Causal Scene Narration with Runtime Safety Supervision for Vision-Language-Action Driving

Anonymous authors
Paper under double-blind review

Abstract

Vision-Language-Action (VLA) models for autonomous driving must integrate diverse textual inputs, including navigation commands, hazard warnings, and traffic state descriptions, yet current systems often present these as disconnected fragments, forcing the model to discover on its own which environmental constraints are relevant to the current maneuver. We introduce Causal Scene Narration (CSN), which restructures VLA text inputs through intent-constraint alignment, quantitative grounding, and structured separation, at inference time with zero GPU cost. We complement CSN with Simplex-based runtime safety supervision and training-time alignment via Plackett-Luce DPO with negative log-likelihood (NLL) regularization. A multi-town closed-loop CARLA evaluation shows that CSN improves Driving Score by +31.1% on original LMDrive and +24.5% on the preference-aligned variant. A controlled ablation reveals that causal structure accounts for 39.1% of this gain, with the remainder attributable to information content alone. A perception noise ablation confirms that CSN's benefit is robust to realistic sensing errors. Semantic safety supervision improves Infraction Score, while reactive Time-To-Collision monitoring degrades performance, demonstrating that intent-aware monitoring is needed for VLA systems.

1 Introduction

Vision-Language-Action (VLA) models combine visual perception with language-conditioned action prediction for end-to-end autonomous driving (Shao et al., 2024; Tian et al., 2024; Mao et al., 2023). In these systems, camera images and LiDAR point clouds are encoded by vision backbones, while natural language provides navigation goals and scene context. Recent results show that text input quality affects driving performance. TLS-Assist (Schmidt et al., 2025) improved LMDrive's driving score by 14.1% simply by injecting structured traffic light messages, and GraphPilot (Schmidt et al., 2026) achieved 15.6% through scene graph serialization, both without any retraining.

These results are typically attributed to the text carrying more information. We argue instead that the operative variable is *causal structure*: whether the text explicitly links what the agent intends to do with what the environment requires it to consider. In causal inference (Pearl, 2009), observing that two variables co-vary does not tell us whether intervening on one will change the other. Similarly, presenting "Turn left" and "Pedestrian ahead" as co-occurring fragments does not tell the model whether the pedestrian is relevant to the turn. Current VLA systems have three related weaknesses.

First, existing systems generate navigation commands and hazard notices as *causally unrelated fragments*. LMDrive (Shao et al., 2024), for example, presents 'Turn left' and 'Pedestrians ahead' separately. The model must independently discover that the pedestrian is relevant *because* the left turn will cross its trajectory. A human instructor would instead say: 'Turn left, *but* yield to the pedestrian at 12 m.' This causally structured utterance links intent to constraint, which is the same structure that DriveVLM (Tian et al., 2024), DriveLM (Sima et al., 2024), and SteerVLA (Gao et al., 2026) each provide through different mechanisms.

Second, VLA models offer no runtime safety guarantees. They operate as open-loop predictors at each timestep, and once an unsafe action is predicted, there is no mechanism to intercept and correct it before execution. Training-time alignment alone cannot guarantee safety across the full distribution of deployment

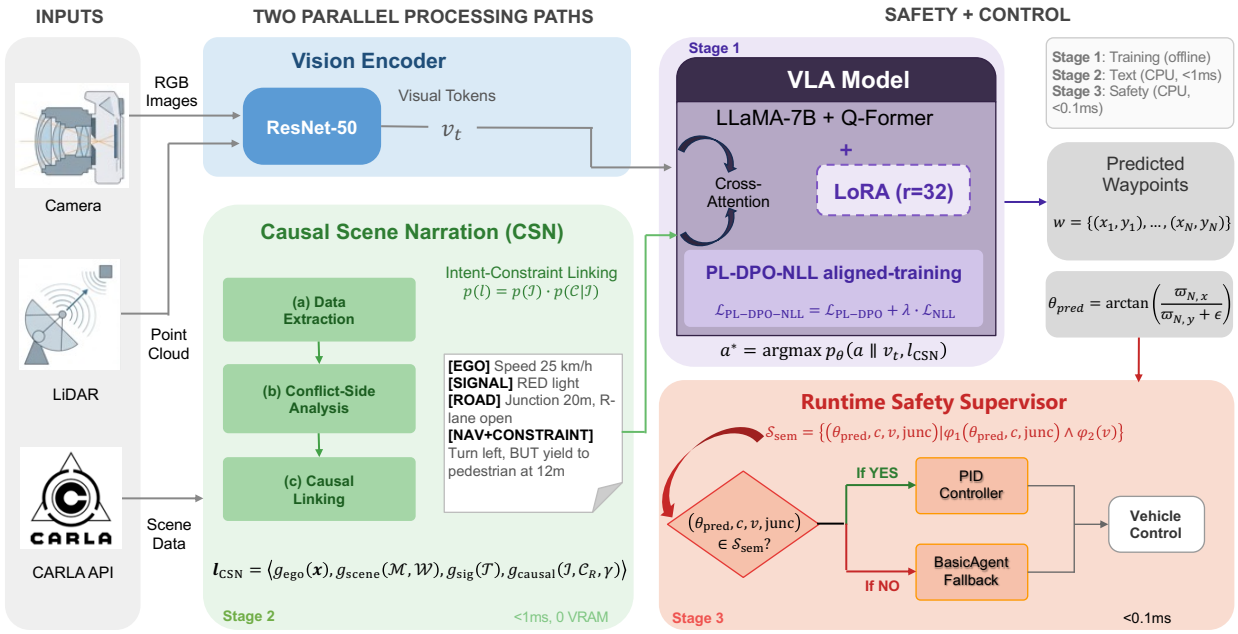

Figure 1: Three-stage framework overview. CSN (Stage 2) restructures driving-environment information into structured text, while the runtime safety supervisor (Stage 3) monitors VLA waypoints against $\mathcal{S}_{\text{sem}}$. Both modules operate on CPU.

scenarios, particularly in rare or adversarial situations that are underrepresented in the training data. The Simplex architecture (Sha, 2001), applied in safety-critical systems such as the Boeing 777 flight controller, provides a well-studied solution: use a simple, verifiable controller to guard against failures of a complex one.

Third, preference-aligned models suffer from distribution shift. Preference optimization methods such as DPO (Rafailov et al., 2023) and Multi-PrefDrive (Li et al., 2025) improve in-distribution driving but can overfit to the training environment, hurting generalization to unseen towns. Because inference-time text enrichment leaves model weights unchanged, it sidesteps this failure mode and can complement or even replace training-time alignment.

We address these limitations at three stages of the VLA pipeline. At inference time, **Causal Scene Narration** (CSN, §4.2) restructures VLA text inputs around intent-constraint causal alignment, quantitative physical grounding, and structured information separation. The resulting text mirrors the perception-prediction-planning reasoning chain rather than presenting disconnected observations, and the entire pipeline runs on CPU with no additional GPU memory. Also at inference time, a **Runtime Safety Supervisor** (§4.3) monitors VLA outputs against classical planner trajectories and intervenes when potentially unsafe actions are detected, providing safety guarantees that training-time alignment alone cannot offer. At training time, **PL-DPO-NLL** (§4.4) combines Plackett-Luce multi-preference ranking with NLL regularization for safety alignment, though our multi-town evaluation reveals that this adaptation introduces distribution shift, which motivates CSN as an alternative that does not modify model weights.

We test this hypothesis through an empirical decomposition (§3.3) showing that when intent and constraints appear as isolated text fragments, the model must discover their relationship through implicit cross-attention, whereas explicitly encoding this relationship via causal connectives lets the model condition on richer structure without any weight change. Our ablation on both weight configurations confirms that 39.1% of CSN's improvement on original LMDrive stems from causal structure rather than information quantity alone, and that this ratio decreases on the preference-aligned variant where the model has already internalized some causal reasoning through training.

Our contributions are:

1. **Causal Scene Narration framework**: We identify the absence of intent-constraint links as a key limitation of VLA text inputs, organize existing approaches in a taxonomy (L0–L3) by their level of structured linking, and propose CSN, a zero-VRAM text enrichment pipeline built on intent-constraint alignment, quantitative physical grounding, and structured information separation, justified by a controlled ablation.

2. **Multi-town evaluation**: A ten-configuration ablation (16 routes, 8 towns, N=5 repetitions, 95% bootstrap confidence intervals [CIs]) shows that CSN benefits both tested weight configurations with overlapping CIs, that semantic safety supervision outperforms reactive TTC on both configurations, and that causal structure accounts for 39.1% of CSN's gain on original LMDrive but only 13.5% on the preference-aligned variant, revealing an interaction between explicit text structure and learned causal reasoning.

2 Related Work

2.1 VLA Models and Text Structure for Driving

End-to-end autonomous driving increasingly uses VLA models for closed-loop control. GPT-Driver (Mao et al., 2023) reformulated motion planning as language modeling, and showed that autoregressive language models can generate plausible driving trajectories from text-encoded scene states. DriveVLM (Tian et al., 2024) introduced a three-stage Chain-of-Thought (CoT) pipeline consisting of scene description, scene analysis, and hierarchical planning, showing that when text mirrors the causal reasoning process, long-tail scenario handling improves substantially. DriveLM (Sima et al., 2024) employed graph-structured visual question answering to create logical dependency chains between perception, prediction, and planning nodes, explicitly encoding causal relationships.

LMDrive (Shao et al., 2024) achieves closed-loop control via Q-Former alignment of visual tokens with text, using LLaVA-v1.5 as backbone. Its template-based instruction planner generates navigation commands and hazard notices as causally disconnected fragments, representing the minimal end of the text structure spectrum. SteerVLA (Gao et al., 2026) showed that replacing sparse routing commands with fine-grained meta-actions improved driving score by 4.77 points on Bench2Drive, with meta-actions carrying implicit causal structure.

Among text-enrichment approaches, those that outperform LMDrive consistently provide text with richer causal structure, whether through CoT decomposition, graph-structured QA, multi-channel separation, or fine-grained meta-actions.

Several studies demonstrate that text enrichment works even without retraining. TLS-Assist (Schmidt et al., 2025) achieved +14.1% DS on LMDrive by injecting structured traffic light messages *without retraining*, showing that LMDrive's pre-trained LLaMA backbone can use structured text never seen during fine-tuning. GraphPilot (Schmidt et al., 2026) achieved +15.6% through scene graph serialization, where relational structure ('pedestrian *is-crossing* ego-lane *conflicts-with* intended left turn') supports the causal structure hypothesis. SimLingo (Renz et al., 2025) found no improvement from post-hoc CoT narration. One possible explanation is that their CoT narrated intended actions without connecting environmental observations to action decisions, though other factors may also contribute.

2.2 Runtime Safety for Autonomous Driving

Existing runtime safety approaches occupy two categories. *Formal frameworks* include Responsibility-Sensitive Safety (RSS) (Shalev-Shwartz et al., 2017), which defines safe distance envelopes and triggers proper responses (braking) when violated; Control Barrier Functions (CBFs) (Ames et al., 2019), which enforce forward invariance of safe sets; and Simplex switching (Sha, 2001; Phan et al., 2020), where a verified safety controller runs alongside an unverified high-performance controller. *Runtime verification* methods in-

clude STL monitoring (Desai et al., 2017) and shield synthesis (Alshiekh et al., 2018), which check controller behavior against temporal logic specifications.

All these methods operate on physical state (distances, velocities) and are not designed to detect *semantic-level* VLA failures such as direction misinterpretation or hallucinated scene elements. Leading E2E driving methods lack explicit runtime safety layers. UniAD (Hu et al., 2023) jointly optimizes perception through planning but errors propagate unchecked. VAD (Jiang et al., 2023) introduces planning constraints that are training-time loss functions not enforced at inference. Chen *et al.* (Chen et al., 2022) and Jaeger *et al.* (Jaeger et al., 2023) document that waypoint predictions fail specifically at junctions due to a "target point shortcut" where models steer toward the nearest GPS waypoint rather than following road geometry. Our semantic monitor addresses one instance of this failure mode, specifically direction inconsistency during junction approach.

Our runtime supervisor instantiates the Simplex architecture (Sha, 2001) with a safety envelope reformulated for the semantic domain (§4.3), targeting direction consistency and liveness rather than physical distance maintenance.

2.3 Training-Time Safety Alignment

DPO (Rafailov et al., 2023) and its variants optimize policies on preference data but face probability collapse, where chosen action likelihoods decrease during optimization (Pang et al., 2024; Razin et al., 2024). Multi-PrefDrive (Li et al., 2025) applied multi-preference tuning to LLM-based autonomous driving, demonstrating improved in-distribution performance through Plackett-Luce ranking (Plackett, 1975) over multiple candidate actions. NLL regularization provides an explicit likelihood floor against probability collapse. We combine both in our PL-DPO-NLL objective (§4.4).

3 Theoretical Foundations

3.1 Text Structure as a Performance Bottleneck

In structural causal models (Pearl, 2009), the presence or absence of a directed arrow between two variables encodes a causal assumption. An arrow from X to Y asserts that X influences Y, while a missing arrow asserts independence. The same logic applies to VLA text inputs. Let $\mathcal{I}$ denote the navigation intent (*e.g.*, "turn left") and $\mathcal{C} = \{c_1, \ldots, c_K\}$ the environmental constraints (*e.g.*, pedestrians, vehicles, traffic lights). The correct driving action $\boldsymbol{a}$ depends not on $\mathcal{I}$ and $\mathcal{C}$ separately, but on their causal interaction, i.e., which constraints are relevant *given* the current intent. A left-turn intent makes a crossing pedestrian safety-critical; the same pedestrian is irrelevant during straight driving.

Template systems present $\mathcal{I}$ and $\mathcal{C}$ as independent text fragments, with no link connecting them. Conceptually, this is equivalent to omitting the edge $\mathcal{I} \to \mathcal{C}$ from a dependency graph: the text encodes both variables but not their relationship. The LLM must recover the missing dependency internally through multi-layer cross-attention (Vaswani et al., 2017), without any explicit signal indicating which constraints are relevant to the current intent. Prior work supports the broader claim that text acts as a reasoning scaffold. LMDrive's own ablation on the LangAuto benchmark (Shao et al., 2024) showed that adding notice instructions significantly reduces collisions, even though the visual information was always available. TLS-Assist (Schmidt et al., 2025) showed the same pattern: the model could always see traffic lights in the image, but without explicit textual mention, those visual features were insufficiently weighted. These results suggest that text directs the model's attention (Wei et al., 2022) rather than just adding information. Whether *causal structure* within the text provides additional benefit beyond information content is the specific question our CSN vs. Flat Text ablation addresses (§5.2).

CSN restores the missing link by explicitly encoding which constraints are relevant *given* the current intent, using linguistic causal connectives to express this conditional relationship, as illustrated in Fig. 2. With K constraints, the model must evaluate $O(K)$ potential pairings to discover which ones matter for the current intent, whereas CSN pre-selects the $R \ll K$ relevant ones.

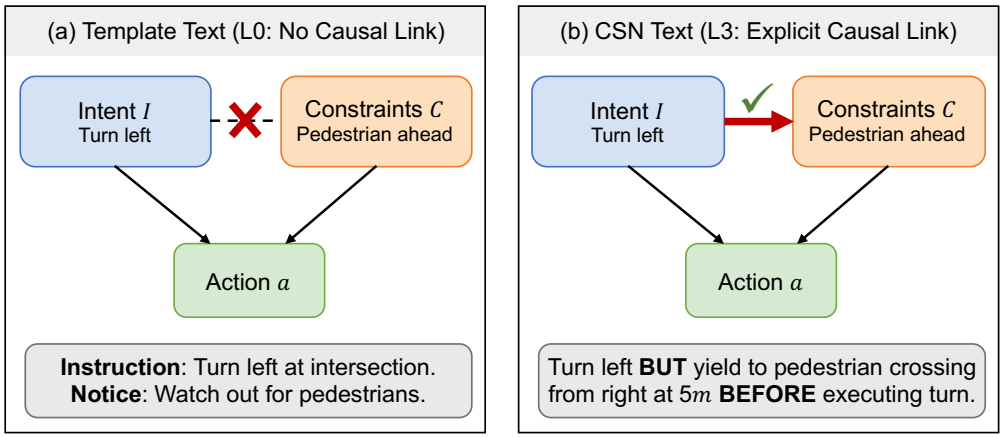

Figure 2: The text structure bottleneck. (a) Template text presents intent $\mathcal{I}$ and constraints $\mathcal{C}$ as independent fragments with no $\mathcal{I} \to \mathcal{C}$ link. (b) CSN restores the causal link via explicit connectives (**BUT**, **BEFORE**).

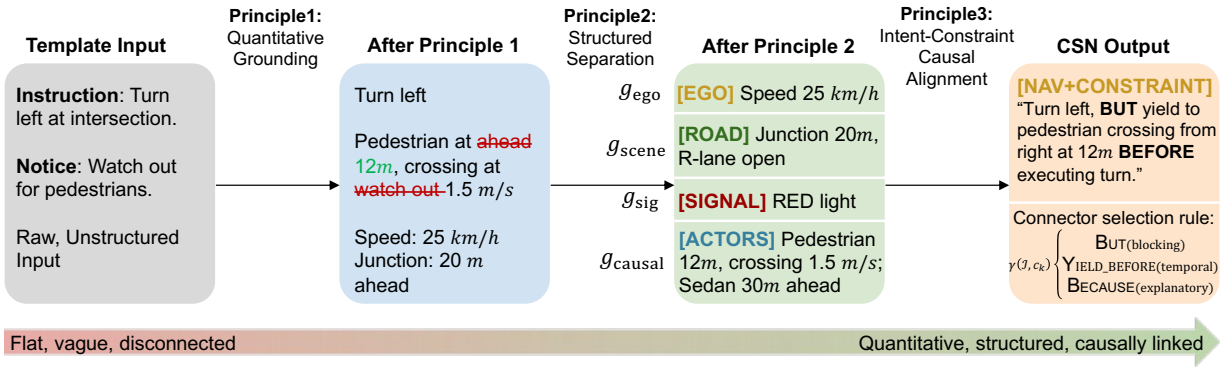

Figure 3: CSN pipeline illustrated on a left-turn scenario. Each principle progressively transforms flat template text into quantitative, structured, and causally linked input.

3.2 Three Principles of CSN

CSN transforms standard template text through three operations, each targeting a specific weakness of flat VLA text inputs. Fig. 3 illustrates the full pipeline on a left-turn scenario.

(1) Quantitative physical grounding. Template terms like 'ahead' and 'watch out' carry no physical dimensions. The LLM cannot tell whether a threat is 10 m or 50 m away, yet this difference dictates the required response (Shalev-Shwartz et al., 2017). CSN replaces all vague qualifiers with exact metric values: distances in meters, speeds in km/h, and timing in seconds. For instance, translating "watch out for pedestrians" into "Pedestrian at 12 m, crossing at 1.5 m/s" allows the model to estimate a 4-second safety window.

(2) Structured information separation. Following findings that structured representation outperforms flat descriptions (Tian et al., 2024; Schmidt et al., 2026), CSN organizes the environment state into a four-part sequence (denoted $\langle\cdot\rangle$ for ordered concatenation) mirroring the perception-prediction-planning chain:

$$\begin{aligned} \boldsymbol{l}_{\text{CSN}} = \big\langle\ & g_{\text{ego}}(\boldsymbol{x}),\ \ g_{\text{scene}}(\mathcal{M}, \mathcal{W}), \\ & g_{\text{sig}}(\mathcal{T}),\ \ g_{\text{causal}}(\mathcal{I}, \mathcal{C}_R, \gamma)\ \big\rangle\,. \end{aligned} \tag{1}$$

Here, g_{ego} encodes ego-state $\boldsymbol{x} = (v, \omega)$; g_{scene} describes road topology $\mathcal{M}$ and weather $\mathcal{W}$; and g_{sig} details traffic signals $\mathcal{T}$. These first three components act as independent observation encoders. The final component,

g_{causal}, is where the reasoning occurs: it fuses the navigation intent $\mathcal{I}$ with a filtered set of relevant constraints $\mathcal{C}_R \subseteq \mathcal{C}$, selecting which detections matter for the current maneuver.

(3) Intent-constraint causal alignment. The third operation links intent to constraints using explicit causal connectives. Human instructors use connectives to reveal the *nature* of a conflict (*e.g.*, "Turn left, BUT..." vs. "Reduce speed BECAUSE..."). CSN mechanizes this by classifying each relevant constraint $c_k \in \mathcal{C}_R$ into one of three conflict types (τ_k) and assigning a connective via a selection function γ:

$$\gamma(\mathcal{I}, c_k) = \begin{cases} \textsc{But} & \text{if } \tau_k = \text{blocking} \\ \textsc{Yield\_Before} & \text{if } \tau_k = \text{temporal} \\ \textsc{Because} & \text{if } \tau_k = \text{explanatory} \end{cases} . \tag{2}$$

To determine τ_k, let z_k denote the spatial zone of the constraint c_k, and $\mathcal{Z}_{\mathcal{I}}$ denote the conflict-side zones for the ego-intent $\mathcal{I}$ (*e.g.*, $\mathcal{Z}_{\text{left-turn}} = \{\text{ahead-left, ahead, left}\}$). The classification follows spatial-temporal rules:

- **Blocking** (τ_k = blocking): A stationary obstacle occupies the intended path ($z_k \in \mathcal{Z}_{\mathcal{I}}$). *Example: a stopped vehicle directly ahead.*
- **Temporal** (τ_k = temporal): A moving actor intersects the intended path ($z_k \in \mathcal{Z}_{\mathcal{I}}$). The conflict has a time dimension and may resolve if the ego yields. *Example: a crossing pedestrian during a left turn.*
- **Explanatory** (τ_k = explanatory): The constraint lies outside the immediate conflict zone ($z_k \notin \mathcal{Z}_{\mathcal{I}}$) but provides necessary context. *Example: a speed limit reduction or a vehicle in an adjacent, non-conflicting lane.*

When a constraint satisfies multiple conditions, priority is assigned as blocking > temporal > explanatory. By embedding these connectives, CSN provides the LLM with explicit causal cues that it is already optimized to process from its natural language pre-training.

3.3 Empirical Decomposition of Text Utility

A central question underlies CSN's design: does the model benefit simply from receiving *more* environmental information, or specifically from the *causal organization* of that information? To answer this, we introduce an intermediate baseline $\boldsymbol{l}_{\text{disc}}$ by setting $\gamma = \varnothing$ in Eq. (1), so $g_{\text{causal}}(\mathcal{I}, \mathcal{C}_R, \varnothing)$ provides the same quantitative facts as CSN (distances, speeds, states) but presents them as disconnected fragments without causal connectives. This yields a clean three-way comparison: template → disconnected → CSN. The total driving-score gain over the baseline decomposes as:

$$\begin{aligned} \Delta\text{DS}_{\text{total}} &= \text{DS}(\boldsymbol{l}_{\text{CSN}}) - \text{DS}(\boldsymbol{l}_{\text{template}}) \\ &= \underbrace{\left[\text{DS}(\boldsymbol{l}_{\text{disc}}) - \text{DS}(\boldsymbol{l}_{\text{template}})\right]}_{\text{Utility}_{\text{info}}} + \underbrace{\left[\text{DS}(\boldsymbol{l}_{\text{CSN}}) - \text{DS}(\boldsymbol{l}_{\text{disc}})\right]}_{\text{Utility}_{\text{struct}}} . \end{aligned} \tag{3}$$

Here, $\text{Utility}_{\text{info}}$ captures the gain from quantitative grounding and structured separation alone, while $\text{Utility}_{\text{struct}}$ isolates the additional gain from explicitly aligning intent with constraints. A positive $\text{Utility}_{\text{struct}}$ demonstrates that VLA performance is bottlenecked not merely by information quantity, but by the model's ability to infer causal dependencies from flat text. Our ablation (§5.2) finds $\text{Utility}_{\text{struct}}/\Delta\text{DS}_{\text{total}} = 39.1\%$ on original LMDrive, confirming that causal structure contributes independently of information quantity.

3.4 Taxonomy of Text Structure Approaches

We organize existing VLA text approaches by their *causal structure level* (Table 1), where L denotes the level of causal linking: L0 provides isolated commands with no environmental context; L1 adds structured factual

Table 1: Taxonomy of VLA text approaches by causal structure level. DS Gain values are from each work's own evaluation setup and are not directly comparable across rows due to differing routes, weather, and CARLA versions.

Approach	Level	VRAM	Retrain	DS Gain	Key Mechanism
LMDrive template (Shao et al., 2024)	L0	0	–	baseline	Isolated instruction + notice
TLS-Assist (Schmidt et al., 2025)	L1	0	No	+14.1%	Structured signal messages
GraphPilot (Schmidt et al., 2026)	L2	0	No	+15.6%	Entity-relationship graph text
SteerVLA (Gao et al., 2026)	L2	0	Yes	+4.77 pts	Fine-grained meta-actions
DriveVLM CoT (Tian et al., 2024)	L3	High	Yes	–	3-stage causal chain
DriveLM graph QA (Sima et al., 2024)	L3	High	Yes	–	Graph dependency chains
CSN (ours)	**L3**	**0**	**No**	**+31.1%**	**Intent-constraint alignment**

information; L2 introduces entity-level relationships, scene graphs, or fine-grained action decomposition; and L3 explicitly models the causal dependence between navigation intent and environmental constraints.

Since DS Gain values come from different evaluation setups, we do not claim a strict correlation between level and performance. However, all Level 3 approaches share the property of explicitly modeling the conditional dependence between constraints and navigation intent, as formalized in §3.1. To our knowledge, CSN is the first L3 approach to achieve this intent-constraint alignment entirely at inference time without additional VRAM or model retraining.

4 Methodology

4.1 System Architecture Overview

Our framework operates at three stages of the VLA pipeline, as shown in Fig. 1. At training time, PL-DPO-NLL (§4.4) fine-tunes the base LLaMA-7B model on preference data for safety alignment. At inference time, Causal Scene Narration (§4.2) restructures text inputs from driving-environment data, and a runtime safety supervisor (§4.3) monitors VLA outputs via direction-conflict detection. Both inference modules operate at zero GPU cost.

4.2 Causal Scene Narration Pipeline

The CSN pipeline converts driving-environment information into structured natural language following the three principles established in §3.2.

4.2.1 Environmental Data Extraction

We extract four categories of environmental data from CARLA's Python API, all computed on CPU: (1) dynamic actor states, including all vehicles and pedestrians within 50 m forward and 15 m lateral, with position and velocity in the ego-vehicle frame, and spatial zone classification (ahead/behind/left/right, near/mid/far); (2) traffic infrastructure, including light state, elapsed timing, and speed limits; (3) road topology, including junction proximity, lane availability, and curvature; and (4) environmental conditions, including precipitation, fog density, wetness, and sun altitude. In this work, we use CARLA's privileged API to isolate and evaluate the impact of text structure independently of perception noise. Replacing this with a vision-based perception stack is discussed in §6.

4.2.2 Causal Narration Generation

Given the navigation command $\mathcal{I}$ and detected constraints $\mathcal{C}$, the algorithm first filters $\mathcal{C}$ for relevance to $\mathcal{I}$ via conflict-side analysis, following GraphPilot (Schmidt et al., 2026). Constraints on the conflict side of the intended maneuver (*e.g.*, ahead-left, ahead, and left for a left turn) receive higher priority than those on non-conflict sides. The filtered constraints are then ranked by urgency based on proximity and dynamic

Table 2: Text input comparison for a left-turn scenario. Causal connectives shown in bold.

(a) Template (LMDrive original)	
Instruction	Turn left at intersection.
Notice	Watch out for pedestrians.
(b) +Flat Text (same facts, no causal links)	
Instruction	Turn left. Speed 25 km/h. Pedestrian 5m right crossing left. Sedan 12m ahead 30 km/h. RED light. Junction 20m.
(c) CSN (causal structure)	
Instruction	Turn left at intersection, **BUT** yield to pedestrian crossing from right at 5m **BEFORE** executing turn. Maintain distance from sedan ahead.
Notice	[EGO] 25/30 km/h. [ROAD] Junction 20m, R-lane open. [SIGNAL] RED. [ACTORS] Sedan 12m ahead 30 km/h. Ped 5m R, crossing L.

state. Finally, each relevant constraint is linked to $\mathcal{I}$ with a causal connective selected by the γ function defined in Eq. (2). The entire pipeline runs in $<$1 ms per frame on CPU.

Table 2 contrasts the three text conditions used in our ablation.

4.3 Runtime Safety Supervision

The Simplex architecture (Sha, 2001) embodies the principle of *using simplicity to control complexity*: a simple, verifiable controller guards against failures of a complex, high-performance one. We instantiate this principle for VLA driving, with safety properties formulated in Signal Temporal Logic (STL) (Desai et al., 2017) and monitored online via lightweight counter-based evaluation.

4.3.1 Simplex Architecture for VLA Driving

The runtime supervisor monitors whether the VLA's output remains inside a semantic safety envelope (Shalev-Shwartz et al., 2017) and triggers a controller switch when violations are detected. We define:

$$\mathcal{S}_{\text{sem}} = \left\{ (\theta_{\text{pred}}, c, v, \texttt{junc}) \mid \varphi_1(\theta_{\text{pred}}, c, \texttt{junc}) \wedge \varphi_2(v) \right\}, \tag{4}$$

where φ_1 enforces direction consistency with junction-aware gating and φ_2 enforces liveness, both formalized below. This is realized through a Simplex switching architecture (Sha, 2001) with three components. In the original Simplex terminology, the *High-Performance Controller* (HPC) is the complex but hard-to-verify subsystem, while the *High-Assurance Controller* (HAC) is the simple, conservative fallback. Our instantiation maps these as follows: the **Advanced Controller (AC)**, corresponding to the HPC, is the VLA model (LMDrive + PL-DPO-NLL LoRA), which outputs waypoint trajectories $\boldsymbol{w}_{\text{VLA}} = \{(x_i, y_i)\}_{i=1}^{N}$ in the ego-vehicle frame. The **Baseline Controller (BC)**, corresponding to the HAC, is CARLA's Traffic Manager, a rule-based planner with access to the HD map and ground-truth actor positions that provides a reliable fallback when the VLA fails. The **Decision Module (DM)** evaluates safety envelope membership $(\theta_{\text{pred}}, c, v, \texttt{junc}) \in \mathcal{S}_{\text{sem}}$ and switches control authority to the BC when the current state exits the envelope.

The switching logic follows bidirectional Simplex (Phan et al., 2020). Upon exiting $\mathcal{S}_{\text{sem}}$, control transfers from AC to BC for a minimum intervention period $T_{\min}$ of 20 steps, approximately 1 s at 20 FPS. The BC maintains control until the semantic safety envelope is re-entered, at which point authority returns to the AC. Unlike classical safety envelopes that monitor physical distances between vehicles, our DM monitors the semantic consistency between the VLA's predicted actions and the intended navigation command.

4.3.2 Safety Specifications

We define two safety properties targeting the dominant VLA failure modes identified by Chen *et al.* (Chen et al., 2022) and Jaeger *et al.* (Jaeger et al., 2023).

Property φ_1: Direction consistency (approach phase). When the route planner issues a turn command $c \in \{\text{LEFT}, \text{RIGHT}\}$, the VLA's predicted waypoints must be consistent with the intended direction. Let θ_{pred} denote the bearing angle of the last predicted waypoint $\boldsymbol{w}_N$ relative to the ego frame. Since predicted waypoints always lie ahead of the ego vehicle ($w_{N,y} > 0$), arctan suffices without the full atan2 range:

$$\theta_{\text{pred}} = \arctan\left(\frac{w_{N,x}}{w_{N,y} + \epsilon}\right). \tag{5}$$

The direction consistency specification, expressed in Signal Temporal Logic (Desai et al., 2017) where $\square$ denotes "always," requires:

$$\varphi_1 \triangleq \square\Big(\neg \texttt{in\_junction} \wedge c = \text{LEFT} \implies \theta_{\text{pred}} < -\theta_{\text{thr}}\Big), \tag{6}$$

and symmetrically for $c = \text{RIGHT}$. The threshold $\theta_{\text{thr}} = 20°$ separates straight-ahead predictions from turning predictions and was selected empirically based on typical CARLA intersection geometries.

φ_1 is evaluated only during the *approach phase*, before the vehicle enters the junction. Once inside the junction, the VLA's predicted waypoints naturally flatten in the rotated ego frame because the model correctly predicts 'go forward' relative to its current heading during mid-turn execution. Without junction-aware gating, this flattening triggers false-positive direction conflicts: the monitor infers STRAIGHT from flattened waypoints while the route command remains LEFT/RIGHT, causing unnecessary takeovers. Junction boundaries are queried from the CARLA HD map.

Property φ_2: Stuck detection. When throttle is applied, the vehicle must eventually move. Here $\diamondsuit_{[0,T]}$ denotes "eventually within T steps":

$$\varphi_2 \triangleq \square\Big(\texttt{throttle} > \tau_{\text{thr}} \implies \diamondsuit_{[0,\, T_{\text{stuck}}]}\, v > v_{\min}\Big), \tag{7}$$

where $\tau_{\text{thr}} = 0.2$, $v_{\min} = 0.1\,\text{m/s}$, and $T_{\text{stuck}} = 30$ frames ($\approx 1.5\,\text{s}$). A typical stuck situation occurs when the VLA predicts forward motion into a stopped vehicle: the PID controller applies throttle, but the vehicle cannot move.

4.3.3 Fallback Policy and Recovery

Upon φ_1 violation, the DM activates the BC with conservative parameters selected to prioritize safety during intervention: auto lane-change disabled, 5 m following distance, 40% speed reduction. The BC's waypoints $\boldsymbol{w}_{\text{BC}}$ replace the VLA output for $T_{\min}$ steps, after which control returns to the AC. During takeover, steering limits are relaxed to 1.2× the normal maximum to enable trajectory correction, while throttle is capped at 0.6× the normal limit to reduce speed during the intervention.

This architecture provides a semantic-level safety guarantee. Conditioned on correct envelope classification by the DM, the system does not execute a VLA action that violates φ_1 or φ_2. This conditional guarantee is absent from all surveyed E2E driving methods (§2.2). The total computational overhead is negligible, under 0.1 ms per step for map query and angle computation.

4.4 Training-Time Safety Alignment (Experimental Condition)

We employ PL-DPO-NLL as an *experimental condition* that provides a second weight configuration for ablation, not as a standalone contribution. It combines Plackett-Luce multi-preference ranking (Plackett, 1975) with NLL regularization to address probability collapse during preference optimization (Rafailov et al., 2023; Pang et al., 2024).

4.4.1 Preference Data

We collect 51,124 Plackett-Luce preference samples from CARLA Town01 across 67 route configurations. Each sample contains one expert (chosen) action and 2–3 rejected actions ranked by risk severity. Scene-type distribution: turns (40.2%), normal driving (27.8%), braking scenarios (14.7%), speed adjustment (6.0%), junctions (5.6%), pedestrian interactions (3.1%), and red-light scenarios (2.7%).

4.4.2 Objective Function

The PL-DPO loss generalizes binary DPO to full rankings over M candidates. Let $x = (\boldsymbol{v}_t, \boldsymbol{l})$ denote the multimodal context (visual features and text input):

$$\mathcal{L}_{\text{PL-DPO}} = -\mathbb{E}_{(x,\, y^{(1:M)})} \left[\sum_{i=1}^{M} \log \frac{\exp(\beta \cdot r_i)}{\sum_{j=i}^{M} \exp(\beta \cdot r_j)} \right], \tag{8}$$

where $r_i = \log \frac{\pi_\theta(y^{(i)}|x)}{\pi_{\text{ref}}(y^{(i)}|x)}$ is the implicit reward, $y^{(1)}$ is the chosen action, and $y^{(2)}, \ldots, y^{(M)}$ are rejected actions ranked by increasing risk. The temperature β is scene-adaptive, with higher values for safety-critical scenes to sharpen the preference distribution: $\beta = 0.35$ for turns, pedestrians, and red lights; $\beta = 0.25$ for braking; $\beta = 0.18$–0.20 for junctions and speed adjustment; and $\beta = 0.12$ for normal driving. These values were selected via grid search on a held-out validation set from Town01.

The full PL-DPO-NLL objective adds explicit likelihood preservation:

$$\mathcal{L}_{\text{PL-DPO-NLL}} = \mathcal{L}_{\text{PL-DPO}} + \lambda \cdot \mathcal{L}_{\text{NLL}}\,, \tag{9}$$

where $\mathcal{L}_{\text{NLL}} = -\log \pi_\theta(y^{(1)} \mid x)$ prevents the absolute probability of correct actions from decreasing during preference optimization. We set $\lambda = 0.1$ based on ablation (higher values cause NLL to dominate the preference signal).

4.4.3 Training Configuration

We apply LoRA adapters ($r = 32$, $\alpha = 32$) to all attention and MLP projections (q, k, v, o, gate, down, up) of LLaMA-7B. Training uses AdamW-8bit with learning rate 10^{-5}, batch size 4 per device with 8 gradient accumulation steps (effective batch 32), 3 epochs, warmup ratio 0.03, and BF16 mixed precision with gradient checkpointing. Training was conducted on 3× NVIDIA RTX 6000 Ada GPUs.

5 Experiments

5.1 Experimental Setup

Our experiments aim to answer three questions. First, does CSN improve driving performance, and is the improvement robust across different weight configurations? Second, how much of CSN's gain comes from causal structure versus additional information? Third, does semantic safety supervision outperform reactive approaches?

We evaluate on CARLA 0.9.10 (Dosovitskiy et al., 2017) in closed-loop mode using LMDrive with a LLaMA-7B (Touvron et al., 2023) backbone and ResNet-50 (He et al., 2016) vision encoder, trained with PL-DPO-NLL LoRA on a single NVIDIA RTX 3090 Ti. The benchmark spans 16 routes across 8 towns drawn from the official Leaderboard route set, including 4 night-time routes, 5 rain routes, 3 dense fog routes, and 4 clear daytime routes. This diversity tests generalization across urban layouts, traffic densities, road topologies, and weather conditions absent from single-town benchmarks. Each configuration is evaluated over N=5 independent repetitions with distinct random seeds. We report the mean and 95% bootstrap CI; non-overlapping CIs between two configurations suggest a meaningful difference.

We evaluate ten configurations organized hierarchically (Table 3). On the original LMDrive without LoRA, we test (1) original only, (2) +CSN, (3) +Flat Text, and (4) +Semantic Safety. On the PL-DPO-NLL variant with LoRA, we test (5) baseline, (6) +TTC Safety, (7) +Semantic Safety, (8) +CSN, (9) +CSN+Safety, and (10) +Flat Text. Flat Text provides the same factual content as CSN but without causal connectives, enabling the decomposition in Eq. (3) on both weight configurations.

We follow the CARLA Leaderboard metrics. Driving Score (DS) measures route completion weighted by infraction penalty and serves as the primary metric. Route Completion (RC) measures the percentage of route distance completed. Infraction Score (IS) is a cumulative penalty multiplier where 1.0 means no infractions. Fig. 4 shows representative evaluation environments.

Figure 4: Evaluation environments. (a) Town01, clear day. (b) Town03, heavy rain. (c) Town05, night. (d) Town07, dense fog.

Table 3: Multi-town ablation (16 routes, 8 towns, N=5). Values are mean ± 95% bootstrap CI. Best **bold**, second underlined; ties share marking.

Configuration	**DS** (↑)	**RC** (↑)	**IS** (↑)	Δ**DS**$_{\text{orig}}$	Δ**DS**$_{\text{base}}$
LMDrive (original)	32.54±3.00	48.3±2.6%	0.729±0.034	—	—
+ CSN	**42.67±2.74**	**56.5±1.7%**	0.787±0.028	+31.1%	—
+ Flat Text	38.71±1.44	48.7±1.9%	**0.828±0.025**	+18.9%	—
+ Semantic Safety	34.10±2.25	45.5±2.1%	0.785±0.038	+4.8%	—
+ PL-DPO-NLL	32.49±3.34	44.9±2.7%	0.754±0.021	−0.1%	—
+ TTC Safety	22.02±4.07	33.7±3.4%	0.658±0.037	−32.3%	−32.2%
+ Semantic Safety	33.17±1.42	44.5±2.5%	0.787±0.022	+1.9%	+2.1%
+ CSN	40.45±3.79	51.9±4.3%	0.789±0.026	+24.3%	+24.5%
+ CSN+Safety	35.74±1.37	48.6±2.1%	0.754±0.020	+9.8%	+10.0%
+ Flat Text	39.38±2.66	49.0±1.3%	0.823±0.047	+21.0%	+21.2%

5.2 Main Results

Table 3 presents the main results across ten configurations and four dimensions: preference training generalization, CSN robustness, safety monitor comparison, and component interaction.

5.2.1 Preference Training Generalization

With N=5, PL-DPO-NLL and the original LMDrive have nearly identical mean DS (32.49 vs. 32.54) with heavily overlapping CIs (Table 3), so preference training neither helps nor hurts on aggregate across towns. Since PL-DPO-NLL is trained on 51,124 preference samples collected exclusively from CARLA Town01, the lack of improvement on unseen towns is consistent with the distribution shift documented in DPO literature (Lin et al., 2024), where preference-optimized models overfit to training-distribution patterns at the expense of out-of-distribution generalization. Any gains on Town01 are offset by losses on the remaining seven towns.

We include PL-DPO-NLL not as a standalone contribution but as an experimental condition that provides a second weight configuration for ablation. Because CSN operates on the input side without modifying model weights, it does not introduce distribution shift.

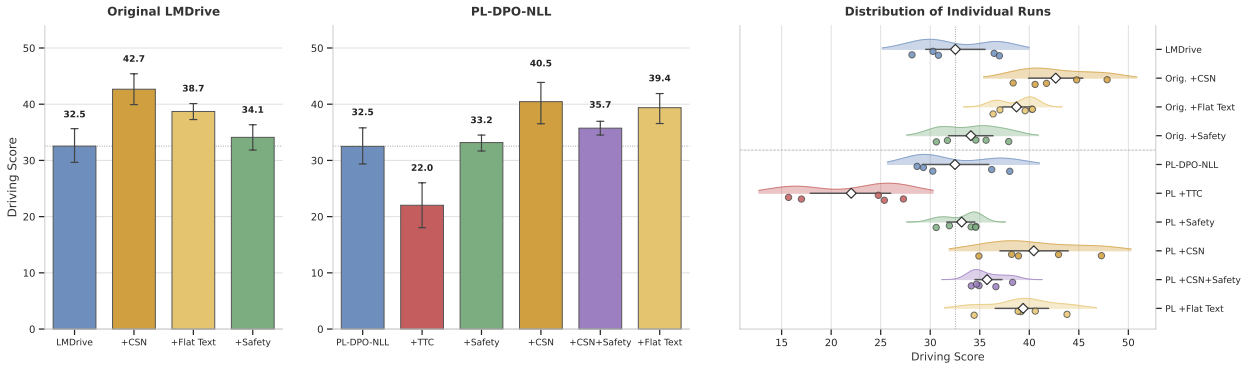

Figure 5: Driving Score comparison across all ten configurations (N=5). Left and center: mean DS with 95% bootstrap CI. Right: raincloud plot showing KDE density (shaded), individual runs (dots), mean (diamond), and 95% CI (whisker). Dotted line: original LMDrive baseline.

5.2.2 CSN Robustness Across Configurations

CSN improves DS by +31.1% on the original LMDrive and +24.5% on PL-DPO-NLL (Table 3). The overlapping 95% CIs between the two CSN-enhanced configurations (42.67±2.74 vs. 40.45±3.79) suggest that CSN provides comparable benefits regardless of LoRA adaptation. IS also improves on both variants, indicating fewer safety violations in novel environments. We note that Flat Text achieves higher IS than CSN despite lower DS (0.828 vs. 0.787 on original LMDrive); this is consistent with Flat Text's lower RC (48.7% vs. 56.5%), as completing fewer route segments naturally reduces infraction opportunities.

5.2.3 Safety Monitor Comparison

The reactive TTC monitor, which triggers emergency braking when Time-To-Collision falls below 2.0 s, achieves the *lowest* DS and IS among all configurations (Table 3). The failure mode is systematic, as frequent false-positive emergency braking causes the vehicle to repeatedly stop and restart, leading to "Agent got blocked" timeout failures. Its wide CIs reflect high variance across repetitions, with performance degrading further as CARLA's stochastic traffic amplifies the over-braking pathology.

In contrast, our semantic supervisor improves IS on both weight configurations: from 0.729 to 0.785 on original LMDrive and from 0.754 to 0.787 on PL-DPO-NLL (Table 3). Its narrow CIs demonstrate that intent-aware monitoring produces *consistently* better outcomes across repetitions. The semantic monitor detects *direction conflicts* between VLA waypoints and the navigation plan rather than reacting to proximity, avoiding the over-braking pathology entirely. For VLA systems, physics-only safety monitors that ignore driving intent degrade rather than improve performance.

5.2.4 Component Interaction Analysis

CSN and semantic safety represent two potentially conflicting safety paradigms. CSN provides *proactive* safety by improving scene understanding so the VLA produces inherently safer decisions, while the semantic supervisor provides *reactive* safety by monitoring outputs and overriding unsafe ones. Their interaction is not additive.

When applied to the unenhanced PL-DPO-NLL baseline, the semantic supervisor has a small positive effect on DS (+0.7, Table 3). However, when layered on top of CSN, the same supervisor *degrades* DS by −4.7 points in the main evaluation (40.45 vs. 35.74, Table 3), a finding confirmed in a separate N=3 replication (CSN alone 43.8, CSN+Safety 36.0, Δ=−7.8). The tight CI of the CSN+Safety configuration (±1.37) confirms that the degradation is systematic rather than stochastic.

To diagnose the mechanism, we logged per-frame intervention frequencies across all safety-enabled runs (N=3 per config, 16 routes each). The direction-conflict detector (φ_1) triggered zero interventions across all 96 route-checks in both baseline+Safety and CSN+Safety. Stuck detection (φ_2) likewise never triggered. The degradation therefore does not arise from explicit safety interventions.

Instead, the cause is passive control clamping. The safety supervisor applies per-frame steering and throttle limits (steer ≤ 0.8, throttle ≤ 0.9) on every timestep regardless of whether an intervention is triggered. Without CSN, the VLA produces conservative waypoints with small steering angles, so the clamp rarely binds. With CSN, the VLA receives structured context about upcoming hazards and produces larger anticipatory steering adjustments—early lane changes, preemptive deceleration curves—that exceed the 0.8 steer limit. The clamp truncates these evasive maneuvers, converting them into incomplete corrections that produce worse trajectories than no correction at all. This asymmetric clamping effect explains why safety supervision helps the baseline (clamp does not bind) but hurts CSN (clamp truncates beneficial evasive actions). Relaxing or removing the passive clamp when CSN is active would likely resolve this conflict.

5.3 Decomposition Results

The +Flat Text ablation provides the same factual content as CSN but without causal connectives (BUT, YIELD BEFORE, BECAUSE), enabling an empirical decomposition of the total performance gain into information quantity ($\text{Utility}_{\text{info}}$) and structural organization ($\text{Utility}_{\text{struct}}$) per Eq. (3). Because we run this ablation on both weight configurations, we can test whether the decomposition ratio generalizes.

Original LMDrive (N=5):

$$\begin{aligned}\Delta\text{DS}_{\text{total}} &= 42.67 - 32.54 = +10.13\\ \text{Utility}_{\text{info}} &= 38.71 - 32.54 = +6.17 \;\; (60.9\%)\\ \text{Utility}_{\text{struct}} &= 42.67 - 38.71 = +3.96 \;\; (39.1\%)\,.\end{aligned}$$

PL-DPO-NLL variant (N=5):

$$\begin{aligned}\Delta\text{DS}_{\text{total}} &= 40.45 - 32.49 = +7.96\\ \text{Utility}_{\text{info}} &= 39.38 - 32.49 = +6.88 \;\; (86.5\%)\\ \text{Utility}_{\text{struct}} &= 40.45 - 39.38 = +1.08 \;\; (13.5\%)\,.\end{aligned}$$

On original LMDrive, causal structure accounts for 39.1% of the total DS improvement as shown in Fig. 6, showing that CSN's benefit is not reducible to providing more information. On PL-DPO-NLL, this drops to 13.5%. Information content contributes comparably in both cases (+6.17 vs. +6.88), but causal structure contributes much less on the preference-aligned variant (+3.96 vs. +1.08). A plausible explanation is that preference learning on 51k ranked driving samples partially internalizes causal reasoning about intent-constraint relationships, reducing the marginal benefit of explicit causal connectives in the text. The original LMDrive, having never seen preference-ranked actions, benefits more from the explicit causal scaffolding that CSN provides.

On both configurations, +Flat Text outperforms the respective baseline with non-overlapping CIs, showing that scene information alone is valuable regardless of weight configuration.

5.4 Discussion

5.4.1 Why Does Causal Structure Help?

Why does reorganizing the same facts into structured sentences help? Consider a speed reduction scenario. Template text presents 'Reduce speed' and 'Wet road' as unrelated fragments, leaving the LLM to infer whether the wet road is relevant. CSN writes 'Reduce speed BECAUSE wet road reduces braking effectiveness at current 45 km/h,' and the connective 'BECAUSE' acts as a direct attention cue. LLaMA has seen millions of such constructions during pre-training on natural text and already knows how to process them. The structured text offloads part of the reasoning from the model's weights to the input.

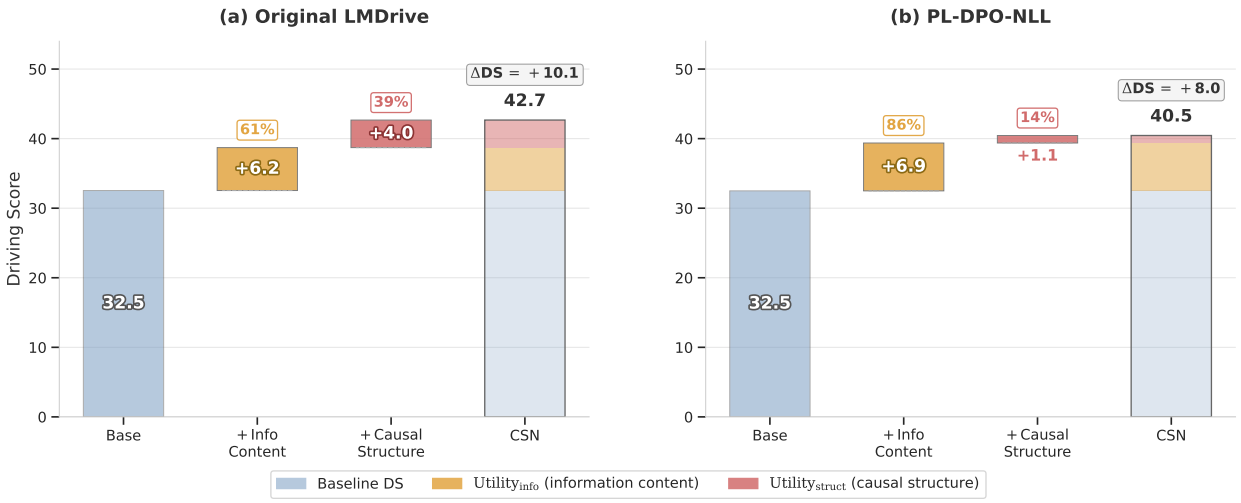

Figure 6: Decomposition of CSN's DS improvement into information content and causal structure contributions on both weight configurations.

Table 4: Perception noise ablation on Original LMDrive + CSN (N=5). We inject Gaussian noise on distance, multiplicative noise on speed, and random actor miss rates. All noise-level CIs overlap with the clean baseline, indicating no statistically significant degradation.

Noise Level	**Dist. σ**	**Speed**	**Miss Rate**	**DS**	**95% CI**
Clean (privileged)	0 m	0%	0%	42.7 ± 3.3	[39.9, 45.4]
Mild	$\pm$1 m	$\pm$10%	0%	45.2 ± 2.9	[42.6, 47.8]
Moderate	$\pm$2 m	$\pm$20%	0%	45.4 ± 2.7	[43.1, 47.6]
Severe	$\pm$5 m	$\pm$20%	10%	46.0 ± 4.1	[42.5, 49.6]
Extreme	$\pm$5 m	$\pm$30%	20%	45.1 ± 2.7	[42.7, 47.4]

5.4.2 Privileged Information and Deployment Considerations

As noted in §4.2, CSN currently uses privileged simulation data. To test whether the improvement survives under realistic perception errors, we inject calibrated noise into CSN's inputs: Gaussian noise on distance measurements ($\sigma \in \{1, 2, 5\}$ m), multiplicative noise on speed readings ($\pm\{10, 20, 30\}$%), and random actor miss rates (0–20%). Table 4 shows the results.

The clean baseline in Table 4 uses the same Original+CSN runs from Table 3; the wider CI ($\pm$3.3 vs. $\pm$2.74) reflects the standard deviation rather than bootstrap CI to enable direct comparison with the noise conditions at N=5. All four noise conditions produce CIs that overlap fully with the clean baseline, confirming that CSN's benefit does not depend on privileged information precision. Even under the extreme condition ($\pm$5 m distance error, $\pm$30% speed noise, 20% actor miss rate), DS remains at 45.1 versus 42.7 for clean inputs. The noisy conditions yield slightly higher mean DS than clean, but all CIs overlap, so this difference is attributable to sampling variance at N=5 rather than a systematic effect. These results are consistent with the hypothesis that CSN's value lies in how information is organized through causal connectives, not in the accuracy of the underlying measurements. The linking algorithm is agnostic to the data source, and these results suggest that replacing privileged data with a vision-based perception pipeline would preserve CSN's effectiveness.

5.4.3 Choice of Evaluation Platform

Our experiments use a single VLA architecture (LMDrive with LLaMA-7B). LMDrive is currently the only open-source VLA that satisfies three requirements simultaneously: (1) it accepts free-form text input that CSN can modify, (2) it supports closed-loop CARLA evaluation with the standard Leaderboard protocol, and (3) its training pipeline is publicly available, enabling the PL-DPO-NLL ablation condition. Other VLA models such as DriveVLM (Tian et al., 2024) and Bench2Drive (Jia et al., 2024) either lack open-source training code, do not accept free-form text, or use proprietary evaluation setups that prevent controlled comparison. CSN's mechanism—restructuring text with causal connectives that LLMs already understand from pretraining—is architecture-agnostic in principle, but validating this claim requires additional open-source VLA platforms with closed-loop evaluation capabilities.

5.4.4 Robustness Across Weight Configurations

As established in §5.2, the overlapping CIs between the original and preference-aligned CSN configurations suggest that CSN's overall benefit does not depend on the specific weight configuration. However, the empirical decomposition reveals an interesting asymmetry: causal structure accounts for 39.1% of CSN's gain on original LMDrive but only 13.5% on PL-DPO-NLL. We note that on PL-DPO-NLL, the CSN vs. Flat Text CIs overlap substantially (40.45 ± 3.79 vs. 39.38 ± 2.66), so the 13.5% figure is not statistically distinguishable from zero. On original LMDrive the overlap is marginal (42.67 ± 2.74 vs. 38.71 ± 1.44), supporting a meaningful structural contribution in that setting. This suggests that preference learning internalizes some of the causal reasoning that explicit text structure otherwise provides. CSN should be compatible with other LMDrive-family checkpoints without further weight updates, though the balance between information and structural contributions may vary. We have not tested whether this transfers to other architectures such as DriveVLM.

5.4.5 Benchmark Difficulty and Weather Effects

The benchmark includes 4 night routes, 5 rain routes, and 3 dense fog routes alongside clear daytime conditions. Baseline DS values (32.54 for original LMDrive, 32.49 for PL-DPO-NLL) are well below the 50–60 DS typical of single-town clear-weather evaluations. CSN's +31.1% improvement holds under these harder conditions.

6 Conclusion

We introduced Causal Scene Narration (CSN), a framework that restructures VLA text inputs around intent-constraint causal alignment at zero GPU cost. Through a multi-town CARLA evaluation (16 routes across 8 towns, N=5 independent repetitions with 95% bootstrap confidence intervals), we establish four findings.

First, CSN substantially improves DS on both the original LMDrive (+31.1%) and the preference-aligned variant (+24.5%), with overlapping CIs consistent with benefits robust across the two weight configurations tested. Second, a controlled ablation on both configurations shows that causal structure accounts for 39.1% of CSN's gain on original LMDrive but only 13.5% on PL-DPO-NLL, suggesting that preference learning partially internalizes causal reasoning and reduces the marginal benefit of explicit text structure. Third, semantic safety supervision improves IS on both weight configurations, while reactive TTC monitoring degrades both DS and IS; VLA safety monitors that rely on physical proximity alone hurt performance, and intent-aware monitoring is needed.

Fourth, a perception noise ablation shows that CSN's benefit is robust to distance errors up to ± 5 m, speed noise up to $\pm 30\%$, and 20% actor miss rates, indicating that the improvement derives from text structure rather than information precision.

An additional observation is that combining CSN with the safety supervisor hurts rather than helps. Intervention logging reveals that the degradation arises not from explicit safety interventions (which never trigger) but from passive control clamping that truncates CSN-guided evasive steering. Relaxing the clamp when CSN is active would likely resolve this conflict.

Limitations. (1) Our evaluation is simulation-based; while noise injection experiments (§5.4.2) show CSN is robust to perception errors, real-world deployment requires integration with an actual perception pipeline. (2) Experiments use a single model architecture and scale (LMDrive with LLaMA-7B). (3) The safety supervisor uses fixed direction-conflict thresholds that do not adapt to CSN-enhanced context quality.

Future Work. (1) Integrating CSN with a vision-based perception pipeline for real-world validation. (2) Extending CSN to other VLA architectures and model scales. (3) Developing adaptive safety thresholds that modulate intervention sensitivity based on CSN context quality. (4) Extending the empirical decomposition to token-level VLA architectures, where action distributions are directly measurable, enabling a strict information-theoretic validation of $\text{Utility}_{\text{struct}}$.

Ethics Statement. CSN aims to improve VLA driving safety through better text conditioning and runtime monitoring. Our safety supervisor provides an additional layer of protection but is not a substitute for comprehensive safety validation. The system is evaluated exclusively in simulation; real-world deployment would require extensive additional testing. All experiments use the open CARLA simulator with no human subjects involved.

Reproducibility Statement. All experiments use the publicly available CARLA 0.9.10 simulator and the open-source LMDrive codebase. Evaluation follows the standard CARLA Leaderboard protocol with 16 routes across 8 towns. Each configuration is run N=5 times with fixed random seeds; we report bootstrap 95% CIs throughout. The CSN text generation pipeline, safety supervisor implementation, evaluation scripts, and all trained LoRA weights will be released upon acceptance.

References

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. *Proc. Conf. AAAI Artif. Intell.*, 32(1), 2018. doi: 10.1609/aaai.v32i1.11797. URL http://dx.doi.org/10.1609/aaai.v32i1.11797.

Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *2019 18th European Control Conference (ECC)*, pp. 3420–3431, 2019. doi: 10.23919/ecc.2019.8796030. URL http://dx.doi.org/10.23919/ECC.2019.8796030.

Li Chen, Xiaosong Jia, Hongyang Li, Yu Qiao, Penghao Wu, and Junchi Yan. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. In *Advances in Neural Information Processing Systems 35*, pp. 6119–6132, 2022. URL https://arxiv.org/pdf/2206.08129.

Ankush Desai, Tommaso Dreossi, and Sanjit A Seshia. Combining model checking and runtime verification for safe robotics. In *Runtime Verification*, pp. 172–189. Springer International Publishing, 2017. doi: 10.1007/978-3-319-67531-2_11. URL http://dx.doi.org/10.1007/978-3-319-67531-2_11.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An Open Urban Driving Simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78, pp. 1–16. PMLR, 2017. URL https://proceedings.mlr.press/v78/dosovitskiy17a.html.

Tian Gao, Celine Tan, Catherine Glossop, Timothy Gao, Jiankai Sun, Kyle Stachowicz, Shirley Wu, Oier Mees, Dorsa Sadigh, Sergey Levine, and Chelsea Finn. SteerVLA: Steering vision-language-action models in long-tail driving scenarios. *arXiv preprint arXiv:2602.08440*, 2026. URL http://arxiv.org/abs/2602.08440.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. URL http://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, and Others. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17853–17862, 2023. URL http://openaccess.thecvf.com/content/CVPR2023/html/Hu_Planning-Oriented_Autonomous_Driving_CVPR_2023_paper.html.

Bernhard Jaeger, Kashyap Chitta, and Andreas Geiger. Hidden Biases of End-to-End Driving Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8240–8249, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Jaeger_Hidden_Biases_of_End-to-End_Driving_Models_ICCV_2023_paper.html.

Xiaosong Jia, Qifeng Li, Junchi Yan, Zhenjie Yang, and Zhiyuan Zhang. Bench2Drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. In *Advances in Neural Information Processing Systems 37*, volume 37, pp. 819–844. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024. doi: 10.52202/079017-0025. URL http://dx.doi.org/10.52202/079017-0025.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. VAD: Vectorized Scene Representation for Efficient Autonomous Driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8350, 2023. URL http://openaccess.thecvf.com/content/ICCV2023/html/Jiang_VAD_Vectorized_Scene_Representation_for_Efficient_Autonomous_Driving_ICCV_2023_paper.html.

Yun Li, Ehsan Javanmardi, Simon Thompson, Kai Katsumata, Alex Orsholits, and Manabu Tsukada. Multi-PrefDrive: Optimizing Large Language Models for Autonomous Driving Through Multi-Preference Tuning. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4347–4354, October 2025. URL https://ieeexplore.ieee.org/abstract/document/11247608/.

Yong Lin, Skyler Seto, Maartje Ter Hoeve, Katherine Metcalf, Barry-John Theobald, Xuan Wang, Yizhe Zhang, Chen Huang, and Tong Zhang. On the limited generalization capability of the implicit reward model induced by direct preference optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 16015–16026, Stroudsburg, PA, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.940. URL http://dx.doi.org/10.18653/v1/2024.findings-emnlp.940.

Jiageng Mao, Yuxi Qian, Junjie Ye, Hang Zhao, and Yue Wang. GPT-Driver: Learning to Drive with GPT. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023. URL http://arxiv.org/abs/2310.01415.

Richard Yuanzhe Pang, Weizhe Yuan, He He, Kyunghyun Cho, Sainbayar Sukhbaatar, and Jason E Weston. Iterative Reasoning Preference Optimization. In *Advances in Neural Information Processing Systems 37*, pp. 116617–116637, 2024. URL https://arxiv.org/abs/2404.19733.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, 2009. ISBN 9780521895606. doi: 10.1017/cbo9780511803161.

Dung T Phan, Radu Grosu, Nils Jansen, Nicola Paoletti, Scott A Smolka, and Scott D Stoller. Neural Simplex Architecture. In *Lecture Notes in Computer Science*, pp. 97–114. Springer International Publishing, 2020. doi: 10.1007/978-3-030-55754-6_6. URL http://dx.doi.org/10.1007/978-3-030-55754-6_6.

R L Plackett. The analysis of permutations. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 24(2):193–202, 1975. URL https://academic.oup.com/jrsssc/article-abstract/24/2/193/6953554.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.18290.

Noam Razin, Sadhika Malladi, Adithya Bhaskar, Danqi Chen, Sanjeev Arora, and Boris Hanin. Unintentional unalignment: Likelihood displacement in direct Preference Optimization. *arXiv preprint arXiv:2410.08847*, 2024. URL http://arxiv.org/abs/2410.08847.

Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. SimLingo: Vision-Only Closed-Loop Autonomous Driving with Language-Action Alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11993–12003, 2025. URL http://openaccess.thecvf.com/content/CVPR2025/html/Renz_SimLingo_Vision-Only_Closed-Loop_Autonomous_Driving_with_Language-Action_Alignment_CVPR_2025_paper.html.

Fabian Schmidt, Noushiq Mohammed Kayilan Abdul Nazar, Markus Enzweiler, and Abhinav Valada. Enhancing LLM-based autonomous driving with modular traffic light and sign recognition. *arXiv preprint arXiv:2511.14391*, 2025. URL http://arxiv.org/abs/2511.14391.

Fabian Schmidt, Markus Enzweiler, and Abhinav Valada. GraphPilot: Grounded scene graph conditioning for language-based autonomous driving. *arXiv preprint arXiv:2511.11266*, 2026. URL http://arxiv.org/abs/2511.11266.

Lui Sha. Using simplicity to control complexity. *IEEE Softw.*, 18(4):20–28, 2001. doi: 10.1109/ms.2001.936213. URL http://dx.doi.org/10.1109/ms.2001.936213.

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. On a formal model of safe and scalable self-driving cars. *arXiv preprint arXiv:1708.06374*, 2017. URL http://arxiv.org/abs/1708.06374.

Hao Shao, Yuxuan Hu, Letian Wang, Steven L Waslander, Yu Liu, and Hongsheng Li. LMDrive: Closed-Loop End-to-End Driving with Large Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL http://arxiv.org/abs/2312.07488.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. DriveLM: Driving with Graph Visual Question Answering. In *European Conference on Computer Vision 2024*, September 2024. URL https://openreview.net/forum?id=AxDZMqrRYS.

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. In *2024 Conference on Robot Learning*, 2024. URL https://arxiv.org/abs/2402.12289.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. URL http://arxiv.org/abs/2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Adv. Neural Inf. Process. Syst.*, 30, 2017. URL https://proceedings.neurips.cc/paper/7181-attention-is-all.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35*, 2022. URL https://arxiv.org/abs/2201.11903.