ALERT: A SELF-EVOLVING AGENT FOR LATENT RISK REASONING IN AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Safety-critical events in autonomous driving often develop progressively before explicit conflicts emerge. However, existing approaches mainly focus on detecting manifested hazards or predicting short-term outcomes, leaving the early-stage risk evolution insufficiently explored. In this work, we introduce ALERT, a self-evolving agent framework for latent risk reasoning that aims to identify how observable cues may evolve into future driving risks. ALERT employs a mechanism-oriented risk routing taxonomy to organize diverse risk formation processes, uses tool-augmented reasoning to acquire driving-specific evidence, and continuously improves its Skill Pool through controlled evolution. To enable systematic evaluation, we construct a latent risk reasoning benchmark from real-world takeover data covering risk stages, risk types, risk subjects and causal reasoning. ALERT outperforms the strongest VLM-only baseline by 11.7% in overall score and further surpasses knowledge-access baselines, particularly in risk-subject identification and reasoning. Controlled Skill evolution yields an additional 4.9% improvement over the warm-up stage, demonstrating the value of accumulating reusable reasoning experience for latent risk understanding.

1 INTRODUCTION

Many safety-critical events in autonomous driving develop progressively before a direct conflict materializes (Westhofen et al., 2023). Once a risk fully emerges, the remaining time and maneuvering space may already be insufficient for effective intervention, making earlier recognition critical for proactive driving. Prior to risk manifestation, observable cues may already indicate a plausible evolution toward a future conflict. For instance, a construction-occupied neighboring lane may induce forced merging, a narrow road may lead to insufficient passing space for oncoming vehicles, and an occlusion may conceal a road user who could subsequently emerge. We refer to such early risk situations as latent risks. Recognizing latent risks requires not only identifying current scene cues, but also reasoning about how these cues may evolve and affect ego vehicle, enabling a transition from reactive hazard response to proactive risk anticipation.

However, latent-risk reasoning remains challenging. Traditional risk assessment methods, such as time-to-collision (Minderhoud & Bovy, 2001), driving risk field (Wang et al., 2015) and trajectory prediction (Huang et al., 2022), mainly rely on geometric or kinematic cues, limiting their ability to capture semantically driven early risks and reason about their future evolution. Recent vision-language models (VLMs) offer a promising way to understand semantic cues, but reliable latent-risk reasoning requires structured motion and map evidence that is difficult to obtain from images alone (Tian et al., 2024). Tool-augmented agents can query such information on demand and ground their reasoning in more reliable evidence (Yao et al., 2022); however, the resulting reasoning experience is typically not retained as reusable procedural knowledge, leaving each new scene to be analyzed largely from scratch.

Agent Skills provide a natural way to preserve such reasoning experience as reusable procedural knowledge by storing risk-specific cues, tool-use strategies and analysis procedures as external knowledge (Wang et al., 2023a). However, manually designed Skills remain static and cannot cover the diverse and evolving patterns of latent-risk scenarios. Simply allowing LLMs to revise Skills from success and failure cases is also risky, as incidental or flawed patterns may be consolidated into general rules and introduce unreliable knowledge into the Skill Pool, thereby degrading future

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.

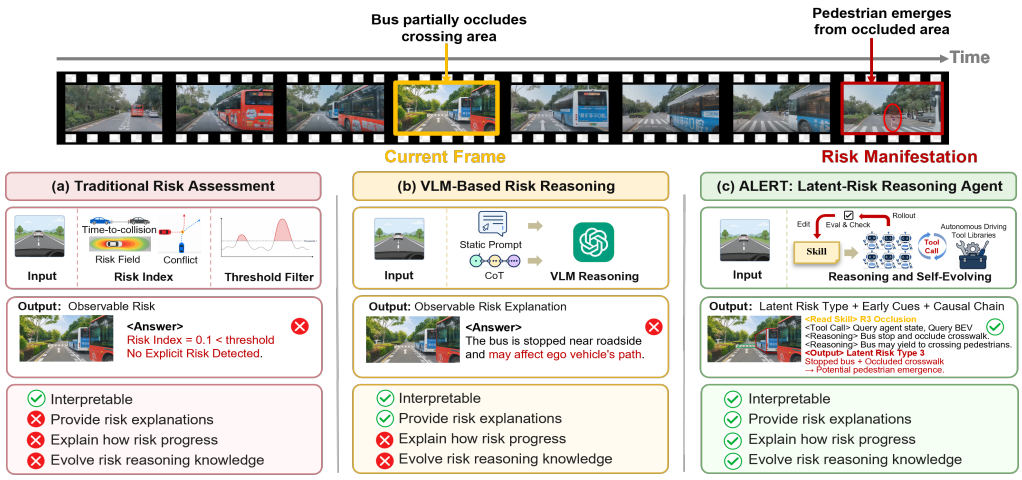


Figure 1: **Comparison of Risk Assessment Paradigms**. Existing methods mainly assess observable or near-conflict risks, whereas ALERT reasons over pre-conflict cues, latent risk mechanisms, and their evolution toward future conflicts, while continuously evolving risk reasoning knowledge.

reasoning (Yang et al., 2026). Therefore, reliable latent-risk reasoning requires not only reusable Skills, but also a controlled evolution mechanism that can accumulate useful experience while evaluating candidate updates before they enter the shared Skill Pool.

To address these challenges, we propose ALERT, a task-specialized agent architecture for latent risk reasoning in autonomous driving, equipped with controlled Skill evolution. ALERT represents accumulated driving experience as reusable risk-specific procedural Skills, organizes latent risks through mechanism-oriented routing, and grounds Skill execution with domain-specific driving tools. More importantly, each candidate Skill update is evaluated before being merged into shared knowledge, reducing the risk of retaining unstable reasoning patterns. In this way, ALERT progressively accumulates reusable risk reasoning strategies from past cases.

Our main contributions are summarized as follows:

- **A mechanism-oriented formulation of latent-risk reasoning.** We formulate latent risk as a pre-conflict reasoning problem and organize risk states by their formation mechanisms, enabling structured analysis of how observable cues may evolve into future conflicts.
- **An event-grounded benchmark for latent-risk reasoning.** We construct a benchmark by backtracking from safety-critical events to characterize early risk states and their evolution, using future observations only for privileged annotation while restricting evaluated models to current observations.
- **A task-specialized agent with controlled Skill evolution.** We propose **ALERT**, a latent-risk reasoning agent that combines risk routing, hierarchical Skills, and tool-grounded reasoning to distinguish plausible risk evolutions. We further develop a controlled Skill evolution process that transforms reasoning feedback into reusable risk knowledge through evaluation, conservative editing, counterfactual validation and merging.

2 Related Works

2.1 Risk Assessment and Critical Scenario Mining

Risk assessment. Traditional rule-based methods detect driving risk with explicit safety indicators, such as time-to-collision (Minderhoud & Bovy, 2001), post-encroachment time (Cooper, 1984) and potential-field-based formulations (Zuo et al., 2025). These methods are effective for near-conflict assessment, but mostly rely on explicit signals and are less sensitive to risks without direct collision trends. To improve anticipation, another line of work estimates future risks through trajectory

prediction (Madjid et al., 2025), intention prediction (Wei & Wang, 2025) and interaction modeling (Wang et al., 2022). While these methods better capture future interactions, their risk judgment still largely depends on predicted path conflicts or occupancy overlap. Latent risks may arise from contextual cues that cannot be fully captured by predicted trajectories alone, making their evolution difficult to anticipate.

Safety-critical scenario mining. Many studies have mined critical clips from real-world driving logs for system evaluation and closed-loop improvement. Zhao et al. (2025) constructed a corner case dataset from naturalistic driving data and introduced scenario marginality metrics to quantify abnormality. Liu et al. (2026) proposed a collaborative evolution framework to explore failure modes of autonomous driving systems. Weber et al. (2022) developed an unsupervised approach to discover critical scenario types missed by predefined rules. These methods are effective at discovering high-risk events, but mainly focus on outcomes such as takeovers and near misses, paying less attention to how risks develop beforehand. Latent risk recognition instead focuses on this neglected earlier progression, aiming to detect how early cues initiate and propagate risk.

2.2 VLM-based Risk Reasoning and Agentic Skill Evolution

VLM-based Risk Reasoning. Recent VLMs have shown strong potential for driving scene understanding and risk reasoning by combining visual perception with language-based inference. Wang et al. (2023b) utilized LLMs as behavior planners with safety verifiers for context-aware safe driving. Zhou et al. (2026) introduced knowledge-driven safety reasoning for risk-sensitive decision making. Xia et al. (2026) proposed KnowVal by combining driving knowledge retrieval with value-guided trajectory evaluation. However, latent risk reasoning requires precise motion, map and spatial information that is difficult to infer reliably from images alone. Tool-augmented agents, including AgentThink (Qian et al., 2025), Agent-Driver (Mao et al., 2023) and MTRDrive (Luo et al., 2025), improve evidence grounding by querying structured driving information during reasoning. While these tools improve evidence grounding, the resulting reasoning experience is typically not retained as reusable procedural knowledge for future analysis.

Agentic Skill Evolution. Recent studies have explored feedback-based self-reflection, self-correction and procedural memory to help agents improve over time. Self-Refine (Madaan et al., 2023) iteratively refines model outputs through feedback without additional training. Reflexion (Shinn et al., 2023) stores verbal reflections in a contextual memory buffer. CRITIC (Gou et al., 2024) enables self-correction through tool-interactive critiquing. Beyond revising a single output or prompt, agent skills store task experience as reusable external knowledge. ExpeL (Zhao et al., 2024) utilizes agents to autonomously collect experience and recall insights during reasoning. SkillX (Wang et al., 2026) maintains and updates external skill libraries for embodied agents. CoEvoSkills (Zhang et al., 2026) jointly evolves skill generators and agent validators. These studies demonstrate the potential of procedural skills for persistent experience accumulation, but are mostly designed for general tasks. For latent risk reasoning, skill evolution requires preserving risk-specific knowledge and preventing case-specific observations from being generalized into unreliable rules.

3 Methodology

3.1 Task Definition

We formulate latent risk reasoning as a structured pre-conflict driving reasoning task. Given a driving scene $\mathbf{x_t} = (I^t_{\text{bev}}, I^t_{\text{front}})$ at time t, where I^t_{bev} and I^t_{front} denote the bird's-eye-view (BEV) and front-view observations respectively, the goal is to first determine whether a driving risk is present, and further reason about the risk when identified. We denote the reasoning agent as $\mathcal{A}$, for each scene the agent produces a structured prediction

$$\hat{y}_t = \mathcal{A}(\mathbf{x_t}) := (\hat{l}_t, \hat{c}_t, \hat{O}_t, \hat{R}_t) \tag{1}$$

where $\hat{l}_t$ denotes predicted risk stage including no risk, latent risk and explicit risk, $\hat{c}_t$ denotes predicted risk type, $\hat{O}_t$ denotes key risk subjects, $\hat{R}_t$ denotes supporting reasoning. For no-risk scenes, $\hat{c}_t$ and $\hat{O}_t$ are left unassigned. The corresponding reference annotation is written as

$$y^*_t := (l^*_t, c^*_t, O^*_t, R^*_t), \tag{2}$$

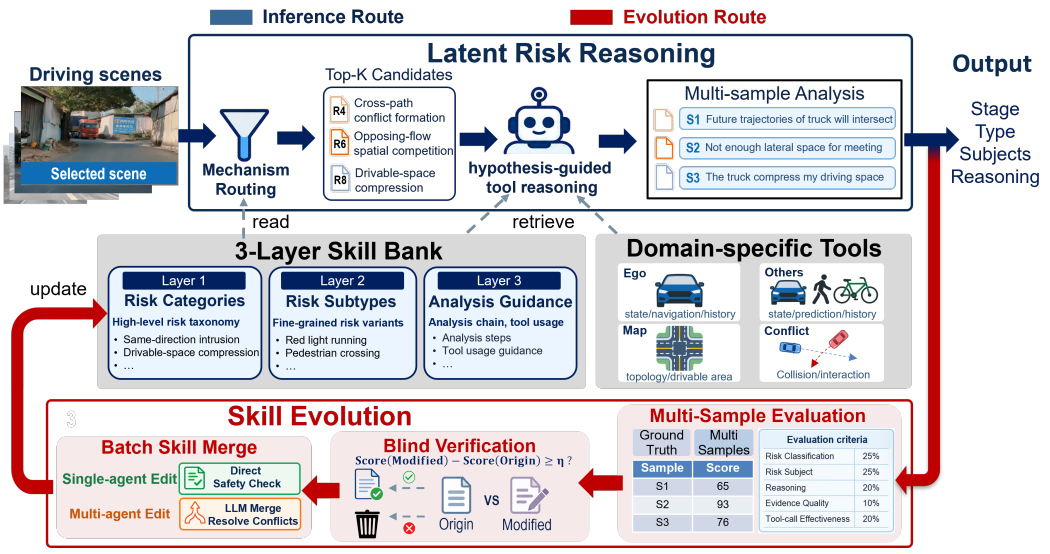


Figure 2: **The overall framework**. ALERT consists of an inference route and a controlled Skill-evolution route. The inference route performs mechanism routing and hypothesis-guided tool reasoning. During evolution, sampled trajectories are jointly evaluated, candidate Skill edits are compared against their original versions, and only accepted edits are merged into the shared Skill Bank. The three-layer Skill Bank organizes risk categories, risk subtypes and analysis guidance, and is updated across training batches.

Given a labeled dataset $\mathcal{D} = \{(\mathbf{x_i}, y_i^*)\}_{i=1}^N$, we evaluate an agent by comparing its structured predictions with reference annotations across risk stage, risk type, risk subjects and reasoning quality, where evaluation protocol is introduced in Section 3.3.

3.2 Latent Risk Taxonomy

We first provide an operational definition of latent risk. A scene is considered to contain a latent risk when non-manifestation, inferability and ego-relevance are jointly satisfied. Non-manifestation requires that no direct conflict or immediate safety constraint has manifested, inferability requires observable early cues supporting a plausible future risk evolution, and ego-relevance requires such evolution may affect ego vehicle's future motion or safety margin. We summarize it as

$$\text{LatentRisk}(\mathbf{x_t}) = \mathbb{I}[\mathcal{M}(x_t) \wedge \mathcal{I}(x_t) \wedge \mathcal{E}(x_t)] \tag{3}$$

where $\mathcal{M}(x_t)$, $\mathcal{I}(x_t)$ and $\mathcal{E}(x_t)$ denote non-manifestation, inferability and ego-relevance.

The above definition determines whether a risk is latent, but does not distinguish different mechanisms of risk evolution. Similar outcomes may arise from different mechanisms and require different reasoning strategies. Therefore, we construct a Latent Risk Routing Taxonomy by tracing where risk-relevant uncertainty or constraints first arise and how they propagate toward ego vehicle. Specifically, the perception–prediction–decision–control chain characterizes dominant source of risk-relevant uncertainty or constraints, providing a coarse-grained mechanism-oriented organization of this progression: perception risks stem from degraded or hidden scene information, prediction risks from uncertain interactions among traffic participants, decision risks from progressive constraints on ego vehicle's feasible driving space, and control risks from environmental conditions that reduce safe control feasibility. Following these formation pathways, we define 10 routing categories summarized in Table 1 (full taxonomy is shown in Appendix A), with each specified by its formation mechanism, activation conditions, exclusion rules and risk-stage boundaries. When multiple cues coexist, routing follows the dominant mechanism that most directly explains ego-relevant risk evolution. This mechanism-oriented taxonomy provides a consistent basis for benchmark annotation, risk routing and subsequent reasoning.

Table 1: Mechanism-oriented latent risk routing taxonomy

Level	ID	Routing Category	Dominant formation mechanism
Perception	R1	Illumination-induced degradation	Abnormal illumination degrades reliability of ego-relevant visual cues.
	R2	Medium-induced visibility degradation	Atmospheric or airborne media reduce reliable scene visibility.
	R3	Occlusion-induced potential emergence	Occlusion removes visual access to ego-relevant regions and may conceal participants entering ego's future path.
Prediction	R4	Cross-path conflict formation	Visible traffic participants may cross ego's future path.
	R5	Same-direction intrusion	Same-direction interactions may cause nearby participants to enter ego's future driving space.
	R6	Opposing-flow spatial competition	Oncoming traffic may compete with ego for limited passing space.
	R7	Behavior-induced uncertainty	Abnormal behavior increases uncertainty beyond common motion patterns.
Decision	R8	Drivable-space compression	Hard or soft constraints reduce ego's feasible driving space without changing path connectivity.
	R9	Path-connectivity restructuring	Changes in road connectivity require ego vehicle to re-evaluate its feasible path.
Control	R10	Control-feasibility degradation	Environmental conditions reduce feasible margin of vehicle stability.

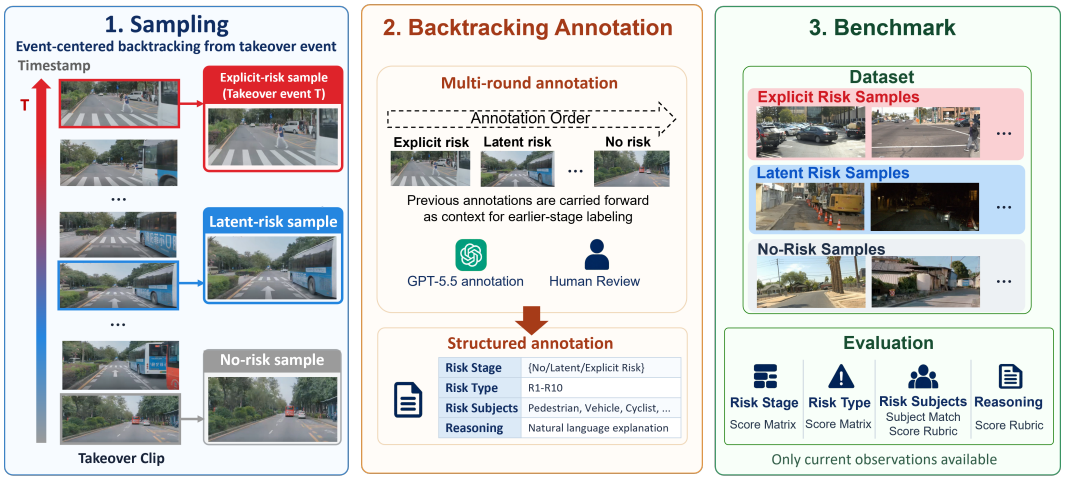


Figure 3: **Annotation pipeline**. Starting from explicit risk at takeover moment, we trace backward to identify earlier scene evidence, constructing samples at different risk stages with risk-evolution annotations.

3.3 Latent Risk Reasoning Benchmark

Event-anchored sampling. We construct a dataset from real-world safety-critical driving logs containing human takeover events as observable risk manifestations. Following accident evolution theory (Svenson, 1991), we view a safety-critical event as a progressive process from normal driving, through latent risk development, to explicit risk manifestation. For each takeover clip, the takeover timestamp T serves as a temporal anchor for an explicit-risk candidate. We then trace backward from T and sample candidate scenes at progressively earlier offsets to cover different phases of risk development. These offsets provide temporal priors, while the final Risk Stage is determined through subsequent human review and revision based on the observed evidence.

Structured annotation. GPT-5.5 is utilized as a privileged annotation model to construct structured annotations for each sampled scene. For latent and explicit risk samples, the annotator observes current scene together with the realized future sequence up to takeover moment, and distills the observed risk evolution into ground-truth risk type, risk subjects and reasoning for the current scene. For no-risk samples, the annotator generates reasoning explaining why current evidence does not support a valid driving risk. Importantly, future observations are available only during annotation process, whereas evaluated models receive only current scene during benchmark evaluation. This separation provides hindsight-grounded supervision while preventing future-information leakage during inference. The final dataset contains 2,849 training and 458 testing takeover clips. Training-set annotations are manually reviewed through sampling, while all test-set annotations undergo human review to ensure consistency.

Benchmark evaluation. We evaluate structured risk reasoning along four dimensions: risk stage, risk type, risk subjects and reasoning. Risk stage and risk type are evaluated through confusion-based metrics. Specifically, risk stage uses a predefined confusion matrix over no-risk, latent-risk and explicit-risk stages, while risk type adopts a mechanism-aware scene-specific confusion set, where exact matches receive full credit and confused categories receive partial credit. Risk subjects and reasoning are evaluated through an LLM critic with rubric-based scoring. The critic derives scene-specific criteria from ground-truth annotations and visual observations, and each criterion is evaluated on a three-level match scale. For risk subjects, predicted subjects are first semantically aligned with the ground-truth list, after which matched subjects are scored against corresponding criteria and unmatched predictions receive an exponential discount penalty. The final benchmark score is computed as a weighted combination of four dimensions:

$$\mathrm{Score}(\hat{y}, y^*) = \lambda_l S_{\mathrm{stage}} + \lambda_c S_{\mathrm{type}} + \lambda_o S_{\mathrm{subject}} + \lambda_r S_{\mathrm{reason}} \tag{4}$$

where S_{stage}, S_{type}, S_{subject} and S_{reason} denote the scores for risk stage, risk type, risk subjects and reasoning, respectively, and λ_l, λ_c, λ_o, λ_r are their corresponding weights set to 0.2, 0.25, 0.25, 0.3, respectively. For no-risk scenes, only risk-stage prediction is evaluated, since risk type and risk subjects are undefined and reasoning is highly non-unique. More details of sampling, annotation and scoring are provided in Appendix B.

3.4 Latent-Risk Reasoning Agent with Controlled Skill Evolution

Overview. ALERT augments a frozen reasoning agent $\mathcal{A}$ with an external Skill Pool $\mathcal{S}$. Given driving scenes and annotations $(\mathbf{x}_t, y_t^*) \sim \mathcal{D}$, the desired objective is

$$\mathcal{S}^* = \arg\max_{\mathcal{S}} \; \mathbb{E}_{(\mathbf{x}_t, y_t^*)\sim\mathcal{D}} \left[\mathrm{Score}\big(\mathcal{A}(\mathbf{x}_t; \mathcal{S}), y_t^*\big)\right], \tag{5}$$

where $\mathcal{A}(\mathbf{x}_t; \mathcal{S})$ is the Skill-conditioned reasoning agent and Score is the benchmark metric defined in Eq. 4. Evolution proceeds over M scene batches $\mathcal{B}_1, \ldots, \mathcal{B}_M$ following Figure 2. Within each batch, scene-level reasoning, evaluation, and Skill editing run in parallel, while accepted edits are merged into $\mathcal{S}^{(n+1)}$ at the batch boundary, forming a propose–evaluate–merge process.

Mechanism routing and hypothesis-guided tool reasoning. Before a conflict manifests, a latent-risk scene may support multiple plausible mechanisms of risk evolution. ALERT therefore treats routing as hypothesis generation rather than committing prematurely to a single category. For each scene $\mathbf{x_t} \in \mathcal{B}_n$, Risk Routing Agent generates ranked risk hypotheses with corresponding rationales:

$$\mathcal{H}_t := \{(c_t^i, r_t^i)\}_{i=1}^{N_h} = \mathrm{Route}(\mathbf{x_t}; \mathcal{S}^{(n)}), \tag{6}$$

where c_t^i denotes the i-th candidate risk category and r_t^i provides the routing rationale explaining why this category is considered plausible, and N_h is the number of routed risk hypotheses. Each hypothesis represents a possible mechanism of how current scene may evolve into future risk.

Conditioned on each hypothesis $(c_t^i, r_t^i) \in \mathcal{H}_t$, ALERT retrieves the corresponding category- and subtype-level Skills $\mathcal{S}_{t,c_t^i}$ and independently samples K_i tool-augmented reasoning trajectories:

$$\hat{y}_{t,i}^{(k)} \sim \mathcal{A}\big(\mathbf{x_t}, c_t^i, r_t^i, \mathcal{S}_{t,c_t^i}, \mathcal{T}\big), \qquad (c_t^i, r_t^i) \in \mathcal{H}_t,\ k = 1, \ldots, K_i. \tag{7}$$

The retrieved Skills provide observable activation and exclusion cues, evidence checks, and analysis procedures, while the driving-specific tool library $\mathcal{T}$ provides structured information about ego and agent states, map geometry, drivable corridors, and conflict relations. Together, they guide the Tool Analysis Agents to actively acquire evidence for evaluating the corresponding risk hypothesis rather than directly prescribing the final prediction. Since the conclusion is grounded in acquired evidence, a reasoning trajectory may reject its initially assigned hypothesis when observations are inconsistent with it, thereby reducing sensitivity to early routing errors.

Risk-oriented multi-dimensional evaluation. The Evaluation Agent jointly evaluates all sampled trajectories of the same scene against the benchmark criteria and annotations:

$$\{g_{t,i}^{(k)}\} = \mathrm{Eval}\big(\{\hat{y}_{t,i}^{(k)}\}, y_t^*\big), \qquad g_{t,i}^{(k)} := (s_{t,i}^{(k)}, d_{t,i}^{(k)}) \tag{8}$$

where $s_{t,i}^{(k)} \in [0, 100]$ denotes the multi-dimensional reasoning score of the k-th trajectory under the i-th routed hypothesis, evaluated along risk classification, risk subjects, reasoning, evidence quality

and tool-call effectiveness, and textual diagnosis $d_{t,i}^{(k)}$ summarizes the strengths and failure modes of the trajectory for subsequent Skill evolution. For latent-risk scenarios, the reasoning dimension particularly evaluates whether agent can infer how current cues may evolve into future conflicts. The sampled trajectories provide two complementary signals. Within the same risk hypothesis, differences between high- and low-scoring trajectories highlight which evidence checks, reasoning steps and tool strategies are associated with better judgments, making edit signal less dependent on absolute LLM scores. Across competing hypotheses, failure cases reveal ambiguities between mechanistically similar risk categories. When a routed hypothesis lacks sufficient supporting evidence, the evaluator tends to reject it in favor of a better-supported alternative or no-risk judgment rather than forcing the assigned category. Such errors can therefore improve not only reasoning procedures but also the activation and exclusion boundaries used for subsequent routing.

Conservative risk skill editing. The Skill Evolution Agent receives evaluation context including sampled trajectories, scores, textual diagnoses and tool traces, and produces edit operations

$$E_t = \mathrm{Evo}\big(\mathcal{S}^{(n)}, \{\hat{y}_{t,i}^{(k)}\}, \{g_{t,i}^{(k)}\}, y_t^*\big). \tag{9}$$

The edits are first applied to an isolated copy $\tilde{\mathcal{S}}_t$ before validation. To prevent scene-specific observations from being directly consolidated into shared knowledge, ALERT restricts Skill edits to reusable risk mechanisms supported by observable perception or map cues. The edited Skills describe how to identify and distinguish risk mechanisms rather than prescribe ego actions, keeping the accumulated knowledge aligned with the reasoning task. New subtype Skills are introduced only when a recurring risk pattern cannot be adequately represented by current subtype Skills and can be clearly distinguished from neighboring subtypes. In this way, Skill evolution expands procedural knowledge only when necessary while reducing the risk of memorizing isolated corner cases.

Task-local counterfactual validation. Evaluating every fine-grained Skill edit across sufficiently broad held-out scenes would multiply the evolution cost. We therefore use task-local validation as a minimal consistency check: an edit should at least improve the scene that motivated it without materially degrading its target-risk routing. The original Skill Pool $\mathcal{S}^{(n)}$ and an edited copy $\tilde{\mathcal{S}}_t$ are compared on the same scene while keeping the underlying reasoning model fixed. To reduce sensitivity to stochastic reasoning, ALERT independently resamples multiple trajectories under both Skill states and accepts the edit according to:

$$\mathrm{Accept}(E_t) = \mathbb{I}\left[\left(\frac{1}{2}\Delta_t^{\mathrm{avg}} + \frac{1}{2}\Delta_t^{\mathrm{best}} > \eta\right) \wedge \left(G_t^{\mathrm{route}} \geq -\epsilon_r\right)\right]. \tag{10}$$

Here, Δ_t^{avg} and Δ_t^{best} measure the average and best-trajectory score improvements, respectively, while G_t^{route} measures the change in the ground-truth risk category's routing priority.

Conflict-aware batch merging. Parallel scene-level evolution may generate concurrent edits to the same risk Skills. At the end of each batch, qualified edits are consolidated into the shared Skill Pool:

$$\mathcal{S}^{(n+1)} = \mathrm{Merge}\big(\mathcal{S}^{(n)}, \{E_t \mid t \in \mathcal{B}_n, \mathrm{Accept}(E_t)\}\big). \tag{11}$$

Edits affecting different Skills are directly incorporated, while conflicting modifications within the same risk category are resolved by a Skill Merge Agent. The merger preserves reusable risk mechanisms and consolidates overlapping updates without collapsing distinctions between similar risk categories. This maintains the category boundaries required for reliable latent-risk routing while keeping the Skill Pool concise.

Deployment inference. After offline evolution, ALERT freezes the final Skill Pool $\mathcal{S}^*$ and performs risk routing and Skill-guided tool reasoning on each new scene without further Skill updates or model parameter changes. This separation allows ALERT to continuously accumulate latent-risk knowledge during development while maintaining a stable and reproducible inference process. More details of latent-risk reasoning agent framework are provided in Appendix C.

4 Experiments

4.1 Setup

Benchmark and splits. We conduct experiments on latent risk reasoning benchmark introduced in Section 3.3. From 2,849 training clips, we construct a 1,600-clip subset for Skill evolution using

Table 2: Main results on latent risk reasoning benchmark.

Method	Stage	Type	Subject	Reasoning	Overall
VLM-only Baselines					
Qwen3.7-Plus	80.30	75.46	47.22	56.76	63.76
Qwen3.7-Flash	79.22	71.75	52.49	62.92	65.78
GPT-5.6-Sol	75.87	74.22	42.59	52.38	60.09
DeepSeek-V4.1-Flash	85.60	69.75	55.91	67.33	68.73
Compact Model Baselines					
Qwen3.5-4B (0-shot)	66.54	54.18	20.01	37.06	42.97
Qwen3.5-4B + SFT	79.21	77.61	46.69	61.46	65.36
Knowledge-Access Baselines					
ExpeL-style (Zhao et al., 2024)	79.42	**79.30**	55.76	70.85	70.90
HyDE-style (Gao et al., 2023)	85.79	**79.30**	63.42	73.86	75.00
ALERT (Ours)	**87.05**	79.28	**66.86**	**76.01**	**76.75**

subtype-aware sampling, with a higher relative proportion of rare risk subtypes to reduce long-tail imbalance. The test split retains its natural distribution and is reserved exclusively for final evaluation. During testing, all methods are restricted to information available at the current time, and future observations are not provided.

Baselines. ALERT uses a frozen Qwen3.7-Plus reasoning backbone. We compare ALERT with three groups of baselines. *(i) VLM-only baselines* remove Skills and tools and reason directly from front-view and BEV observations using Qwen3.7-Plus, Qwen3.7-Flash, GPT-5.6 Sol, and DeepSeek-V4.1-Flash, providing a reference for model-only reasoning capability. *(ii) Compact model baselines* use Qwen3.5-4B with zero-shot inference and supervised fine-tuning to evaluate parameter adaptation on a smaller deployable model. *(iii) Knowledge-access baselines* share the same Qwen3.7-Plus backbone, routed hypotheses, and driving tools with ALERT, but replace hierarchical Skill access with retrieval. ExpeL-style retrieves similar successful trajectories and tool traces, while HyDE-style retrieves relevant Skill chunks from router-generated summaries.

Implementation details. During evolution, ALERT retains up to three candidate risk categories and samples five reasoning trajectories per scene at temperature 0.8, allocating three to the top-ranked hypothesis and one to each remaining hypothesis, with at most 8 tool-call rounds per trajectory. Training uses batches of 32 scenes, sets the validation threshold to $\eta = 0$, and performs Skill Pool grooming every 25 batches. At test time, the evolved Skill Pool is frozen and read-only. ALERT performs risk routing followed by near-greedy Skill-guided tool reasoning to produce the final structured prediction.

Evaluation metrics. We evaluate all methods following protocol defined in Eq. 4 and report scores for risk stage, risk type, reasoning, risk subjects and overall score. All methods use the same GPT-5.5 critic, evaluation prompts and ground-truth annotations, and predictions from the same scene are jointly evaluated to reduce the variance of LLM-based scoring.

4.2 Main Results

As shown in Table 2, ALERT achieves the best overall performance, with its gains mainly arising from Risk Stage, Risk Subject and Reasoning, while remaining comparable to knowledge-access baselines on Risk Type. This suggests that the main benefit of ALERT is to identify relevant risk subjects and explain how current cues may evolve into future conflicts once a plausible risk mechanism has been identified. In comparison, VLM-only methods show less consistent performance across evaluation dimensions, indicating that stronger backbone models alone do not reliably yield complete structured risk understanding. SFT substantially improves the compact model, but still trails knowledge-augmented methods. Notably, although HyDE-style is already competitive, ALERT further improves subject identification and reasoning under similar risk-type performance, suggesting

Table 3: Ablation study on Skill evolution and main components of ALERT. All values denote average score changes relative to full ALERT.

Variant	Stage	Type	Subject	Reasoning	Overall
Warm-up (3 batches)	-0.70	-1.09	-6.56	-5.07	-3.57
Intermediate (100 batches)	-0.70	-0.35	-0.58	+0.20	-0.31
ALERT (200 batches)	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
w/o Subtype Skills	-7.31	-0.41	-9.49	-5.74	-5.66
w/o Tools & w/o BEV	-6.79	-1.77	-14.54	-14.33	-9.73
w/o Tools & w/ BEV	-5.12	-1.05	-13.90	-14.07	-8.98

that hierarchical Skill organization is more effective than flat retrieval for converting reusable experience into fine-grained latent-risk reasoning.

4.3 Ablation Study

Effect of Skill evolution. As the Skill Pool evolves from the warm-up stage to the full model, overall performance steadily improves, with the largest gains concentrated in Risk Subject and Reasoning, which require finer-grained scene understanding. The intermediate model already approaches the final performance, suggesting that much of the useful knowledge is acquired relatively early, while later evolution yields smaller additional gains, mainly in subject identification. In contrast, Risk Stage and Risk Type change only modestly, indicating that Skill evolution mainly contributes by accumulating more detailed knowledge of scene relations and risk evolution. To further examine how controlled evolution contributes to these improvements, we retrospectively analyze the training logs. Counterfactual validation rejects 45.7% of candidate edits, with rejected edits yielding an average score change of -3.65 points under trajectory resampling, compared with a 9.97-point improvement for accepted edits. These results illustrate how the validation gate filters locally detrimental updates before they enter the shared Skill Pool.

Effect of components. Removing Subtype Skills causes a clear performance drop, particularly in Risk Subject and Reasoning, showing that fine-grained subtype knowledge helps distinguish specific risk mechanisms and organize more complete reasoning. Removing Tools leads to a substantial performance drop, particularly in Risk Subject and Reasoning, highlighting the importance of structured spatial, motion, and interaction evidence. Further removing BEV consistently degrades all evaluation dimensions, indicating that BEV provides complementary visual information even in the absence of structured driving tools.

5 Conclusion

We propose ALERT, a self-evolving agent framework for latent risk reasoning in autonomous driving. ALERT organizes risk evolution through a mechanism-oriented routing taxonomy, grounds reasoning with driving-specific tools, and progressively improves its Skill Pool through controlled evolution. We further construct a latent risk reasoning benchmark from real-world takeover data, enabling evaluation across different risk development stages with structured annotations of risk types, subjects, and reasoning. Experiments demonstrate that ALERT improves risk identification, causal reasoning, and evidence-grounded analysis by accumulating reusable reasoning experience. These results highlight the potential of self-evolving agents for proactive safety understanding in autonomous driving.

AI use statement

Generative AI tools were used to assist with benchmark annotation, literature discovery and polish writing. In the research pipeline, LLMs were used to generate structured annotations and reasoning trajectories from real-world driving data, as well as to support risk reasoning, trajectory evaluation and Skill evolution. All AI-assisted research materials were reviewed by the authors. Generated

annotations underwent manual review, and LLM-based evaluation was examined through human assessment, independent judging and prediction-order robustness tests. The authors verified the experimental configurations, reported results, and research claims using the actual implementation and original experimental records. We take full responsibility for final content of this work.

Ethics statement

This work aims to improve autonomous driving safety through early recognition of latent risks and reasoning about their evolution. We construct our benchmark from real-world driving takeover data, and any use or release of these data must comply with applicable privacy, confidentiality, and licensing requirements. Similarly, any future code release will comply with relevant intellectual property and licensing requirements. Sensitive information, internal interfaces, and access credentials will be removed before release to prevent unauthorized data access or information leakage.

Reproducibility Statement

We provide detailed descriptions of the latent-risk taxonomy, benchmark and ALERT framework in Section 3 and Appendices A– C. Experimental settings, baseline configurations, Skill evolution details, and computational costs are documented in Section 4 and Appendix D. Our benchmark is constructed from real-world driving takeover data and is subject to data access restrictions. Therefore, exact reproduction of the reported benchmark results requires authorized access to the underlying data. Nevertheless, the documented methods and experimental procedures are intended to facilitate reimplementation of ALERT and evaluation on other driving datasets.

References

PJ Cooper. Experience with traffic conflicts in canada with emphasis on "post encroachment time" techniques. In *International calibration study of traffic conflict techniques*, pp. 75–96. Springer, 1984.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1762–1777, 2023.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Nan Duan, Weizhu Chen, et al. Critic: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, volume 2024, pp. 57734–57811, 2024.

Yanjun Huang, Jiatong Du, Ziru Yang, Zewei Zhou, Lin Zhang, and Hong Chen. A survey on trajectory-prediction methods for autonomous driving. *IEEE transactions on intelligent vehicles*, 7(3):652–674, 2022.

Jiangfan Liu, Yongkang Guo, Fangzhi Zhong, Tianyuan Zhang, Zonglei Jing, Siyuan Liang, Jiakai Wang, Mingchuan Zhang, Aishan Liu, and Xianglong Liu. Adversarial generation and collaborative evolution of safety-critical scenarios for autonomous vehicles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 38926–38934, 2026.

Ziang Luo, Kangan Qian, Jiahua Wang, Yuechen Luo, Jinyu Miao, Zheng Fu, Yunlong Wang, Sicong Jiang, Zilin Huang, Yifei Hu, et al. Mtrdrive: Memory-tool synergistic reasoning for robust autonomous driving in corner cases. *arXiv preprint arXiv:2509.20843*, 2025.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Nadya Abdel Madjid, Abdulrahman Ahmad, Murad Mebrahtu, Yousef Babaa, Abdelmoamen Nasser, Sumbal Malik, Bilal Hassan, Naoufel Werghi, Jorge Dias, and Majid Khonji. Trajectory prediction for autonomous driving: Progress, limitations, and future directions. *Information Fusion*, pp. 103588, 2025.

Jiageng Mao, Junjie Ye, Yuxi Qian, Marco Pavone, and Yue Wang. A language agent for autonomous driving. *arXiv preprint arXiv:2311.10813*, 2023.

Michiel M Minderhoud and Piet HL Bovy. Extended time-to-collision measures for road traffic safety assessment. *Accident Analysis & Prevention*, 33(1):89–97, 2001.

Kangan Qian, Sicong Jiang, Yang Zhong, Ziang Luo, Zilin Huang, Tianze Zhu, Kun Jiang, Mengmeng Yang, Zheng Fu, Jinyu Miao, et al. Agentthink: A unified framework for tool-augmented chain-of-thought reasoning in vision-language models for autonomous driving. In *EMNLP (Findings)*, pp. 10663–10682, 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Ola Svenson. The accident evolution and barrier function (aeb) model applied to incident analysis in the processing industries. *Risk Analysis*, 11(3):499–507, 1991.

Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models. *arXiv preprint arXiv:2402.12289*, 2024.

Chenxi Wang, Zhuoyun Yu, Xin Xie, Wuguannan Yao, Runnan Fang, Shuofei Qiao, Kexin Cao, Guozhou Zheng, Xiang Qi, Peng Zhang, et al. Skillx: Automatically constructing skill knowledge bases for agents. *arXiv preprint arXiv:2604.04804*, 2026.

Denggui Wang, Weiping Fu, Qingyuan Song, and Jincao Zhou. Potential risk assessment for safe driving of autonomous vehicles under occluded vision. *Scientific reports*, 12(1):4981, 2022.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023a.

Jianqiang Wang, Jian Wu, and Yang Li. The driving safety field based on driver–vehicle–road interactions. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):2203–2214, 2015.

Yixuan Wang, Ruochen Jiao, Sinong Simon Zhan, Chengtian Lang, Chao Huang, Zhaoran Wang, Zhuoran Yang, and Qi Zhu. Empowering autonomous driving with large language models: A safety perspective. *arXiv preprint arXiv:2312.00812*, 2023b.

Nico Weber, Christoph Thiem, and Ulrich Konigorski. Toward unsupervised test scenario extraction for automated driving systems from urban naturalistic road traffic data. *arXiv preprint arXiv:2202.06608*, 2022.

Wen Wei and Jiankun Wang. Intention-based and risk-aware trajectory prediction for autonomous driving in complex traffic scenarios. In *2025 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 345–350. IEEE, 2025.

Lukas Westhofen, Christian Neurohr, Tjark Koopmann, Martin Butz, Barbara Schütt, Fabian Utesch, Birte Neurohr, Christian Gutenkunst, and Eckard Böde. Criticality metrics for automated driving: A review and suitability analysis of the state of the art: L. westhofen et al. *Archives of Computational Methods in Engineering*, 30(1):1–35, 2023.

Zhongyu Xia, Wenhao Chen, Yongtao Wang, and Ming-Hsuan Yang. Knowval: A knowledge-augmented and value-guided autonomous driving system. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3740–3749, 2026.

Shidong Yang, Ziyu Ma, Tongwen Huang, Xucong Wang, Renda Li, Yiming Hu, Yong Wang, and Xiangxiang Chu. Skillforge: Evolving verifiable skills for reinforcement learning agents. *arXiv preprint arXiv:2608.24747*, 2026.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Hanrong Zhang, Shicheng Fan, Henry Peng Zou, Yankai Chen, Zhenting Wang, Jiayu Zhou, Chengze Li, Wei-Chieh Huang, Yifei Yao, Kening Zheng, et al. Coevoskills: Self-evolving agent skills via co-evolutionary verification. *arXiv preprint arXiv:2604.01687*, 2026.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

Jian Zhao, Wenxu Li, Bing Zhu, Peixing Zhang, Zhaozheng Hu, and Jie Meng. Corner case dataset for autonomous vehicle testing based on naturalistic driving data. *Smart Cities*, 8(4):129, 2025.

Zhiyuan Zhou, Heye Huang, Boqi Li, Shiyue Zhao, Yao Mu, and Jianqiang Wang. Safedrive: Knowledge-and data-driven risk-sensitive decision-making for autonomous vehicles with large language models. *Accident Analysis & Prevention*, 224:108299, 2026.

Dachuan Zuo, Zilin Bian, Fan Zuo, and Kaan Ozbay. Composite safety potential field for highway driving risk assessment. *Accident Analysis & Prevention*, 220:108080, 2025.

A Details on Latent Risk Taxonomy

The routing taxonomy is designed as an operational mechanism space for latent-risk patterns considered in our benchmark. Each category is specified by its definition, activation condition, exclusion boundary and risk-stage boundary, as summarized in Table 4. Figure 4 illustrates examples of typical scene patterns from R1 to R10. As a risk evolves, its dominant mechanism may change. For example, an occlusion-induced emergence risk (R3) may transition into a crossing-path conflict (R4) once the previously hidden participant becomes fully observable.

Table 4: Operational specification of latent risk routing taxonomy.

ID	Definition	Activation Condition	Exclusion Boundary	Risk-stage Boundary
R1	**Illumination-induced degradation.** Key driving-relevant visual information becomes unreliable because of abnormal illumination.	Activated when illumination conditions reduce reliable observability of ego-relevant road regions, traffic participants, lane boundaries, or traffic-control cues.	Excluded when lighting variation is visually present but does not affect ego-relevant perception, or when visibility degradation is mainly caused by atmospheric media or geometric occlusion.	Latent only in current benchmark. If a downstream interaction risk becomes explicit, routing follows the corresponding interaction mechanism.
R2	**Medium-induced visibility degradation.** Key driving-relevant visual information becomes unreliable due to atmospheric or airborne media.	Activated when weather or airborne media reduce reliable perception of ego-relevant scene elements.	Excluded when the medium is present but visibility remains sufficient, or when the dominant degradation is caused by illumination or occlusion.	Latent only in current benchmark. If a downstream interaction risk becomes explicit, routing follows the corresponding interaction mechanism.
R3	**Occlusion-induced potential emergence.** An ego-relevant driving region is fully or partially occluded, leaving open the possibility that a hidden traffic participant may emerge into ego vehicle's future path.	Activated when an occluder hides an ego-relevant region and observable contextual cues support the plausible emergence of a hidden traffic participant into the ego vehicle's future driving space.	Excluded when occluded region is irrelevant to ego motion, when no plausible scene context supports the presence of hidden participants, or when the participant is sufficiently observable.	Latent only in current benchmark. Once participant becomes observable and a concrete interaction mechanism is identifiable, routing switches to corresponding interaction category.
R4	**Crossing-path conflict formation.** A visible traffic participant may enter ego vehicle's future path through a crossing motion pattern.	Activated when participant has a plausible transverse, turning or crossing motion that may intersect ego vehicle's future path.	Excluded for same-direction intrusion, opposing-flow competition, or occlusion-based hypotheses without a visible participant.	Latent when participant remains outside ego vehicle's future path but observable evidence supports a plausible future crossing. Explicit when crossing interaction already occupies or creates an immediate constraint on ego vehicle's future path.
R5	**Same-direction intrusion.** A neighboring or merging traffic participant with a broadly aligned traffic flow may enter or encroach upon ego vehicle's future driving space.	Activated when relative motion indicates a plausible transition from a neighboring or merging traffic flow into ego vehicle's future corridor.	Excluded for crossing traffic, opposing traffic, or static constraints that compress the road without active participant intrusion.	Latent when observable motion cues indicate a plausible future intrusion while participant remains outside ego vehicle's future corridor. Explicit when intrusion has entered or creates immediate constraint on ego vehicle's future corridor.
R6	**Opposing-flow spatial competition.** Opposing traffic may compete with ego vehicle for limited shared driving space.	Activated when ego vehicle and opposing participant are likely to enter the same spatially constrained region.	Excluded when opposing traffic remains spatially separated with sufficient passing space, or when dominant risk arises from passive road compression without active opposing-flow interaction.	Latent when an upcoming spatial competition is supported by observable evidence but sufficient separation still remains. Explicit when opposing interaction creates an immediate spatial constraint and available passing clearance becomes insufficient.
R7	**Behavior-induced uncertainty.** A traffic participant exhibits unexplained abnormal behavior that increases uncertainty about an ego-relevant future interaction.	Activated when participant shows abnormal motion, such as abrupt braking, weaving, or unstable movement, that may affect ego vehicle but cannot be explained by a clear scene cause.	Excluded when behavior can be explained by a specific crossing, same-direction intrusion or opposing-flow interaction, which should be routed to R4, R5 or R6 respectively.	Latent when abnormal behavior creates uncertainty about a plausible future interaction but has not directly constrained ego vehicle. Explicit when the unexplained behavior already creates immediate constraint on ego vehicle.
R8	**Drivable-space compression.** Static, structural, or passive constraints may progressively reduce the ego vehicle's feasible driving space while path connectivity remains unchanged.	Activated when such constraints reduce the width or freedom of ego vehicle's feasible corridor.	Excluded when constraint is primarily caused by an actively intruding traffic participant, or when main change concerns path connectivity rather than available space.	Latent when an upcoming compression is observable but current feasible corridor remains sufficient. Explicit when ego vehicle reaches constrained region and available driving space becomes an immediate limitation.
R9	**Path-connectivity restructuring.** Changes in road topology, lane connectivity, or temporary road organization may alter the ego vehicle's future feasible paths and require path re-selection.	Activated when such changes alter the set of feasible future paths available to the ego vehicle.	Excluded when path connectivity remains unchanged and only available driving width or freedom is reduced.	Latent when a connectivity change is observable ahead while current path remains valid. Explicit when continuation along current path becomes infeasible or an immediate path choice is required.
R10	**Control-feasibility degradation.** Road geometry, surface conditions, or environmental factors may reduce the margin for safe vehicle control.	Activated when observable road or environmental conditions indicate a reduced margin for stable and safe vehicle control.	Excluded when dominant risk concerns visibility, traffic interaction, drivable-space compression, or path connectivity rather than vehicle controllability.	Latent only in current benchmark. The category captures observable precursors to reduced control feasibility.



(a) R1 Illumination degradation (b) R2 Visibility degradation (c) R3 Occluded emergence (d) R4 Cross-path conflict (e) R5 Same-direction intrusion



(f) R6 Opposing-flow competition (g) R7 Behavior uncertainty (h) R8 Space compression (i) R9 Path restructuring (j) R10 Control degradation

Figure 4: Representative scene patterns of R1-R10 in latent risk taxonomy.

Some scenes may exhibit multiple risk cues simultaneously. We therefore apply a small set of mechanism-level tie-breaking rules to keep routing consistent, as shown in Table 5. The key principle is to prioritize the most specific observable interaction mechanism over a more generic source of uncertainty. These rules are intended to provide a consistent operational interface for annotation, routing and Skill retrieval.

Table 5: Tie-breaking rules for commonly confused routing categories.

Competing Categories	**Routing Principle**
R1 vs. R2	Distinguish the source of visibility degradation. Degradation primarily caused by abnormal illumination is routed to R1, whereas degradation caused by atmospheric or airborne media is routed to R2.
R3 vs. R4/R5/R6	R3 is used when occlusion prevents potentially conflicting participant or its interaction geometry from being sufficiently observed. Once participant becomes sufficiently observable and a concrete interaction mechanism can be identified, routing follows R4, R5 or R6.
R4 vs. R5 vs. R6	Route according to dominant interaction geometry: crossing-path interaction to R4, same-direction intrusion to R5, and opposing-flow spatial competition to R6.
R7 vs. R4/R5/R6	R7 is used when abnormal behavior cannot be explained by a scene cause. If behavior is attributable to a reasonable crossing-path, same-direction intrusion, or opposing-flow interaction, the scene is routed to R4, R5 or R6 respectively.
R5 vs. R8	Active intrusion by a neighboring or merging traffic participant is routed to R5, whereas passive or structural constraints that compress the ego vehicle's feasible space without active intrusion are routed to R8.
R6 vs. R8	R6 is used when an opposing participant actively creates spatial competition for limited shared driving space. R8 is used when the dominant risk arises from passive or structural compression of the ego vehicle's feasible space without an active opposing-flow conflict.
R8 vs. R9	R8 describes a reduction in feasible driving space while path connectivity remains unchanged, whereas R9 is used when feasible path connectivity changes and path re-selection becomes necessary.
R2 vs. R10	Route according to affected driving capability. Weather or airborne media that primarily degrade visibility are routed to R2, whereas road-surface or environmental conditions that primarily reduce safe control feasibility are routed to R10.

B Details on Latent Risk Reasoning Benchmark

B.1 Benchmark Statistics

Table 6 summarizes the scale and label distribution of the constructed benchmark, which contains 3,307 takeover clips, including 2,849 clips for training and 458 clips for testing. The split is performed at the clip level before scene sampling, so scenes from the same takeover event never appear in both sets. For each clip, we sample 4 scenes corresponding to explicit risk, late-stage latent risk, early-stage latent risk and no risk, resulting in 12,875 valid scenes in total after human review. We preserve the empirical frequency of 10 risk mechanisms observed in the collected takeover events. All 10 categories are represented in both training and testing sets, while their frequencies exhibit a clear long-tailed distribution. For example, R4 and R5 account for a large fraction of training risk-type annotations, whereas R5 and R8 are more frequent in test set. This distribution provides broad coverage of the proposed taxonomy.

Table 6: Statistics of latent risk reasoning benchmark after human review.

Statistic	**Train**	**Test**	**Total**
Dataset scale			
Takeover Clips	2,849	458	3,307
Scenes	11,050	1,825	12,875
Risk-stage distribution			
Explicit Risk	2,242	458	2,700
Latent Risk	5,148	899	6,047
No Risk	3,660	468	4,128
Risk-type distribution in risk scenes			
R1: Illumination-induced Degradation	274	102	376
R2: Medium-induced Visibility Degradation	162	54	216
R3: Occlusion-induced Potential Emergence	382	42	424
R4: Crossing-path Conflict Formation	2,649	108	2,757
R5: Same-direction Intrusion / Merging	1,777	532	2,309
R6: Opposing-flow Spatial Competition	742	50	792
R7: Behavior-induced Uncertainty	77	64	141
R8: Drivable-space Compression	1,050	346	1,396
R9: Path-connectivity Restructuring	200	31	231
R10: Control-feasibility Degradation	77	28	105
Risk-type Instances	**7,390**	**1,357**	**8,747**

B.2 Benchmark Construction and Annotation

Sampling Details. The benchmark is constructed from an industrial-scale real-world driving takeover dataset, where each clip contains an annotated human takeover timestamp T used as the temporal anchor for an explicit-risk candidate. For the same clip, we sample three candidate scenes before T: the lead-time priors for No Risk, Early-stage Latent Risk, and Late-stage Latent Risk follow $\mathcal{N}(\mu = 10\,\mathrm{s}, \sigma = 1\,\mathrm{s})$, $\mathcal{N}(\mu = 2\,\mathrm{s}, \sigma = 0.5\,\mathrm{s})$, and $\mathcal{N}(\mu = 1\,\mathrm{s}, \sigma = 0.2\,\mathrm{s})$, respectively. The approximately 10-s no-risk offset provides a temporally separated yet context-matched negative from the same clip, discouraging shortcut predictions based on static scene appearance rather than evolving risk evidence. These distributions provide temporal priors rather than fixed stage labels. During backtracking annotation, the expected Risk Stage is provided to the privileged annotation model as a recommendation rather than a mandatory constraint. The model may assign a different stage when the observed evidence does not support the recommended stage. Final labels are determined through evidence-based annotation and subsequent human quality control.

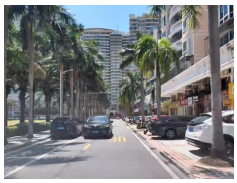
Annotation Details. For each retained clip, annotation is performed through a takeover-centered backtracking process, starting from takeover scene and progressively moving to earlier sampled scenes. The dialogue history is preserved across annotation rounds, such that when an earlier scene

is annotated, the annotator retains access to observations and annotations from the later stages of the same event. Such privileged context is used to verify how the current scene subsequently evolves and to support consistent annotation across different risk stages. At each round, the annotator receives current front-view and BEV images and outputs structured labels including Risk Stage, Risk Type, Risk Subjects, and corresponding reasoning. For risk-bearing scenes, the annotator additionally reports its confidence in the assigned Risk Type at three levels: high, medium, or low, reflecting the strength of supporting evidence and ambiguity between plausible risk mechanisms. For No Risk scenes, Risk Type and Risk Subjects are left unassigned, while an auxiliary rationale is retained for manual verification but excluded from quantitative evaluation. Importantly, this backtracking context is available only during benchmark annotation, whereas evaluated models receive only current-time observations. The complete annotation prompt is shown in Fig. 5, and an example output is shown in Fig. 6.

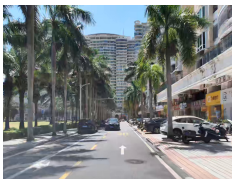
Human Review. We adopt a confidence-guided human review strategy for training-set annotations and exhaustive human review for test-set annotations. In training set, annotations with low Risk Type confidence or disagreement between the assigned and recommended Risk Stages are reviewed by human. All test-set annotations undergo human review. Reviewers inspect current observations together with corresponding future sequence and verify the assigned Risk Stage, Risk Type, Risk Subjects and Reasoning. Particular attention is paid to whether the annotated stage satisfies the stage definition, whether the selected risk type follows the routing rules in Appendix A, and whether the identified subjects and reasoning are supported by the observed risk evolution. Annotations with incorrect or inconsistent labels are revised, while scenes that cannot be reliably assigned to intended risk stage are discarded.

```
# Shared system instruction
Role: Autonomous-driving risk analysis expert
Input modality: Front-view image + local BEV image
Risk taxonomy: R1--R10 TAXONOMY AND ROUTING RULES
Risk stages: Explicit / Latent / No Risk
Task: Analyze the risk type and its evolution according to the predefined taxonomy.

# Stage and confidence assessment
- Suggested Risk Stage is a temporal prior, not a mandatory label.
- Assign the actual stage based on observable scene evidence.
- If evidence doesn't support suggested stage, revise and explain why.
- For risk-bearing scenes, report Risk Type confidence:
   High: Clear evidence with little ambiguity.
   Medium: Supported, but alternative mechanisms remain plausible.
   Low: Weak or ambiguous evidence.
- For No Risk, leave Risk Type, Subjects, and confidence unassigned.

# Multi-turn backtracking annotation
Annotation order: Takeover frame → late-stage candidate → early-stage candidate →
no-risk candidate
Dialogue history: Preserved across all annotation rounds. Earlier scenes retain access
to observations and annotations from later stages.

Round 1: Takeover Frame at T
# Annotate the manifested risk at takeover moment
Input: BEV_T, Front_T
Suggested Risk Stage: Explicit Risk
Instruction:
   - Determine the actual Risk Stage and Risk Type.
   - Describe the current risk situation.
   - Identify the subjects directly involved in the risk.
   - Report Risk Type confidence if a risk is identified.
Output:
   { signs_visible, risk_subjects, risk_level, risk_type, risk_type_confidence, description
}

Round 2–3: Late and Early Latent Candidates
# Backtrack to identify early risk cues
Input: Current BEV + Front image + preserved dialogue history
Suggested Risk Stage: Latent Risk
Instruction:
   - Determine whether early signs of the later risk are visible.
   - Assign the actual Risk Stage based on current evidence.
   - Identify currently visible or inferable Risk Subjects.
   - Infer risk evolution from current scene evidence.
   - Do not treat future-only observations as current evidence.
   - Report Risk Type confidence if a risk is identified.
Output:
   { signs_visible, risk_subjects, risk_level, risk_type, risk_type_confidence, reasoning }

Final Round: No-Risk Candidate
# Backtrack further to assess the absence of valid risk cues
Input: Current BEV + Front image + preserved dialogue history
Suggested Risk Stage: No Risk
Instruction:
   - Assess whether current evidence supports a valid driving risk.
   - Revise the suggested stage if risk evidence is present.
   - Explain the assigned stage using current scene evidence.
Output:
   { risk_level, reasoning }
```

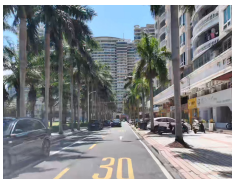
Figure 5: Prompt template for takeover-centered backtracking annotation. Temporal priors provide suggested risk stages, which the annotation model may revise based on observable evidence. Risk-type confidence guides subsequent human review, while all test-set annotations undergo exhaustive verification.



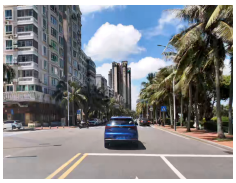
(a) Explicit Risk



(b) Late-stage Latent Risk



(c) Early-stage Latent Risk



(d) No Risk

Round 1: Explicit Risk
`risk_level`: Explicit
`risk_type`: R6 (High Confidence)
`signs_visible`: true
`reasoning`: An oncoming vehicle is approaching close to the road center on a narrow street where roadside parked vehicles limit lateral clearance, creating an immediate spatial conflict with the ego vehicle's forward corridor.
`risk_subjects`:

[1] **Oncoming dark-colored car** (BEV: true): ahead-left, approximately 15–20 m from the ego vehicle near the road center; directly competes with the ego vehicle for limited passing space.

[2] **Roadside parked vehicles** (BEV: true): along both sides of the road ahead, especially the right curb; restrict lateral avoidance space and intensify the opposing-flow conflict.

Round 2: Late-stage Latent Risk
`risk_level`: Latent
`risk_type`: R6 (High Confidence)
`signs_visible`: true
`reasoning`: The visible oncoming sedan is approaching along the left side of a roadway where parked vehicles constrain the ego vehicle's right-side clearance. There is not yet a clear occupation of the ego corridor, but continued convergence could create spatial competition and insufficient passing clearance, indicating a potential opposing-traffic-flow conflict.
`risk_subjects`:

[1] **Oncoming dark-colored sedan** (BEV: true): ahead-left in the opposing lane; its proximity to the road center may reduce safe lateral separation.

[2] **Roadside parked vehicles** (BEV: true): along both sides of the ego lane; reduce the ego vehicle's available clearance when passing opposing traffic.

Round 3: Early-stage Latent Risk
`risk_level`: Latent
`risk_type`: R6 (High Confidence)
`signs_visible`: true
`reasoning`: The oncoming dark sedan is clearly visible both in the front view and far ahead-left in the BEV. It is currently on the opposing side, but its proximity to the center line and the limited separation during the approach indicate possible spatial competition with the ego vehicle's forward corridor. This is a potential opposing-traffic-flow spatial conflict.
`risk_subjects`:

[1] **Oncoming dark sedan** (BEV: true): ahead-left near the center line; its close approach may reduce lateral passing clearance.

Round 4: No Risk
`risk_level`: No risk
`reasoning`: The ego vehicle has a clear, continuous lane and is proceeding under a green traffic signal. The blue vehicle ahead remains centered in the same lane with adequate separation and shows no visible abnormal behavior.

Figure 6: Example of takeover-centered backtracking annotation.

B.3 Evaluation Protocol

For Risk Type, Risk Subject and Reasoning, we construct scene-specific evaluation rubrics from ground-truth annotations, as illustrated in Fig. 7. The Risk Type rubric defines a small set of plausible confusable categories for each scene, while Risk Subject and Reasoning rubrics specify weighted semantic criteria for open-ended evaluation. Model predictions are then scored using fixed prompts shown in Fig. 8, with the same frozen rubrics shared across all methods for each scene.

Risk Stage. Risk stage is categorized as No Risk, Latent Risk and Explicit Risk. Since different stage errors have different severity, we use an asymmetric scoring scheme shown in Table 7. This asymmetric design accounts for the varying severity of different risk-stage prediction errors.

Table 7: Scoring scheme for risk-stage prediction.

Ground Truth / Prediction	No Risk	Latent Risk	Explicit Risk
No Risk	100	50	0
Latent Risk	0	100	50
Explicit Risk	0	50	100

Risk Type. For risk-positive scenes, we evaluate Risk Type based on ground-truth category and its scene-specific set of confusable risk types. Exact matches receive full points, while predictions in confusable types receive 30% of points, and all other predictions receive no points. Rather than using a fixed similarity between risk categories, such design assigns partial credit only to categories that exhibit plausible mechanism-level ambiguity, distinguishing exact matches, reasonable confusions and clearly incorrect predictions.

Risk Subject. Given the scene-specific subject rubric, Risk Subject is evaluated through semantic matching between the predicted and ground-truth subjects. For each ground-truth subject, we first identify the most semantically compatible predicted subject according to its description and spatial location, while enforcing one-to-one matching and excluding clearly incompatible subject types. Each matched pair is then evaluated along three dimensions, including *description*, *position* and *effect to ego*. For each dimension k, the evaluator assigns a matching level $q_{i,k} \in \{0, 0.5, 1\}$, corresponding to incorrect, partially correct, and correct predictions. Given the rubric weight $w_{i,k}$, the score of the i-th ground-truth subject is computed as

$$s_i = 100 \cdot \frac{\sum_k w_{i,k} q_{i,k}}{\sum_k w_{i,k}}. \tag{12}$$

Unmatched ground-truth subjects receive a score of 0. To further penalize redundant or hallucinated subjects, we apply an exponential penalty according to the number of unmatched predicted subjects. The final Risk Subject score is

$$S_{\text{subject}} = \frac{\sum_{i=1}^{N_{\text{gt}}} s_i}{N_{\text{gt}}} \cdot 0.9^{N_{\text{extra}}}, \tag{13}$$

where N_{gt} denotes the number of ground-truth risk subjects and N_{extra} denotes the number of unmatched predicted subjects. Such design jointly measures the completeness and accuracy of risk-subject identification while penalizing unsupported subject predictions.

Reasoning. Given the scene-specific reasoning rubric, Reasoning is evaluated by matching the predicted reasoning against a set of weighted semantic points. For each rubric item j, the evaluator determines whether the predicted reasoning is incorrect, partially correct, or correct, corresponding to a matching level $q_j \in \{0, 0.5, 1\}$. Let w_j denote the maximum weight assigned to the j-th semantic point. The Reasoning score is computed as

$$S_{\text{reason}} = 100 \cdot \frac{\sum_{j=1}^{M} w_j q_j}{\sum_{j=1}^{M} w_j}, \tag{14}$$

where M is the number of semantic points in the rubric. This scoring scheme evaluates whether the prediction captures the key evidence and risk-evolution relations in the reference reasoning, while allowing semantically equivalent expressions without requiring lexical overlap.

```
# Input shared by all rubric types
Current scene: Front-view image + BEV image
Ground-truth annotation: GT ANNOTATION
Risk taxonomy: R1--R10 TAXONOMY AND ROUTING RULES

# General principles
- Construct the rubric before evaluating any model prediction.
- Do not use model predictions during rubric construction.
- Abstract semantic requirements rather than copying exact GT
wording.
- Accept aliases and semantically equivalent expressions.
- Keep each scoring point concise and independently judgeable.
```

Risk Type Confusion Rubric

```
# Identify scene-specific confusable risk categories
Requirements:
  - Use the ground-truth Risk Type as reference category.
  - Select only categories that are plausibly confusable in current
scene.
  - Include at most 2 confusable Risk Types.
  - Do not include ground-truth Risk Type itself.
  - Provide a brief reason for each selected category.
  - Return an empty list if no category is reasonably confusable.

Output: [{confusable type, reason}, ...]
```

Reasoning Rubric

```
# Decompose the GT reasoning into weighted semantic requirements
Requirements:
  - Extract 3--4 key semantic points from the ground-truth reasoning.
  - Each point should capture one distinct piece of risk-relevant
evidence or evolution.
  - Assign an importance weight to each point.
  - The weights must sum to 100.

Output: [{pointID, criterion, weight}, ...]
```

Risk Subject Rubric

```
# Construct a separate rubric for each GT risk subject
Requirements:
  - Do not merge multiple ground-truth subjects.
  - Evaluate each subject along three dimensions:
    description: semantic identity or object description;
    position: spatial location or relation to the ego vehicle;
    effect: contribution to the current or potential risk.
  - Assign a weight to each dimension according to its importance.
  - The weights for each subject must sum to 100.

Output: [{subjectID, subject name, desc:(criterion,weight),
       position:(criterion,weight), effect:(criterion,weight)}, ...]
```

Figure 7: Prompt template for constructing scene-specific evaluation rubrics.

```
# Inputs
Scene: Front-view image + BEV image
Ground truth: GT ANNOTATION
Rubric: SCENE-SPECIFIC RUBRIC
Predictions: MODEL PREDICTIONS
```

Reasoning Scoring

```
# Evaluate each semantic point independently
- Compare each prediction with every reasoning rubric point.
- Judge semantic meaning rather than lexical overlap.
- Assign one matching level to each point:
  100%: correctly and sufficiently covered;
  50%: partially covered but incomplete;
  0%: missing, incorrect, or contradicted.
- Semantically equivalent expressions receive the same credit.
Output: [{point, level, justification}, ...]
```

Risk Subject Scoring

```
# Match subjects first, then score matched pairs
- Perform one-to-one semantic matching between GT and predicted
subjects.
- Match only subjects with compatible identity and spatial location.
- Do not force-match clearly incompatible subject types.
- Score each matched subject on description, position, and effect.
- For each field, assign 100%, 50%, or 0% using the same criteria
above.
- Mark missing GT subjects and unmatched predicted subjects
explicitly.
Output: [{subject, match/unmatch, description level, position level,
effect level}, ...]
```

Figure 8: Fixed scoring prompts for Reasoning and Risk Subject.

B.4 Evaluation Reliability

Risk Stage is evaluated using a predefined deterministic scoring matrix, while Risk Type scoring is deterministic given the scene-specific confusion rubrics. We therefore focus our LLM-judge reliability analysis on Risk Subject and Reasoning. We examine the evaluator from three perspectives: agreement with human judgment, robustness to the choice of judge model, and robustness to prediction order. For human agreement, we sample a 10% subset of the test set and ask 5 human evaluators to independently score Risk Subject and Reasoning using the same scene-specific rubrics. Their scores are averaged to obtain human reference scores. For cross-judge robustness, we use GPT-5.5 as primary judge and Qwen3.7-Plus as an independent judge, keeping evaluation rubrics and scoring criteria identical. For prediction-order robustness, we randomly shuffle the order of jointly evaluated predictions within each scene and repeat the evaluation using the same judge and rubrics. In all cases, we compute within-scene Spearman's rank correlation coefficient:

$$\rho_{\text{scene}}(e_1, e_2) = \frac{1}{|\mathcal{S}'|} \sum_{s \in \mathcal{S}'} \text{Spearman}\left(\{z_{s,m}^{(e_1)}\}_{m=1}^{M}, \{z_{s,m}^{(e_2)}\}_{m=1}^{M} \right), \tag{15}$$

where S' denotes test scenes, $z_{s,m}^{(e)}$ denotes the score assigned by evaluation setting e to method m on scene s. A higher ρ_{scene} indicates stronger consistency in the relative ranking of different methods. As shown in Table 8, results suggest that the proposed evaluation protocol provides stable relative comparisons across methods.

Table 8: Reliability analysis of the LLM-based evaluation protocol.

Reliability Test	Risk Subject	Reasoning
Human vs. GPT-5.5 Judge	0.904	0.892
GPT-5.5 vs. Qwen3.7-Plus Judge	0.929	0.958
Origin vs. Shuffled Prediction Order	0.940	0.943

C Details of Agent Framework

This section provides additional implementation details of the ALERT agent framework, including Skill organization, tool-guided reasoning, trajectory evaluation, Skill evolution, validation, merging, and test-time inference.

C.1 Skill Pool and Driving Tool Library

The Skill Pool mirrors the latent risk taxonomy: a routing index (`SKILL.md`) lists all categories with one-line activation cues, and each category directory holds subtype Skills specifying activation conditions, exclusion rules, analysis procedures, error-handling recovery rules and conditional tool guidance, together with a shared convention file that fixes entry formats and exact tool names. The driving tool library exposes ego and agent states, map geometry, drivable corridors and conflict relations. All tools are restricted to current and historical observations, as summarized in Table 9: no tool accesses future frames, and benchmark ground truth is visible only to the privileged annotation pipeline, never to the agent.

Table 9: Driving tool library and input provenance. Trajectories marked † are model-issued multi-modal predictions extrapolated from history, not ground-truth future. File write operations are available only to the Skill Evolution Agent, never during reasoning or inference.

Tool	Function	Source	Temporal scope
Ego state			
`get_ego_current_state`	Position, velocity, acceleration, yaw, shape	Sensor context	Current
`get_ego_history`	Past ego trajectory	Sensor context	History
`summarize_ego_motion_pattern`	Speed / acceleration / heading patterns	Derived	History
`get_ego_planned_trajectory`	Current planned reference path	Planner	Current plan
Surrounding agents			
`query_nearby_agents`	List agents around ego	Perception	Current
`get_agent_state/history`	Agent kinematics and past motion	Perception	Current / History
`summarize_agent_motion_pattern`	Agent behavior summary	Derived	History
`get_agent_predicted_trajectory`†	Multi-modal future motion modes	History-derived	Prediction
Map			
`get_lane_geometry/topology`	Lane boundaries and connectivity	HD map	Static
`locate_point_on_map`	Map element containing a query point	HD map	Static
`project_trajectory_to_map`	Map-project a motion trajectory	HD map	Derived
Corridor & conflict			
`build_ego_corridor_bundle`	Time-indexed ego occupancy envelope	Derived	Derived
`check_trajectory_corridor_intrusion`	Test a trajectory against the corridor	Derived	Derived
`check_batch_predicted_conflicts`	Batch cross-agent conflict screening	Derived	Derived
Skill knowledge (read-only during reasoning)			
`read/glob/grep`	Access Skill files	Skill Pool	Current pool

C.2 Skill-Guided Tool Reasoning

The Routing Agent reads the routing index and outputs a primary category with up to two alternatives. For each candidate, Tool Analysis Agents sample reasoning trajectories under the retrieved Skill. Each tool call must specify the hypothesis being examined. As evidence accumulates, an agent may reject its assigned category or issue a no-risk veto if the evidence contradicts the initial hypothesis. Fig. 9 shows the prompt structure and two representative trajectories, including one where evidence overturns the initial routing.

```
# Shared system instruction
Role: Skill-guided risk analysis agent. Input: front-view + local
BEV image. The routed category's SKILL.md is preloaded; every tool
call must declare which hypothesis it examines and whether the result
is decisive.

Trajectory 1: hypothesis confirmed (GT: latent R5)
# Scene DI20241111160108D2862947445, routed: R5 primary
query_nearby_agents → black sedan, front-left adjacent lane, ∼11 m
ahead
get_agent_predicted_trajectory → inward drift of 0.9 m toward the ego
lane over 2 s (examines: intrusion cue; decisive)
get_agent_current_state → edge-to-edge clearance ∼1.25 m while ego
closes at 6.5 m/s (examines: corridor margin; decisive)
check_batch_predicted_conflicts → predicted corridor overlap
(examines: conflict formation; decisive)
Output: latent R5; early cues: {inward drift, tight clearance,
predicted overlap}

Trajectory 2: evidence overturns routing (GT: no risk)
# Scene DI20241111200954D2868723585, routed: R5 primary
query_nearby_agents → white SUV, left adjacent lane, 7.8 m ahead
get_agent_predicted_trajectory → lateral offset 4.9→5.2 m over 2 s,
drifting away from the ego corridor (examines: intrusion cue;
decisive against)
get_agent_current_state → edge-to-edge clearance ∼3.15 m, ample
(examines: corridor margin; decisive against)
Output: no vehicle-specific intrusion cue; hypothesis rejected,
risk_level = none.
```

Figure 9: Real sampled trajectories from training under the same routed R5 Skill: tool evidence confirms the intrusion hypothesis in one scene and overturns it in another, with each call declaring the hypothesis it examines. Values are tool-returned observations.

C.3 Multi-Dimensional Evaluation

The Evaluation Agent jointly scores all sampled trajectories of a scene in one conversation, so that high- and low-scoring trajectories serve as within-scene controls. Scores aggregate five dimensions: risk classification (25), risk subjects (25), reasoning (20), evidence quality (10) and tool-call effectiveness (20). The first three reuse the benchmark dimensions of Eq. 4. Evidence quality has no ground-truth anchor and scores self-consistency and the honesty of confidence labels. Tool-call effectiveness rewards decisive, on-demand verification chains rather than enumerative calling. No-risk outputs are scored symmetrically against the ground-truth stage, and a wrong stage does not double-count against subject scoring. Diagnostic feedback is organized into five categories that directly guide subsequent Skill editing. Fig. 10 shows the abridged prompt.

C.4 Conservative Skill Editing

The Evolution Agent receives trajectories, scores, diagnoses and tool traces, and edits an isolated copy of the responsible Skill files through surgical file operations. Edits are constrained to reusable mechanisms: activation cues must be observable from sensors or map data; scene identifiers, coordinates and annotation text are forbidden and automatically audited; entries supported by successful cases may only be condensed; and a new subtype requires verified mechanism-level distinctness from existing subtypes. Fig. 11 shows a validated edit.

```
# System instruction (abridged)
Role: Reasoning Quality Evaluation Agent
Input: ALL SAMPLED TRAJECTORIES OF ONE SCENE + GT ANNOTATION
Task: Score all trajectories jointly across 5 dimensions (total 100
points):
  - Risk classification (25): risk_level AND risk type, scored
jointly.
  - Risk subjects (25): identification + effect to ego; hallucinated
subjects penalized.
  - Reasoning (20): risk mechanism; for latent scenes, the current
→ near-term → future evolution chain.
  - Evidence quality (10): no GT anchor; directionality,
evidence-to-conclusion coherence, confidence honesty.
  - Tool-call effectiveness (20): decisive on-demand chains over
enumerative calling.
# Symmetric no-risk handling
  - Correct veto (GT=none, pred=none): 90--100.
  - Missed detection / false alarm: 0--10.
  - A wrong stage must not double-count against subject scoring.
# Diagnosis types (one per trajectory)
  risk_perception / early_warning / causal_reasoning /
description_accuracy / tool_optimization
# Output
  {scores, diagnosis} as strict JSON; on judge failure the unit is
    re-queued, never silently scored.
```

Figure 10: Training-time Evaluation Agent prompt (abridged).

```
# Edit constraints (abridged)
Target reusable mechanisms, not scene-specific details. Activation
cues: observable from sensors or map data. Forbidden and audited:
scene_id, coordinates, annotation text. Entries supported by
successful cases: condense only, never delete.
```

Edit example: R4_crossing_conflict / SKILL.md, Output Check (batches 1–2)

```
# Before
OC-8: Do not treat ego deceleration or planned trajectory
adjustments as eliminating the risk when a dynamic target is actively
entering the ego corridor. Ego deceleration often indicates the
planner is responding to an immediate conflict, corroborating the
explicit nature of the risk rather than mitigating it to 'none' or
latent.
# After (validated edit)
OC-8: Do not treat ego deceleration or planned trajectory
adjustments as eliminating the risk when a dynamic target is actively
entering the ego corridor. Ego deceleration indicates the planner
is responding to a conflict, corroborating the existence of risk
rather than mitigating it to 'none'. However, do not automatically
escalate to explicit solely because the ego is decelerating; early
deceleration for a distant crossing target is a valid response to a
latent risk. Explicit classification still requires the target to
be currently occupying or immediately intersecting the ego's forward
passable space.
```

Figure 11: A real in-place edit of an existing entry (OC-8) early in training: the rule is tightened in both directions—deceleration must not justify a no-risk output, but must not alone escalate the stage to explicit either; stage escalation still requires an observable spatial criterion.

C.5 Task-Local Counterfactual Validation

Each candidate edit is re-evaluated through independently resampled trajectories (temperature 0.8) under both the original and edited Skill states. To reduce evaluator bias, the two sets are jointly scored in shuffled order without revealing their identities. The acceptance score equally combines a mean-based improvement term and a best-trajectory improvement term. An edit is accepted only if the combined improvement is positive and routing of the ground-truth category is not severely degraded; otherwise the original pool snapshot is restored. Appendix D.2 reports the full statistics of this gate over training; a large fraction of proposed edits is rejected under task-local resampling.

C.6 Conflict-Aware Batch Merging

At each batch boundary, files modified by a single scene are updated directly, while files modified by multiple scenes are consolidated by a Merge Agent within each risk category. The Merge Agent relies on actual edit operations (old→new) rather than model self-reports. It preserves non-overlapping edits, renumbers conflicting entry identifiers, and condenses overlapping statements. Every 25 batches, a grooming pass further merges duplicate lessons across sections and condenses bloated entries. Throughout this process, substantive lessons must be preserved, and only clear errors may be deleted. This keeps the Skill Pool manageable without losing validated knowledge.

D Additional Experimental Details and Analysis

D.1 Baseline and Evolution Configurations

Evolution configuration. Skill evolution uses a category-aware subset of 1,600 clips selected from the 2,849-clip training split, with increased representation of rare risk subtypes. The subset is organized into 200 batches, each containing eight clips and their four sampled scenes (32 scenes in total). The first three batches cover diverse risk categories and serve as a warm-up stage for initializing the Skill Pool. The backbone remains frozen, with up to three candidate categories and five sampled trajectories per scene at temperature 0.8 and at most eight tool-call rounds per trajectory. Counterfactual validation resamples trajectories under both Skill states at temperature 0.8 and rejects edits that severely degrade the ground-truth category's routing priority. The test split is not used during Skill evolution and retains its natural category distribution.

Knowledge-access baselines. Both retrieval arms share the backbone, routed hypotheses and reasoning pipeline of ALERT, replacing only the hierarchical Skill access with dense retrieval injected into the same guidance slot. Queries are built from the routing summary and candidate reasons without extra LLM calls, and retrieval is performed per candidate. The HyDE-style arm retrieves from the final evolved Skill Pool, chunked into 378 section-level entries; the ExpeL-style arm retrieves from 4,683 summarized successful trajectories (overall score $\geq$80 with correct risk classification, selected from 10,204 training records), reranked by similarity and hierarchy level. Both use text-embedding-v3 embeddings with top-5 injection. Since the HyDE-style arm draws from the same final pool as ALERT, this comparison isolates hierarchical navigation versus flat retrieval under identical knowledge content.

VLM-only baselines. Closed-source models receive the same structured task prompt, dual-view inputs, and high-level R1–R10 category definitions as ALERT, but have no access to subtype Skills, evolved guidance, or driving tools. The output schema is kept identical across methods.

Student SFT. The SFT variant fine-tunes Qwen3.5-4B with LoRA (rank 16, vision tower frozen) on the same training records: routing dialogues upsampled twice, plus Skill-guided reasoning trajectories with judge score $\geq$60 (5,977 kept after length filtering), split at the clip level to prevent leakage. Training runs for 2 epochs at learning rate 10^{-4} with cosine decay.

D.2 Counterfactual Validation Statistics

Table 10 summarizes the operating outcomes of the gate over the full training run. It rejects 45.7% of the $5{,}011$ candidate edits, with the rejection rate stabilizing at 40–48% after the initial evolution stage. Accepted and rejected proposals have mean resampled score changes of $+9.97$ and -3.65,

Table 10: Outcomes of task-local counterfactual validation over all 5,011 edit proposals raised during training. Δ is the blind-judge score change of the edited Skill Pool copy versus the original pool on freshly re-sampled trajectories. Rank is the position of the GT category among the ($\leq$3) routing hypotheses, with absence counted as rank 4; positive rank change means the GT category moves up.

Outcome	# Proposals	Ratio	Mean Δ	Mean rank $\Delta \uparrow$
Accepted	2,720	54.3%	$+9.97$	$+0.28$
Rejected	2,291	45.7%	-3.65	-0.05

respectively. Among the 731 cases where the ground-truth category is initially absent from the routed hypotheses, accepted edits recover it in 302 cases, whereas rejected edits recover none.

D.3 Training Cost

The full training run consumes ∼9,312M tokens over ∼340K LLM calls in ∼118 hours on a single machine with 32-way scene-level parallelism; multi-step tool reasoning accounts for 73% of the cost and skill editing for 21%, and ∼80% of tokens hit the prompt cache. Table 11 details the per-stage breakdown; a single scene costs ∼1.47M tokens end to end.

Table 11: Per-stage LLM cost of the full training run. Tokens in millions (M).

Stage	**Calls**	**Tokens (M)**	**Cache hit**
Routing	∼11.7K	197	31%
Tool reasoning	∼259K	6,770	83%
Evaluation	∼11.3K	129	5%
Skill editing	∼40.5K	1,930	83%
Counterfactual validation	∼9.4K	117	5%
Batch merging	3.3K	105	77%
Grooming	2.8K	64	91%
Total	∼340K	∼9,312	∼80%

D.4 Qualitative and Failure Analysis

Figure 12–14 provide representative examples of ALERT's latent-risk reasoning behavior. Figure 12 shows that ALERT can distinguish an emerging crossing tendency from an already manifested conflict through Skill-guided motion. Figure 13 illustrates how Skill evolution improves risk routing and evidence acquisition by redirecting attention from a salient but irrelevant lighting hypothesis to a subtle R5 intrusion risk. Figure 14 further shows that, even when the correct risk stage and mechanism are identified, ALERT may still miss secondary risk subjects and lead to incomplete reasoning. Together, these examples highlight both the benefits of evolved procedural knowledge and the remaining challenges in fine-grained latent-risk interpretation.



Current Scene. A pedestrian is tending to cross the ego lane.

DeepSeek-V4.1-Flash Overall Score: 84

Prediction: R4 / **Explicit Risk**

Risk Subject: Pedestrian crossing ahead-left.

Reasoning: The pedestrian is already crossing the marked crosswalk toward ego lane. The visible crossing behavior is an immediate conflict and Explicit Risk.

Reasoning Chain:
Read Category Definition → Observe Pedestrian → path conflict inferred → Explicit R4

ALERT Overall Score: 100

Prediction: R4 / **Latent Risk**

Risk Subject: Pedestrian ahead-left at ∼17 m.

Reasoning: The pedestrian shows a clear tendency to cross toward the ego lane. Motion prediction confirms lateral convergence, while corridor checking shows that the pedestrian has not yet entered the current ego corridor. The conflict is therefore evolving but not yet manifested.

Reasoning Chain:
Read Category Definition → Read Pedestrian Subskill in R4 → Query State → Predict Trajectory → Check Corridor → **Latent R4**

Figure 12: **Case study of early crossing-risk reasoning.** Both methods identify the pedestrian as the relevant risk subject, but the baseline prematurely treats the observed crossing behavior as an explicit conflict. ALERT further verifies the pedestrian's crossing tendency and current corridor relation, leading to the correct latent-risk judgment.



Current Scene. A large truck closely in the front-left adjacent lane under strong sunlight.

Before Evolution **Overall Score: 17.5**

Prediction: No Risk (R1 hypothesis rejected)

Risk Subject: None.

Reasoning: Strong sunlight is visible, but the road, lane markings, and surrounding vehicles remain clearly observable. Since the lighting does not degrade driving-relevant perception, I reject the R1 hypothesis and classify the scene as No Risk.

Reasoning Chain:
Strong sunlight → Read R1 Skill → Check visibility degradation → R1 rejected → No Risk

After Evolution **Overall Score: 98**

Prediction: R5 / **Latent Risk**

Risk Subject: Cement mixer in the front-left adjacent lane.

Reasoning: The truck has only ∼1.35 m lateral clearance and a ∼6.3 m longitudinal gap from the ego vehicle. Its predicted trajectory shows a subtle inward drift toward ego corridor, while the faster ego vehicle is rapidly closing the gap. The truck has not yet intruded into the ego lane, indicating an emerging rather than explicit R5 conflict.

Reasoning Chain:
Read Category Definition → Read Adjacent-Proximity Subskill in R5 → Measure Clearance → Predict Inward Drift → Check Relative Motion → **Latent R5**

Figure 13: **Case study of Skill evolution.** Before evolution, the agent focuses on the salient sunlight and terminates after rejecting an R1 visibility risk, missing the nearby truck interaction. After evolution, ALERT identifies the front-left cement mixer as the relevant risk subject and combines spatial clearance, predicted inward drift, and relative motion to recognize the emerging R5 intrusion risk. This example illustrates how evolved procedural knowledge improves both risk routing and evidence-guided latent-risk reasoning.



Current Scene. Parked vehicles constrain ego driving space, with a pedestrian wandering near the parked-vehicle row, unsure whether he will cross ego at front.

ALERT Failure Case **Overall Score: 65**

Prediction: R8 / **Latent Risk**

Risk Subjects: Parked vehicles at the right roadside.

Reasoning: Multiple parked vehicles reduce the available lateral margin along the right side of the future corridor. The current path remains passable, but the available driving space becomes progressively constrained ahead, indicating latent drivable-space compression.

Reasoning Chain:
Read Category Definition → Read Parking-Compression Subskill in R8 → Query Nearby Agents → Build Ego Corridor → Verify Reduced Clearance → **Latent R8**

Failure: Although the risk mechanism is correctly identified, the reasoning fails to include the pedestrian beside the parked-vehicle row, resulting in incomplete coverage of the factors contributing to the reduced lateral clearance.

Figure 14: **Failure case of incomplete latent-risk reasoning.** ALERT correctly recognizes latent R8 drivable-space compression, but focuses on the parked vehicles and spatial constraints while overlooking the nearby pedestrian. The case shows that correct stage and mechanism recognition does not always guarantee complete identification of all risk-relevant subjects and causal factors.