---|---|---|
| NSVS-TL standalone | 22.0 | 4.9 | 53.7 |
| NSVS-TL + pipeline | 12.2 | 3.9 | 70.7 |
| STRIVE-D | **26.8** | **8.6** | **73.2** |




Figure 6: **Qualitative retrieval comparison.** Each row shows the top retrieved results per model for two queries. For *bus cut-in*, baselines retrieve videos containing buses but miss the cut-in motion. For *car collision with city guardrails*, baselines retrieve either collisions without guardrails or guardrails without collisions. STRIVE-D correctly identifies both the object and the event in each case.

buses are simply present in traffic with no encroachment into the ego lane. For "*car collision with city guardrails*," baselines retrieve either collisions that do not involve guardrails, or scenes with guardrails but no collision event, capturing one half of the compositional query while missing the other. STRIVE-D grounds both the object type and the motion event, returning videos that satisfy the specified conditions in these examples.

## 5 Conclusion

We present STRIVE-D, a calibrated structured-grounding framework for natural-language retrieval over driving video. The framework rests on a single diagnosis: large language models reliably translate a query into the structural form of an executable rule, but are unreliable at fitting that rule's numerical thresholds to real driving data; this miscalibration is what limits LLM-written rules in practice. STRIVE-D addresses it by factoring the two competencies: an LLM proposes rule structure, while a learning-to-rank objective recovers the numerical thresholds under distant supervision from captions on an auxiliary in-domain corpus, organizing the calibrated rules into a reusable library that a confidence-gated policy reuses or adapts at query time. Across three driving benchmarks, STRIVE-D consistently improves over the strongest embedding-based and structured baselines. Several limitations remain: the rule schema lacks explicit temporal sequencing and scoped negation, retrieval depends on upstream detection and tracking, transfer of a calibrated library across perception stacks is untested, and caption errors introduce noise into calibration labels, so thresholds for events with few positives may be unreliable. Despite these limitations, these components together turn LLM-written rules from structurally plausible but numerically miscalibrated proposals into a calibrated, interpretable retrieval primitive grounded in the empirical signal distribution of an unlabeled in-domain corpus.

## Acknowledgments and Disclosure of Funding

This work was supported by NSF grants CMMI-2326309 and CNS-2312395. The authors declare no additional competing interests.

## References

[1] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[2] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[3] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[4] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

[5] Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Qwen3-VL-Embedding and Qwen3-VL-Reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026.

[6] Minkyu Choi, Harsh Goel, Mohammad Omama, Yunhao Yang, Sahil Shah, and Sandeep Chinchali. Towards neuro-symbolic video understanding. In *European Conference on Computer Vision*, pages 220–236. Springer, 2024.

[7] Sahil Shah, SP Sharan, Harsh Goel, Minkyu Choi, Mustafa Munir, Manvik Pasula, Radu Marculescu, and Sandeep Chinchali. NeuS-QA: Grounding long-form video understanding in temporal logic and neuro-symbolic reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

[8] Dídac Surís, Sachit Menon, and Carl Vondrick. ViperGPT: Visual inference via Python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[9] Lili Liang, Guanglu Sun, Jin Qiu, and Lizhong Zhang. Neural-symbolic VideoQA: Learning compositional spatio-temporal reasoning for real-world video question answering. arXiv preprint arXiv:2404.04007, 2024.

[10] Tie-Yan Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

[11] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, 2009.

[12] Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations (ICLR)*, 2024.

[13] Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625:468–475, 2024.

[14] Yuqi Wang, Ke Cheng, Jiawei He, Qitai Wang, Hengchen Dai, Yuntao Chen, Fei Xia, and Zhaoxiang Zhang. DrivingDojo dataset: Advancing interactive and knowledge-enriched driving world model. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2024.

[15] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[16] Xinyi Xu, Hongsong Wang, Guo-Sen Xie, Caifeng Shan, and Fang Zhao. HiTeA: Hierarchical temporal alignment for training-free long-video temporal grounding. In *The Fourteenth International Conference on Learning Representations*, 2026.

[17] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. CLIP4Clip: An empirical study of CLIP for end to end video clip retrieval and captioning. *Neurocomputing*, 508:293–304, 2022.

[18] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-CLIP: End-to-end multi-grained contrastive learning for video-text retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia (ACM MM)*, 2022.

[19] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. VideoLLaMA 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024.

[20] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. VideoChat: Chat-centric video understanding. *Science China Information Sciences*, 2025.

[21] 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 Findings)*, 2025.

[22] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, April 2009.

[23] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. SPLADE: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 2288–2292, 2021.

[24] Dahun Kim, AJ Piergiovanni, Ganesh Mallya, and Anelia Angelova. VideoComp: Advancing fine-grained compositional and temporal alignment in video-text models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 29060–29070, 2025.

[25] Animesh Gupta, Jay Parmar, Ishan Rajendrakumar Dave, and Mubarak Shah. From play to replay: Composed video retrieval for temporally fine-grained videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[26] Yiqing Shen, Chenxiao Fan, Chenjia Li, and Mathias Unberath. Reasoning text-to-video retrieval via digital twin video representations and large language models. arXiv preprint arXiv:2511.12371, 2025.

[27] Manyi Yao, Bingbing Zhuang, Sparsh Garg, Amit Roy-Chowdhury, Christian Shelton, Manmohan Chandraker, and Abhishek Aich. iFinder: Structured zero-shot vision-based LLM grounding for dash-cam video reasoning. In *Proceedings of the 39th Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[28] Johnathan Leung, Guansen Tong, Parasara Sridhar Duggirala, and Praneeth Chakravarthula. From road to code: Neuro-symbolic program synthesis for autonomous driving scene translation and analysis. In George Pappas, Pradeep Ravikumar, and Sanjit A. Seshia, editors, *Proceedings of the International Conference on Neuro-symbolic Systems*, volume 288 of *Proceedings of Machine Learning Research*, pages 331–351. PMLR, 28–30 May 2025.

[29] Yuxuan Yan, Shiqi Jiang, Ting Cao, Yifan Yang, Qianqian Yang, Yuanchao Shu, Yuqing Yang, and Lili Qiu. AVA: Towards agentic video analytics with vision language models. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2026.

[30] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. DriveLM: Driving with graph visual question answering. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

[31] Licheng Wen, Daocheng Fu, Xin Li, Xinyu Cai, Tao Ma, Pinlong Cai, Min Dou, Botian Shi, Liang He, and Yu Qiao. DiLu: A knowledge-driven approach to autonomous driving with large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

[32] Oded Maler and Dejan Nickovic. Monitoring temporal properties of continuous signals. In *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems (FORMATS/FTRTFT)*, volume 3253 of *Lecture Notes in Computer Science*, pages 152–166. Springer, 2004.

[33] Alexandre Donzé and Oded Maler. Robust satisfaction of temporal logic over real-valued signals. In *Formal Modeling and Analysis of Timed Systems (FORMATS)*, volume 6246 of *Lecture Notes in Computer Science*, pages 92–106. Springer, 2010.

[34] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 758–759, New York, NY, USA, 2009. Association for Computing Machinery.

[35] Daniel C. Moura, Shizhan Zhu, and Orly Zvitia. Nexar dashcam collision prediction dataset and challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2025.

[36] OpenAI. Introducing GPT-4.1 in the API. `https://openai.com/index/gpt-4-1/`, April 2025.

[37] Wentao Bao, Qi Yu, and Yu Kong. Uncertainty-based traffic accident anticipation with spatio-temporal relational learning. In *ACM Multimedia Conference*, October 2020.

[38] Jianwu Fang, Lei-lei Li, Junfei Zhou, Junbin Xiao, Hongkai Yu, Chen Lv, Jianru Xue, and Tat-Seng Chua. Abductive ego-view accident video understanding for safe driving perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22030–22040, June 2024.

[39] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[40] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[41] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[42] Alexander Veicht, Paul-Edouard Sarlin, Philipp Lindenberger, and Marc Pollefeys. GeoCalib: Learning single-image calibration with geometric optimization. In *European Conference on Computer Vision (ECCV)*, 2024.

[43] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24185–24198, 2024.

[44] Zachary Teed and Jia Deng. DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:16558–16569, 2021.

[45] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[46] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. ByteTrack: Multi-object tracking by associating every detection box. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[47] Dongkwon Jin and Chang-Su Kim. OMR: Occlusion-aware memory-based refinement for video lane detection. In *European Conference on Computer Vision (ECCV)*, 2024.

[48] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3D v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2024.

[49] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[50] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision (ECCV)*, 2020.

[51] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. GPT-4o system card. arXiv preprint arXiv:2410.21276, 2024.

[52] OpenAI. GPT-4o mini: Advancing cost-efficient intelligence. `https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/`, July 2024.

# A Experimental Setup

## A.1 Perception Pipeline

We adopt the perception pipeline of iFinder [27] with a single substitution: the video captioner used at library-construction time is replaced by Qwen2.5-VL-7B [41]. The pipeline produces per-frame structured representations consumed by the symbolic-rule evaluation in Appendix A.3; each record contains a set of tracked objects with object class, 3D distance from the ego vehicle, lane assignment, lateral offset, and additional context fields. The modules and the pretrained model used for each are:

- **Camera intrinsics and lens-distortion estimation.** GeoCalib [42] predicts focal length, principal point, and radial/tangential distortion coefficients per video.
- **Scene understanding and per-object attributes (image VLM).** InternVL [43] extracts global scene context (weather, road structure, time of day) and per-object attributes such as vehicle color and traffic-light state.
- **Video captioning (video VLM, library-construction only).** Qwen2.5-VL-7B [41] produces the per-video event description used to seed library construction. This module is invoked only when building the rule library and is not used at retrieval time.
- **Ego-vehicle camera-pose estimation.** DROID-SLAM [44] provides per-frame camera translations, from which ego heading-rate and speed are derived to populate the ego turn/-motion state.
- **2D object detection.** OWL-V2 [45] provides open-vocabulary, class-conditioned bounding boxes.
- **Multi-object tracking.** ByteTrack [46] associates detections across frames into per-object tracks.
- **Lane detection.** OMR [47] predicts per-frame lane-marking polylines, which are converted to per-object lane assignments by mapping each object's bottom-midpoint into the lane partition.
- **Metric depth.** Metric3D [48] predicts a metric depth map per frame.
- **Foreground segmentation for distance-to-ego.** SAM [49] produces a binary foreground mask within each detected bounding box; the per-object distance is the mean masked depth.
- **3D detection for object orientation.** CenterTrack [50] provides 3D bounding boxes whose yaw angles are matched to the 2D detections via Hungarian assignment.

## A.2 Rule Library Construction

We build the library from 1,147 deduplicated Nexar training videos [35], disjoint from all evaluation benchmarks. Motion events are extracted from captions generated by Qwen2.5-VL-7B [41] using a predefined inventory of 21 categories; a video may belong to multiple categories. For each category, videos whose captions mention the event form the positive set, and all remaining videos form the negative set. We calibrate one symbolic rule per category to maximize the AP of the ranking induced by $s_{\text{mot}}$ over the auxiliary corpus (Appendix A.4), yielding 21 calibrated rules. Table 8 reports the category counts: hard braking and swerving appear in 1,068 and 1,019 videos, respectively.

## A.3 Rule Scoring: Formal Definitions

Given a symbolic rule $r_q$, the motion relevance score $s_{\text{mot}}(v, r_q)$ is computed in three steps; symbol conventions follow Section 3.

**Hard gates.** With $G$ hard-gate constraints $\{(\phi_g, \texttt{op}_g, \theta_g)\}_{g=1}^{G}$, where $\phi_g$ is gate $g$'s target field (*e.g.*, `distance`, `loc_x`) and each $\theta_g$ is a threshold calibrated against the auxiliary set, the hard-gate score is

$$s_{\text{hard}}(v, r_q) = \prod_{g=1}^{G} \mathbf{1}\big[\, \exists o,\ \exists t :\ \texttt{op}_g(\phi_g(o,t), \theta_g) \,\big], \tag{1}$$

where $\phi_g(o,t)$ is the value of $\phi_g$ for object $o$ at frame $t$. A video is gated to $s_{\text{hard}} = 0$ if any single precondition is unsatisfied.

Table 8: Extracted driving event categories in the Nexar training set (1,147 dashcam videos after de-duplication and canonicalization).

| **Event** | **# Videos** |
|---|---|
| Hard Brake | 1,068 |
| Swerving | 1,019 |
| Pedestrian Crossing | 370 |
| Near Miss | 199 |
| Cut-In | 170 |
| Cut-Out | 166 |
| Traffic Disruption | 151 |
| Lane Change | 96 |
| Object Approach | 78 |
| Collision | 62 |
| Emergency Vehicle | 31 |
| Evasive Maneuver | 20 |
| Sudden Speed Change | 18 |
| Lateral Movement | 10 |
| Sudden Stop | 9 |
| Loss of Control | 8 |
| Rollover | 8 |
| Dynamic Object Crossing | 8 |
| Object Departure | 3 |
| Reverse | 1 |
| Sharp Turn | 1 |
| **Total** | **1,147** |

**Soft scores.** For each soft constraint $c$ with target field $\phi_c$, expected magnitude $\delta_c$, and window length $\Delta T_c$, the per-object score for object $o$ is the maximum within-window change in $\phi_c$, normalized by $\delta_c$:

$$s_c^{(o)} = \min\left(1,\ \frac{\max_{0<t_2-t_1\le\Delta T_c} |\phi_c(o,t_2) - \phi_c(o,t_1)|}{\delta_c}\right). \tag{2}$$

When constraint $c$ specifies a signed `direction` (*e.g.*, `negative` for an approaching object), the absolute value is replaced by the corresponding one-sided difference (*e.g.*, $\max(0,\ \phi_c(o,t_1) - \phi_c(o,t_2))$); Equation (2) shows the unsigned default. The aggregate per-object soft score is the weighted average $s_{\text{obj}}^{(o)} = \big(\sum_c w_c\, s_c^{(o)}\big)/\big(\sum_c w_c\big) \in [0,1]$, with weights $w_c$ calibrated jointly with the other numerical constraints.

**Entity aggregation and final score.** The entity selector aggregates per-object scores to a video-level score:

$$s_{\text{ent}}(v, r_q) = \begin{cases} \max_i s_{\text{obj}}^{(i)} & (\textit{any}\text{-mode}) \\ \frac{1}{2}\Big(\max_i s_{\text{obj}}^{(i)} + \overline{s_{\text{obj}}}\Big) & (\textit{consensus}\text{-mode}) \end{cases}, \tag{3}$$

where $\overline{s_{\text{obj}}}$ is the mean across candidate objects. The final motion relevance score is the product $s_{\text{mot}}(v, r_q) = s_{\text{hard}}(v, r_q) \cdot s_{\text{ent}}(v, r_q)$. Hard gates and soft scores are aggregated independently at the video level, so a video can satisfy hard gates via one object and accrue soft scores via another; this decoupling is intentional, with hard gates acting as a global admissibility filter and soft scores capturing graded evidence anywhere in the video.

### A.4 L2R Calibration: Precise Definitions

**Selection objective.** At iteration $\tau$ of the calibration loop, candidate rule $r_{e_m}^{(\tau)}$ is scored by the average precision (AP) of the ranking $\pi$ that $s_{\text{mot}}$ induces on $\mathcal{V}_m^+ \cup \mathcal{V}_m^-$:

$$\text{AP}(r_{e_m}^{(\tau)}, \mathcal{V}_m^+, \mathcal{V}_m^-) = \frac{1}{|\mathcal{V}_m^+|} \sum_{k:\, v'_{\pi_k} \in \mathcal{V}_m^+} \frac{|\{j \le k : v'_{\pi_j} \in \mathcal{V}_m^+\}|}{k}, \tag{4}$$

where $j$ is the inner running index over $\{1, \ldots, k\}$. AP is the only quantity used to compare proposals; the per-field summary statistics provided to the LLM at proposal time are conditioning inputs only and do not appear in this objective.

**Best-so-far feedback.** We denote the best-scoring rule among the first $\tau$ proposals by

$$r_{e_m}^{(\tau)*} := \arg\max_{\tau' \le \tau} \mathrm{AP}(r_{e_m}^{(\tau')}, \mathcal{V}_m^+, \mathcal{V}_m^-). \tag{5}$$

When $\mathrm{AP}(r_{e_m}^{(\tau)}, \mathcal{V}_m^+, \mathcal{V}_m^-) < \gamma_{\text{AP}}$, the LLM is fed the previous AP together with $r_{e_m}^{(\tau-1)*}$ as an anchor and asked to propose a revision; this repeats up to $\mathcal{T} = 20$ iterations.

**Calibrated library.** After calibration, the best assignment per event is recorded as

$$\mathcal{L} = \{(e_m, r_{e_m}^{(\mathcal{T})*})\}_{m=1}^{M}, \tag{6}$$

and used at retrieval time according to the confidence-aware policy in Section 3.4.

**Search space and proposal conditioning.** For each event, the proposal loop searches all four classes of numerical constraints jointly: hard-gate thresholds $\theta_g$, soft-score expected magnitudes $\delta_c$, soft-score weights $w_c$, and window lengths $\Delta T_c$. To keep proposals on the right scale, each LLM proposal step is conditioned on per-field summary statistics (mean, standard deviation, and quantiles) computed from the auxiliary set's per-frame structured representations; these statistics anchor initial guesses but do not enter the AP objective.

### A.5 Multi-Source Fusion: Formal Definition

Per-source rankings from the symbolic-rule path, dense embedding, and sparse lexical retrieval are merged via query-conditioned weighted reciprocal rank fusion [34]:

$$s_{\text{fusion}}(v) = \sum_{s \in \{\text{rule, dense, bm25}\}} \frac{w_s(q)}{k_{\text{RRF}} + \text{rank}_s(v)}, \quad k_{\text{RRF}} = 60, \tag{7}$$

where $\text{rank}_s(v) \in \{1, 2, \ldots\}$ is the position of $v$ in source $s$'s ranking, $w_s(q) \in \mathbb{R}_{\ge 0}$ is the query-conditioned weight produced by the fusion router (Appendix D.7), and $k_{\text{RRF}}$ is the standard RRF smoothing constant [34]. The final ranking is obtained by sorting the candidate pool by $s_{\text{fusion}}$ in descending order; a post-fusion reranker is then applied over the top-$K'$ items (Section 4.1).

### A.6 DrivingDojo Event Annotation

We annotate 583 videos from DrivingDojo [14] with structured event labels and release these annotations at `https://github.com/nec-labs-ma/strive-d-data-annotation`. Each annotation specifies the event type, the object class (ego or surrounding vehicle), and the direction of motion where applicable. Of the 583 annotated videos, 236 are ground-truth positives for at least one query; the remaining videos serve as non-relevant candidates in the retrieval pool. The benchmark comprises 41 distinct queries, with each query having between 1 and 138 relevant videos (a video may be relevant to multiple queries).

As shown in Table 9, the queries fall into four event types: *Cut-In*, *Cut-Out*, *Lane Change*, and *Turn*. Cut-In and Cut-Out queries are further stratified by the surrounding vehicle's object class (car, motorcycle, two-wheeled vehicle, bicycle, truck, and bus), plus a class-agnostic variant; each class-specific query also includes left and right directional sub-queries. Lane Change and Turn queries involve the ego vehicle and likewise include directional variants. Each query is expressed as a free-form natural-language description specifying the event type, object class, and direction where applicable.

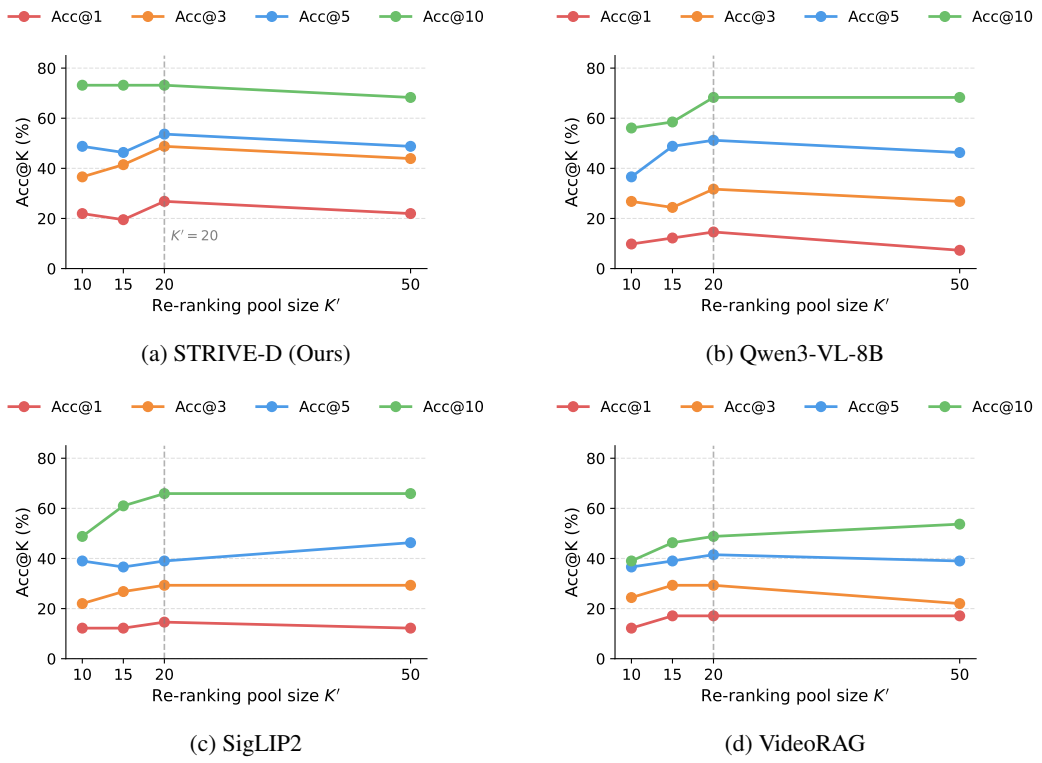
**Annotation protocol.** Each video is annotated with its event type. For events involving a surrounding vehicle, the annotation also records the vehicle's object class, color (white, grey, black, red, brown, or other), and direction relative to the ego vehicle (left, right, or front). Cut-in labels distinguish entry from the left or right, and cut-out labels distinguish departure to the left or right. Lane-change and turn labels specify left or right motion of the ego vehicle. Two annotators labeled every video, and two additional annotators checked the annotations for quality.

Table 9: Query categories in the DrivingDojo [14] annotation release. Cut-In/Cut-Out are stratified by object class (car, motorcycle, two-wheeled, bicycle, truck, bus) plus an class-agnostic variant, each with any/left/right direction variants. Video count range is across object-class queries; 236 unique videos total.

| Event | Queries | # Videos |
|---|---|---|
| Cut-In (surrounding vehicle) | 18 | 2–138 |
| Cut-Out (surrounding vehicle) | 17 | 1–80 |
| Lane Change (ego) | 3 | 47–98 |
| Turn (ego) | 3 | 38–92 |
| **Total / unique videos** | **41** | **236** |

# B Additional Experiments

## B.1 Effect of Candidate Pool Size $K'$




(a) STRIVE-D (Ours)

(b) Qwen3-VL-8B

(c) SigLIP2

(d) VideoRAG

Figure 7: Sensitivity to re-ranking pool size $K'$ on DrivingDojo, evaluated for STRIVE-D and reranker-equipped baselines. Every method either peaks at or is at the plateau by $K' = 20$, so we adopt this value uniformly throughout the main paper.

We study how the candidate pool size $K'$ affects retrieval performance for STRIVE-D and each reranker-equipped baseline (Figure 7). All evaluated methods attain strong, near-peak performance at $K' = 20$, with further increases yielding only marginal change. The Acc@$k$ values reported for the reranker-equipped baselines in Table 1 are lifted from this sweep at $K' = 20$ for fair comparison; we therefore fix $K' = 20$ throughout the main paper.

## B.2 Structured Grounding Choice: Symbolic Rules vs. NSVS-TL within Our Pipeline

The main-paper comparison in Table 1 shows that STRIVE-D outperforms NSVS-TL [6], but does not isolate *which part* of our framework is responsible for that gap: the calibrated symbolic

Table 10: **Structured grounding choice (DrivingDojo, %).** Replacing our calibrated symbolic rules with NSVS-TL's temporal-logic grounding within an otherwise identical fusion pipeline (dense + sparse + query-conditioned router + reranker) loses 14.6 pp Acc@1 and 4.7 pp mAP. The precision gain is attributable to the calibrated symbolic-rule component rather than to generic structured grounding.

| Variant | MRR↑ | mAP↑ | Acc@1 | Acc@3 | Acc@5 | Acc@10 |
|---|---|---|---|---|---|---|
| NSVS-TL standalone [6] | 31.8 | 4.9 | 22.0 | 34.2 | 43.9 | 53.7 |
| NSVS-TL within our pipeline | 28.7 | 3.9 | 12.2 | 34.2 | 53.7 | 70.7 |
| Full STRIVE-D | **38.7** | **8.6** | **26.8** | **48.8** | **53.7** | **73.2** |

rules, the multi-source fusion, the reranker, or some combination. To answer this, we compare three structured-retrieval variants on DrivingDojo [14]: NSVS-TL as a standalone retriever in its as-published configuration; NSVS-TL substituted into our pipeline (replacing the symbolic-rule component while keeping dense retrieval, sparse retrieval, the query-conditioned router, and the reranker identical); and full STRIVE-D.

Table 10 shows that full STRIVE-D outperforms NSVS+pipeline by 14.6 pp Acc@1, 4.7 pp mAP, and 10.0 pp MRR despite identical fusion machinery. The gap concentrates at top-of-list precision; broad-recall metrics (Acc@5, Acc@10) are closer because dense and sparse retrieval establish a recall floor independent of the symbolic choice. NSVS+pipeline also drops below NSVS-TL standalone at Acc@1: NSVS-TL emits binary verdicts that our continuous query-conditioned weighted fusion cannot leverage as effectively as graded scores, whereas our symbolic rules produce graded scores by design. This isolates the contribution of the calibrated symbolic rules beyond generic structured grounding: the gain is attributable to the rules' calibrated graded scoring, not to the surrounding fusion machinery.

Beyond accuracy, online inference cost separates the two paradigms decisively: NSVS-TL averages 4,868.05 s per query because each query requires running its full temporal-logic verification over the corpus, whereas STRIVE-D averages 3.10 s per query, a $\sim$1,500$\times$ reduction (Section 4.3). The cross-dataset NSVS-TL numbers in Table 1 were obtained at this prohibitive per-query cost; STRIVE-D reaches the reported accuracy without it.

### B.3 Per-stage Latency Breakdown

Table 11: **STRIVE-D per-stage latency breakdown (mean per query, DrivingDojo).** The two LLM stages, rule synthesis (run only when no library rule matches the query) and multi-source fusion, together account for $\sim$87% of the total 3,096.66 ms (3.10 s); the three retrieval branches (sparse, dense, rule matching) cost $\sim$384 ms combined. Both LLM stages are amenable to caching across queries that share an event category.

| Stage | Latency (ms) |
|---|---|
| Sparse retrieval (BM25 [22]) | 1.27 |
| Dense retrieval (Qwen3-VL-Embedding-8B [5]) | 128.54 |
| Rule matching | 254.16 |
| Rule synthesis | 1,987.43 |
| Multi-source fusion | 725.26 |
| **Total** | 3,096.66 |

Table 11 reports the per-stage online latency breakdown of STRIVE-D on DrivingDojo [14], complementing the cross-method comparison in Table 2 of the main paper. The two LLM stages, rule synthesis (invoked only on the exemplar-conditioned fallback path, when no library rule matches the query) and multi-source fusion, together account for $\sim$87% of the total per-query latency. Both stages are amenable to caching across queries that share an event category, which our timings do not exploit; the remaining stages (rule matching against per-frame records, dense retrieval, BM25 [22] sparse retrieval) are inexpensive.

### B.4 Variability across Runs

Several components of our pipeline are stochastic, including VLM-based video captioning (Appendix A.1), event extraction from captions, and the LLM proposer in the calibration loop (Appendix A.4). To assess the cumulative variability across these sources, we re-ran library construction and retrieval five times with independent samples, holding all other settings fixed, and report mean $\pm$ one standard deviation (1-$\sigma$) across the five runs in Table 12. At top-10, the candidate set typically saturates the relevant items, so the ranking is largely insensitive to small rule perturbations; the relative ordering against the strongest baseline is preserved in every run.

Table 12: **Variability of STRIVE-D across 5 independent runs.** Mean $\pm$ one standard deviation (1-$\sigma$); variability captures the cumulative effect of all stochastic components in the pipeline (VLM captioning, event extraction, LLM proposer). Standard deviations stay below 1.3 points for Acc@$\{1, 3, 5\}$ and below 0.3 for Acc@10 on all three benchmarks.

| Dataset | Acc@1 | Acc@3 | Acc@5 | Acc@10 |
|---|---|---|---|---|
| MMAU [38] | $26.5 \pm 1.1$ | $53.1 \pm 1.1$ | $59.2 \pm 1.1$ | $65.3 \pm 0.0$ |
| DrivingDojo [14] | $26.8 \pm 1.3$ | $48.8 \pm 1.1$ | $53.7 \pm 1.1$ | $73.2 \pm 0.0$ |
| CarCrash [37] | $28.9 \pm 1.2$ | $60.0 \pm 1.1$ | $73.3 \pm 0.9$ | $75.6 \pm 0.3$ |

### B.5 Proposer Comparison: Protocol

We compare calibration proposers with each rule's entity selector and the fields used by its hard gates and soft scores fixed. All calibrated variants optimize AP on the Nexar auxiliary set [35] (Equation (4)), with a budget of 20 candidate parameter assignments per rule. The assignment with the highest auxiliary AP is retained.

- **Uncalibrated baseline:** thresholds are fixed at $40\%$ of each field's range, with no search.
- **Random search:** sample 20 assignments, drawing thresholds uniformly from their allowed ranges and sampling weights, directions, and gate settings at random.
- **Bayesian search:** evaluate 10 random assignments, followed by 10 proposals concentrated around the best assignment found so far.
- **Grid search:** evaluate the 20 combinations of minimum change at $\{15, 30, 45, 60\}\%$ of each field's range and distance gate in $\{20, 30, 40, 50, 60\}$ m.
- **LLM proposer:** generate assignments conditioned on per-field statistics and best-so-far feedback, following Appendix A.4.

Random and Bayesian search are each evaluated over three runs; Table 4 reports mean $\pm$ standard deviation. Each resulting library is evaluated with the same matcher, adapter, fusion, and reranker.

### B.6 Perception Robustness: Protocol and Full Results

We perturb one module at a time in the precomputed perception records and rerun DrivingDojo [14] retrieval with five random seeds. The calibrated library, remaining fields, and dense and sparse paths are fixed.

**Independent perturbations (Table 13).** We randomly select $P\%$ of detected-object entries for removal, tracks for fragmentation, or vehicles for depth perturbation. Selected tracks are split at random points, with new identities assigned to their tails. Depth perturbation multiplies the selected vehicles' distances by factors drawn from $\mathcal{N}(1, (P/100)^2)$.

**Temporally correlated perturbations (Table 14).** False detections are added at $P\%$ of the original detection count and persist for 1–5 frames. Their classes are sampled from the clip, and their boxes and depths are copied from real objects with small noise. Consecutive misses remove $P\%$ of each object's detections in one contiguous block with a random start. Track interruptions remove $P\%$ of tracks for $k \in \{10, 20\}$ frames, after which they resume under new identities.

Table 13: **Independent perception corruption** (DrivingDojo Acc@10, %; mean ± std over 5 seeds; uncorrupted = 73.2).

| Corrupted module | $P = 0$ | $P = 25$ | $P = 50$ |
|---|---|---|---|
| Depth | 73.2 | $72.7 \pm 3.2$ | $72.2 \pm 3.3$ |
| Detection | 73.2 | $71.2 \pm 3.2$ | $70.2 \pm 2.0$ |
| Tracking | 73.2 | $70.7 \pm 1.7$ | $69.8 \pm 1.3$ |

Table 14: **Temporally correlated perception corruption** (DrivingDojo Acc@10, %; mean ± std over 5 seeds; uncorrupted = 73.2).

| Corruption | $P = 25$ | $P = 50$ | $P = 75$ |
|---|---|---|---|
| False detections | $68.3 \pm 3.5$ | $72.4 \pm 3.0$ | $69.1 \pm 3.0$ |
| Consecutive missed detections | $67.5 \pm 2.3$ | $64.2 \pm 2.3$ | $62.6 \pm 7.5$ |
| Track re-establishment ($k = 10$) | $71.7 \pm 1.3$ | $69.8 \pm 1.3$ | $66.7 \pm 1.2$ |
| Track re-establishment ($k = 20$) | $70.7 \pm 3.5$ | $64.2 \pm 4.1$ | $63.4 \pm 3.1$ |

Table 15: **Rule-generator and reranker comparison (DrivingDojo, %).** We vary the rule-generator LLM or the reranker while keeping the other components fixed.

| **Method** | **MRR** | **mAP** | **Acc@1** | **Acc@3** | **Acc@5** | **Acc@10** |
|---|---|---|---|---|---|---|
| *Rule Generator Choice* | | | | | | |
| Ours (w/ GPT-4o-mini) | 30.08 | 3.47 | 17.07 | 36.59 | 51.22 | 73.17 |
| Ours (default, GPT-4.1) | **38.7** | **8.6** | **26.8** | **48.8** | **53.7** | **73.2** |
| Ours (w/ GPT-4o) | 32.01 | 3.47 | 21.95 | 36.59 | 48.78 | 73.17 |
| *Reranker* | | | | | | |
| Ours (default, 2B Reranker) | **38.7** | **8.6** | **26.8** | **48.8** | **53.7** | **73.2** |
| Ours (w/ 8B Reranker) | 33.86 | 5.29 | 17.07 | 39.02 | 53.66 | 73.17 |

### B.7 Choice of Rule-Generator LLM and Reranker

To assess sensitivity to model choice, Table 15 compares rule-generator LLMs and Qwen3-VL rerankers [5] on DrivingDojo [14]. Acc@10 remains 73.2% across the tested variants, while Acc@1 varies with the generator: 26.8% for GPT-4.1 [36], 22.0% for GPT-4o [51], and 17.1% for GPT-4o-mini [52]. Replacing the default 2B reranker with the 8B model lowers Acc@1, MRR, and mAP, with Acc@5 and Acc@10 unchanged. For the embedding baselines, Table 16 reports the effect of adding Qwen3-VL-Reranker, which raises Acc@1 for both model sizes on all three datasets.

## C Additional Qualitative Results

Figure 8 presents additional qualitative comparisons on four complex queries from the DrivingDojo [14] dataset. Our method retrieves videos matching the query, while vision-language baselines (SigLIP2 [4], Qwen3-VL [5]) return less relevant results on these examples.

**Cross-dataset retrievals beyond ground-truth annotations.** Figure 9 presents examples in which our method retrieves videos that fall outside the ground-truth annotations, yet appear visually consistent with the query. For instance, the query *"car skids in snow"* retrieves snowy road-incident videos from CarCrash [37], DrivingDojo [14], and MM-AU [38]; the query *"motorcycle cutin"* retrieves motorcycles encroaching on the ego lane across all three datasets. These examples also illustrate that the same calibrated rule library can be applied across dataset boundaries without dataset-specific adaptation.

## D LLM Prompts

This section documents the seven LLM prompt templates used by STRIVE-D, grouped by pipeline stage. Library-construction prompts (Appendices D.1 to D.4) run once on the auxiliary set to caption

Table 16: **Effect of Qwen3-VL-Reranker on the embedding baselines (%).** For each dataset, we report Qwen3-VL-2B and Qwen3-VL-8B [5] with and without their corresponding Qwen3-VL-Reranker. The +Reranker rows use $K' = 20$ and match the main results (Table 1).

| Method | MRR(↑) | mAP(↑) | Acc@1 | Acc@3 | Acc@5 | Acc@10 |
|---|---|---|---|---|---|---|
| Dataset: *DrivingDojo*[14] | | | | | | |
| QWen3-vl-2B | 10.0 | 2.3 | 4.3 | 14.6 | 24.4 | 46.3 |
| QWen3-vl-2B + Reranker | 12.2 | 1.4 | 4.9 | 26.8 | 46.3 | 65.9 |
| QWen3-vl-8B | 15.8 | 1.2 | 4.9 | 19.5 | 31.7 | 56.1 |
| QWen3-vl-8B + Reranker | 20.1 | 2.2 | 14.6 | 31.7 | 51.2 | 68.3 |
| Dataset: *MM-AU* [38] | | | | | | |
| QWen3-vl-2B | 17.8 | 4.0 | 8.2 | 20.4 | 30.6 | 46.9 |
| QWen3-vl-2B + Reranker | 26.9 | 6.0 | 18.4 | 30.6 | 42.9 | 49.0 |
| QWen3-vl-8B | 34.6 | 8.6 | 18.4 | 32.7 | 44.9 | 53.1 |
| QWen3-vl-8B + Reranker | 32.7 | 6.3 | 20.4 | 38.8 | 55.1 | 61.2 |
| Dataset: *CarCrashDataset* [37] | | | | | | |
| QWen3-vl-2B | 31.9 | 24.8 | 17.8 | 37.8 | 48.9 | 64.4 |
| QWen3-vl-2B + Reranker | 38.3 | 32.1 | 26.7 | 44.4 | 55.6 | 64.4 |
| QWen3-vl-8B | 34.9 | 27.5 | 20.0 | 44.4 | 55.6 | 66.7 |
| QWen3-vl-8B + Reranker | 40.1 | 33.5 | 28.9 | 46.7 | 62.2 | 66.7 |

videos, extract candidate events, and calibrate the rules' numerical constraints. Retrieval-time prompts (Appendices D.5 to D.7) run per query to match against the calibrated library, synthesize an adapted rule when no library entry matches, and produce per-source weights that combine the symbolic, dense, and sparse retrieval signals through weighted reciprocal rank fusion [34].

### D.1 Video Captioner

The captioner runs at library construction only, on each video in the auxiliary set, to produce the natural-language description $\mathcal{C}'$ that downstream event extraction (Appendix D.2) parses for weak-supervision labels. The prompt targets kinematic content (spacing, movement, signals, scene adjustments) rather than generic scene description; even with this prompting, captions remain a noisy motion-event signal, which motivates the AP-averaged calibration objective (Appendix A.4). Listing 2 shows the prompt.

Listing 2: Video captioner system prompt: chronological motion-targeted description used at library construction.

```
You are an autonomous driving expert, specializing in recognizing traffic scenes.

You receive a series of traffic images captured from the perspective of the ego car. Your
     task is to analyze the video and provide a detailed, chronological description of the
     scene.

For the scene description, for each significant event or change, describe:
- Changes in Spacing: Note how the distance between vehicles and stable objects changes
     over time. Indicate if vehicles get closer to or farther from these objects.
- Movement Dynamics: Explain the movement of each vehicle, including their direction, any
     noticeable adjustments in their position relative to other vehicles and stationary
     elements, and whether the vehicle is approaching or stopping.
- Vehicle Signals: Pay attention to whether vehicles have their blinkers or lights on, and
     note their status in your description.
- Scene Adjustments: Describe any significant changes in the arrangement of vehicles and
     objects and their impact on the overall scene.

Ensure that the description captures the sequence of events as they occur in the video,
     detailing vehicle behavior and scene dynamics without referring to individual frames.
```

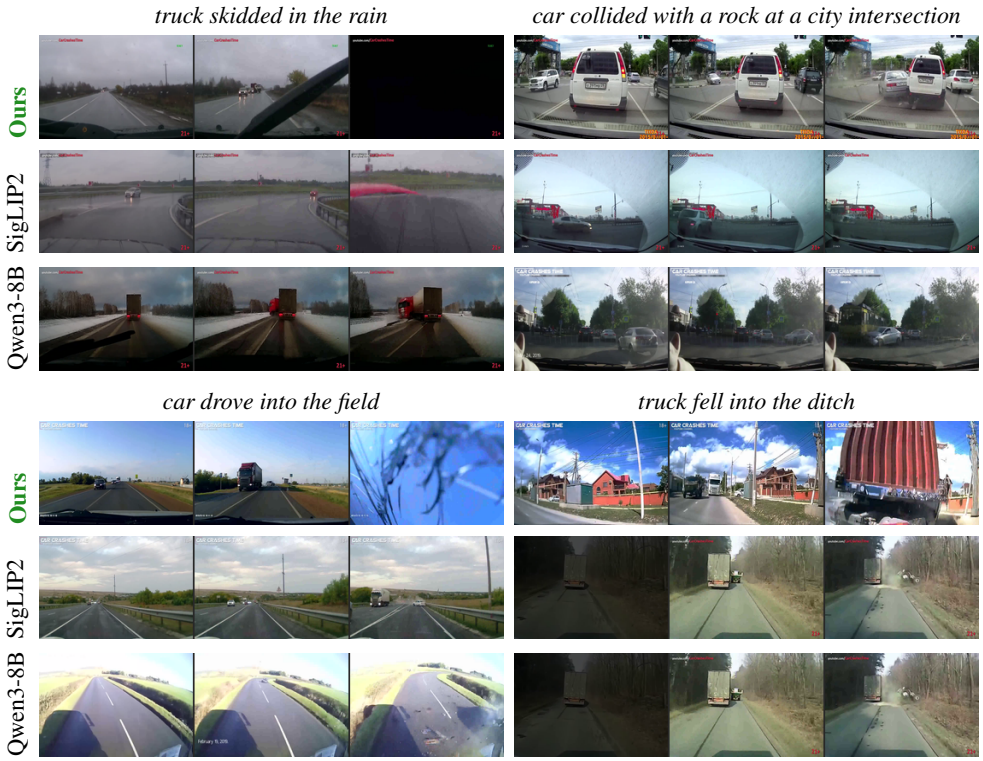



Figure 8: **Qualitative retrieval comparison.** Sampled frames from the top-1 retrieved video per method for four complex natural-language queries on DrivingDojo [14]; our method consistently surfaces semantically relevant videos while vision-language baselines (SigLIP2 [4], Qwen3-VL [5]) struggle with queries involving environmental conditions or rare event types.

### D.2 Event Extraction

For each auxiliary video, the extractor parses the captioner's description into a list of (event_type, subject) candidates with a short evidence quote. After consolidation across the auxiliary set, the deduplicated list forms the event index $\mathcal{E} = \{e_1, \ldots, e_M\}$ used to construct the calibrated library $\mathcal{L}$. Per-event positive and negative video sets $\mathcal{V}_m^+, \mathcal{V}_m^-$ are derived by matching extracted events against each caption: a video is positive for $e_m$ if its extracted events contain $e_m$, and negative otherwise. Listing 3 shows the prompt.

Listing 3: Event extraction system prompt: emits a JSON array of (event_type, subject, confidence, evidence) tuples per caption.

```
You are an expert in driving video analysis and vehicle kinematics.

Given a driving video summary, extract ALL notable kinematic events -- i.e.,
events that involve measurable physical motion that could be captured by
sensors (speed, distance, lateral position, trajectory).

INCLUDE events involving:
- Sudden longitudinal speed changes (hard braking, rapid acceleration, emergency stop)
- Lateral movements (cut-in, cut-out, lane change, swerving)
- Object approach / departure (vehicle closing in, near miss, overtaking)
- Dynamic object crossing (pedestrian, cyclist, animal crossing the road)
- Emergency vehicle passing (siren, flashing lights with motion)
- Construction zone entry / slowing

DO NOT include:
- Static scene descriptions with no motion (parked car, road type, weather alone)
- Object presence without motion relevance (car visible in mirror but not interacting)
- Ambiguous non-kinematic observations
```




Figure 9: **Cross-dataset retrieval results beyond ground-truth annotations.** For each query and dataset (CarCrash [37], DrivingDojo [14], MM-AU [38]), we show sampled frames from the top-1 retrieved video. The retrieved videos are not present in the ground-truth annotations, yet appear visually consistent with the query.

```
Output a JSON array (can be empty []).  Each element must have:
{
  "event_type": "<snake_case label, e.g. hard_brake / cutin / pedestrian_crossing>",
  "subject":    "ego" | "object" | "both",
  "confidence": "high" | "medium" | "low",
  "evidence":   "<short exact quote or paraphrase from the summary>"
}

Return ONLY the JSON array, no explanation.
```

### D.3 L2R Proposer: Initial Proposal

For each event $e_m$ extracted in Appendix D.2, the L2R loop's first iteration calls the proposer with the event's natural-language label and per-field summary statistics from the auxiliary set, producing a complete candidate rule (entity selector, hard gates, soft scores, agg_mode). Subsequent iterations use the revision prompt (Appendix D.4). Threshold values in the few-shot examples are illustrative structures only; the proposer is expected to substitute dataset-specific values from the field-statistics block. Listing 4 shows the prompt.

Listing 4: L2R proposer initial-proposal system prompt: emits a complete symbolic rule conditioned on the event label and per-field summary statistics.

```
You are a driving video retrieval expert. Translate a natural-language query about driving
     behavior into a structured JSON symbolic rule that can deterministically score every
     video in a dataset.

## Dataset context
Dataset: {DATASET_NAME} (dashcam videos, ego-centric, 10 fps, typical length 10-30 s).
Task: rank videos by relevance to the query; the rule will be scored against tracked-object
      time-series extracted from JSON annotations.

## Available fields (per tracked object, per frame)

| Field      | Source            | Unit   | Notes |
|------------|-------------------|--------|-------|
| distance   | 2D monocular depth| m      | Integer-quantized |
| loc_z      | 3D detector       | m      | Forward depth. Only for tracked objects. |
| loc_x      | 3D detector       | m      | Lateral offset: positive = right of ego,
     negative = left. |
```

```
| lane_rel   | VLM text label    | lanes  | Relative lane vs ego: 0=same, +1=one right, -1=
    one left. |
| bbox_area  | 2D detector       | px^2   | Bounding box of objects. |


### agg_mode="any" (object events: cut-in, approach, hard-brake)
Your threshold MUST be calibrated so that only rare individual tracks satisfy it:
- A threshold at the P50 means ~50% of all tracks will satisfy it -- far too permissive.
- For a query matching ~20% of videos: aim for P75-P85 (~15-25% of tracks exceed).
- For a rare event (<10%): target P90-P95.
- Use 'pct_exceeding' rows to read off: "if I set min_delta=X, what fraction of tracks will
     pass?"

Example: distance P50-change-in-4s = 12 m and 57% of tracks exceed 10 m ->
min_delta=10 is too permissive for a rare proximity event. Use 30 m (P80) instead.


## Entity spec
{
  "role": "object",          // "object" = a detected other vehicle/pedestrian
  "class_filter": ["car"],   // null = any class; valid: car, truck, bus, pedestrian,
     motorcycle, bicycle, barrier, traffic_cone
  "selection": "any",        // "any" = evaluate all matching tracks
                             // "closest" = only the nearest object by distance
  "agg_mode": "any"          // HOW to aggregate scores across multiple tracks:
                             // "any"       = max over tracks (default)
                             //               Use when ONE specific object causes the event
                             //               (cut-in, hard-brake, pedestrian crossing)
                             // "consensus" = mean over tracks when min_fraction pass the
    gate
                             //               Use for EGO-MOTION events (lane change, turn):
                             //               when ego changes lane, ALL nearby objects
    shift
                             //               in the same direction simultaneously
}

NOTE: This dataset does NOT have an explicit ego-lane-index field.
To detect ego maneuvers (lane change, turn), use surrounding objects' relative motion as
     proxy:
  - Ego changes lane RIGHT -> nearby objects shift LEFT (lane_rel decreases, loc_x
     decreases)
  - Ego turns LEFT -> nearby objects' loc_x shifts RIGHT
  - For ego-motion queries, ALWAYS set "agg_mode": "consensus" -- the pattern should appear
    across most surrounding vehicles, not just a lucky single track.

## Hard gates (binary, video scores 0 if none of its tracks pass)
{ "field": "distance", "op": "lt", "value": 40.0 }
Operators: "lt" (<), "le" (<=), "gt" (>), "ge" (>=).
Use gates only when there is a clear physical reason (e.g., object must be within 40 m for
     a near-miss).

## Soft Scores (continuous in [0, 1])
soft_scores: weighted evidence terms that produce a graded relevance in [0, 1] per video (
     unlike hard_gates which are binary pass/fail).

Each entry tests one kinematic signal on one field and scores smoothly:
  signal == threshold -> 0.5;  signal == 2x threshold -> 1.0;  wrong direction -> 0.

Per-track scores are aggregated by entity.agg_mode:
  "any"       -> max  (one object suffices: cut-in, brake, pedestrian crossing)
  "consensus" -> mean over tracks, gated by min_fraction (ego-motion: lane change, turn)

The final video score is the weight-weighted average of all soft_scores.
Use weight to express relative importance; use min_fraction (not min_delta) to discriminate
      ego-motion from noise.


## Output format
```

```
Return ONLY a valid JSON object -- no markdown, no comments, no text outside the JSON:

{
  "event": "short_snake_case_event_name",
  "confidence": 0.85,
  "entity": {
    "role": "object",
    "class_filter": ["car", "truck"],
    "selection": "any",
    "agg_mode": "any"
  },
  "hard_gates": [...],
  "soft_scores": [...],
  "scoring_notes": "One sentence explaining the rule's logic, threshold choices, direction,
     and agg_mode."
}

## Few-shot examples
Note: threshold values in these examples are illustrative STRUCTURES only.
Always substitute thresholds from the {FIELD_STATS} section above for the actual dataset.

### Query: "ego car change lane to the right"
{
  "event": "ego_lane_change_right",
  "confidence": 0.80,
  "entity": {
    "role": "object",
    "class_filter": ["car", "truck", "bus"],
    "selection": "any",
    "agg_mode": "consensus"
  },
  "hard_gates": [],
  "soft_scores": [
    {
      "type": "monotonic_trend",
      "field": "loc_x",
      "direction": "decreasing",
      "min_duration_seconds": 2.0,
      "min_fraction": 0.20,
      "weight": 1.0
    }
  ],
  "scoring_notes": "Ego moves right -> ALL nearby objects shift LEFT (loc_x decreasing).
     min_fraction=0.20 requires 20% of tracks to show a 2+ s sustained decrease
     simultaneously. In noise ~25% of tracks show this in some window; but a real lane
     change shifts ALL tracks consistently, so a much higher fraction agree -> better-
     ranked videos. NO temporal_change -- window-based change is dominated by noise for ego
     -motion."
}

### Query: "object approaching ego"
{
  "event": "approach",
  "confidence": 0.90,
  "entity": {
    "role": "object",
    "class_filter": null,
    "selection": "any",
    "agg_mode": "any"
  },
  "hard_gates": [],
  "soft_scores": [
    {
      "type": "monotonic_trend",
      "field": "distance",
      "direction": "decreasing",
      "min_duration_seconds": 3.0,
      "weight": 1.0
```

```
    },
    {
      "type": "temporal_change",
      "field": "distance",
      "min_delta": 30.0,
      "window_seconds": 5.0,
      "direction": "negative",
      "weight": 0.8
    }
  ],
  "scoring_notes": "One specific object approaches. direction=negative on temporal_change
     counts only distance reductions (not random fluctuation). agg_mode=any -- only the
     approaching object needs to show this signal. Threshold at >=P85 of distance change."
}

Now translate the following query. Return ONLY the JSON object.
```

### D.4 L2R Proposer: Revision

When a candidate assignment fails self-consistency checks on the auxiliary set, the proposer is fed back the previous AP, a percentile summary of the score distribution, and a list of issue tags, and asked to revise. Listing 5 shows the message.

Listing 5: L2R proposer revision user message: feeds back the previous AP, score-distribution percentiles, and a list of issue tags, with the best-so-far assignment as the anchor for the next proposal.

```
The previous rule for query '{query}' achieved AP={ap_t} on the
auxiliary set (target >= {tap}).
Best-so-far AP: {ap_best}; use the best-so-far rule as the anchor for
this revision.

Issue tags detected:
{issues}

Score distribution across {N} auxiliary videos:
  P10={p10}  P25={p25}  P50={p50}  P75={p75}  P90={p90}  P95={p95}
  nonzero (>0.05): {nonzero_ratio} of videos

Please revise the rule with these adjustments:
{hints}

Keep the same event type. Return ONLY a valid JSON object.
```

### D.5 Library Matcher

The library matcher maps a natural-language query to one of 21 predefined event categories and returns a scalar confidence score. Queries with confidence $\geq \gamma_h$ ($\gamma_h = 0.6$) are routed directly to the matched library template; queries with confidence $< \gamma_h$ trigger the rule translator, which generates an adapted rule using the highest-confidence library template as a few-shot exemplar. Listing 6 shows the prompt.

Listing 6: Library matcher system prompt: event categories, disambiguation rules, and JSON output format.

```
You are a driving event classifier for a video retrieval system.

Given a natural-language query about a driving incident, identify which of the
following event categories best describes the PRIMARY event. If no category
fits well, return "none".

Event categories:
- hard brake: The EGO vehicle decelerates suddenly/abruptly.
- swerving: The EGO vehicle weaves or makes erratic lateral movements
  (all surroundings shift together -- camera effect).
```

```
  NOT for a surrounding vehicle that moves sideways; use "lateral movement".
- lane change: The EGO or another vehicle smoothly changes lanes.
- evasive maneuver: The EGO driver steers/brakes sharply to avoid an obstacle.
- sudden speed change: The EGO vehicle abruptly accelerates or decelerates.
- sudden stop: The EGO vehicle stops abruptly from speed.
- loss of control: A vehicle loses control: spins out, drifts. If the query
  adds a condition the kinematic rule cannot encode (snow/rain/ice/"for no
  apparent reason"), cap confidence at 0.55.
- rollover: A vehicle rolls over, tips onto its side, or flips.
- reverse: A vehicle is reversing/backing up.
- sharp turn: A vehicle makes a sharp or aggressive turn.
- cutin: A SURROUNDING vehicle cuts in front of the ego car
  (lane-change encroachment). NOT a general collision.
- cutout: A SURROUNDING vehicle ahead moves out of the ego lane.
- lateral movement: A SURROUNDING vehicle (not the ego) moves noticeably sideways.
- object approach: A surrounding object rapidly approaches the ego.
- object departure: A surrounding object moves away from the scene.
- traffic disruption: A vehicle blocks or disrupts traffic flow.
- collision: Direct physical impact: vehicle-vehicle or vehicle-object crash.
- near miss: Close call without actual contact.
- pedestrian crossing: A pedestrian crosses in front of the ego.
- emergency vehicle: Ambulance, fire truck, or police is present.
- dynamic object crossing: Animal, debris, or non-vehicle object crosses the road.

Disambiguation rules:
- "swerving" = EGO weaves; all surroundings shift together (camera).
  "lateral movement" = a SURROUNDING vehicle moves sideways. Not the same.
- "cutin"/"cutout" = lane-change encroachments only -- not crashes.
- "collision" = any direct physical impact.
- "rollover" = rolls, tips, flips, falls on its side.
- Condition-modifier rule: weather/road conditions (snow, rain, ice) or vague
  causes ("for no apparent reason") set confidence <= 0.55, so the system
  generates a fresh adapted rule instead of reusing the library template.
- Return "none" only when the query describes no driving incident at all.

Respond with valid JSON only -- no extra text:
{"event": "<category_or_none>", "confidence": <0.0-1.0>, "reason": "<one sentence>"}
```

### D.6 Rule Adapter

When the library matcher confidence falls below $\gamma_h$, the rule adapter produces a query-specific rule by adjusting the highest-confidence library template under dataset field statistics, treating the template as a few-shot exemplar. The system prompt below (Listing 7) specifies the rule schema, constraint types, and few-shot examples; the user message (Listing 8) prepends the matched template alongside the new query.

Listing 7: Rule adapter system prompt: task definition, field statistics, rule schema, constraint types, and few-shot examples.

```
You are a driving video retrieval expert. Translate a natural-language query
about driving behavior into a structured JSON symbolic rule that can
deterministically score every video in a dataset.

Dataset context
Dataset: [DATASET_NAME] (dashcam videos, ego-centric, 10 fps, 10-30s).
Task: rank videos by relevance to the query; the rule will be scored against
tracked-object time-series extracted from JSON annotations.

Available fields (per tracked object, per frame):
- distance   : forward distance (m), 2D depth
- loc_z      : forward depth (m), 3D detection
- loc_x      : lateral offset (m), 3D detection; +=right of ego
- lane_rel   : lane offset (lanes), VLM; 0=same, +1=right, -1=left
- bbox_area  : bounding-box area (px^2), 2D detection

CRITICAL: Data-driven field statistics for this dataset
```

```
[Automatically generated from the target dataset. Contains, for each field
(distance, loc_x, loc_z, lane_rel) and each time window (2s, 4s, 8s):
percentile distribution of max|Delta value| across all tracked objects, plus
the fraction of tracks exceeding representative thresholds in a 4s window.
Example snippet (DrivingDojo):]

How to use the statistics to set thresholds:
- A threshold at P50 means ~50% of tracks satisfy it -- too permissive for rare events.
- For a query matching ~20% of videos: aim for P75-P85 (~15-25% of tracks exceed).
- For a rare event (<10%): target P90-P95.

Entity spec:
{
  "role": "object",
  "class_filter": ["car"],   // null = any; valid: car, truck, bus,
                             //   pedestrian, motorcycle, bicycle,
                             //   barrier, traffic_cone
  "selection": "any",        // "any" = all matching tracks
  "agg_mode": "any"          // "any"       = max over tracks (ONE object causes event)
                             // "consensus" = mean when min_fraction pass (EGO-MOTION)
}

Constraint types:

temporal_change -- max field change within any sliding window:
{
  "type": "temporal_change",
  "field": "distance",
  "min_delta": 30.0,          // set from field stats (P80-P95)
  "window_seconds": 4.0,
  "direction": null,          // null=absolute | "positive" | "negative"
  "min_fraction": null,       // only for consensus: fraction of tracks >= 0.5
  "weight": 1.0
}

Hard gates (binary; video scores 0 if no track passes):
{ "field": "distance", "op": "lt", "value": 40.0 }
Operators: "lt" (<), "le" (<=), "gt" (>), "ge" (>=).
Use gates only when there is a clear physical reason.

Output format -- return ONLY a valid JSON object, no markdown:
{
  "event": "short_snake_case_event_name",
  "confidence": 0.85,
  "entity": { "role": "object", "class_filter": [...],
              "selection": "any", "agg_mode": "any" },
  "hard_gates": [...],
  "soft_scores": [...],
  "scoring_notes": "One sentence explaining the rule's logic, thresholds, and agg_mode."
}

Few-shot examples (thresholds are dataset-specific; substitute from FIELD_STATS):

Query: "other car cutin"
{
  "event": "cutin",
  "confidence": 0.90,
  "entity": { "role": "object", "class_filter": ["car","truck"],
              "selection": "any", "agg_mode": "any" },
  "hard_gates": [{"field": "distance", "op": "lt", "value": 40.0}],
  "soft_scores": [
    { "type": "temporal_change", "field": "loc_x", "min_delta": 14.0,
      "window_seconds": 4.0, "direction": null, "weight": 1.0 },
    { "type": "temporal_change", "field": "distance", "min_delta": 20.0,
      "window_seconds": 4.0, "direction": "negative", "weight": 0.7 }
  ],
  "scoring_notes": "Lateral shift >= P90 loc_x combined with approach; agg_mode=any."
}
```

```
Query: "ego car change lane to the right"
{
  "event": "ego_lane_change_right",
  "confidence": 0.80,
  "entity": { "role": "object", "class_filter": ["car","truck","bus"],
              "selection": "any", "agg_mode": "consensus" },
  "hard_gates": [],
  "soft_scores": [
    { "type": "temporal_change", "field": "loc_x", "min_delta": 3.5,
      "window_seconds": 4.0, "direction": "negative",
      "min_fraction": 0.20, "weight": 1.0 }
  ],
  "scoring_notes": "Ego right => objects shift left (negative loc_x); min_fraction=0.20 for
     consensus agreement; agg_mode=consensus."
}

Query: "object approaching ego"
{
  "event": "approach",
  "confidence": 0.90,
  "entity": { "role": "object", "class_filter": null,
              "selection": "any", "agg_mode": "any" },
  "hard_gates": [],
  "soft_scores": [
    { "type": "temporal_change", "field": "distance", "min_delta": 30.0,
      "window_seconds": 5.0, "direction": "negative", "weight": 1.0 }
  ],
  "scoring_notes": "direction=negative counts reductions only; agg_mode=any."
}

Now translate the following query. Return ONLY the JSON object.
```

Listing 8: Rule adapter user message: injects the closest library template as an initialization seed alongside the new query.

```
A similar event from the rule library is "[matched_event_key]"
(keyword similarity=[sim]).
Use this rule as a starting point and adapt it to the new query.
Adjust thresholds to match the target dataset's field statistics,
and change entity/class_filter as needed.

Reference rule:
[JSON of the closest library template]

Query: <natural-language query>
```

### D.7 Fusion Router

The fusion router outputs a per-source weight for three retrieval backends: a dense vision-language embedding (Backend A), the symbolic-rule path (Backend B), and a sparse lexical retriever over per-video summaries (Backend C). The prompt frames weight assignment as a perception-alignment decision: which backend's input modality contains the discriminator that separates matching from non-matching videos for this query? The LLM identifies a primary backend whose modality carries the discriminator and optionally adds a supplementary weight to Backend C when the query has surface tokens (a color, a model, a place name) likely to appear in summary text. Three geometric facts about the dashcam setup (ego-motion produces global pixel shifts; surrounding-object motion produces local pixel changes whose visibility depends on end-of-motion location; summary text rarely contains technical event names) anchor the decision in physical priors rather than free-form reasoning. Listing 9 shows the prompt.

Listing 9: Fusion router prompt: outputs per-source weights for dense (Backend A), symbolic rule (Backend B), and sparse lexical (Backend C) retrieval.

```
For this query, assign per-source weights for fusion across three retrieval
backends.  Each backend produces its own ranking; the weights you output
determine how much each ranking contributes to the final ordering.
The three backends, identified by neutral labels A, B, C, are each defined
by what they perceive and nothing more:

  - Backend A : perceives pixels.  Ranks each candidate video by the
                visual similarity between its dashcam frames and the
                implied content of the query, as a human would judge it
                by looking at the frames.  Returns degenerate ranks when
                matching and non-matching frames look the same to a
                human.

  - Backend B : has no access to pixels.  Perceives, for each video, a
                per-frame numeric record of detected objects (each
                object's position relative to the ego camera, distance,
                over time).  Ranks each candidate by whether this
                numeric record matches a pattern derived from the query.
                Returns informative ranks precisely when the visual
                frames are NOT enough to discriminate matching from
                non-matching cases -- when the discriminator is in the
                numeric record only.

  - Backend C : perceives a single natural-language summary string per
                video.  Ranks candidates by token overlap with the
                query.  Useful only when the discriminating words in
                the query also literally appear in the matching summaries
                (and not in non-matching ones).

Geometric facts about the dashcam (apply these directly):

  Fact 1.  The camera is rigidly mounted on the ego vehicle.  When EGO
  moves (rotates or translates), every pixel in the frame shifts coherently
  -- lane markings sweep, the horizon tilts, every off-road landmark slides.
  This is a global change in pixels that Backend A picks up immediately.

  Fact 2.  When the ego is straight-and-steady and another object moves,
  only the pixels covering THAT object change.  Whether Backend A picks up
  this local pixel change depends on where the object ends up:
    - End position in the central viewport (large, close, ahead): the
      object dominates the frame, matching frames look distinct from
      non-matching ones, Backend A discriminates.
    - End position off-centre (alongside ego in a parallel lane, no longer
      interacting with ego): matching frames look the same as ordinary
      multi-lane driving, Backend A's signal is degenerate; only Backend B
      (which sees the numeric record) can discriminate.

  Fact 3.  Dashcam summary text is generic ("a car drives along ...").
  Technical event names ("cutin", "cutout", "lane change") rarely appear
  in summaries even when the event occurs, so Backend C is informative only
  when the query has a non-technical token (a color, a model, a place).

Reasoning protocol:

Step 1 -- Decode the physical motion in ONE short sentence:
  (a) the moving subject (ego, or a specific other object class)
  (b) the direction of motion in ego's reference frame
  (c) where the subject is at the END of the motion

Step 2 -- Identify the PRIMARY backend whose perception aligns with the
discriminator for this query.  Apply Facts 1 and 2 to decide which
backend's input modality contains the discriminator that separates
matching from non-matching videos.  The primary backend should receive
the highest weight.

Step 3 -- Decide whether to add a SUPPLEMENTARY weight to Backend C.
Inspect the query's surface tokens: if the query contains non-technical
```

```
tokens (a color, a model, a place name) likely to appear literally in
summary text (Fact 3), assign Backend C a small supplementary weight;
otherwise leave it at zero.

Step 4 -- Output JSON of the form
    {"dense": <float>, "rule": <float>, "bm25": <float>}
where the value at each label is the weight you have assigned to that
backend (Backend A <-> "dense", Backend B <-> "rule", Backend C <-> "bm25").

Query: "{query}"

Output Step 1, Step 2 (cite which Fact and why), Step 3 (count surface
tokens and decide whether Backend C should receive supplementary weight),
and on a final line output ONLY the JSON object.
```

# E Limitations and Broader Impact

## E.1 Limitations

Our rule schema lacks explicit temporal sequencing and scoped negation, so queries requiring these operators rely on dense or lexical retrieval, or approximate synthesized rules. Retrieval also depends on upstream detection and tracking, particularly under temporally correlated failures (Section 4.3); transfer of a calibrated library across perception stacks remains untested. VLM caption errors introduce noise into calibration labels, and thresholds for events with few caption-derived positives may be unreliable (Table 8). Linear scans of per-frame records and up to two LLM calls per query may limit scalability beyond the evaluated corpora (up to $1{,}953$ videos). Finally, the contribution of the LLM fusion router is not isolated, as we have not compared it with uniform or validation-tuned fixed weights.

## E.2 Broader Impact

STRIVE-D targets natural-language retrieval over driving video; plausible positive uses include scenario mining for autonomous-driving development, retrieval of rare or safety-critical events for regulatory and post-incident analysis, and curation of training or evaluation corpora for downstream perception and planning. The same capability could be repurposed for driver surveillance or privacy-eroding video mining, and biases in the upstream perception pipeline (*e.g.*, demographic disparities in detection or tracking) propagate into retrieval results without being corrected by our calibration. To mitigate these risks, we recommend two safeguards: (i) restricting source corpora to data with appropriate consent and provenance, and (ii) auditing the upstream perception pipeline for such biases before deployment. More broadly, our results suggest that numerical calibration is a binding constraint in applying LLM-written rules to physically grounded retrieval, that data-driven calibration closes this gap without sacrificing the interpretability of rule-based scoring, and that this separation of rule construction and numerical calibration should extend to other domains with similar per-frame structured representations, such as surgical, sports, or industrial monitoring. Each application would require a separate assessment of domain-specific risks.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The retrieval and robustness claims are evaluated in Section 4; limitations are discussed in Appendix E.1.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Appendix E.1.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper presents an empirical retrieval framework and does not claim theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide full method details, hyperparameters, and LLM prompts in Sections 3 and 4.1 and Appendices A and D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Releasing the code is contingent upon approval from the authors' organization; the human-annotated DrivingDojo event-grounding annotations introduced in Appendix A.6 are released at `https://github.com/nec-labs-ma/strive-d-data-annotation`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We use pretrained models without fine-tuning and calibrate symbolic-rule parameters on a separate auxiliary corpus. Data splits, hyperparameters, calibration procedures, and prompts are described in Section 4.1 and Appendices A, B.5 and D.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Appendix B.4 reports the mean and one standard deviation across five runs that repeat library construction and retrieval, capturing variation in captioning, event extraction, and LLM proposals. These are run-to-run standard deviations, not confidence intervals.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Section 4.1 specifies the GPU and its memory, and Section 4.3 and Appendix B.3 reports online latency. We do not provide the full preprocessing and calibration costs or the total compute used for all experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We strictly followed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader impacts are discussed in Appendix E.2 of the appendix.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The new asset consists of event annotations for existing driving videos, rather than a new pretrained model or a newly scraped video corpus. Potential privacy and misuse concerns are discussed in Appendix E.2.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: We credit the original assets and summarize available license information in Table 17. The paper does not establish licensing terms for every code dependency: the public VideoRAG and NSVS-TL repositories do not state explicit code licenses, and the BM25 implementation is not identified.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

Table 17: Reported licenses and sources for the principal datasets, retrieval models, and code assets. Linked sources distinguish dataset terms from code licenses; the inventory does not establish licensing terms for every dependency.

| Asset | License or terms reported by the source |
|---|---|
| DrivingDojo [14] | Apache-2.0 in the current official dataset-card metadata; access requires acceptance of the dataset conditions. |
| MM-AU [38] | CC-BY-NC-4.0 in the official dataset card. |
| CarCrashDataset [37] | MIT in the official dataset repository. |
| Nexar [35] | Nexar Open Data License, including attribution, redistribution, and ethical-use conditions. |
| GPT models [36, 51, 52] | Accessed through the OpenAI API, subject to the applicable OpenAI service terms. |
| Qwen3-VL-Embedding [5] | Apache-2.0 in the official model card. |
| SigLIP [39] | Apache-2.0 in the official model card. |
| SigLIP2 [4] | Apache-2.0 in the official model card. |
| VideoRAG [21] | No explicit code license stated in the public repository. |
| NSVS-TL [6] | No explicit code license stated in the public repository. |
| BM25 [22] | Algorithm reference; the implementation package and its license are not specified in the paper. |

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Appendix A.6 documents the new event annotations, query inventory, and annotation protocol; the annotations and usage documentation are released at `https://github.com/nec-labs-ma/strive-d-data-annotation`. The open-access status is described in item 5 above.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: [N/A]

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: [N/A]

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Section 4.1 and Appendices A.1 and B.7 identifies the language and vision-language models used in the method and experiments; Appendix D provides the prompts.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.