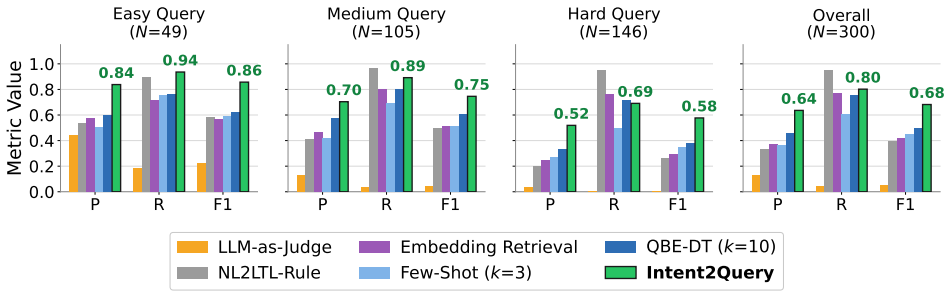
INTENT2QUERY: FROM NATURAL LANGUAGE INTENT TO EXECUTABLE SPATIOTEMPORAL QUERIES FOR MULTI-AGENT TRAJECTORY RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieving semantically meaningful spatiotemporal trajectories from large-scale driving data is essential for analyzing complex traffic behaviors and improving autonomous systems, yet remains challenging. Multi-agent interactions evolve over continuous space and time, making retrieval inherently difficult. Meanwhile, existing systems lack both formally verifiable query constraints and an accessible interface aligned with natural language intent, which is typically more intuitive and flexible than example-driven similarity mining. Moreover, such intent is frequently ambiguous and requires semantic grounding together with quantitative, verifiable evidence to ensure correctness. We introduce **Intent2Query**, a framework that translates natural language (NL) intent into a formally defined spatiotemporal query language and executes retrieval programs under Boolean and quantitative semantics. The proposed domain-specific language (DSL) is grounded in signal temporal logic (STL), enabling precise specification of multi-agent behaviors and spatiotemporal reasoning with quantitative robustness. To ensure reliability, parsing is constrained by schema and grammar rules, followed by deterministic validation, guaranteeing executable queries. We further construct a human-annotated scene-level benchmark over intersection LiDAR data to evaluate both parsing fidelity and retrieval performance. Results show that **Intent2Query** outperforms all ten baselines on all three datasets and at every difficulty level, improving F1 from 0.496 to 0.682 on our benchmark with non-overlapping confidence intervals and nearly doubling the precision of an untyped temporal-logic parser at comparable recall (0.636 v.s. 0.330), highlighting the necessity of typed formal semantic grounding for reliable spatiotemporal retrieval.



1 INTRODUCTION

The rapid advancement of autonomous driving systems and advanced driver-assistance technologies has led to an unprecedented growth in high-fidelity spatiotemporal data. Large-scale trajectory datasets, collected through multi-modal sensor suites such as LiDAR, radar, and cameras, form the foundation of modern intelligent transportation research. Public datasets including highD Krajewski et al. (2018), INTERACTION Zhan et al. (2019), Argoverse Chang et al. (2019), nuScenes Caesar et al. (2020), and the Waymo Open Motion dataset Sun et al. (2020) contain billions of kinematic states,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

capturing the complex and unscripted dynamics of multi-agent traffic interactions. These datasets provide critical empirical evidence for safety validation, rare-event discovery, and the development of interaction-aware learning systems. However, a fundamental bottleneck persists: the lack of effective mechanisms for semantically querying and retrieving specific spatiotemporal interaction scenarios. Identifying a precise multi-agent behavior, such as a vehicle braking sharply due to an encroaching pedestrian while another vehicle performs an unprotected left turn, requires reasoning over high-dimensional, continuous spatiotemporal signals with intricate temporal dependencies. In practice, researchers are often forced to rely on brittle, low-level scripts that approximate spatial conditions but fail to capture the compositional and temporal structure of such events.

Natural language offers the most intuitive and accessible interface for interacting with these datasets. The ability to describe a traffic scenario in plain language and retrieve corresponding real-world instances would significantly accelerate the development and validation of autonomous systems. However, direct text-based retrieval methods, which map natural language queries to trajectory data using embedding-based similarity search, lack formal guarantees of correctness and struggle with temporal compositionality, often conflating distinct event sequences that share similar spatial patterns. Rule-based systems, while precise, require manually engineered query logic for each task, do not scale to the combinatorial diversity of query intents, and offer limited flexibility. This gap motivates the central research question: *How can we translate unconstrained natural language into a formally grounded, executable spatiotemporal query, while guaranteeing retrieval correctness, efficiency, and interpretability over large-scale trajectory data?*

To address this challenge, we introduce **Intent2Query**, a neural-symbolic framework that bridges natural language and formal query execution through a structured intermediate representation. The key idea is to decompose the problem into a natural language $\rightarrow$ formal query $\rightarrow$ retrieval pipeline. Natural language serves as an expressive interface for specifying intent, while a domain-specific language (DSL) provides a compositional, constrained, and formally grounded representation of spatiotemporal behaviors. This separation allows us to preserve the flexibility of natural language while enforcing correctness and executability through formal semantics. Additionally, our DSL is grounded in signal temporal logic (STL), enabling precise specification of multi-agent interactions over continuous trajectories, along with quantitative robustness scores that measure the degree of satisfaction. To ensure reliability, we employ grammar-constrained semantic parsing that enforces syntactic and structural correctness during query generation, followed by deterministic validation to guarantee type safety and executability. This design ensures parsed queries correspond to a well-defined program, and retrieved results are accompanied by a verifiable, witness-backed explanation.

Contributions. **Intent2Query** establishes a neural-symbolic paradigm for spatiotemporal trajectory retrieval that (1) provides an intuitive natural language interface, (2) ensures precision and compositional parsing of executable DSL query, and (3) enables verifiable, evidence-backed retrieval via robustness-aware semantics. By integrating language understanding with formal reasoning, our framework enables scalable and trustworthy querying of complex multi-agent traffic data.

2 Related Work

Trajectory retrieval and scenario search. Prior work on trajectory retrieval treats the problem as either query-by-example or representation-based ranking. Mell et al. learns formal trajectory queries from labeled examples and demonstrates the value of quantitative semantics, but does not address natural-language intent. Zhan et al. introduces a benchmark for contextual multi-intent retrieval on Argoverse, framing the task as similarity search rather than intent-to-program translation. In autonomous driving, systems such as the natural language-based automatic identification system (AIS) Guo et al. (2025), Driving-RAG Chang et al. (2026), CARIM Ki et al. (2025), and BEV-TSR Tang et al. (2025) improve accessibility via textualization, captioning, and shared text–scene embeddings. However, these methods remain similarity-driven: they do not produce executable queries, enforce compositional spatiotemporal constraints, or provide witness-backed evidence.

Scenario representation, indexing, and formal specification. A complementary line of work focuses on infrastructure for large-scale scenario management. MetaScenario Chang et al. (2022) provides a unified framework for scenario description, storage, annotation, and indexing, while Sayed Mohamed Bakli E et al. demonstrates scalable spatiotemporal trajectory querying using distributed

SQL. These systems enable efficient organization and retrieval but rely on structured schemas or hand-authored queries. Declarative languages further support formal reasoning: RoadLogic Bartocci et al. (2026) compiles OpenSCENARIO ASAM (2026) specifications into executable simulations with solver-backed guarantees.

Natural language to formal logic and ambiguity-aware semantic parsing. NL2LTL Fuggitti & Chakraborti (2023) and nl2spec Cosler et al. (2023) show that LLMs can translate natural language into temporal logic, with interactive tools supporting refinement and repair. AMBROSIA Saparina & Lapata (2024) highlights ambiguity as a core challenge for executable semantic parsing, even in structured database settings. These works motivate the use of a formal intermediate representation in **Intent2Query**, but do not address multi-agent continuous trajectories, map-grounded traffic predicates, variable binding, or retrieval-oriented witness semantics. Relatedly, UltraQuery Galkin et al. (2024) demonstrates compositional logical reasoning with strong zero-shot generalization, but operates on discrete knowledge graphs rather than continuous spatiotemporal data.

Constrained decoding and neural-symbolic reasoning. **Intent2Query** builds on advances in reliable structured generation and neural-symbolic inference. Geng et al. shows that context-free grammars enable flexible, unified constraints for LLM outputs, including input-dependent grammars without finetuning. Baranowski & Hochgeschwender integrates LR parsers Knuth (1965) into neural decoders to enforce syntactic validity with minimal overhead, while Park et al. improves scalability by reducing preprocessing cost while preserving soundness. These works demonstrate that explicit grammatical constraints enhance validity and reliability. SEM-CTRL Albinhassan et al. (2025) further enforces semantic constraints, and Logic-LM Pan et al. (2023) and Logic-LM++ Kirtania et al. (2024) combine LLMs with symbolic solvers for faithful structured reasoning.

3 Problem Formulation

Problem. We study the problem of retrieving spatiotemporal multi-agent traffic scenes from large-scale trajectory datasets $\mathcal{D}$ using NL intent ℓ and DSL query q.

Scene model. Let $\mathcal{D} = \{\sigma_1, \ldots, \sigma_N\}$ denote a corpus of scenes, where each scene is a finite multi-agent trajectory trace. A scene is defined as $\sigma = (\mathcal{A}, \mathcal{Z}, \tau, \mathbf{y})$, where $\mathcal{A} = \{a_1, a_2, \ldots, a_M\}$ is a finite set of agents, $\mathcal{Z}$ is a set of map zones, $\tau = (t_0, \ldots, t_T)$ is a discrete time grid, and $\mathbf{y}(a, t)$ gives the continuous state (e.g., position, velocity) of agent $a \in A$ at time $t \in \tau$. Each agent has a semantic type $type(a) \in \mathcal{T}$, where $\mathcal{T} = \{\texttt{Vehicle}, \texttt{Pedestrian}, \texttt{Cyclist}, \ldots\}$.

Natural language to executable queries. Given a NL intent ℓ, a semantic parser p maps it to an executable query $q = p(\ell) \in \mathcal{L}$, where $\mathcal{L}$ is a well-defined spatiotemporal multi-agent query DSL. A query has the form $q = \exists \mathcal{X}[\mathcal{D}].\phi$, where $\mathcal{X} = \{x_1, \ldots, x_n\}$ is a set of typed agent variables in scene base $\mathcal{D}$, and ϕ is a temporal logic formula over these variables. A variable assignment $\nu : \mathcal{X} \to \mathcal{A}$ maps each variable $x \in \mathcal{X}$ to a concrete agent $\nu(x) \in \mathcal{A}$ in the scene, subject to type constraints. The formula ϕ is constructed from atomic predicates over continuous signals, geometric relations, map-aware predicates, and event macros, composed using Boolean and temporal operators.

Retrieval execution. For a scene σ and a variable assignment $\nu : \mathcal{X} \to \mathcal{A}$, the satisfaction relation $(\sigma, \nu, t) \models \phi$ determines whether ϕ holds at time t. To enable graded retrieval, we define a quantitative robustness function $\rho(\sigma, \nu, \phi, t) \in \mathbb{R}$ that measures the degree of satisfaction. The scene-level score is computed as $Score(\sigma, q) = \max_\nu \max_{t \in \tau} \rho(\sigma, \nu, \phi, t)$, and a scene is considered a match if $Score(\sigma, q) > 0$. The retrieval objective is therefore $\mathcal{O}_r(x) = \{\sigma \in \mathcal{D} \mid Score(\sigma, q) > 0\}$.

4 Method

Figure 1 shows user interaction and components of **Intent2Query**.

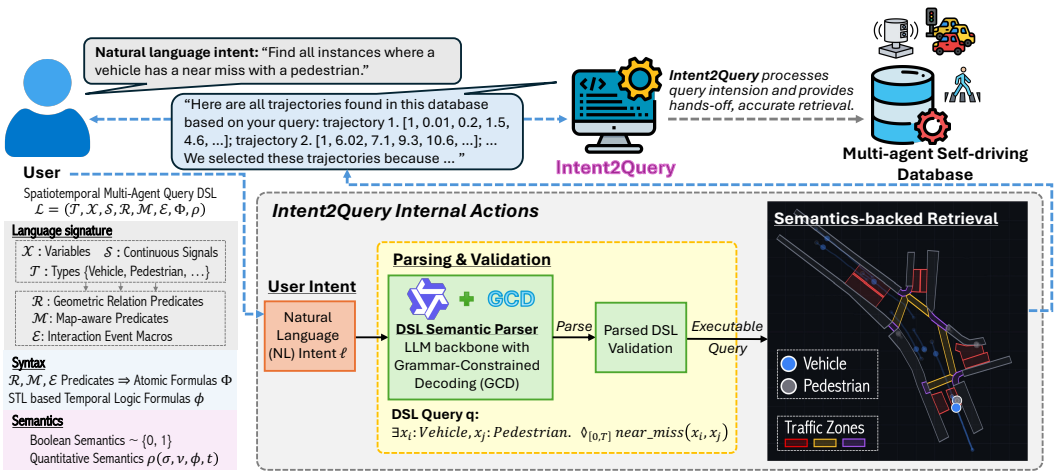


Figure 1: User interaction and components of Intent2Query.

4.1 Spatiotemporal Multi-Agent Query DSL

We formalize a typed, map-aware spatiotemporal query language $\mathcal{L}$ for retrieving traffic scenes from empirical multi-agent trajectory corpora $\mathcal{D}$. Formally, the DSL is defined as

$$\mathcal{L} = (\mathcal{T}, \mathcal{X}, \mathcal{S}, \mathcal{R}, \mathcal{M}, \mathcal{E}, \Phi, \rho),$$

where $\mathcal{T}$ is the set of types, $\mathcal{X}$ is the set of typed variables, $\mathcal{S}$ is the signal algebra, $\mathcal{R}$ is the set of geometric relation predicates, $\mathcal{M}$ is the set of map-aware predicates, $\mathcal{E}$ is the set of interaction event macros, Φ is the temporal formula language, and ρ is the quantitative semantics.

Language signature. A variable is written as $x : \tau_x$ with $\tau_x \in \mathcal{T}$, and $\mathcal{X} = \{x_1, \ldots, x_n\}$ denotes the declared variables.

Continuous signals $\mathcal{S}$ over agents and time contains unary signals and pairwise signals. For instance, unary signals contain position $pos(x,t) \in \mathbb{R}^2$, velocity $vel(x,t) \in \mathbb{R}^2$, heading $heading(x,t) \in \mathbb{R}$, speed $speed(x,t) \in \mathbb{R}$, acceleration $accel(x,t) \in \mathbb{R}$, and yaw rate $yaw_rate(x,t) \in \mathbb{R}$. Pairwise signals include distance $distance(x_i, x_j, t) \in \mathbb{R}$, relative velocity $rel_vel(x_i, x_j, t) \in \mathbb{R}^2$, closing speed $closing_speed(x_i, x_j, t) \in \mathbb{R}$, and time-to-collision $ttc(x_i, x_j, t) \in \mathbb{R} \cup \{+\infty\}$. All signals are derived from trajectory data.

Predicates over agents and time include $\mathcal{R}$, $\mathcal{M}$, and $\mathcal{E}$. These predicates enable compact, compositional, and interpretable representations of complex multi-agent behaviors.

- Geometric relation predicates $\mathcal{R}$ include proximity, approach, and collision risk relations such as $near(x_i, x_j, t)$, $approaching(x_i, x_j, t)$, and $conflict_proximity(x_i, x_j, t)$, defined via thresholded continuous signals (e.g., distance, closing speed, and time-to-collision).
- Map-aware predicates $\mathcal{M}$ capture both region membership and directional spatial relations. We define predicates $inside$, $ahead_of$, $behind$, $left_of$, and $right_of$, where $inside$ encodes zone membership based on polygonal geometry, and the directional predicates are defined from relative positions under a heading-aligned local coordinate frame, capturing longitudinal and lateral relationships between agents and map regions. For example, $inside(x, z, t) := [pos(x,t) \in geom(z)]$ indicates that agent x is inside zone z. These predicates enable reasoning over both semantic map regions and directional interactions in structured traffic environments.
- Interaction event macros $\mathcal{E}$ are derived operators that lift low-level predicates into higher-level temporal abstractions, including unary motion events such as $moving(x)$, $stop(x)$, $decelerating(x)$, and $hard_brake(x)$, temporal behaviors such as $turn_left(x)$ and $turn_right(x)$ defined over yaw rate, and pairwise interaction events such as $near_miss(x_i, x_j)$ based on joint distance and safety constraints.

More complex behaviors combine map context and temporal composition, more examplesIntent2Query are in Appendix A.

Syntax. Let $\sim \in \{<, \leq, >, \geq, =\}$. Atomic formulas are

$$\pi ::= f(\bar{x}, t) \sim c \mid r(\bar{x}, t) \mid m(\bar{x}, t) \mid e(\bar{x}),$$

where atomic formulas $\pi \in \Phi$ are constructed from the components of the DSL signature, consisting of (i) signal comparisons $s(\bar{x}, t) \sim c$ with $s \in \mathcal{S}$ and c as a threshold, (ii) geometric relation predicates $r(\bar{x}, t)$ with $r \in \mathcal{R}$, (iii) map-aware predicates $m(\bar{x}, t)$ with $m \in \mathcal{M}$, and (iv) interaction event macros $e(\bar{x})$ with $e \in \mathcal{E}$. $\bar{x}$ indicates one or more typed variables, and t is the time.

Temporal logic formulas are generated by

$$\phi ::= \pi \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \Diamond_{[a,b]}\phi \mid \Box_{[a,b]}\phi \mid \phi_1 \mathcal{U}_{[a,b]} \phi_2,$$

where $0 \leq a \leq b < \infty$. A query is defined as

$$q ::= \exists x_1 : \tau_1, \ldots, x_n : \tau_n \, [D].\, \phi,$$

which specifies a set of typed agent variables together with a temporal formula. For example, the natural-language query "a vehicle brakes hard" can be expressed as

$$\exists x : \texttt{Vehicle}.\, \Diamond_{[0,T]} hard_brake(x),$$

where the existential quantifier declares the agent variable and the formula ϕ encodes the temporal condition. Additional NL–to–DSL query mapping benchmark are provided in Appendix B.

Boolean Semantics. Given a scene σ, assignment ν, and time t, the satisfaction relation $(\sigma, \nu, t) \models \phi$ is defined inductively.

For signal comparisons, let $\sim \in \{<, \leq, >, \geq, =\}$:

$$(\sigma, \nu, t) \models f(\bar{x}, t) \sim c \iff f(\nu(\bar{x}), t) \sim c.$$

For geometric relation and map-aware predicates:

$$(\sigma, \nu, t) \models r(\bar{x}, t) \iff r(\nu(\bar{x}), t) = 1, \quad (\sigma, \nu, t) \models m(\bar{x}, t) \iff m(\nu(\bar{x}), t) = 1.$$

Event macros are evaluated via expansion:

$$(\sigma, \nu, t) \models e(\bar{x}) \iff (\sigma, \nu, t) \models expand(e(\bar{x})).$$

For bounded temporal operators:

$$(\sigma, \nu, t) \models \Diamond_{[a,b]}\phi \iff \exists t' \in [t + a, t + b] : (\sigma, \nu, t') \models \phi,$$

$$(\sigma, \nu, t) \models \Box_{[a,b]}\phi \iff \forall t' \in [t + a, t + b] : (\sigma, \nu, t') \models \phi,$$

$$(\sigma, \nu, t) \models \phi_1 \mathcal{U}_{[a,b]} \phi_2 \iff \exists t' \in [t + a, t + b] \text{ s.t. } (\sigma, \nu, t') \models \phi_2$$
$$\forall t'' \in [t, t'] : \ (\sigma, \nu, t'') \models \phi_1.$$

Quantitative Semantics. To enable graded evaluation, each formula is associated with a robustness score $\rho(\sigma, \nu, \phi, t) \in \mathbb{R} \cup \{\pm\infty\}$.

For signal comparisons,

$$\rho(\sigma, \nu, f(\bar{x}, t) \geq c, t) = f(\nu(\bar{x}), t) - c, \quad \rho(\sigma, \nu, f(\bar{x}, t) \leq c, t) = c - f(\nu(\bar{x}), t).$$

For Boolean predicates $\pi \in \mathcal{R} \cup \mathcal{M}$,

$$\rho(\sigma, \nu, \pi, t) = \begin{cases} +\kappa & \text{if } (\sigma, \nu, t) \models \pi, \\ -\kappa & \text{otherwise,} \end{cases}$$

with a fixed constant $\kappa > 0$.

Robustness for composite formulas follows standard STL semantics:

$$\rho(\sigma, \nu, \neg\phi, t) = -\rho(\sigma, \nu, \phi, t),$$
$$\rho(\sigma, \nu, \phi_1 \wedge \phi_2, t) = \min\big(\rho(\sigma, \nu, \phi_1, t), \rho(\sigma, \nu, \phi_2, t)\big),$$
$$\rho(\sigma, \nu, \phi_1 \vee \phi_2, t) = \max\big(\rho(\sigma, \nu, \phi_1, t), \rho(\sigma, \nu, \phi_2, t)\big).$$

For temporal operators,

$$\rho(\sigma, \nu, \Diamond_{[a,b]}\phi, t) = \max_{t' \in [t+a, t+b]} \rho(\sigma, \nu, \phi, t'),$$
$$\rho(\sigma, \nu, \Box_{[a,b]}\phi, t) = \min_{t' \in [t+a, t+b]} \rho(\sigma, \nu, \phi, t'),$$
$$\rho(\sigma, \nu, \phi_1 \mathcal{U}_{[a,b]}\phi_2, t) = \max_{t' \in [t+a, t+b]} \min\left(\rho(\sigma, \nu, \phi_2, t'), \min_{t'' \in [t, t']} \rho(\sigma, \nu, \phi_1, t'')\right).$$

4.2 Retrieval Execution

Given a parsed query $q = \exists \mathcal{X}[\mathcal{D}].\phi$, **Intent2Query** retrieves matching scenes by evaluating the temporal formula ϕ over a database of multi-agent trajectories using the aforementioned semantics. The execution pipeline consists of three stages: (i) macro expansion, (ii) variable binding, and (iii) robustness evaluation. First, high-level event macros are expanded into equivalent formulas over primitive signals and predicates. Second, for each scene σ, the system enumerates all valid assignments ν that bind query variables $\mathcal{X}$ to agent trajectories consistent with type constraints. Third, for each (σ, ν) pair, the formula ϕ is evaluated over time to compute a quantitative robustness trace. We adopt standard discrete-time STL robustness semantics Donzé (2010); Deshmukh et al. (2017). Each atomic predicate is evaluated as a real-valued signal over time, and composite formulas are evaluated recursively using max–min operations corresponding to temporal operators. For a given assignment ν, this yields a robustness signal $\rho(\sigma, \nu, \phi, t)$ over the time horizon. Finally, scenes with $Score(\sigma, q) > 0$ are returned and ranked by their scores, enabling graded retrieval based on the degree of satisfaction.

4.3 Grammar-Constrained Neural Semantic Parsing

Our DSL is a structured, typed query language with a fixed predicate inventory and temporal operators, naturally defined by a formal grammar. We adopt a neural-symbolic approach with grammar-constrained semantic parsing, mapping natural language intents directly to executable DSL queries while ensuring well-formed outputs. The LLM performs semantic interpretation, while the DSL schema constrains generation to syntactically and semantically valid programs, mitigating schema drift (e.g., invalid structures, missing bindings) and semantic drift (e.g., unsupported predicates, type inconsistencies). Given input ℓ and DSL grammar $\mathcal{G}$, the parser generates $\hat{z} = f_\theta(\ell; \mathcal{G})$, which is transformed into a validated program $z^\star = \mathcal{V}(\mathcal{R}^D(\mathcal{N}(\hat{z})))$ via normalization $\mathcal{N}$, deterministic repair $\mathcal{R}^D$, and grammar-level validation $\mathcal{V}$. Only validated outputs are compiled for execution. The parser is conditioned on a dynamically constructed DSL registry specifying predicates, types, and operators, ensuring consistency and reducing hallucination as the language evolves. Strict validation enforces binding consistency, predicate arity, type compatibility, and temporal constraints, with lightweight repairs applied only when unambiguous. This guarantees that all accepted programs are executable and semantically grounded.

5 Experiments

We evaluate **Intent2Query** along three axes: (i) *parsing performance*, measuring whether NL intents are translated into faithful and executable DSL programs; (ii) *retrieval performance*, measuring whether DSL queries retrieve the correct traffic scenes; and (iii) *cross-dataset generalization*, measuring whether the same parser, executor, and DSL transfer to trajectory corpora the system was not designed around. We additionally report a component ablation in Appendix D; a runtime and scalability analysis appears in Appendix E. All experiments were conducted on a Mac with M1 Pro chip and 32 GB memory.

5.1 EXPERIMENTAL SETUP

Benchmark construction. We construct a benchmark of 300 unique NL traffic-scene search intents paired with DSL annotations and scene-level relevance labels. 30 intents are written directly by domain experts; the remaining 270 are generated under expert supervision using LLM generator, then expert-verified. Queries are enumerated from DSL grammar coverage following a documented protocol, and were authored before inspecting the scene corpus, so the benchmark is not screened for queries the corpus happens to satisfy. The benchmark queries span three difficulty levels of easy, medium, and hard, and they are organized into six semantic categories of 50 queries each: unary motion, pairwise geometry, safety-oriented, temporal composition, map-aware zone queries, and interaction-event macros; detailed difficulty statistics across categories and datasets, together with representative examples for each difficulty level and semantic category, are provided in Appendix B.

Scene corpora and cross-dataset evaluation. We design the query benchmark around LADDMS Barbour et al. (2025) because it provides the broadest scenario coverage. To evaluate generalization beyond LADDMS, we additionally test on the public NGSIM Coifman & Li (2017) and INTERACTION Zhan et al. (2019) datasets. Because these datasets lack some semantic information required by the full benchmark, evaluation is restricted to their applicable query subsets, as summarized in Table 1. LADDMS supports both retrieval and ranking evaluation using human-verified graded relevance annotations (Appendix F), whereas NGSIM and INTERACTION are used only to assess cross-dataset retrieval generalization because such ranking annotations are unavailable.

Table 1: Statistics of the three evaluation corpora.

Statistic	**LADDMS**	**NGSIM**	**INTERACTION**
Domain	urban intersection	highway	urban intersection
Agent tracks (instances)	16,918	5,985	492
Total trajectory-hours	101.2 h	10.7 h	1.0 h
Agent types	veh/ped/cyc	vehicle only	veh/ped
Map zone annotations	Yes	No	No
Missing categories	—	map-aware, interaction	map-aware, interaction
Applicable queries	300	86	200
Easy / Medium / Hard	49 / 105 / 146	23 / 37 / 26	42 / 83 / 75

5.2 PARSING PERFORMANCE

Evaluation metrics. We evaluate NL→DSL parsing using complementary metrics that capture correctness at different levels. *DSL Validity (DSLV)* measures the fraction of outputs that satisfy the DSL grammar and semantic constraints. *Executable Rate (Exec)* measures whether the parsed program can be successfully executed, reflecting end-to-end functional correctness. *Predicate–Argument F1 (PAF1)* evaluates compositional structure by matching predicted predicates and arguments against the gold program. *Semantic Frame Accuracy (FrAc)* measures whether the predicted program captures the correct high-level intent. For efficiency, we report Latency and Overhead, where Overhead denotes the relative latency increase of GCD over the corresponding unconstrained LLM inference.

Comparisons. Rule-based parsing uses hand-crafted regex templates and lexicon matching for deterministic, zero-latency translation of supported query patterns. Additionally, we use the Qwen2.5 series models at varying scales (0.5B, 1.5B, and 7B) to compare unconstrained LLM parsing against grammar-constrained decoding (GCD), where the same LLM parser is guided by formal grammar constraints during decoding.

Result discussion. Table 2 shows that GCD substantially improves parsing for small and mid-scale models, especially on difficult queries. At 1.5B, DSL validity rises from 0.14 to 0.86 on Easy queries and from 0.00 to 0.79 on Hard queries, with corresponding gains in execution and semantic metrics. Similar trends hold at 0.5B, where unconstrained decoding largely fails but GCD recovers meaningful performance. These results indicate that grammar constraints are most effective when model capacity is limited. At 7B, however, unconstrained decoding is already near ceiling, and GCD can hurt Hard-query performance because locally valid prefix constraints may block globally correct completions. Thus, GCD primarily improves the reliability of smaller parsers rather than providing a uniform gain across model scales.

Table 2: Parsing quality and efficiency across query difficulty levels.

		Easy Query						**Medium Query**						**Hard Query**					
Parser	Model	DSLV	Exec	PAF1	FrAc	Lat. (ms)	Over. (%)	DSLV	Exec	PAF1	FrAc	Lat. (ms)	Over. (%)	DSLV	Exec	PAF1	FrAc	Lat. (ms)	Over. (%)
Rule	—	0.57	0.57	**0.43**	0.33	0	—	0.78	0.78	0.67	0.48	0	—	0.29	0.29	0.10	0.10	0	—
LLM	0.5b	0.00	0.00	0.00	0.00	1040	—	0.00	0.00	0.00	0.00	782	—	0.00	0.00	0.00	0.00	852	—
+GCD	0.5b	0.43	0.43	0.14	0.14	6751	+549	0.56	0.56	0.32	0.22	7947	+917	0.50	0.50	0.18	0.29	5913	+594
LLM	1.5b	0.14	0.14	0.14	0.10	1462	—	0.11	0.11	0.00	0.04	1298	—	0.00	0.00	0.00	0.00	1449	—
+GCD	1.5b	0.86	0.86	0.29	0.38	4513	+209	0.44	0.44	0.33	0.30	5727	+341	0.79	0.79	0.31	0.43	6576	+354
LLM	7b	**1.00**	**1.00**	0.29	**0.43**	5532	—	**1.00**	**1.00**	**0.78**	**0.59**	6067	—	**0.86**	**0.86**	**0.31**	**0.43**	7327	—
+GCD	7b	1.00	1.00	0.38	**0.43**	7330	+33	0.89	0.89	0.50	0.52	13684	+126	0.50	0.50	0.25	0.36	15742	+115

5.3 Retrieval Performance

Evaluation metrics. We evaluate retrieval across difficulty levels using macro-averaged *Precision*, *Recall*, and *F1*. To assess ranking quality, we report *Precision@*k and *Recall@*k for $k \in \{1, 5, 10\}$ and *nDCG@10*.

Baselines. We compare **Intent2Query** against ten baselines spanning five families. *Feature-based*: Few-shot similarity retrieval over scene feature vectors with $k \in \{1, 3, 10\}$ labeled examples. *LLM*: LLM-as-Judge, which prompts Qwen2.5-7B to assess query–scene relevance directly from textual scene summaries. *Neural retrieval*: a bi-encoder (Embedding Retrieval) that scores structured scene captions against the query, and a cross-encoder reranking pipeline (Embed + Rerank) applied to its top-100 candidates. *Query-by-example*: given k=10 positive and negative scenes per query, QBE-DT (decision tree), QBE-SVM, and QBE-STL (STL template mining) learn a per-query scene classifier. *Formal methods*: NL2LTL-Rule translates NL into vanilla Linear Temporal Logic (LTL) without typed bindings, map predicates, or quantitative robustness, isolating the contribution of our DSL extensions from that of formal execution as such.

Across difficulty levels and methods discussion. Table 3 shows that **Intent2Query** achieves the highest F1 at every difficulty level, with an overall 95% confidence interval $[0.647, 0.715]$ that does not overlap with the strongest baseline, QBE-DT $[0.453, 0.537]$. The advantage is largest on Hard queries (0.577 versus 0.377), where temporal composition, multi-agent reasoning, and map-aware constraints are most important. NL2LTL-Rule attains high recall but much lower precision, highlighting the value of typed and map-aware semantics, while neural and LLM-based baselines degrade substantially on compositional queries. Oracle-DSL performance is comparable to the deployed parser, indicating that parsing is not the dominant bottleneck.

Table 3: Retrieval Precision (P), Recall (R), and F1 by difficulty on LADDMS.

	Easy (40 queries)			**Medium** (105 queries)			**Hard** (146 queries)			**Overall** (300 queries)		
Method	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Few-Shot (k=1)	0.486	0.719	0.566	0.403	0.657	0.491	0.252	0.463	0.323	0.343	0.573	$0.422^{+0.035}_{-0.035}$
Few-Shot (k=3)	0.504	0.756	0.591	0.420	0.692	0.514	0.271	0.499	0.348	0.361	0.608	$0.446^{+0.035}_{-0.034}$
Few-Shot (k=10)	0.500	0.749	0.586	0.412	0.677	0.503	0.263	0.487	0.337	0.354	0.596	$0.436^{+0.036}_{-0.035}$
LLM-as-Judge	0.438	0.182	0.220	0.128	0.039	0.047	0.033	0.002	0.003	0.132	0.044	$0.054^{+0.020}_{-0.018}$
Embedding Retrieval	0.577	0.712	0.567	0.462	0.799	0.511	0.243	0.762	0.297	0.374	0.767	$0.416^{+0.036}_{-0.036}$
Embed + Rerank	0.635	0.245	0.237	0.502	0.213	0.208	0.241	0.204	0.167	0.397	0.214	$0.193^{+0.022}_{-0.021}$
QBE-DT (k=10)	0.600	0.765	0.623	0.574	0.802	0.603	0.334	0.715	0.377	0.461	0.754	$0.496^{+0.041}_{-0.043}$
QBE-SVM (k=10)	0.485	0.786	0.515	0.466	0.812	0.505	0.254	0.758	0.300	0.366	0.781	$0.407^{+0.041}_{-0.041}$
QBE-STL (k=10)	0.524	0.806	0.551	0.494	0.798	0.517	0.285	0.773	0.335	0.397	0.787	$0.434^{+0.041}_{-0.042}$
NL2LTL-Rule	0.539	0.894	0.581	0.414	0.967	0.496	0.199	0.950	0.262	0.330	0.947	$0.396^{+0.039}_{-0.039}$
Intent2Query	**0.838**	**0.936**	**0.857**	**0.704**	**0.892**	**0.746**	**0.519**	**0.691**	**0.577**	**0.636**	**0.802**	$\mathbf{0.682}^{+0.033}_{-0.035}$
w/ oracle DSL	*0.894*	*0.926*	*0.870*	*0.698*	*0.804*	*0.685*	*0.632*	*0.651*	*0.581*	*0.698*	*0.749*	$0.665^{+0.038}_{-0.040}$

Across datasets discussion. Table 4 shows that **Intent2Query** achieves the highest F1 across datasets. In contrast, the strongest baseline changes across datasets: QBE-DT performs best in-domain but degrades sharply off-domain, whereas NL2LTL-Rule transfers better but remains weaker overall. The smaller margin on NGSIM is expected because its vehicle-only, map-free setting removes many of the typed and map-aware cases where **Intent2Query** provides the largest benefit. These results indicate that the structured representation generalizes more consistently across domains than learned retrieval baselines.

Table 4: Cross-dataset Retrieval Performance.

	LADDMS (300 queries)			**NGSIM** (86 queries)			**INTERACTION** (200 queries)		
Method	P	R	F1	P	R	F1	P	R	F1
Few-Shot (k=1)	0.343	0.573	$0.422^{+0.035}_{-0.035}$	0.911	0.525	$0.560^{+0.084}_{-0.082}$	0.297	0.450	$0.251^{+0.042}_{-0.042}$
Few-Shot (k=3)	0.361	0.608	$0.446^{+0.035}_{-0.034}$	0.928	0.733	$0.793^{+0.050}_{-0.053}$	0.279	0.333	$0.266^{+0.041}_{-0.041}$
Few-Shot (k=10)	0.354	0.596	$0.436^{+0.036}_{-0.035}$	0.934	0.843	$0.866^{+0.046}_{-0.052}$	0.258	0.237	$0.241^{+0.046}_{-0.044}$
LLM-as-Judge	0.132	0.044	$0.054^{+0.020}_{-0.018}$	0.449	0.168	$0.191^{+0.071}_{-0.065}$	0.156	0.107	$0.101^{+0.031}_{-0.030}$
Embedding Retrieval	0.374	0.767	$0.416^{+0.036}_{-0.036}$	0.942	0.497	$0.648^{+0.019}_{-0.026}$	0.295	0.369	$0.275^{+0.039}_{-0.038}$
Embed + Rerank	0.397	0.214	$0.193^{+0.022}_{-0.021}$	0.944	0.286	$0.437^{+0.012}_{-0.017}$	0.319	0.402	$0.305^{+0.041}_{-0.041}$
QBE-DT (k=10)	0.461	0.754	$0.496^{+0.041}_{-0.043}$	0.240	0.197	$0.215^{+0.083}_{-0.078}$	0.223	0.244	$0.224^{+0.049}_{-0.048}$
QBE-SVM (k=10)	0.366	0.781	$0.407^{+0.041}_{-0.041}$	0.229	0.181	$0.200^{+0.077}_{-0.073}$	0.211	0.240	$0.213^{+0.046}_{-0.045}$
QBE-STL (k=10)	0.397	0.787	$0.434^{+0.041}_{-0.042}$	0.237	0.233	$0.235^{+0.090}_{-0.087}$	0.240	0.278	$0.249^{+0.053}_{-0.053}$
NL2LTL-Rule	0.330	0.947	$0.396^{+0.039}_{-0.039}$	0.932	0.988	$0.951^{+0.029}_{-0.039}$	0.296	0.618	$0.359^{+0.051}_{-0.050}$
Intent2Query	**0.636**	**0.802**	$\mathbf{0.682}^{+0.033}_{-0.035}$	**0.954**	**0.987**	$\mathbf{0.964}^{+0.025}_{-0.035}$	**0.386**	**0.508**	$\mathbf{0.388}^{+0.055}_{-0.055}$

Ranking quality discussion on LADDMS. Table 5 further shows that **Intent2Query** provides the strongest ranking quality. The gain in nDCG@10 is especially important because methods that produce only binary matches can achieve high recall while failing to rank the most relevant scenes near the top. Ranking performance decreases with query difficulty, but **Intent2Query** remains strongest across all levels.

Table 5: Ranking quality on LADDMS across query difficulty levels and methods.

	Precision@10				**Recall@10**				**nDCG@10**			
Method	Easy	Med	Hard	Overall	Easy	Med	Hard	Overall	Easy	Med	Hard	Overall
Few-Shot (k=1)	0.619	0.544	0.355	0.464	0.028	0.036	0.056	0.044	0.621	0.551	0.358	$0.468^{+0.041}_{-0.041}$
Few-Shot (k=3)	0.721	0.622	0.422	0.540	0.041	0.049	0.070	0.058	0.720	0.626	0.428	$0.545^{+0.043}_{-0.044}$
Few-Shot (k=10)	0.724	0.622	0.429	0.545	0.043	0.047	0.083	0.064	0.724	0.631	0.435	$0.551^{+0.045}_{-0.047}$
LLM-as-Judge	0.438	0.128	0.033	0.132	0.066	0.018	0.001	0.018	0.352	0.080	0.015	$0.093^{+0.028}_{-0.027}$
Embedding Retrieval	0.686	0.542	0.268	0.432	0.071	0.073	0.063	0.068	0.693	0.553	0.269	$0.437^{+0.044}_{-0.045}$
Embed + Rerank	0.678	0.538	0.276	0.433	0.071	0.070	0.062	0.066	0.690	0.540	0.284	$0.440^{+0.045}_{-0.045}$
QBE-DT (k=10)	0.607	0.564	0.331	0.457	0.105	0.078	0.080	0.083	0.572	0.547	0.312	$0.437^{+0.042}_{-0.042}$
QBE-SVM (k=10)	0.568	0.541	0.333	0.444	0.038	0.042	0.056	0.048	0.679	0.669	0.429	$0.554^{+0.046}_{-0.047}$
QBE-STL (k=10)	0.601	0.569	0.347	0.466	0.041	0.043	0.060	0.051	0.613	0.581	0.364	$0.480^{+0.045}_{-0.046}$
NL2LTL-Rule	0.539	0.414	0.199	0.330	0.032	0.037	0.022	0.029	0.537	0.415	0.199	$0.330^{+0.039}_{-0.039}$
Intent2Query	**0.929**	**0.743**	**0.563**	**0.686**	**0.136**	**0.168**	**0.191**	**0.174**	**0.883**	**0.780**	**0.620**	$\mathbf{0.719}^{+0.039}_{-0.040}$

6 Conclusion

We presented **Intent2Query**, a framework that bridges natural-language intent and executable formal queries over multi-agent trajectory data through a typed DSL with STL robustness semantics. Across 300 query intents, ten baselines spanning five method families, and three trajectory corpora, **Intent2Query** achieves the strongest retrieval and ranking performance. By combining typed, map-aware semantics with quantitative robustness, **Intent2Query** supports accurate, compositional, interpretable and rank-aware scene retrieval while transferring across datasets without modification.

AI USE STATEMENT

In this work, we did not use generative AI tools for tasks requiring mandatory disclosure, including research methodology design, hypothesis formulation, theoretical development, implementation of the proposed method, dataset construction, result interpretation, or qualitative analysis. We used generative AI tools only for tasks for which disclosure is recommended: grammar and readability editing, assistance with formatting and compiling LaTeX tables, and identifying potentially relevant baseline methods during literature search. All AI-assisted text, tables, and literature suggestions were manually reviewed by the authors; baseline methods and associated claims were independently verified against the original papers before inclusion. We take responsibility for the final content of this work, including all text, claims, citations, and artifacts produced with the aid of generative AI.

REFERENCES

Mohammad Albinhassan, Pranava Madhyastha, and Alessandra Russo. Sem-ctrl: Semantically controlled decoding. *arXiv preprint arXiv:2503.01804*, 2025.

ASAM. Asam openscenario® dsl. `https://www.asam.net/standards/detail/openscenario-dsl/`, 2026.

Artur Baranowski and Nico Hochgeschwender. Grammar-constrained neural semantic parsing with lr parsers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 1275–1279, 2021.

William Barbour, Matthew Bunting, Derek Gloudemans, Daniel B Work, and Jonathan Sprinkle. Persistent monitoring and analysis at a corridor scale with lidar. In *2025 IEEE 28th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 646–651. IEEE, 2025.

Ezio Bartocci, Alessio Gambi, Felix Gigler, Cristinel Mateis, and Dejan Ničković. Declarative scenario-based testing with roadlogic. *arXiv preprint arXiv:2603.09455*, 2026.

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*, pp. 11621–11631, 2020.

Cheng Chang, Dongpu Cao, Long Chen, Kui Su, Kuifeng Su, Yuelong Su, Fei-Yue Wang, Jue Wang, Ping Wang, Junqing Wei, et al. Metascenario: A framework for driving scenario data description, storage and indexing. *IEEE Transactions on Intelligent Vehicles*, 8(2):1156–1175, 2022.

Cheng Chang, Jingwei Ge, Jiazhe Guo, Zelin Guo, Binghong Jiang, and Li Li. Driving-rag: Driving scenarios embedding, search, and rag applications. *Automotive Innovation*, pp. 1–13, 2026.

Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8748–8757, 2019.

Benjamin Coifman and Lizhe Li. A critical evaluation of the next generation simulation (ngsim) vehicle trajectory dataset. *Transportation Research Part B: Methodological*, 105:362–377, 2017.

Matthias Cosler, Christopher Hahn, Daniel Mendoza, Frederik Schmitt, and Caroline Trippel. nl2spec: Interactively translating unstructured natural language to temporal logics with large language models. In *International Conference on Computer Aided Verification*, pp. 383–396. Springer, 2023.

Jyotirmoy V Deshmukh, Alexandre Donzé, Shromona Ghosh, Xiaoqing Jin, Garvit Juniwal, and Sanjit A Seshia. Robust online monitoring of signal temporal logic. *Formal Methods in System Design*, 51(1):5–30, 2017.

Alexandre Donzé. Breach, a toolbox for verification and parameter synthesis of hybrid systems. In *International Conference on Computer Aided Verification*, pp. 167–170. Springer, 2010.

Francesco Fuggitti and Tathagata Chakraborti. Nl2ltl–a python package for converting natural language (nl) instructions to linear temporal logic (ltl) formulas. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 16428–16430, 2023.

Mikhail Galkin, Jincheng Zhou, Bruno Ribeiro, Jian Tang, and Zhaocheng Zhu. A foundation model for zero-shot logical query reasoning. *Advances in Neural Information Processing Systems*, 37: 54137–54160, 2024.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. Grammar-constrained decoding for structured nlp tasks without finetuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10932–10952, 2023.

Xuan Guo, Shutong Yu, Jinxue Zhang, Huanyu Bi, Xiaohui Chen, and Junnan Liu. A natural language-based automatic identification system trajectory query approach using large language models. *ISPRS International Journal of Geo-Information*, 14(5):204, 2025.

Minjoo Ki, Daejung Kim, Kisung Kim, Seon Joo Kim, and Jinhan Lee. Carim: Caption-based autonomous driving scene retrieval via inclusive text matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22036–22045, 2025.

Shashank Kirtania, Priyanshu Gupta, and Arjun Radhakrishna. Logic-lm++: Multi-step refinement for symbolic formulations. In *Proceedings of the 2nd Workshop on Natural Language Reasoning and Structured Explanations (@ ACL 2024)*, pp. 56–63, 2024.

Donald E Knuth. On the translation of languages from left to right. *Information and control*, 8(6): 607–639, 1965.

Robert Krajewski, Julian Bock, Laurent Kloeker, and Lutz Eckstein. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. In *2018 21st international conference on intelligent transportation systems (ITSC)*, pp. 2118–2125. IEEE, 2018.

Stephen Mell, Favyen Bastani, Steve Zdancewic, and Osbert Bastani. Synthesizing trajectory queries from examples. In *International Conference on Computer Aided Verification*, pp. 459–484. Springer, 2023.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 3806–3824, 2023.

Kanghee Park, Timothy Zhou, and Loris D'Antoni. Flexible and efficient grammar-constrained decoding. *arXiv preprint arXiv:2502.05111*, 2025.

Irina Saparina and Mirella Lapata. Ambrosia: A benchmark for parsing ambiguous questions into database queries. *Advances in Neural Information Processing Systems*, 37:90600–90628, 2024.

Mohamed Sayed Mohamed Bakli E, Mahmoud Sakr, and Esteban Zimanyi. Distributed spatiotemporal trajectory query processing in sql. 2020.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2446–2454, 2020.

Tao Tang, Dafeng Wei, Zhengyu Jia, Tian Gao, Changwei Cai, Chengkai Hou, Peng Jia, Kun Zhan, Haiyang Sun, Fan JingChen, et al. Bev-tsr: Text-scene retrieval in bev space for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7275–7283, 2025.

Eric Zhan, Jagjeet Singh, Yisong Yue, and Andrew Hartnett. The argoverse trajectory retrieval benchmark.

Wei Zhan, Liting Sun, Di Wang, Haojie Shi, Aubrey Clausse, Maximilian Naumann, Julius Kummerle, Hendrik Konigshof, Christoph Stiller, Arnaud de La Fortelle, et al. Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. *arXiv preprint arXiv:1910.03088*, 2019.

A Full Signal and Predicate Definitions

A.1 Notations

Tables 6 and 7 provide a comprehensive reference to all notations used throughout the paper, covering the scene model, query semantics, and the DSL's signals, predicates, and event macros.

Table 6: Core notation summary for Intent2Query.

Symbol	Description
Scene model	
$\mathcal{D} = \{\sigma_1, \ldots, \sigma_N\}$	Corpus of multi-agent trajectory scenes
$\sigma = (\mathcal{A}, \mathcal{Z}, \tau, \mathbf{y})$	A scene (finite multi-agent trajectory trace)
$\mathcal{A} = \{a_1, \ldots, a_M\}$	Finite set of agents in a scene
$\mathcal{Z}$	Set of map zones in a scene
$\tau = (t_0, \ldots, t_T)$	Discrete time grid
$\mathbf{y}(a, t)$	Continuous state (position, velocity, ...) of agent a at time t
$\mathcal{T} = \{\text{VEHICLE}, \text{PEDESTRIAN}, \text{CYCLIST}, \ldots\}$	Set of agent semantic types
$\text{type}(a) \in \mathcal{T}$	Semantic type of agent a
$\text{geom}(z) \subset \mathbb{R}^2$	Polygonal geometry of zone z
Natural language to executable queries	
ℓ	Natural language (NL) intent
p	Semantic parser, $q = p(\ell)$
$\mathcal{L} = (\mathcal{T}, \mathcal{X}, \mathcal{S}, \mathcal{R}, \mathcal{M}, \mathcal{E}, \Phi, \rho)$	Spatiotemporal multi-agent query DSL
$q = \exists \mathcal{X}[\mathcal{D}].\, \phi$	DSL query: typed variables and a temporal formula
$\mathcal{X} = \{x_1, \ldots, x_n\}$	Set of typed agent variables
$x : \tau_x$ with $\tau_x \in \mathcal{T}$	A typed variable declaration
$\nu : \mathcal{X} \to \mathcal{A}$	Variable assignment binding query variables to agents
ϕ	Temporal logic formula over typed variables
Semantics and retrieval	
$(\sigma, \nu, t) \models \phi$	Boolean satisfaction relation
$\rho(\sigma, \nu, \phi, t) \in \mathbb{R} \cup \{\pm\infty\}$	Quantitative robustness score
$\kappa > 0$	Robustness constant for Boolean predicates
$\text{Score}(\sigma, q) = \max_\nu \max_{t \in \tau} \rho(\sigma, \nu, \phi, t)$	Scene-level robustness score
$\sigma \models q \iff \text{Score}(\sigma, q) > 0$	Scene-level satisfaction
$\mathcal{O}_r(x) = \{\sigma \in \mathcal{D} \mid Score(\sigma, q) > 0\}$	Retrieval objective
Grammar-constrained semantic parsing	
f_θ	Neural parser with parameters θ
$\mathcal{G}$	DSL grammar used to constrain decoding
$\hat{z} = f_\theta(\ell; \mathcal{G})$	Grammar-conditioned structured output
$\mathcal{N}$	Normalization step
$\mathcal{R}^D$	Deterministic repair step
$\mathcal{V}$	Grammar- and DSL-level validation step
$z^\star = \mathcal{V}(\mathcal{R}_D(\mathcal{N}(\hat{z})))$	Final validated, executable program

This section provides the complete formal definitions of continuous signals $\mathcal{S}$, geometric predicates $\mathcal{R}$, map-aware predicates $\mathcal{M}$, and event macros $\mathcal{E}$ used in the DSL.

A.2 Continuous Signals $\mathcal{S}$

We define continuous signals over agents and time. Unary signals include

$$pos(x, t) \in \mathbb{R}^2, \quad vel(x, t) \in \mathbb{R}^2,$$
$$heading(x, t) \in \mathbb{R}, \quad speed(x, t) = \|vel(x, t)\|_2 \in \mathbb{R},$$

Table 7: Continuous signals, predicates, and event macros in the Intent2Query DSL.

Symbol	Description
Unary continuous signals (over agent x at time t)	
$\mathrm{pos}(x,t) \in \mathbb{R}^2$	Position
$\mathrm{vel}(x,t) \in \mathbb{R}^2$	Velocity
$\mathrm{heading}(x,t) \in \mathbb{R}$	Heading angle (direction)
$\mathrm{speed}(x,t) = \|\mathrm{vel}(x,t)\|_2 \in \mathbb{R}$	Speed (Euclidean norm of velocity; magnitude)
$\mathrm{accel}(x,t) \in \mathbb{R}$	Acceleration (finite-difference of speed)
$\mathrm{yaw_rate}(x,t) \in \mathbb{R}$	Yaw rate (finite-difference of unwrapped heading)
Pairwise continuous signals (over x_i, x_j at time t)	
$\mathrm{distance}(x_i,x_j,t) \in \mathbb{R}$	Euclidean distance between agents
$\mathrm{rel_vel}(x_i,x_j,t) \in \mathbb{R}^2$	Relative velocity
$\mathrm{closing_speed}(x_i,x_j,t) \in \mathbb{R}$	Closing speed (non-negative projection along baseline)
$\mathrm{ttc}(x_i,x_j,t) \in \mathbb{R} \cup \{+\infty\}$	Time-to-collision
Geometric relation predicates $\mathcal{R}$	
$\mathrm{near}(x_i,x_j,t)$	Distance below threshold δ_{near}
$\mathrm{approaching}(x_i,x_j,t)$	Closing speed at least δ_{app}
$\mathrm{conflict_proximity}(x_i,x_j,t)$	Distance $\le \delta_d$ and TTC $\le \delta_{\mathrm{ttc}}$
Map-aware predicates $\mathcal{M}$	
$\mathrm{inside}(x,z,t) := [\mathrm{pos}(x,t) \in \mathrm{geom}(z)]$	Agent x lies inside zone z
ahead_of, behind, left_of, right_of	Directional relations in heading-aligned frame
$\mathrm{in_crosswalk}(x,t)$	$\exists z$ with $\mathrm{type}(z) =$ xwalk and $\mathrm{inside}(x,z,t)$
$\mathrm{in_sidewalk}(x,t)$	$\exists z$ with $\mathrm{type}(z) =$ sidewalk and $\mathrm{inside}(x,z,t)$
$\mathrm{in_pedwait}(x,t)$	$\exists z$ with $\mathrm{type}(z) =$ pedwait and $\mathrm{inside}(x,z,t)$
$\mathrm{on_stopbar}(x,t)$	$\exists z$ with $\mathrm{type}(z) =$ stopbar and $\mathrm{inside}(x,z,t)$
Interaction event macros $\mathcal{E}$	
$\mathrm{moving}(x)$, $\mathrm{stop}(x)$	Speed above ϵ_m / below ϵ_s
$\mathrm{decelerating}(x)$, $\mathrm{hard_brake}(x)$	Acceleration $< -\alpha$ / $\le -\beta$
$\mathrm{turn_left}(x)$, $\mathrm{turn_right}(x)$	Yaw rate exceeds θ_L / falls below $-\theta_R$ within $[0,\Delta]$
$\mathrm{near_miss}(x_i,x_j)$	Distance $\le d_{\mathrm{nm}}$ and TTC $\le \tau_{\mathrm{nm}}$ within $[0,\Delta]$
$\mathrm{cross_from_sidewalk}(x)$	$\mathrm{in_sidewalk}(x)\,\mathcal{U}_{[0,\Delta]}\,\mathrm{in_crosswalk}(x)$
$\mathrm{stop_at_stopbar}(x)$	$\Diamond_{[0,\Delta]}\big(\mathrm{on_stopbar}(x) \wedge \mathrm{stop}(x)\big)$
$\mathrm{yield_to}(x_i,x_j)$	Multi-agent yielding macro (Appendix A.5)

$$accel(x,t) \approx \frac{speed(x,t+\Delta t) - speed(x,t)}{\Delta t} \in \mathbb{R},$$

$$yaw_rate(x,t) \approx \frac{\mathrm{unwrap}(heading(x,t+\Delta t) - heading(x,t))}{\Delta t} \in \mathbb{R}.$$

Pairwise signals include

$$distance(x_i,x_j,t) = \|pos(x_i,t) - pos(x_j,t)\|_2,$$

$$rel_vel(x_i,x_j,t) = vel(x_i,t) - vel(x_j,t),$$

$$closing_speed(x_i,x_j,t) = \max\left(0, \frac{(pos(x_j,t) - pos(x_i,t))^\top (vel(x_i,t) - vel(x_j,t))}{\|pos(x_j,t) - pos(x_i,t)\|_2 + \epsilon}\right),$$

$$ttc(x_i, x_j, t) = \begin{cases} \dfrac{distance(x_i, x_j, t)}{closing_speed(x_i, x_j, t)} & \text{if } closing_speed(x_i, x_j, t) > 0, \\ +\infty & \text{otherwise.} \end{cases}$$

A.3 Geometric Relation Predicates $\mathcal{R}$

Let $\delta_{near}, \delta_{app}, \delta_d, \delta_{ttc} > 0$ be thresholds. We define

$$near(x, y, t) := [distance(x, y, t) \le \delta_{near}],$$
$$approaching(x, y, t) := [closing_speed(x, y, t) \ge \delta_{app}],$$
$$conflict_proximity(x, y, t) := [distance(x, y, t) \le \delta_d \wedge ttc(x, y, t) \le \delta_{ttc}].$$

A.4 Map-aware Predicates $\mathcal{M}$

Let Z_i denote the set of polygonal zones for intersection i. For a zone $z \in Z_i$ with geometry $geom(z) \subset \mathbb{R}^2$, define

$$inside(x, z, t) := [pos(x, t) \in geom(z)].$$

We define semantic predicates:

$$in_crosswalk(x, t) := \exists z \in Z_i \text{ s.t. } type(z) = \texttt{xwalk} \wedge inside(x, z, t),$$
$$in_sidewalk(x, t) := \exists z \in Z_i \text{ s.t. } type(z) = \texttt{sidewalk} \wedge inside(x, z, t),$$
$$in_pedwait(x, t) := \exists z \in Z_i \text{ s.t. } type(z) = \texttt{pedwait} \wedge inside(x, z, t),$$
$$on_stopbar(x, t) := \exists z \in Z_i \text{ s.t. } type(z) = \texttt{stopbar} \wedge inside(x, z, t).$$

A.5 Interaction Event Macros $\mathcal{E}$

Unary motion events:

$$moving(x) := [speed(x, t) > \epsilon_m], \quad stop(x) := [speed(x, t) \le \epsilon_s],$$
$$decelerating(x) := [accel(x, t) < -\alpha], \quad hard_brake(x) := [accel(x, t) \le -\beta].$$

Turning events:

$$turn_left(x) := \Diamond_{[0,\Delta]}(yaw_rate(x) \ge \theta_L), \quad turn_right(x) := \Diamond_{[0,\Delta]}(yaw_rate(x) \le -\theta_R).$$

Pairwise safety events:

$$near_miss(x_i, x_j) := \Diamond_{[0,\Delta]}(distance(x_i, x_j) \le d_{nm} \wedge ttc(x_i, x_j) \le \tau_{nm}).$$

Map-aware events:

$$cross_from_sidewalk(x) := in_sidewalk(x)\, \mathcal{U}_{[0,\Delta]}\, in_crosswalk(x),$$
$$stop_at_stopbar(x) := \Diamond_{[0,\Delta]}(on_stopbar(x) \wedge stop(x)).$$

Multi-agent interaction:

$$yield_to(x_i, x_j) := \Diamond_{[0,\Delta]}\big(\Diamond_{[0,\delta_1]} on_stopbar(x_i) \ \wedge \ (in_pedwait(x_j)\, \mathcal{U}_{[0,\delta_2]}\, in_crosswalk(x_i))$$
$$\wedge \ \Diamond_{[0,\delta_3]} hard_brake(x_i) \ \wedge \ \Box_{[0,\delta_4]} \neg collision(x_i, x_j)\big).$$

B Benchmark Query Mappings

We list the 30 expert-authored gold NL–DSL mappings, organized into six semantic categories with five examples per category. These are the intents for which human-written gold DSL exists and on which the parsing metrics of Table 2 are computed; the full 300-intent benchmark extends each category to 50. The examples include paraphrases that vary in lexical choice, syntactic form, and explicitness (for instance Q1 and Q2, or Q11 and Q12, map to the same program from different surface forms). In all examples, temporal horizons and thresholds are configurable; we write them symbolically for brevity.

Difficulty criterion. Each intent is tagged EASY, MEDIUM, or HARD. Difficulty is a property of the target DSL program, not of the English sentence, and is assigned along three axes:

- **Binding complexity** — how many typed variables the program quantifies over, and whether their types must be distinguished to avoid over-retrieval.
- **Predicate complexity** — whether a predicate is evaluated pointwise from a *directly measured* signal (speed, acceleration, pairwise distance, zone membership), is *derived* from relative motion or geometry (closing velocity, relative heading), or is *predictive* and requires extrapolating future state (time-to-collision, conflict).
- **Compositional depth** — how many of the three composition dimensions (temporal operators, pairwise relations, map grounding) the program combines.

The three levels are then defined as follows:

- **Easy Query**: a single directly measured predicate over one or two agents, with at most one temporal operator and no composition across dimensions.
- **Medium Query**: elevated on exactly one axis — a sequential composition ($\mathcal{U}$) of two simple predicates, *or* one derived relational predicate, *or* a conjunction of zone membership with a kinematic state.
- **Hard Query**: elevated on two or more axes, or requiring a predictive predicate. This covers orientation-dependent relations, time-to-collision reasoning, composition across two or more of {temporal, relational, map}, and event macros that expand into multi-agent multi-phase patterns.

Unary motion.

Q1 [EASY]**:** "A vehicle brakes hard."
$\exists x : \texttt{Vehicle}.\ \Diamond_{[0,T]} hard_brake(x).$

Q2 [EASY]**:** "A car makes a sharp stop."
$\exists x : \texttt{Vehicle}.\ \Diamond_{[0,T]} hard_brake(x).$

Q3 [EASY]**:** "A vehicle turns left."
$\exists x : \texttt{Vehicle}.\ \Diamond_{[0,T]} turn_left(x).$

Q4 [MEDIUM]**:** "A vehicle is moving before it stops."
$\exists x : \texttt{Vehicle}.\ moving(x)\ \mathcal{U}_{[0,T]}\ stop(x).$

Q5 [MEDIUM]**:** "A vehicle decelerates to a stop."
$\exists x : \texttt{Vehicle}.\ \Diamond_{[0,T]}(decelerating(x)\ \mathcal{U}_{[0,T]}\ stop(x)).$

Pairwise geometry.

Q6 [EASY]**:** "A vehicle is near a pedestrian."
$\exists x_i : \texttt{Vehicle}, x_j : \texttt{Pedestrian}.\ \Diamond_{[0,T]} near(x_i, x_j).$

Q7 [EASY]**:** "A pedestrian is close to a car."
$\exists x_i : \texttt{Pedestrian}, x_j : \texttt{Vehicle}.\ \Diamond_{[0,T]} near(x_i, x_j).$

Q8 [MEDIUM]**:** "A vehicle approaches another vehicle."
$\exists x_i : \texttt{Vehicle}, x_j : \texttt{Vehicle}.\ \Diamond_{[0,T]} approaching(x_i, x_j).$

Q9 [HARD]**:** "A cyclist is beside a vehicle on the left."
$\exists x_i : \texttt{Cyclist}, x_j : \texttt{Vehicle}.\ \Diamond_{[0,T]} left_of(x_i, x_j).$

Q10 [MEDIUM]**:** "A car is behind another car."
$\exists x_i : \texttt{Vehicle}, x_j : \texttt{Vehicle}.\ \Diamond_{[0,T]} behind(x_i, x_j).$

Safety oriented.

Q11 [MEDIUM]**:** "A vehicle has a near miss with a pedestrian."
$\exists x_i : \texttt{Vehicle}, x_j : \texttt{Pedestrian}.\ \Diamond_{[0,T]} near_miss(x_i, x_j).$

Q12 [MEDIUM]**:** "A car comes dangerously close to a cyclist."
$\exists x_i : \texttt{Vehicle}, x_j : \texttt{Cyclist}.\ \Diamond_{[0,T]} near_miss(x_i, x_j).$

Q13 [HARD]**:** "Two vehicles enter a low-TTC conflict."
$\exists x_i : \texttt{Vehicle}, x_j : \texttt{Vehicle}.\ \Diamond_{[0,T]} conflict_proximity(x_i, x_j).$

Q14 [HARD]**:** "A pedestrian and a vehicle are in a collision-risk situation."
$\exists x_i : \texttt{Pedestrian}, x_j : \texttt{Vehicle}.\ \Diamond_{[0,T]} conflict_proximity(x_i, x_j).$

Q15 [HARD]**:** "A vehicle hard-brakes while close to a pedestrian."
$\exists x_i : \texttt{Vehicle}, x_j : \texttt{Pedestrian}.\ \Diamond_{[0,T]} (hard_brake(x_i) \wedge near(x_i, x_j)).$

Temporal composition.

Q16 [MEDIUM]**:** "A vehicle turns left and then stops."
$\exists x : \texttt{Vehicle}.\ turn_left(x)\ \mathcal{U}_{[0,T]}\ stop(x).$

Q17 [MEDIUM]**:** "A car moves, then brakes hard."
$\exists x : \texttt{Vehicle}.\ moving(x)\ \mathcal{U}_{[0,T]}\ hard_brake(x).$

Q18 [HARD]**:** "A vehicle approaches a pedestrian before stopping."
$\exists x_i : \texttt{Vehicle}, x_j : \texttt{Pedestrian}.\ approaching(x_i, x_j)\ \mathcal{U}_{[0,T]}\ stop(x_i).$

Q19 [EASY]**:** "A vehicle remains stopped for several seconds."
$\exists x : \texttt{Vehicle}.\ \Diamond_{[0,T]} \Box_{[0,\Delta]} stop(x).$

Q20 [HARD]**:** "A pedestrian is near a vehicle while the vehicle is braking."
$\exists x_i : \texttt{Pedestrian}, x_j : \texttt{Vehicle}.\ \Diamond_{[0,T]} (near(x_i, x_j) \wedge hard_brake(x_j)).$

Map-aware zone queries.

Q21 [HARD]**:** "A pedestrian leaves the waiting zone and enters the crosswalk."
$\exists x : \texttt{Pedestrian}.\ in_pedwait(x)\ \mathcal{U}_{[0,T]}\ in_crosswalk(x).$

Q22 [HARD]**:** "A pedestrian crosses from the sidewalk into the crosswalk."
$\exists x : \texttt{Pedestrian}.\ cross_from_sidewalk(x).$

Q23 [MEDIUM]**:** "A vehicle stops at the stop bar."
$\exists x : \texttt{Vehicle}.\ \Diamond_{[0,T]} (on_stopbar(x) \wedge stop(x)).$

Q24 [EASY]**:** "A pedestrian is inside the crosswalk."
$\exists x : \texttt{Pedestrian}.\ \Diamond_{[0,T]} in_crosswalk(x).$

Q25 [HARD]**:** "A vehicle reaches the stop bar before braking."
$\exists x : \texttt{Vehicle}.\ \Diamond_{[0,T]} on_stopbar(x)\ \mathcal{U}_{[0,T]}\ hard_brake(x).$

Interaction event macros.

Q26 [HARD]**:** "A vehicle yields to a pedestrian."
$\exists x_i : \texttt{Vehicle}, x_j : \texttt{Pedestrian}.\ yield_to(x_i, x_j).$

Q27 [HARD]**:** "A car slows for a pedestrian entering the crosswalk."
$\exists x_i : \texttt{Vehicle}, x_j : \texttt{Pedestrian}.\ yield_to(x_i, x_j).$

Q28 [HARD]**:** "A pedestrian crosses while a vehicle stops at the stop bar."
$\exists x_i : \texttt{Pedestrian}, x_j : \texttt{Vehicle}.\ \Diamond_{[0,T]}(in_crosswalk(x_i)$
$\wedge\ stop_at_stopbar(x_j)).$

Q29 [HARD]**:** "A vehicle gives way as a pedestrian moves from waiting zone into crosswalk."
$\exists x_i : \texttt{Vehicle}, x_j : \texttt{Pedestrian}.\ yield_to(x_i, x_j).$

Q30 [HARD]**:** "A pedestrian and a vehicle interact at the crosswalk without collision."
$\exists x_i : \texttt{Pedestrian}, x_j : \texttt{Vehicle}.\ \Diamond_{[0,T]}(in_crosswalk(x_i) \wedge near(x_i, x_j)$
$\wedge\ \neg collision(x_i, x_j)).$

Difficulty by category and corpus. Difficulty is not uniform across categories, and the applicable subsets of the two external corpora are not uniform either. Table 8 gives the full category × difficulty × corpus breakdown of the 300-intent benchmark, refining the per-corpus totals of Table 1. Three points follow from it.

Table 8: Query category × difficulty × corpus for the full 300-intent benchmark.

	LADDMS (300)				**NGSIM** (86)				**INTERACTION** (200)			
Category	Easy	Med	Hard	All	Easy	Med	Hard	All	Easy	Med	Hard	All
Unary motion	28	21	1	50	18	15	1	34	28	21	1	50
Pairwise geometry	13	21	16	50	4	4	4	12	13	21	16	50
Safety oriented	0	13	37	50	0	2	14	16	0	13	37	50
Temporal composition	1	28	21	50	1	16	7	24	1	28	21	50
Map-aware zones	7	18	25	50	—	—	—	—	—	—	—	—
Interaction macros	0	4	46	50	—	—	—	—	—	—	—	—
Total	**49**	**105**	**146**	**300**	**23**	**37**	**26**	**86**	**42**	**83**	**75**	**200**

C Retrieval Performance by Query Difficulty and Dataset

Table 9 provides the full retrieval breakdown by method, dataset, and query difficulty, complementing the aggregate results in Tables 3 and 4. Table 10 compares **Intent2Query** against the strongest baseline in each dataset–difficulty cell using a paired bootstrap over queries (B=10,000, seed 42). **Intent2Query** has the highest F1 point estimate in all nine cells, with statistically significant gains in seven; the two non-significant cases occur at the Easy and Hard extremes of INTERACTION, where the margin over NL2LTL-Rule is not distinguishable at the available subset sizes. On NGSIM, the gains are statistically reliable but small in magnitude, consistent with its omission of the two hardest query categories.

Figures 2–4 plot all ten baselines and **Intent2Query** on the three corpora. Together they show the precision/recall trade-off that the aggregate F1 column of Table 4 compresses. **Intent2Query** has the highest Precision and the highest F1 on all three corpora. NL2LTL-Rule has the highest Recall on all three, which is a consequence of untyped binding rather than a strength: it retrieves nearly every relevant scene together with a large number of irrelevant ones, so its precision is roughly half that of **Intent2Query** on LADDMS (0.330 versus 0.636) and its F1 is correspondingly lower (0.396 versus 0.682). Reading the three figures together is the clearest statement of why F1, not Recall, is the appropriate headline metric for this task.

Table 9: Retrieval performance by method, dataset, and query difficulty.

	Easy			Medium			Hard			Overall		
Method	P	R	F1	P	R	F1	P	R	F1	P	R	F1
LADDMS (N_{easy}=49, N_{med}=105, N_{hard}=146)												
Few-Shot (k=1)	0.486	0.719	0.566	0.403	0.657	0.491	0.252	0.463	0.323	0.343	0.573	0.422
Few-Shot (k=3)	0.504	0.756	0.591	0.420	0.692	0.514	0.271	0.499	0.348	0.361	0.608	0.446
Few-Shot (k=10)	0.500	0.749	0.586	0.412	0.677	0.503	0.263	0.487	0.337	0.354	0.596	0.436
LLM-as-Judge	0.438	0.182	0.220	0.128	0.039	0.047	0.033	0.002	0.003	0.132	0.044	0.054
Embedding Retrieval	0.577	0.712	0.567	0.462	0.799	0.511	0.243	0.762	0.297	0.374	0.767	0.416
Embed + Rerank	0.635	0.245	0.237	0.502	0.213	0.208	0.241	0.204	0.167	0.397	0.214	0.193
QBE-DT (k=10)	0.600	0.765	0.623	0.574	0.802	0.603	0.334	0.715	0.377	0.461	0.754	0.496
QBE-SVM (k=10)	0.485	0.786	0.515	0.466	0.812	0.505	0.254	0.758	0.300	0.366	0.781	0.407
QBE-STL (k=10)	0.524	0.806	0.551	0.494	0.798	0.517	0.285	0.773	0.335	0.397	0.787	0.434
NL2LTL-Rule	0.539	0.894	0.581	0.414	0.967	0.496	0.199	0.950	0.262	0.330	0.947	0.396
Intent2Query	**0.838**	**0.936**	**0.857**	**0.704**	**0.892**	**0.746**	**0.519**	**0.691**	**0.577**	**0.636**	**0.802**	**0.682**
NGSIM (N_{easy}=23, N_{med}=37, N_{hard}=26)												
Few-Shot (k=1)	0.925	0.461	0.515	0.893	0.548	0.569	0.924	0.548	0.588	0.911	0.525	0.560
Few-Shot (k=3)	0.949	0.878	0.903	0.909	0.696	0.756	0.937	0.658	0.748	0.928	0.733	0.793
Few-Shot (k=10)	0.946	0.898	0.908	0.921	0.859	0.879	0.941	0.773	0.812	0.934	0.843	0.866
LLM-as-Judge	0.607	0.297	0.355	0.467	0.193	0.202	0.285	0.017	0.031	0.449	0.168	0.191
Embedding Retrieval	0.939	0.504	0.648	0.929	0.484	0.636	0.962	0.507	0.664	0.942	0.497	0.648
Embed + Rerank	0.944	0.292	0.441	0.931	0.279	0.429	0.962	0.291	0.447	0.944	0.286	0.437
QBE-DT (k=10)	0.129	0.123	0.126	0.214	0.175	0.191	0.376	0.295	0.328	0.240	0.197	0.215
QBE-SVM (k=10)	0.089	0.086	0.083	0.213	0.168	0.187	0.377	0.285	0.322	0.229	0.181	0.200
QBE-STL (k=10)	0.130	0.130	0.130	0.206	0.205	0.205	0.375	0.364	0.369	0.237	0.233	0.235
NL2LTL-Rule	0.936	1.000	0.956	0.914	0.972	0.929	0.955	1.000	0.976	0.932	0.988	0.951
Intent2Query	**0.995**	**0.997**	**0.996**	**0.920**	**0.972**	**0.932**	**0.966**	**0.999**	**0.981**	**0.954**	**0.987**	**0.964**
INTERACTION (N_{easy}=42, N_{med}=83, N_{hard}=75)												
Few-Shot (k=1)	0.444	0.387	0.277	0.329	0.484	0.293	0.178	0.447	0.189	0.297	0.450	0.251
Few-Shot (k=3)	0.450	0.391	0.378	0.283	0.330	0.268	0.181	0.302	0.202	0.279	0.333	0.266
Few-Shot (k=10)	0.429	0.366	0.384	0.261	0.246	0.248	0.159	0.154	0.154	0.258	0.237	0.241
LLM-as-Judge	0.272	0.200	0.199	0.195	0.123	0.118	0.047	0.038	0.027	0.156	0.107	0.101
Embedding Retrieval	0.467	0.420	0.401	0.316	0.406	0.300	0.175	0.301	0.178	0.295	0.369	0.275
Embed + Rerank	0.499	0.462	0.437	0.339	0.414	0.327	0.197	0.354	0.208	0.319	0.402	0.305
QBE-DT (k=10)	0.349	0.363	0.341	0.236	0.257	0.240	0.139	0.162	0.140	0.223	0.244	0.224
QBE-SVM (k=10)	0.306	0.346	0.307	0.235	0.253	0.233	0.133	0.165	0.137	0.211	0.240	0.213
QBE-STL (k=10)	0.369	0.410	0.377	0.257	0.284	0.262	0.148	0.199	0.162	0.240	0.278	0.249
NL2LTL-Rule	0.480	0.736	0.546	0.307	0.660	0.376	0.181	0.505	0.235	0.296	0.618	0.359
Intent2Query	**0.619**	**0.635**	**0.587**	**0.382**	**0.574**	**0.412**	**0.259**	**0.364**	**0.250**	**0.386**	**0.508**	**0.388**

Table 10: Paired bootstrap comparison of **Intent2Query** against the strongest baseline.

Dataset	Difficulty	N	**Intent2Query** F1	Best baseline	ΔF1	95% CI	Sig.
LADDMS	Easy	49	0.857	QBE-DT (0.623)	+0.234	[+0.144, +0.330]	✓
	Medium	105	0.746	QBE-DT (0.603)	+0.143	[+0.081, +0.205]	✓
	Hard	146	0.577	QBE-DT (0.377)	+0.200	[+0.143, +0.258]	✓
NGSIM	Easy	23	0.996	NL2LTL-Rule (0.956)	+0.040	[+0.003, +0.096]	✓
	Medium	37	0.932	NL2LTL-Rule (0.929)	+0.003	[+0.001, +0.004]	✓
	Hard	26	0.981	NL2LTL-Rule (0.976)	+0.005	[+0.002, +0.008]	✓
INTERACTION	Easy	42	0.587	NL2LTL-Rule (0.546)	+0.041	[-0.040, +0.123]	–
	Medium	83	0.412	NL2LTL-Rule (0.376)	+0.036	[+0.002, +0.068]	✓
	Hard	75	0.250	NL2LTL-Rule (0.235)	+0.015	[-0.019, +0.054]	–

D Ablation Studies

Ablation setting. To quantify the contribution of each DSL component and executor design, we perform controlled ablations on the **Intent2Query** system, removing or simplifying one component at a time: interaction event macros, map-aware predicates, pairwise geometric predicates, temporal operators, typed bindings, and quantitative ranking. We also compare full robustness-based ranking against Boolean-only matching. All eight configurations are evaluated on all 300 queries over the full 940-scene corpus, and the ablated configurations execute human-written gold DSL so that retrieval behavior is isolated from parsing errors.

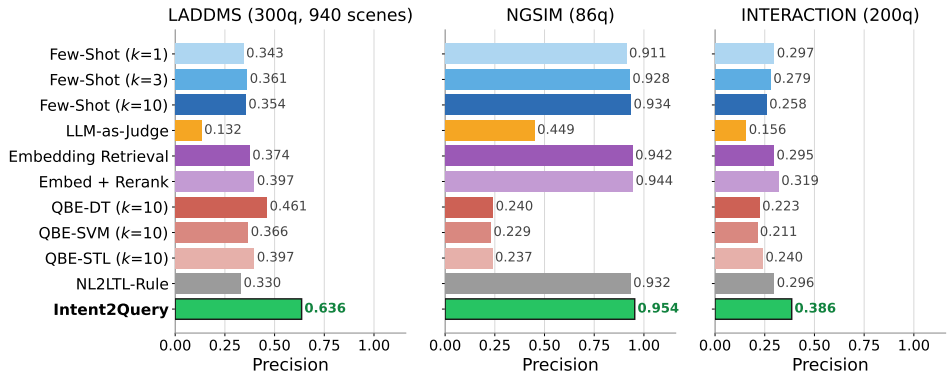


Figure 2: Cross-dataset Precision for all ten baselines and **Intent2Query**.

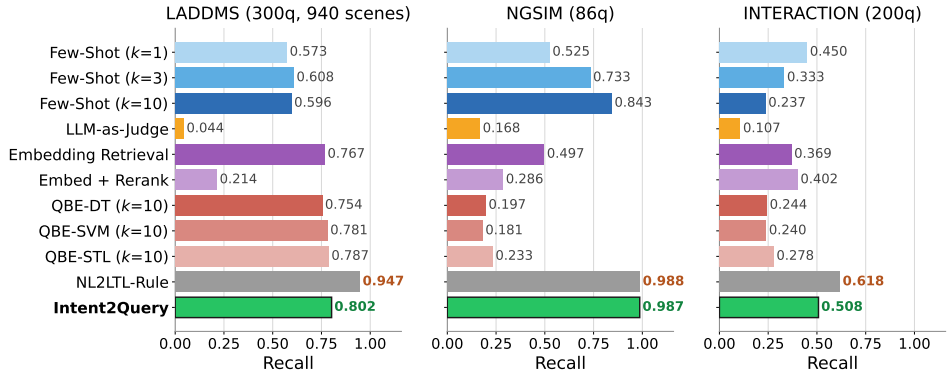


Figure 3: Cross-dataset Recall for all ten baselines and **Intent2Query**.

Result discussion. Table 11 and Figure 5 show that the DSL components contribute along three complementary dimensions: retrieval accuracy, compositional expressiveness, and ranking/auditability.

Retrieval-critical: typed bindings and map grounding. Typed bindings and map predicates provide the largest retrieval gains, validating the need for domain-specific semantics beyond generic formal execution. Removing type constraints causes the largest drop in both F1 and nDCG@10, mainly through over-retrieval: without explicit types, a query such as "a pedestrian crosses" can also bind vehicles or cyclists. Removing map predicates similarly harms queries involving regions such as crosswalks and stop bars. These results show that typed entities and map grounding are essential for precise scene retrieval.

Expressiveness: relational and temporal composition. Pairwise geometry and temporal operators enable compositional queries involving multi-agent relations and event sequencing. Their impact is concentrated on harder queries, with pairwise predicates affecting both retrieval and ranking. The smaller aggregate effect of temporal operators reflects their narrower use: simple temporal intents can often be expressed through conjunctions, whereas genuinely sequential patterns require explicit temporal composition.

Ranking and explainability: quantitative robustness and macros. Quantitative STL robustness primarily improves ranking rather than set retrieval. Replacing it with Boolean matching leaves F1 nearly unchanged because the same scenes are retrieved, but lowers nDCG@10 by removing continuous match-strength information. Macros likewise have limited effect on F1 because they expand to equivalent DSL programs, but they make generated queries concise, human-readable, and auditable; for example, `hard_brake(v)` is easier to verify than its expanded kinematic and temporal constraints.

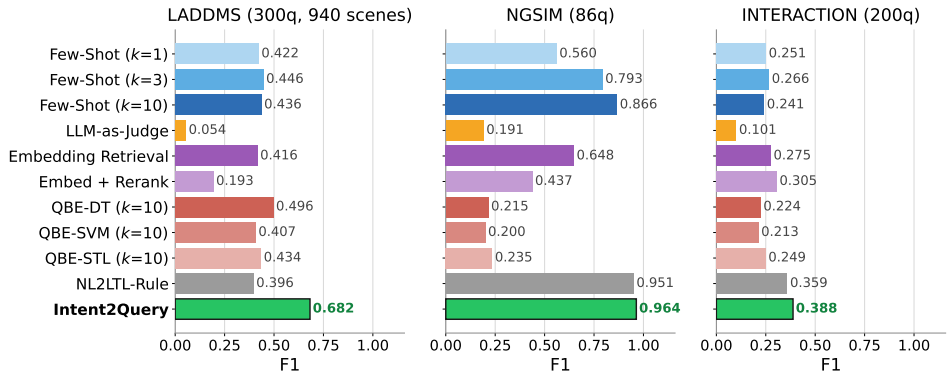


Figure 4: Cross-dataset F1 for all ten baselines and **Intent2Query**.

Table 11: Component ablation over full queries on LADDMS.

Configuration	P	R	F1	ΔF1	95% CI (F1)	nDCG@10	ΔnDCG
Full (Intent2Query)	**0.636**	0.802	**0.682**	—	[0.647, 0.715]	**0.719**	—
No types	0.271	0.661	0.309	**−0.373**	[0.266, 0.352]	0.348	**−0.371**
No map	0.528	**0.824**	0.557	**−0.125**	[0.515, 0.598]	0.616	**−0.103**
No pairwise	0.611	0.812	0.636	**−0.046**	[0.595, 0.675]	0.647	**−0.072**
No temporal	0.684	0.693	0.653	−0.029	[0.609, 0.694]	0.713	−0.006
No macros	0.696	0.759	0.676	−0.006	[0.636, 0.715]	0.714	−0.005
Boolean-only	0.701	0.765	0.682	0.000	[0.643, 0.720]	0.690	**−0.029**
No ranking	0.701	0.766	0.682	0.000	[0.643, 0.720]	0.690	**−0.029**

E Runtime and Scalability

Table 12 shows that runtime is dominated by execution rather than parsing. STL evaluation scales linearly with the number of scenes, but binding enumeration grows rapidly with agents per scene, producing the long-tail latency and most execution timeouts for queries with multiple pairwise variables. Because scenes are independent, corpus-level execution is readily parallelizable, while agent-level binding remains the main scalability bottleneck. Type- and map-aware indexing could substantially prune invalid bindings and is a natural direction for future optimization. At its current operating point, **Intent2Query** is therefore well suited to batch corpus mining and interactive search over pre-filtered candidates, but not yet unindexed million-scene retrieval.

Table 12: Runtime analysis.

Stage	Component	Latency
Parsing (per query)		
	Rule-based extractor	$<$1 ms
	GCD (Qwen2.5-0.5B, CPU)	11.7 s
	GCD (Qwen2.5-7B, CPU)	13.3 s
STL evaluation (per query)		
	Median	20.1 s
	Mean	68.9 s
	95th percentile	300 s
End-to-end throughput		
	Per query (parse + execute)	$\sim$69 s

F Human-in-the-loop Annotation Tool

Figure 6 shows the interface used for expert verification of query–scene relevance labels. The tool visualizes the trajectories of multiple agents in each scene and presents the corresponding query to the annotator, who assigns one of four relevance levels: exact, partial, negative, or unsure. Annotators

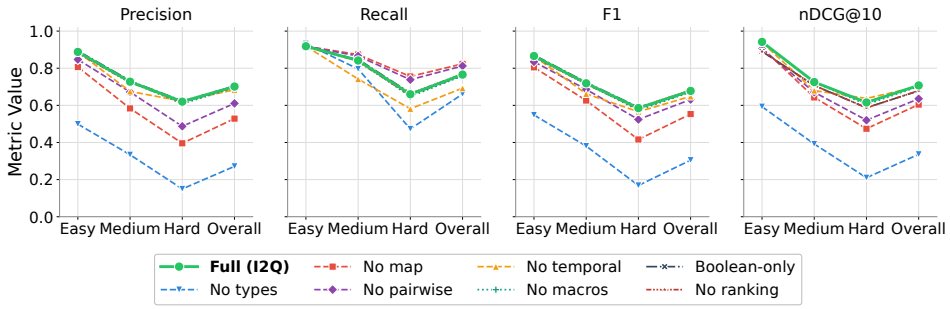


Figure 5: Impact of removing each DSL component, across difficulty levels on full queries.

may also provide optional witness annotations, including the relevant agent bindings and temporal intervals. These annotations are used to construct the graded relevance ground truth for the LADDMS ranking evaluation.

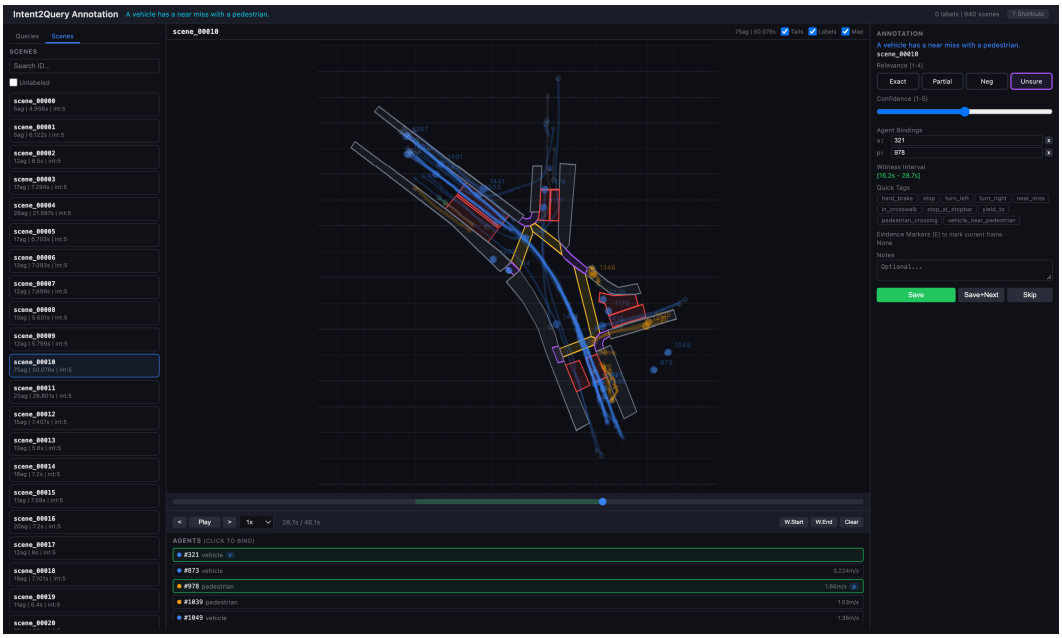


Figure 6: Human-in-the-loop annotation interface.