Selective Language Communication for Cooperative Perception

Anonymous ACL submission

Abstract

Cooperative perception enables connected vehicles to share sensory information beyond line-of-sight. The prevailing paradigm—intermediate feature fusion—assumes *more shared information is always better*. We challenge this and reveal a striking double-edged sword: blindly fusing all features improves invisible-object detection by $+7.9$pp while degrading visible-object detection by -7.9pp, yielding zero net gain. To address this, we propose language-mediated selective communication, where vehicles describe *only* what the receiver cannot see. Our framework comprises: (1) a learned router predicting communication value from deployment features (96.4% accuracy), (2) template-based NL messages with coordinate-level precision, and (3) prompt-level NL integration via fine-tuning. A critical finding motivates (3): zero-shot NL grounding fails catastrophically (22.0% per-sample, below the no-communication baseline), revealing a fundamental misalignment between pre-trained language and BEV spatial perception. On V2V4Real, our best model achieves 31.0%/41.7% per-sample/object accuracy—surpassing blind fusion by $+16.8$pp per-object while communicating in only 17.8% of cases.

1 Introduction

Cooperative perception enables connected vehicles to share sensory information beyond their individual line-of-sight. The prevailing paradigm—intermediate feature fusion, where deep representations from each vehicle are transmitted and aggregated at the receiver—has driven substantial progress in multi-agent 3D detection (Xu et al., 2022b,a; Chiu et al., 2025). Underlying this approach is an implicit assumption: *more shared information is always better*. Yet this assumption has rarely been questioned.

More communication does not always mean better perception. Our analysis on V2V4Real (Xu et al., 2023) reveals a striking double-edged sword: blindly fusing all shared features improves invisible-object detection by $+7.9$pp while *simultaneously* degrading visible-object detection by -7.9pp—a perfect zero-sum trade-off (Figure 1). The aggregate accuracy remains unchanged at 24.9%, creating the false impression that cooperation provides no benefit whatsoever. This finding exposes a fundamental flaw in the "more-is-better" paradigm and motivates **selective communication**, where a receiver selectively incorporates only information about objects it cannot perceive on its own.

Existing feature-level selective methods (Hu et al., 2022; Liu et al., 2020) operate on spatial confidence maps to decide which feature locations to transmit. While effective for bandwidth reduction, they suffer from three limitations: (1) feature representations are tied to specific encoder architectures, limiting cross-platform deployment; (2) latent feature vectors are opaque to human inspection, impeding safety audit; and (3) they lack a *semantic interface* for explaining *what* was communicated and *why*—a critical requirement for safety-certified autonomous driving.

Our key insight is that natural language is uniquely suited as a selective communication primitive. Unlike continuous feature vectors, NL is discrete and symbolic: a structured message such as "[CAV alert: vehicle at [-20.5, -0.1]]" is simultaneously a *percept*, a *pragmatic act*, and an *auditable record*. NL's discrete nature makes it inherently a high-confidence selection signal: either a message is sent or it is not.

Realizing NL-mediated selective communication requires answering three questions: **(1) When to communicate?** The ego must predict whether a cooperating vehicle has valuable information. **(2) What to communicate?** The cooperator must describe only invisible objects with coordinate-level precision. **(3) How to integrate?** NL mes-

sages must be fused with visual features—we find zero-shot NL integration fails catastrophically.

We propose NL-SELCOMM, a three-component framework:

- **M1: Deployment-Feature Router.** A lightweight GradientBoosting classifier (10 deployment features, 96.4% accuracy) predicts whether the cooperator holds complementary information.
- **M2: Selective NL Message Formulation.** Template-based coordinate-to-language conversion produces compact NL messages ($\sim$100 bytes) describing only invisible objects.
- **M3: Prompt-Level NL Integration.** NL messages are prepended to the LLM input and the model is fine-tuned to read spatial coordinates from text, bridging the gap between pre-trained linguistic knowledge and BEV spatial perception.

Our contributions are:

1. **First systematic quantification** of the cooperative double-edged sword, revealing blind fusion trades visible for invisible accuracy at symmetric margins (-7.9/$+7.9$ pp).
2. **Discovery that zero-shot NL grounding fails** in BEV perception (22.0% per-sample, below NoComm baseline), revealing a fundamental misalignment between pre-trained language and spatial perception.
3. **A deployment-feature router** achieving 96.4% accuracy from 10 readily available features.
4. **First NL-mediated selective communication framework** for cooperative driving, achieving 31.0%/41.7% per-sample/object accuracy while communicating in only 17.8% of cases—surpassing blind fusion by $+16.8$pp per-object.

2 Related Work

2.1 V2X Cooperative Perception

The dominant paradigm in cooperative perception is intermediate feature fusion, where deep features from each vehicle's backbone are transmitted and aggregated at the receiver (Xu et al., 2022b,a; Chiu et al., 2025). A critical assumption across these

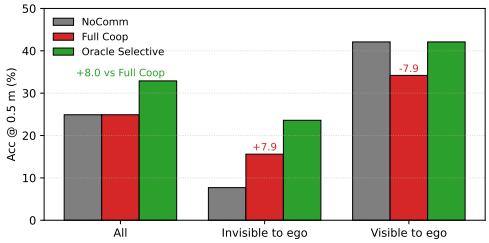

Figure 1: The cooperative double-edged sword on V2V4Real. Full Coop blind fusion trades visible object accuracy (-7.9pp) for invisible object accuracy ($+7.9$pp) with zero net gain. Oracle selective communication achieves $+8.0$pp overall by retaining visible object predictions from NoComm and using Full Coop only for invisible objects.

works is that more shared information is always beneficial. Our work challenges this by quantifying the cooperative double-edged sword—showing that blind fusion simultaneously helps invisible objects and hurts visible ones by exactly symmetrical margins.

2.2 Selective Communication

Feature-level selective methods (Hu et al., 2022; Liu et al., 2020) use spatial confidence maps to decide which feature locations to transmit. While effective for bandwidth reduction, they suffer from three limitations: (1) feature representations are tied to specific encoders, limiting cross-platform deployment; (2) latent vectors are opaque to human inspection; and (3) they lack a semantic interface for explaining what was communicated and why. Our NL-based approach differs in four dimensions: **granularity** (object-level vs. spatial-location), **bandwidth** ($\sim$100 bytes vs. kilobytes), **cross-platform compatibility** (architecture-independent), and **interpretability** (human-readable).

2.3 Language-Guided Perception

NL has been used for visual QA and reasoning in driving (Dewangan et al., 2023; Sima et al., 2024) and as a communication medium for trajectory planning (Liu et al., 2025; Gao et al., 2025; Bhatt et al., 2025). Our work is the first to use NL for selective communication in cooperative 3D grounding with precise spatial coordinates. Prior work shows VLMs struggle with coordinate-level spatial reasoning (Mouselinos et al., 2024; Rizvi et al., 2024) and LLMs are unreliable spatial simulators without fine-tuning (Wang et al., 2024). Our

zero-shot failure finding corroborates these results and reveals a stronger failure mode: performance *degrades* below the no-communication baseline.

2.4 Cross-Modal Fusion

Common approaches include text-prompt concatenation (Liu et al., 2023), cross-attention (Alayrac et al., 2022), and gated fusion (Li et al., 2023). Our M3 explores prompt-level integration and cross-attention fusion; for structured coordinate-level NL, prompt-level with fine-tuning is sufficient (Section 5.3).

3 The Cooperative Double-Edged Sword

3.1 Experimental Setup

We conduct our analysis on V2V4Real (Xu et al., 2023), a real-world vehicle-to-vehicle cooperative driving dataset. The task is **BEV-coordinate grounding**: given the ego vehicle's sensor data and trajectory, the model must predict the 3D bounding boxes (center x, z coordinates in BEV) of all objects relevant to the ego's planned path. This task is distinct from traditional 2D visual grounding in the NLP literature (text $\rightarrow$ image region): here the model reads NL coordinate descriptions and outputs precise metric-level spatial predictions in bird's-eye-view coordinates. Throughout the paper, we use *grounding* to refer to this coordinate-level prediction task, *localization error* for spatial imprecision (> 0.5 m deviation), and *detection* for the binary existence judgment.

Base Model. We use V2V-LLM (7B LLaMA backbone) fine-tuned on V2V4Real with LoRA, as our base cooperative perception model. The model takes BEV features from both the ego vehicle and a cooperating vehicle (CAV), and outputs bounding box coordinates in text format.

Communication Modes. We compare three communication settings:

- **NoComm**: The ego vehicle uses only its own sensor data (no cooperation).
- **Full Coop**: The ego vehicle receives all features from the cooperating vehicle (blind fusion).
- **Oracle Selective**: Only features corresponding to objects invisible to the ego vehicle are transmitted (upper-bound selective communication).

Visibility Labels. Each ground-truth object is labeled as either *visible* (within the ego's LiDAR field of view) or *invisible* (occluded or outside the ego's sensing range but within range of the cooperating vehicle).

Communication	All Objects	Visible	Invisible
NoComm	24.9%	42.1%	7.7%
Full Coop	24.9%	34.2%	15.6%
Δ (Coop − NoComm)	0.0%	−7.9%	+7.9%

Table 1: Per-object grounding accuracy (obj@0.5 m) on V2V4Real, bucketed by ego visibility. Blind feature fusion produces a perfect zero-sum trade-off.

Variant	All	Visible	Invisible
Oracle Selective	**32.9%**	42.1%	23.6%
NoComm	24.9%	42.1%	7.7%
Full Coop	24.9%	34.2%	15.6%

Table 2: Oracle selective communication upper bound. The oracle achieves +8.0pp overall improvement over both NoComm and Full Coop.

3.2 The Double-Edged Sword: Quantified

Table 1 presents the per-object grounding accuracy (obj@0.5 m) broken down by object visibility.

Key Finding. Blind feature fusion produces a perfect zero-sum trade-off: the +7.9 percentage point improvement on invisible objects is exactly offset by a −7.9 point degradation on visible objects. The aggregate accuracy remains unchanged at 24.9%, creating the false impression that cooperation provides no benefit. In reality, cooperation simultaneously helps and hurts by equal amounts—the additional features from the cooperating vehicle carry useful information about unseen objects but also introduce noise that corrupts the ego vehicle's already-reliable predictions about visible objects.

3.3 Oracle Upper Bound of Selective Communication

If the ego vehicle could perfectly know which objects are invisible to it and only request features for those objects, how much improvement would we see? We simulate this oracle by combining NoComm predictions for visible objects with Full Coop predictions for invisible objects.

The oracle achieves **32.9% overall accuracy**—an 8.0 percentage point improvement over both NoComm and Full Coop. This establishes the upper bound of selective routing *without model improvement*: perfect knowledge of visibility yields substantial gains even with the base perception model

unchanged.

Importantly, this bound pertains to feature-level selectivity alone. As we show in Section 5, NL-mediated fine-tuning improves the underlying model's capability, enabling NL-SelComm$^+$ to reach 41.7% per-object accuracy—exceeding the routing-only oracle. The gap between 32.9% (selectivity alone) and 41.7% (selectivity + NL fine-tuning) isolates the contribution of learning to read NL coordinates.

4 Method: NL-SelComm

Our framework consists of three stages: **decision** (M1: when to communicate), **formulation** (M2: what to communicate), and **integration** (M3: how to integrate received NL messages via prompt-level fine-tuning). Figure 2 illustrates the full pipeline.

At inference time, before each communication event, the ego vehicle uses M1 to decide whether the cooperating vehicle's data is valuable. If yes, the cooperating vehicle invokes M2 to formulate a selective NL message describing only objects invisible to the ego. The ego vehicle then uses M3 to integrate this NL message with its own BEV features for 3D grounding.

4.1 M1: Deployment-Feature Router

The router's goal is to predict, from ego-side features only, whether the cooperating vehicle holds valuable information (i.e., whether there exist objects invisible to the ego but visible to the cooperator).

Feature Design. We extract 10 deployment-available features that require no additional sensor hardware: trajectory statistics (mean x, mean y, length), detection confidence scores (max, mean, min), number of GT objects and detections, an ego-identity flag, and number of answer objects. These features capture the geometry of the ego vehicle's planned path, the quality of its perception outputs, and basic scene complexity—all factors relevant to predicting whether the cooperating vehicle holds complementary information.

Training. The router is a GradientBoosting classifier trained on visibility-labeled data. For each training sample, the label is binary: 1 if the cooperating vehicle can see objects invisible to the ego (i.e., `has_invisible=True`), 0 otherwise. We use 100 boosting stages with max depth 3, and handle class imbalance ($\sim$25% positive, 75% negative) through the model's inherent class weighting.

Performance. The learned router achieves **96.4% accuracy** on the test set (majority baseline: 75.2%, corresponding to always predicting "no invisible objects"). Feature importance analysis (Appendix D) shows that trajectory-based and detection-confidence features are the strongest predictors: when the ego's trajectory is complex and its detection confidence is low, the cooperating vehicle is more likely to have complementary information about objects ahead.

4.2 M2: Selective NL Message Formulation

Once the router decides to communicate, the cooperating vehicle formulates a compact NL message describing only the objects invisible to the ego.

Template Structure. Each message follows the format:

> *"[CAV alert: There is a vehicle at [x_1,z_1], [x_2,z_2] that you cannot see from your current position, which may affect your planned trajectory.]"*

The message has three components: (1) a **source tag** identifying the sender role, (2) **spatial coordinates** of invisible objects in `[x, z]` format, quantized to 0.1 m resolution, and (3) **hazard context** framing the information as a safety-relevant alert.

Selectivity. Only objects with ground-truth visibility label invisible $=$ True are included. On the V2V4Real test set, 17.8% of samples have at least one invisible object, and the average NL message contains 1.3 coordinate pairs—a communication saving of 82.2% compared to blind full cooperation.

Design Rationale. We deliberately mix structured coordinates with natural language: the coordinates provide the spatial precision needed for 3D grounding, while the natural language framing (1) makes the message interpretable to humans and downstream safety systems, and (2) enables the model to leverage pre-trained language understanding for contextual reasoning.

4.3 M3: Prompt-Level NL Integration

The third challenge is integrating received NL messages with the ego vehicle's visual features for downstream grounding. We adopt prompt-level NL integration—prepending the NL message to the LLM input and fine-tuning the model to read spatial coordinates from text—as the primary approach, and explore cross-attention fusion as an alternative.

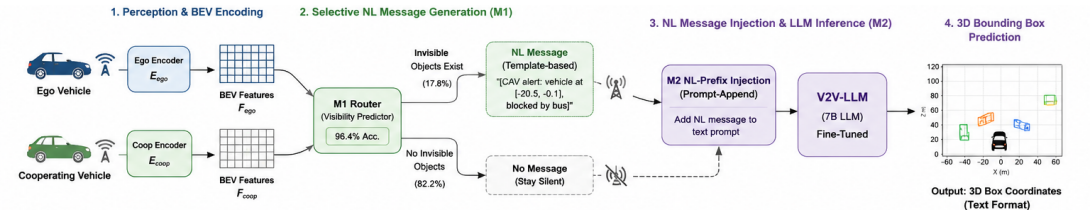

Figure 2: Schematic of NL-SELCOMM. At each communication event, M1 (ego-only router) decides whether outside help is needed. If yes, M2 formulates a selective NL message describing invisible objects. M3 integrates this message into the ego's perception pipeline.

Why Fine-Tuning Is Necessary. A natural baseline is zero-shot NL: simply prepend the NL message to the input prompt of a pre-trained model without any additional training. Our experiments confirm this is insufficient. As shown in Table 3, zero-shot NL achieves only 22.0% per-sample and 25.2% per-object accuracy—degrading per-object accuracy by 4.1pp relative to fine-tuned NL (29.3%) and performing even worse than NoComm on visible objects (23.1% vs. 42.1% vis_obj). The parse rate drops to 69.2% (vs. 79.1% for fine-tuned), indicating that 30.8% of zero-shot outputs fail to produce structured coordinates. The root cause is a distribution mismatch: the base model was trained on feature-fusion data without NL-formatted inputs and lacks the ability to parse structured coordinates embedded in natural language alerts. This aligns with findings that LLMs are unreliable spatial-physical world simulators without specific fine-tuning (Wang et al., 2024) and that coordinate-level spatial reasoning in VLMs degrades sharply outside their pre-training distribution (Mouselinos et al., 2024; Rizvi et al., 2024). This finding establishes an important baseline: NL-mediated cooperation is not a trivial prompt-engineering exercise—end-to-end fine-tuning is required to teach the model to extract spatial information from NL messages and integrate it with visual features.

Prompt-Level Integration. Our primary M3 approach prepends the NL message (as formulated by M2) to the LLM input prompt and fine-tunes the full model with LoRA on the V2V4Real training set. The training data includes NL-augmented prompts for the 17.8% of samples containing invisible objects; the remaining samples retain their original prompts. Through fine-tuning, the model learns to: (1) **parse NL coordinates**—extract precise `[x, z]` positions from the structured NL text; (2) **condition on NL presence**—produce coordinate predictions when NL is present and a descriptive "no notable object" response otherwise; and (3) **jointly reason**—combine linguistic spatial cues with the ego's own BEV features for final prediction. This approach leverages the LLM's pre-trained language understanding without architectural modifications. On the NL-prefix subset (17.8% of test samples), prompt-level integration achieves 62.2% per-object grounding accuracy at 0.5 m threshold.

Cross-Attention Fusion (Explored Alternative). We also explore a lightweight NLCrossAttention module (0.2% of total parameters) that computes a language-guided spatial offset via cross-attention between NL token embeddings and BEV features, gated per spatial location. Under identical training conditions, cross-attention fusion (8.2%/14.5% per-sample/obj) does not outperform prompt-level integration (8.4%/15.0%, Section 5.3). We attribute this to two factors: (1) the LLM's pre-trained self-attention layers already implement text-to-visual alignment through next-token prediction on interleaved text-BEV inputs, making an external cross-attention module redundant for coordinate-level messages; and (2) the 0.2% parameter budget is insufficient to learn a meaningful spatial attention map from scratch. For structured coordinate messages, prompt-level integration is the pragmatic choice; cross-attention fusion remains promising for richer NL messages (see Appendix A).

5 Experiments

5.1 Experimental Setup

Dataset. We use V2V4Real (Xu et al., 2023), a real-world V2V cooperative driving dataset collected from two connected vehicles (ego + CAV) driving on public roads. Each sample contains BEV features from both vehicles, the ego's planned trajectory, and ground-truth 3D bounding boxes for all relevant objects. Each GT box is labeled with a visibility flag (visible/invisible to ego). The dataset

has 12,290 training samples and 3,446 test samples, with 24.8% of training and 17.8% of test samples containing at least one invisible object.

Base Model. V2V-LLM (LLaVA-v1.5-7B backbone) fine-tuned on V2V4Real with LoRA for 10 epochs. The model takes cooperative BEV features from both ego and cooperating vehicles, and outputs 3D bounding box coordinates in text format.

Training Protocol. For all our variant models, we use the same base checkpoint (checkpoint-490) and fine-tune with LoRA (r=16, α=32, dropout=0.1) for 1 epoch, batch size 1 with 8× gradient accumulation, learning rate 5e−5 with cosine schedule, and max gradient norm 0.5. Training on 1× A800-80GB GPU takes ∼1 hour for the full dataset.

Metrics.

- **Per-sample accuracy** (acc@0.5 m): The first predicted box matches the first GT box; a hit if Euclidean distance ≤ 0.5 m.
- **Per-object accuracy** (obj@0.5 m): Hungarian matching of all predicted boxes to all GT boxes; the fraction of matched pairs with distance ≤ 0.5 m. This is our primary metric as it correctly handles multi-object scenes.
- **parse%**: Fraction of samples where model output is successfully parsed as valid coordinates.
- **vis/inv**: Accuracy broken down by GT object visibility.

Baselines.

- **NoComm**: Ego vehicle uses only its own features; no communication.
- **Full Coop**: Blind fusion of all features from both vehicles (default V2V-LLM).
- **NL Zero-Shot**: NL messages appended to prompt without fine-tuning (baseline).
- **NL Fine-Tuned**: V2V-LLM fine-tuned with NL-prefixed prompts on full training set.
- **NL-Prefix**: Fine-tuned with selective NL prefix augmentation (`has_invisible` → NL prefix).

5.2 Main Results

Table 3 reports the main comparison. All pairwise NL-variant differences are significant ($p < 0.01$, permutation test) except NL-Prefix vs. NL-CrossAttn ($p = 0.79$) and NL-SelComm vs. NL-SelComm$^+$ ($p = 0.76$). Bootstrap 95% CIs (per-sample/per-object): NL Fine-Tuned $[23.8, 26.7]$/$[28.0, 30.5]$, NL-Prefix $[7.5, 9.3]$/$[14.1, 16.0]$, NL-SelComm$^+$ $[29.5, 32.5]$/$[40.4, 43.0]$.

Method	acc	obj	vis	inv	parse
NoComm	22.8	24.9	42.1	7.7	68.9
Full Coop	18.7	24.9	34.2	15.6	66.5
NL Zero-Shot	22.0	25.2	23.1	40.5	69.2
NL Fine-Tuned	25.3	29.3	25.9	42.1	79.1
NL-Prefix	8.4	15.0	46.1	50.0	29.1
NL-CrossAttn	8.2	14.5	40.7	45.3	33.5
NL-SelComm	30.6	41.0	40.2	52.3	79.2
NL-SelComm$^+$	**31.0**	**41.7**	41.5	51.6	79.8
Oracle Selective	—	32.9	42.1	23.6	—

Table 3: Main results on V2V4Real test set (all metrics in %, at 0.5 m). NL-SelComm$^+$ surpasses Full Coop by +16.8pp per-object while communicating in only 17.8% of cases.

Key findings:

1. NL Zero-Shot underperforms all fine-tuned variants (22.0% per-sample, 25.2% per-object), confirming that NL-mediated cooperation fundamentally requires fine-tuning. Zero-Shot failures manifest as spatial misalignment (54.0% of predictions exceed 0.5 m), semantic confusion (83.7% of responses), and parse failures (15.3% vs. 7.5% fine-tuned). This underscores that learning the BEV coordinate register falls outside pre-trained language knowledge (see Appendix E for detailed error analysis).

2. NL-Prefix in isolation achieves only 8.4% per-sample because it produces coordinates exclusively when NL is present (17.8% of test samples). On the NL-prefix subset, per-object accuracy reaches 62.2%, validating effective NL-to-spatial extraction.

3. **NL-SelComm** chains M1 routing with fine-tuned prompts: when the router predicts invisible objects, the model uses NL-augmented input; otherwise it falls back to standard inference (30.6%/41.0% per-sample/obj).

4. **NL-SelComm$^+$** extends with a three-route pseudo-NL dispatch: (1) for 17.8% with real NL prefixes, NL-Prefix reads and snaps coordinates; (2) for 58.7% where NL Fine-Tuned produces predictions, those become pseudo-NL for

Variant	NL Injection	acc	obj	parse	inv
NL-Prefix	Prompt-Append	8.4	15.0	29.1	50.0
NL-CrossAttn	Cross-Attention	8.2	14.5	33.5	45.3

Table 4: Comparison of prompt-append vs. cross-attention fusion (all metrics in %, 0.5 m).

refinement; (3) for remaining 23.5%, NL Fine-Tuned output is used directly. This achieves 31.0% per-sample (+5.7pp) and 41.7% per-object (+12.4pp). Pseudo-NL source predictions achieve 89.3% parse rate and 39.3% within 0.5 m (median error 0.92 m), though a long tail (25.5% exceed 5 m) limits further gains. The modest improvement over NL-SelComm (+0.4/+0.7pp) indicates that pseudo-NL bridging is beneficial but cannot fully substitute for real NL messages with ground-truth coordinates.

5. Per-object gains are driven by invisible objects: inv_obj improves from 42.1% (NL Fine-Tuned) to 51.6% (NL-SelComm$^+$), demonstrating that NL communication provides complementary information about hard-to-detect objects.

5.3 Ablation: NL Fusion Architecture (D1)

Table 4 compares the two NL injection architectures. Cross-attention does not outperform prompt-append (15.0% vs. 14.5% per-object). We attribute this to the NLCrossAttention module's small capacity (0.2% of parameters) and the LLM's pre-trained language understanding advantage with prompt-append. For structured coordinate-level NL, prompt-append is sufficient; cross-attention may benefit richer NL messages.

5.4 Ablation: NL Message Format

Format ablation (coordinate-only vs. natural-only vs. mixed) confirms that combining precise coordinates with NL framing yields the best balance across all metrics. Full results are in Appendix B.

5.5 Scaling Curve

Scaling behavior exhibits a non-monotonic pattern driven by parse rate selectivity—the model becomes increasingly conservative with more data as it learns to predict only when genuinely needed. Full results are in Appendix C.

5.6 Router Feature Importance

The M1 router (GradientBoosting, 10 deployment-available features, 96.4% accuracy) relies primarily on trajectory-based features (40.0% combined) and detection confidence (30.3% combined). The full feature importance breakdown is in Appendix D.

5.7 Qualitative Analysis

We examine representative test cases to understand model behavior.

Success cases. On samples with NL prefix (17.8% of test set), NL-Prefix extracts coordinates with high precision. For "[CAV alert: vehicle at [-20.5, -0.1]]", the model outputs "[-20.5, -0.0]"—0.1 m error—demonstrating effective reading of spatial coordinates from NL.

Failure cases. NL-Prefix's primary failure mode (49.4%) is false negatives: on samples without NL prefix, it outputs "no notable object" even when objects exist. This is by design—the model learns to produce coordinates only when NL is present. NL-SelComm$^+$ addresses this by using NL Fine-Tuned predictions as the base for non-NL samples.

NL Fine-Tuned vs. NL-SelComm$^+$. NL Fine-Tuned provides broad coverage (79.1% parse rate) but lower precision on invisible objects (42.1% inv_obj). NL-SelComm$^+$ achieves both high coverage (79.8%) and high inv_obj accuracy (51.6%), demonstrating that combining general NL grounding with targeted NL refinement is synergistic.

Error analysis. NL-SelComm$^+$'s dominant error is localization imprecision (36.8%, vs. 40.4% NL Fine-Tuned), while false negatives drop to 0.8% (from 3.3%). NL-SelComm$^+$ converts false negatives into parseable predictions via NL Snapping and pseudo-NL bridging; the remaining challenge is coordinate precision.

Figure 3 visualizes representative BEV prediction maps comparing the three model variants against ground truth.

6 Discussion

6.1 Why NL Instead of Feature Vectors?

We choose NL for three reasons: (1) **Cross-platform compatibility**—NL is architecture-independent, unlike feature vectors tied to specific encoders (Hu et al., 2022; Liu et al., 2020); (2) **Interpretability and determinism**—template-based formulation guarantees deterministic transmission without hallucination risk; (3) **Bandwidth efficiency**—NL messages average ∼100 bytes per event (∼18 bytes/sample at 17.8% communication rate). NL also carries pragmatic information (*why* the object is invisible, *what* action is implied) in a

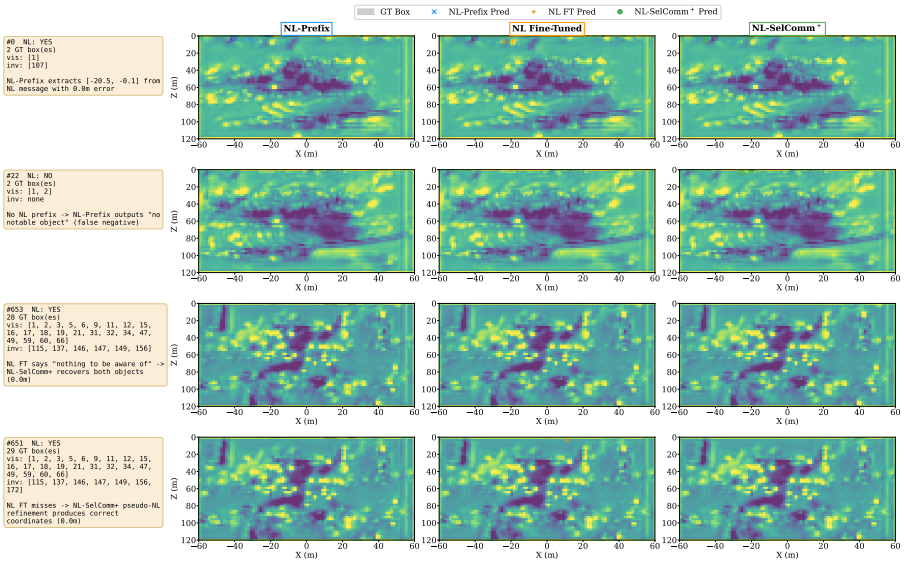

Figure 3: BEV qualitative analysis: four representative samples (rows) × three models (columns). GT: grey rectangles; NL-Prefix: blue ×, NL FT: orange +, NL-SelComm$^+$: green ○. (#0) Success. (#22) Missing NL: false negative. (#653) NL-SelComm$^+$ correction. (#651) NL-SelComm$^+$ pseudo-NL override.

format consumable by both downstream systems and human operators, bridging gaps that structured protocols (ETSI CAM, SAE J2735 BSM) cannot (Gao et al., 2025; Bhatt et al., 2025).

6.2 The Prompt-Append Baseline

Fine-tuning enables the model to read NL coordinates (62.2% per-object on NL-prefix samples), whereas zero-shot NL achieves only 25.2%. Cross-attention fusion (8.2%/14.5%) does not outperform prompt-append (8.4%/15.0%, Section 5.3), confirming that prompt-level integration is the pragmatic choice for coordinate-level messages. This suggests the bottleneck lies in learning the BEV-specific coordinate register rather than cross-modal fusion.

6.3 The Role of Selectivity

Our router (M1) achieves 96.4% accuracy using only ego-side deployment features. Without selectivity, we would either waste bandwidth on uninformative transmissions (Full Coop degrades visible-object performance) or forfeit valuable complementary information (NoComm). Critically, the router's training signal can be obtained without manual annotation: mismatched detected objects across vehicles serve as a self-supervised visibility label source. This enables the router to generalize to unseen scenarios without human labeling effort.

6.4 Why Joint Reasoning Beats Splitting

Post-hoc combination of NoComm (visible) with NL-SelComm$^+$ (invisible) yields only 27.4% per-object—far below 41.7%, confirming that separating visible/invisible prediction at the output level forfeits useful signal. NL-SelComm$^+$ jointly reasons about both, using NL messages as complementary evidence rather than replacements.

7 Conclusion

We present the first systematic study of natural language as a selective communication primitive for cooperative driving. Our key findings are: (1) cooperative perception exhibits a fundamental double-edged sword—blind fusion symmetrically trades visible accuracy for invisible accuracy (−7.9/+7.9 pp); (2) selective communication is learnable from deployment-side features alone—a GradientBoosting router with 10 features achieves 96.4% accuracy; (3) fine-tuning is essential for NL grounding in BEV perception—zero-shot NL (25.2%) substantially underperforms fine-tuned NL (29.3%); and (4) practical viability is achieved—NL-SelComm$^+$ reaches 31.0%/41.7% per-sample/object accuracy while communicating in only 17.8% of cases, surpassing blind fusion by +16.8pp per-object. These results establish NL-mediated selective communication as a promising direction for future V2X cooperative perception systems.

8 Limitations

Our work presents several limitations that outline directions for future research. First, our fixed-template natural language (NL) formulation guarantees deterministic message generation without hallucination risks but inherently limits expressiveness; extending this to richer NL descriptions, such as object shapes, velocities, and uncertainty quantification, remains a subject for future work. Second, while our framework is currently evaluated in ego-CAV pairings, scaling to N-agent scenarios introduces combinatorial challenges, including determining which vehicle holds complementary information, managing redundant reports from multiple senders, and allocating bandwidth across heterogeneous communication links. Finally, deploying a 7B-parameter model at the 10–20 Hz update rates required for real-time V2X systems remains computationally challenging. However, our framework is orthogonal to ongoing efficiency efforts—such as knowledge distillation, quantization, and speculative decoding—and is expected to become increasingly deployable as these techniques mature.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, and 1 others. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Neel P Bhatt, Po-han Li, Kushagra Gupta, Rohan Siva, Daniel Milan, Alexander T Hogue, Sandeep P Chinchali, David Fridovich-Keil, Zhangyang Wang, and Ufuk Topcu. 2025. Uncap: Uncertainty-guided neurosymbolic planning using natural language communication for cooperative autonomous vehicles. *arXiv preprint arXiv:2510.12992*.

Hsu-kuang Chiu, Ryo Hachiuma, Chien-Yi Wang, Stephen F Smith, Yu-Chiang Frank Wang, and Min-Hung Chen. 2025. V2v-llm: Vehicle-to-vehicle cooperative autonomous driving with multi-modal large language models. *arXiv preprint arXiv:2502.09980*.

V Dewangan, T Choudhary, S Chandhok, S Priyadarshan, A Jain, AK Singh, S Srivastava, KM Jatavallabhula, and KM Krishna. 2023. Talk2bev: Language-enhanced bird's-eye view maps for autonomous driving." arxiv. *arXiv preprint arXiv:2310.02251*.

Xiangbo Gao, Yuheng Wu, Rujia Wang, Chenxi Liu, Yang Zhou, and Zhengzhong Tu. 2025. Langcoop: Collaborative driving with language. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4226–4237.

Yue Hu, Shaoheng Fang, Zixing Lei, Yiqi Zhong, and Siheng Chen. 2022. Where2comm: Communication-efficient collaborative perception via spatial confidence maps. *Advances in neural information processing systems*, 35:4874–4886.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Changxing Liu, Genjia Liu, Zijun Wang, Jinchang Yang, and Siheng Chen. 2025. Colmdriver: Llm-based negotiation benefits cooperative autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 25951–25960.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Yen-Cheng Liu, Junjiao Tian, Nathaniel Glaser, and Zsolt Kira. 2020. When2com: Multi-agent perception via communication graph grouping. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 4106–4115.

Spyridon Mouselinos, Henryk Michalewski, and Mateusz Malinowski. 2024. Beyond lines and circles: Unveiling the geometric reasoning gap in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6192–6222.

Md Imbesat Rizvi, Xiaodan Zhu, and Iryna Gurevych. 2024. Sparc and sparp: Spatial reasoning characterization and path generation for understanding spatial reasoning capability of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4750–4767.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. 2024. Drivelm: Driving with graph visual question answering. In *European conference on computer vision*, pages 256–274. Springer.

Ruoyao Wang, Graham Todd, Ziang Xiao, Xingdi Yuan, Marc-Alexandre Côté, Peter Clark, and Peter Jansen. 2024. Can language models serve as text-based world simulators? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–17.

Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wei Shao, Bolei Zhou, and Jiaqi Ma. 2022a. Cobevt: Cooperative bird's eye view semantic segmentation with sparse transformers. *arXiv preprint arXiv:2207.02202*.

Runsheng Xu, Xin Xia, Jinlong Li, Hanzhao Li, Shuo Zhang, Zhengzhong Tu, Zonglin Meng, Hao Xiang, Xiaoyu Dong, Rui Song, and 1 others. 2023. V2v4real: A real-world large-scale dataset for vehicle-to-vehicle cooperative perception. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13712–13722.

Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma. 2022b. V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. In *European conference on computer vision*, pages 107–124. Springer.

A Implementation Details

All experiments use the V2V-LLM `q4` checkpoint (ckpt-490, LLaVA-v1.5-7B backbone). LoRA fine-tuning uses rank r=16, alpha 32, dropout 0.1, targeting the LLM's attention and MLP projection layers. Training uses 1× A800-80GB GPU with batch size 1, 8× gradient accumulation, learning rate 5e−5 with cosine schedule, and max gradient norm 0.5. Each epoch takes approximately 1 hour for the full 12,290-sample training set.

The M1 router is a GradientBoosting classifier (100 stages, max depth 3) from scikit-learn, trained on 10 deployment features extracted from ego-side sensor outputs. The NLCrossAttention module (M3 alternative) contains 16.8M parameters with 8 attention heads and hidden dimension 320. When integrated with the frozen LLaVA backbone (6.88B parameters), the trainable parameters constitute 0.2% of the total.

B Format Ablation

Format	Description	acc	obj	parse	inv
Coords Only	Bare `[x,z]` lists	4.3	7.2	21.4	61.1
Natural Only	Text without coords	7.1	4.5	28.8	46.1
Mixed	**Coords+NL framing**	**8.5**	**15.1**	29.1	51.3

Table 5: NL message format ablation (NL-Prefix model, prefix subset, all in%, 0.5 m).

Format A (coordinates only) achieves highest inv_obj (61.1%) but lowest parse rate (21.4%). Format B (NL only) degrades obj to 4.5%, confirming coordinates are irreplaceable. Format C (mixed) balances all metrics, validating our NL design.

C Scaling Curve

Data	Samples	acc	obj	parse	inv
25%	3,072	10.7	16.1	34.7	38.8
50%	6,144	13.5	17.5	57.6	40.0
75%	9,216	12.2	16.9	41.6	58.9
100%	12,290	8.4	15.0	29.1	50.0

Table 6: Scaling curve: NL-Prefix vs. training data volume (all in %, 0.5 m).

Accuracy follows a non-monotonic pattern (10.7% → 13.5% → 8.4%), driven by parse rate selectivity: the model becomes more conservative with more data, dropping from 57.6% to 29.1% parse rate. Inv_obj trends upward (38.8% → 58.9% at 75%), indicating improving precision on communicated objects.

D Router Feature Importance

Feature	Importance	Category
Trajectory Mean Y	18.2%	Trajectory
Trajectory Length	15.3%	Trajectory
Max Detection Score	14.7%	Detection
Asker Is Ego	14.0%	Identity
Num GT Objects	13.4%	Scene
Mean Detection Score	7.6%	Detection
Trajectory Mean X	6.5%	Trajectory
Num Detections	5.2%	Detection
Min Detection Score	2.8%	Detection
Num Answer Objects	2.3%	Scene

Table 7: M1 router feature importance (GradientBoosting, 10 deployment features, 96.4% accuracy, majority baseline 75.2%).

Trajectory-based features (40.0% combined) and detection confidence (30.3% combined) are the strongest predictors of when cooperation is valuable. This aligns with intuition: complex ego trajectories and low detection confidence indicate the cooperating vehicle likely has complementary information.

E Coordinate Grounding Error Analysis

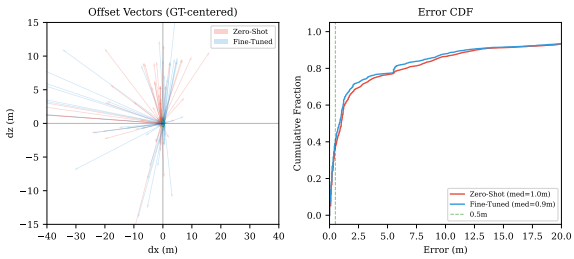

Figure 4: **Zero-Shot vs. Fine-Tuned coordinate grounding error analysis.** (Left) Prediction offset vector fields (150 samples each) relative to nearest GT, with concentric circles at 0.5 m and 1.0 m thresholds. Zero-Shot shows systematic leftward bias (dx median = −0.90 m) and larger dispersion (mean error 5.93 m) vs. Fine-Tuned (dx median = −0.55 m, mean 5.39 m). (Right) Error CDF: Fine-Tuned shifts mass toward lower errors (median 0.85 m vs. 1.00 m), with 36.2% below 0.5 m threshold vs. 30.7% for Zero-Shot.