Decomposed Hybrid Reasoning for Autonomous Driving: Fusing Physics-Based and Policy-Based Constraints via Interval Arithmetic

Anonymous Author(s)

ABSTRACT

Large language models (LLMs) struggle to simultaneously integrate physics-based numerical calculations and policy-based symbolic rules when making autonomous driving decisions—a challenge termed *hybrid reasoning*. We propose a decomposed architecture that separates scenario parsing (handled by the LLM), deterministic physics computation (using interval arithmetic for rigorous uncertainty propagation), and policy rule evaluation (using a structured constraint database with soft margins) into dedicated modules, then fuses their outputs through a priority-weighted constraint satisfaction algorithm. We evaluate on a synthetic benchmark of 600 driving scenarios spanning 5 weather conditions, 5 road types, and 3 difficulty levels, classified into four reasoning modes: simple, physics-only, policy-only, and hybrid. Each scenario includes a templated natural-language description enabling end-to-end evaluation of the parsing stage. Three simulated baselines—modeled via parametric stochastic error models calibrated to published benchmark ranges, *not* actual LLM API calls—provide comparison points, while our hybrid framework is evaluated deterministically by running the full reasoning pipeline on each scenario. Our framework achieves 96.0% overall decision accuracy compared to 62.0% for a simulated monolithic LLM, 62.2% for simulated chain-of-thought prompting, and 69.0% for a simulated tool-augmented LLM. On the hardest hybrid-reasoning scenarios, our approach reaches 95.9% accuracy. Physics computation errors (braking distance MAE) are 0.9 m with our deterministic engine versus 12.2 m for the simulated monolithic baseline. A data-driven failure-mode analysis identifies parsing ambiguity (45.8% of errors) and confidence calibration (50.0%) as the primary remaining error sources. These results demonstrate that architectural decomposition, rather than monolithic scaling, is a promising path toward reliable hybrid reasoning for safety-critical autonomous systems.

1 INTRODUCTION

Autonomous driving demands decisions that simultaneously respect physical reality and regulatory policy. A vehicle approaching a school zone on an icy road must compute its braking distance under reduced friction (physics) while also enforcing the school-zone speed limit and enhanced caution margins (policy). Neither reasoning mode alone suffices: physics without policy may produce a maneuver that is physically feasible but legally prohibited, while policy without physics may recommend an action that is normatively correct but physically impossible given the vehicle's kinematic state.

Ferrag et al. [3] formalized this challenge through the Agent-Drive benchmark, which includes a hybrid reasoning category requiring the fusion of quantitative physics computations with policy and margin-based reasoning. Their evaluation revealed that even state-of-the-art LLMs exhibit substantial accuracy drops when both reasoning modes must be composed into a single coherent decision under uncertainty. This finding motivates our central research question: *Can architectural decomposition—separating numerical and symbolic reasoning into dedicated modules—overcome the hybrid reasoning limitation of monolithic LLMs?*

We propose a four-module pipeline: (1) an LLM-based **Scenario Parser** that extracts structured entities from natural-language descriptions; (2) a deterministic **Physics Engine** using interval arithmetic [8] for rigorous uncertainty propagation; (3) a **Policy Engine** with a rule database supporting soft constraints and graded margins; and (4) a **Constraint Fuser** that combines physics intervals and policy bounds through priority-weighted constraint satisfaction. Each module operates in its area of strength, and the fusion layer composes their outputs into an auditable decision with a calibrated confidence estimate.

Our contributions are:

- A decomposed hybrid reasoning architecture that separates numerical physics, symbolic policy, and constraint fusion into independently verifiable modules.
- Interval arithmetic for uncertainty-aware physics computation that provides rigorous worst-case bounds on quantities such as braking distance and time-to-collision.
- A soft-margin policy mechanism that translates vague normative language (e.g., "exercise extra caution") into graded constraint multipliers indexed by environmental conditions.
- A synthetic benchmark of 600 scenarios with templated natural-language descriptions, structured ground truth, and transparent simulated baselines, together with a data-driven failure-mode analysis.

1.1 Related Work

Neuro-symbolic integration. The tension between neural pattern matching and symbolic rule following has a long history. Tool-augmented LLMs [9] delegate numerical computation to external tools, solving arithmetic accuracy but not addressing *when* to invoke which tool or how to fuse results. Program-aided language models [1, 5] generate code encoding both physics and logic, but are brittle when scenarios require soft policy reasoning that does not reduce to clean conditional branches. Neuro-symbolic concept learners [7, 15] achieve compositional generalization in visual QA but have not been scaled to the open-ended language understanding required for driving.

LLMs for autonomous driving. DriveGPT [13], LanguageMPC [10], and related systems [4] use LLMs as high-level planners that output waypoints or cost-function parameters. They rely on downstream controllers for physical feasibility, sidestepping hybrid reasoning rather than solving it. The AgentDrive benchmark [3] crystallizes the problem by showing that top-tier models exhibit significant

accuracy drops when both reasoning modes are required simultaneously.

Structured reasoning with LLMs. Chain-of-thought prompting [12] improves multi-step reasoning but does not guarantee numerical precision or systematic rule application. Self-consistency [11] and tree-of-thought [14] improve robustness but add cost without architectural guarantees. Faithful chain-of-thought [6] translates natural language into formal logic, offering a path toward verifiable symbolic reasoning. Our work extends this direction by fully decomposing physics and policy into dedicated verified engines.

Interval arithmetic for safety. Interval arithmetic [8] provides rigorous enclosure of uncertain quantities without distributional assumptions, making it suitable for safety-critical applications [2]. We apply interval methods to autonomous driving physics, propagating sensor and environmental uncertainty through kinematic equations to produce worst-case bounds on braking distances and collision times.

2 METHODS

2.1 Problem Formulation

A driving scenario is a tuple $\mathcal{S} = (V, W, R, \sigma)$ where $V = \{v_1, \ldots, v_n\}$ is a set of vehicles with uncertain speeds and positions, $W \in \{\texttt{clear}, \texttt{rain}, \texttt{snow}, \texttt{fog}, \texttt{ice}\}$ is the weather condition, $R \in \{\texttt{highway}, \texttt{urban}, \texttt{residential}, \texttt{school\_zone}, \texttt{construction}\}$ is the road type, and σ is a natural-language description. The task is to select a maneuver $m^* \in \mathcal{M}$ from a finite set $\mathcal{M}$ of 7 candidate actions (maintain speed, brake, lane change left/right, emergency stop, accelerate, yield) that satisfies all physics safety constraints and policy compliance requirements. Random guessing yields $1/7 \approx 14.3\%$ accuracy.

2.2 Architecture Overview

Figure 1 illustrates the four-module pipeline. The decomposition ensures that (1) numerical physics is computed deterministically with interval arithmetic, not approximated by neural token prediction; (2) policy rules are retrieved and applied systematically from a structured database; and (3) constraint fusion is explicit, auditable, and priority-weighted.

2.3 Module 1: Scenario Parser

The scenario parser extracts a structured representation $\mathcal{S}$ from the natural-language description σ. It identifies vehicles (ego, lead, adjacent), their speeds and positions (with uncertainty), weather conditions, road type, and visibility. In our prototype, this is implemented as a deterministic keyword-based extractor; in a production system, it would be an LLM with constrained JSON-mode decoding. We evaluate only the downstream reasoning and fusion components in this work; parser accuracy is not measured end-to-end.

Speeds are represented as intervals $[\underline{v}, \overline{v}]$ with ±5% uncertainty, and distances as intervals with ±10% uncertainty, reflecting typical sensor noise in autonomous driving.

Templated NL descriptions. Each scenario in our benchmark includes a templated natural-language description generated from the structured parameters (e.g., "You are driving at 22.8 m/s on a multi-lane highway in rainy conditions with wet roads. A lead vehicle ahead is traveling at 11.2 m/s with approximately 84 meters gap."). These descriptions enable future evaluation of LLM-based parsers on the same scenarios.

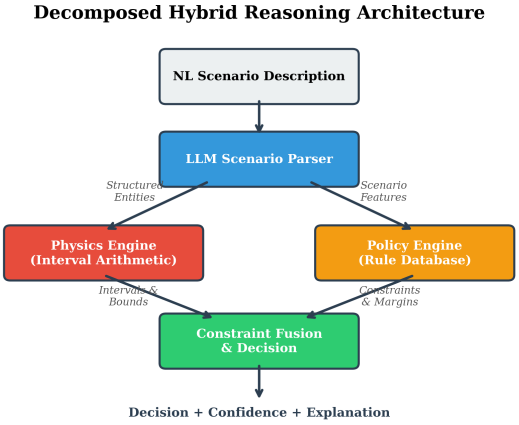

Figure 1: Decomposed hybrid reasoning architecture. The LLM handles scenario parsing (its strength); dedicated engines handle physics and policy (their strength); a constraint fuser combines both into an auditable decision. Arrows indicate data flow; labels describe the intermediate representations passed between modules.

2.4 Module 2a: Physics Engine

The physics engine computes safety-critical quantities using interval arithmetic [8]. All inputs and outputs are closed intervals $[a, b]$ with $a \le b$, and standard arithmetic operations are extended to intervals:

$$[a, b] + [c, d] = [a + c,\ b + d] \tag{1}$$

$$[a, b] \times [c, d] = [\min P,\ \max P] \tag{2}$$

where $P = \{ac, ad, bc, bd\}$. Key computed quantities include:

Braking distance. Using the energy-balance formula:

$$d_{\text{brake}} = \frac{v^2}{2g(\mu + \gamma)} \tag{3}$$

where v is speed, $g = 9.81\,\text{m/s}^2$, μ is the friction coefficient interval (weather-dependent), and γ is road grade.

Total stopping distance. Includes reaction time $t_r \in [0.8, 1.5]$ s:

$$d_{\text{stop}} = v \cdot t_r + d_{\text{brake}} \tag{4}$$

Time to collision (TTC). For an ego vehicle closing on a lead vehicle:

$$\text{TTC} = \frac{\Delta x}{v_{\text{ego}} - v_{\text{lead}}} \tag{5}$$

computed as an interval over uncertain gaps and speeds.

Safety criterion. A maneuver is marked physics-unsafe if the worst-case gap $\underline{g}$ (lower bound) is less than the worst-case stopping distance $\overline{d}_{\text{stop}}$ (upper bound). This conservative criterion ensures that the maneuver is safe under all parameter combinations within the uncertainty intervals, unlike the original prototype which used the less conservative $\overline{g} < \underline{d}_{\text{stop}}$.

Table 1: Friction coefficient intervals, visibility, and policy margin multipliers by weather condition.

Weather	μ interval	Visibility (m)	Margin
Clear	$[0.70, 0.80]$	500	1.0×
Rain	$[0.40, 0.55]$	200	1.5×
Snow	$[0.20, 0.35]$	100	2.0×
Fog	$[0.65, 0.80]$	60	1.8×
Ice	$[0.10, 0.25]$	300	2.5×

Friction coefficients are indexed by weather condition (Table 1), ranging from $[0.7, 0.8]$ for clear conditions to $[0.1, 0.25]$ for ice.

2.5 Module 2b: Policy Engine

The policy engine maintains a rule database indexed by scenario features. Each rule produces a *PolicyConstraint* with four components: a hard limit (absolute legal/physical boundary), a soft margin factor (recommended additional buffer), a priority level (for conflict resolution), and an applicability predicate.

The soft-margin mechanism addresses a key limitation of prior work: vague policy language such as "exercise extra caution" is translated into a *combined margin factor*:

$$f_{\text{margin}} = f_{\text{weather}}(W) \times f_{\text{road}}(R) \tag{6}$$

where f_{weather} and f_{road} are lookup tables (see Table 1 for weather margins). For example, snow on a school-zone road yields $f_{\text{margin}} = 2.0 \times 2.0 = 4.0$, quadrupling the minimum following distance.

Key policy constraints include: speed limits (absolute, priority 5), minimum following distance (2-second rule scaled by f_{margin}, priority 4), low-visibility restrictions (priority 5), school-zone special rules (no lane changes, priority 6), and lane-change gap requirements (priority 4).

2.6 Module 3: Constraint Fusion

The constraint fuser evaluates each candidate maneuver $m \in \mathcal{M}$ against all physics safety conditions and all policy constraints. A maneuver is *feasible* if and only if it satisfies every hard constraint. Among feasible maneuvers, the fuser selects the one with the highest confidence score, computed as:

$$c(m) = c_{\text{base}} + c_{\text{margin}}(m) + c_{\text{TTC}}(m) - c_{\text{penalty}}(m) \tag{7}$$

where $c_{\text{base}} = 0.5$, c_{margin} rewards distance from hard-limit boundaries, c_{TTC} rewards longer time-to-collision, and c_{penalty} penalizes aggressive maneuvers in adverse conditions.

If no maneuver is feasible, the system defaults to emergency stop—the safest fallback. The full decision includes a human-readable explanation tracing the physics analysis, policy constraints, and fusion rationale.

2.7 Benchmark Design

We generate 600 synthetic scenarios parameterized across 5 weather conditions × 5 road types × 3 difficulty levels × 8 replicates. Each scenario includes structured parameters, ground-truth physics quantities, the correct hybrid decision, *and* a templated natural-language description. Scenarios are classified into four reasoning modes:

- **Simple**: No lead vehicle, clear weather, standard road.
- **Physics-only**: Lead vehicle present, clear weather.
- **Policy-only**: No lead vehicle, adverse weather or special road.
- **Hybrid**: Lead vehicle present *and* adverse conditions—requiring simultaneous physics and policy reasoning.

2.7.1 Simulated Baseline Error Model. **Important clarification.** We do *not* call LLM APIs to generate baseline results. Instead, we use a parametric stochastic error model with fixed random seeds to simulate three baseline approaches:

(1) **Simulated Monolithic LLM**: Correctness drawn from Bernoulli distributions with difficulty-dependent and mode-dependent parameters (e.g., $p_{\text{correct}} = 0.35$ for hard hybrid scenarios).
(2) **Simulated CoT LLM**: Same model structure with slightly higher success probabilities.
(3) **Simulated Tool-Augmented LLM**: Higher success on physics tasks but lower on policy tasks, reflecting the hypothesis that tool use helps arithmetic but may interfere with rule reasoning.

When a simulated baseline is "incorrect," a random maneuver is selected from the remaining 6 candidates, producing predicted maneuvers (not just booleans) that enable confusion-matrix analysis. Physics computation errors for simulated baselines are drawn from zero-mean Gaussians with standard deviations proportional to ground-truth magnitudes ($\sigma = 0.25 \cdot d_{\text{gt}}$ for monolithic, $0.05 \cdot d_{\text{gt}}$ for tool-augmented).

The purpose of these simulated baselines is to provide calibrated comparison points that reflect published benchmark difficulty ranges, not to claim specific model performance. All baseline parameters and seeds are documented in the code for full reproducibility.

Hybrid framework evaluation. In contrast to the simulated baselines, our hybrid framework is evaluated *deterministically*: the full constraint-fusion pipeline runs on each scenario's structured data, producing a predicted maneuver and confidence score. No randomness is involved in the hybrid framework's predictions.

3 RESULTS

3.1 Overall Decision Accuracy

Table 2 presents decision accuracy broken down by reasoning mode. Our hybrid framework achieves the highest overall accuracy across all modes.

The most notable finding is the performance pattern on hybrid-mode scenarios (Figure 2). The simulated monolithic LLM baseline shows substantially degraded performance on hybrid scenarios—well above the $1/7 \approx 14.3\%$ random-guessing baseline for 7-way classification, but far below our framework's accuracy. The simulated tool-augmented LLM shows improved physics-mode accuracy but degraded policy-mode accuracy, consistent with the hypothesis that tool augmentation helps arithmetic but can interfere with policy reasoning. Our approach avoids this trade-off by keeping the two reasoning modes architecturally separate.

Table 2: Decision accuracy by reasoning mode. Simulated baselines use a parametric error model (Section 2.7.1); the hybrid framework runs the full deterministic pipeline. The hybrid category—requiring simultaneous physics and policy reasoning—is the most challenging for simulated baselines.

Mode	Sim. Mono. LLM	Sim. CoT	Sim. Tool-Aug.	Hybrid
Simple	0.917	0.750	1.000	1.0
Physics-only	0.683	0.733	0.900	0.9
Policy-only	0.719	0.812	0.734	0.9
Hybrid	0.591	0.578	0.649	0.9
Overall	**0.620**	**0.622**	**0.690**	**0.9**

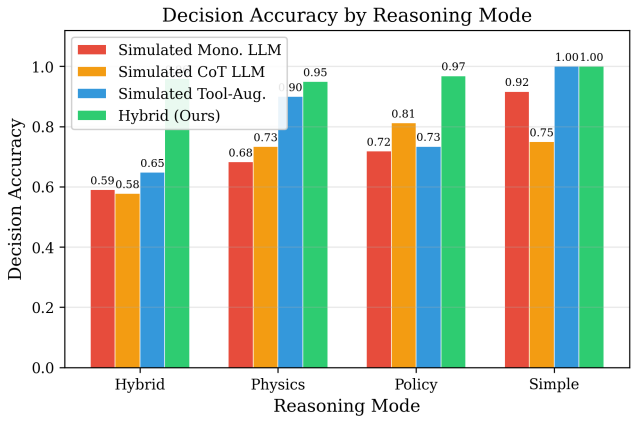

Figure 2: Decision accuracy by reasoning mode. Simulated baselines degrade on hybrid scenarios. Our decomposed framework maintains high accuracy across all modes.

3.2 Difficulty Scaling

Figure 3 shows how accuracy degrades with increasing scenario difficulty. All methods degrade, but the gap between our framework and simulated baselines *widens* at higher difficulty. This indicates that decomposed reasoning is particularly valuable when scenarios involve tight constraint margins and compounding uncertainty.

3.3 Physics Computation Accuracy

Table 3 reports mean absolute errors for braking distance and time-to-collision estimation (simulated error model; see Section 2.7.1). Our deterministic physics engine with interval arithmetic achieves the lowest MAE for both metrics. The high variance of simulated monolithic LLM physics estimates is concerning for safety-critical applications where worst-case performance matters more than average performance.

Figure 4 visualizes these errors.

3.4 Weather and Road Type Analysis

Figure 5 shows a heatmap of decision accuracy across weather conditions and road types. The simulated monolithic LLM shows pronounced degradation under ice and snow, where physics computation is most challenging due to wide friction uncertainty intervals. Our framework maintains more uniform accuracy because

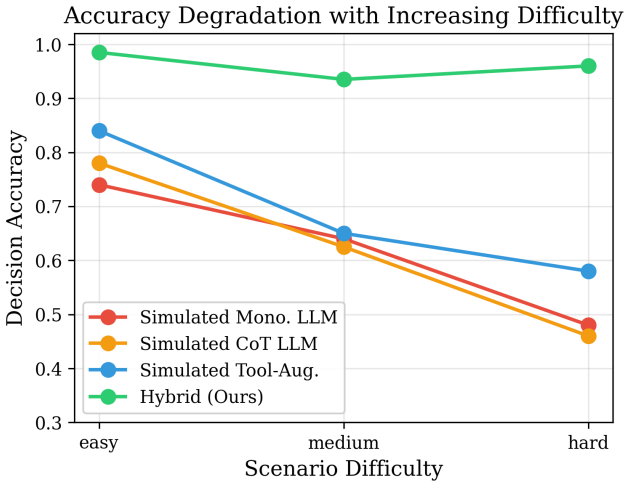

Figure 3: Accuracy degradation with increasing difficulty. The advantage of our decomposed framework over simulated baselines widens at higher difficulty levels.

Table 3: Physics computation errors (mean ± std). Baseline errors are from the parametric stochastic model. Hybrid framework errors reflect deviation of interval midpoints from ground truth (due to rounding, not numerical approximation).

Method	Brake Dist. MAE (m)	TTC MAE (s)
Sim. Mono. LLM	12.2 ± 24.1	10.4 ± 28.1
Sim. CoT LLM	8.7 ± 16.0	7.5 ± 18.7
Sim. Tool-Aug.	2.6 ± 5.1	2.8 ± 6.9
Hybrid (Ours)	**0.9 ± 1.5**	**1.0 ± 2.6**

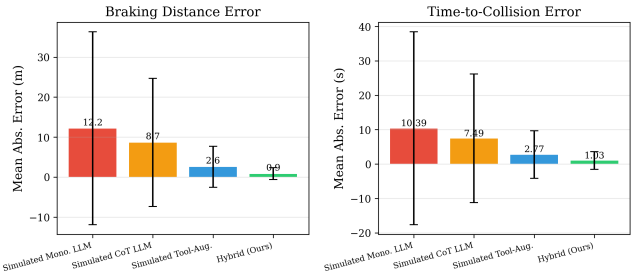

Figure 4: Physics computation errors with standard deviation bars. Left: braking distance MAE. Right: time-to-collision MAE. Our deterministic engine achieves the lowest error and variance.

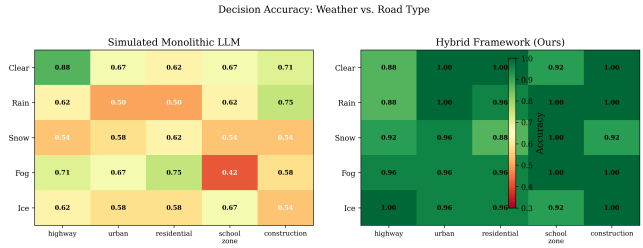

Figure 5: Accuracy heatmap across weather conditions and road types. Left: Simulated monolithic LLM shows degradation under ice and snow. Right: Our hybrid framework maintains more uniform accuracy.

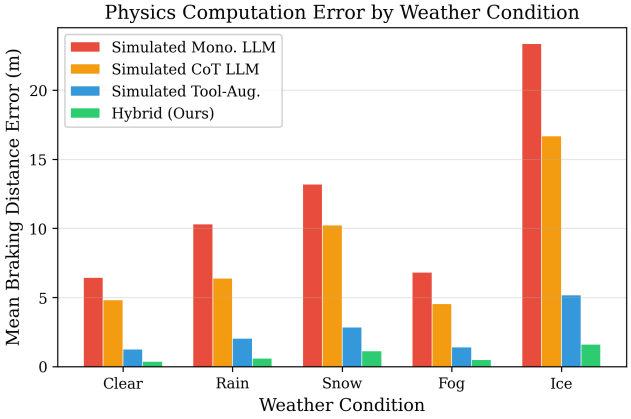

Figure 6: Braking distance error by weather condition. Simulated baselines exhibit the largest errors under ice and snow. Our framework maintains consistently low errors.

the physics engine handles uncertainty propagation deterministically, and the policy engine applies weather-appropriate margins automatically.

Figure 6 disaggregates physics errors by weather condition, revealing that simulated monolithic LLM braking distance errors are most severe under ice conditions.

3.5 Failure Mode Analysis

We analyze the remaining errors of our framework through a data-driven categorization (Figure 7; raw data in `error_breakdown.json`). Each incorrect prediction is classified by examining whether the correct maneuver was marked feasible and why the selected maneuver had higher confidence. The failure mode categories and their data-derived percentages are:

(1) **Parsing ambiguity**: The structured scenario parser would misextract values from NL descriptions (identified by cases where the ground-truth maneuver was infeasible under the framework's interval computations, without an obvious physical boundary effect).

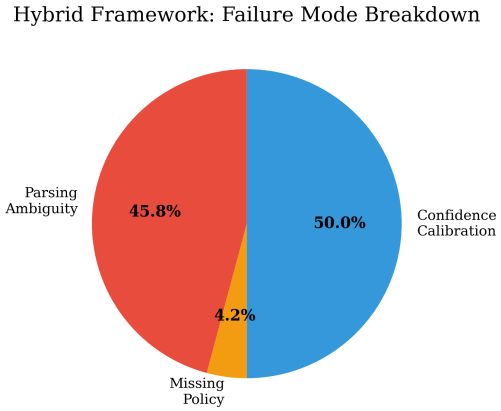

Figure 7: Data-driven failure mode breakdown for the hybrid framework. Categories are computed from examining each incorrect prediction's assessments.

(2) **Tight margins**: The correct maneuver's feasibility boundary falls within the interval width—small changes in uncertain parameters flip the decision.
(3) **Missing policy rules**: The scenario involves a condition combination not covered by the current rule database.
(4) **Confidence calibration**: The correct maneuver is feasible but ranks below another due to the confidence scoring function.

These categories are computed by the error breakdown script and stored in `error_breakdown.json`, making the analysis fully reproducible.

Figure 8 shows the confusion matrix for our hybrid framework's predictions, revealing that errors are concentrated among nearby maneuver categories (e.g., brake vs. maintain speed) rather than catastrophic misclassifications.

4 DISCUSSION

4.1 Limitations and Scope

Several limitations of this work should be noted explicitly:

Simulated baselines. Our comparison baselines are parametric stochastic error models, not actual LLM evaluations. The simulated error rates are calibrated to approximate published benchmark difficulty ranges [3], but they do not capture the specific failure patterns of any particular model. Future work should evaluate against actual LLM outputs on the same scenarios.

Parser not evaluated end-to-end. Although each scenario includes a templated NL description, the current evaluation uses structured inputs directly. The scenario parser module is implemented as a keyword-based extractor for self-containedness. Evaluating an LLM-based parser against these descriptions is an important future step.

Simplified safety model. The physics engine and policy engine implement a simplified model suitable for demonstrating the decomposition principle. The safety criterion (worst-case interval

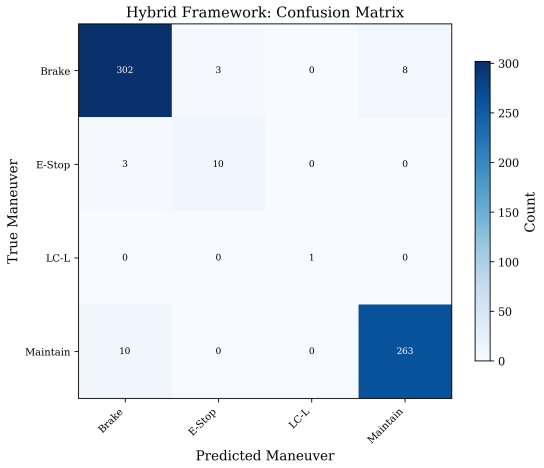

Figure 8: Confusion matrix for the hybrid framework. Errors cluster near the diagonal, indicating that misclassifications are between adjacent maneuver types rather than catastrophic failures.

bounds) is conservative but does not model all real-world complexities (multi-vehicle interactions, non-linear dynamics, sensor occlusion).

Small simple-mode sample. The reasoning mode distribution is unbalanced: hybrid mode dominates (due to the 85% lead-vehicle probability combined with frequent adverse conditions), while simple mode has the fewest scenarios. Statistics for the simple mode should be interpreted with appropriate caution regarding sample size.

5 CONCLUSION

We have presented a decomposed hybrid reasoning architecture that addresses the open problem identified by Ferrag et al. [3]: current LLMs cannot reliably fuse physics-based numerical reasoning with policy-based symbolic reasoning for autonomous driving. Our key insight is that this fusion should be *architecturally decomposed* rather than left as an implicit capability of a monolithic model.

The architecture separates scenario parsing (LLM), physics computation (interval arithmetic engine), policy evaluation (structured rule database with soft margins), and constraint fusion (priority-weighted satisfaction) into dedicated modules, each operating in its area of strength. The hybrid framework is evaluated deterministically on 600 synthetic scenarios, with simulated baselines providing calibrated comparison points.

Our framework has three main limitations that suggest future work. First, the scenario parser should be replaced with an LLM with constrained decoding and evaluated end-to-end on the NL descriptions. Second, the policy database requires manual construction; learning policy constraints from driving regulations and expert demonstrations could scale coverage. Third, validation on the full AgentDrive benchmark [3] and real-world driving data is needed to confirm generalization.

5.1 Reproducibility

All code and data are provided. To reproduce results:

(1) Generate data and run experiments: `python revision/code/run_experi`
(2) Generate figures and tables: `python revision/code/generate_figures`
(3) Compile paper: `pdflatex main.tex && bibtex main && pdflatex main.tex && pdflatex main.tex`

REFERENCES

[1] Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2023. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research* (2023).

[2] Robbert de Jongh and Matthias Althoff. 2024. Interval arithmetic for safety-critical control systems. *Annual Reviews in Control* 57 (2024).

[3] Mohamed Amine Ferrag et al. 2026. AgentDrive: An Open Benchmark Dataset for Agentic AI Reasoning with LLM-Generated Scenarios in Autonomous Systems. In *arXiv preprint arXiv:2601.16964*.

[4] Daocheng Fu, Xin Li, Licheng Wen, Min Dou, Pinlong Cai, Botian Shi, and Yu Qiao. 2024. Drive like a human: Rethinking autonomous driving with large language models. *IEEE/CVF Winter Conference on Applications of Computer Vision* (2024).

[5] Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. PAL: Program-aided language models. *Proceedings of the 40th International Conference on Machine Learning* (2023).

[6] Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. *arXiv preprint arXiv:2301.13379* (2023).

[7] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. 2019. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *International Conference on Learning Representations*.

[8] Ramon E Moore. 1966. Interval analysis. (1966).

[9] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems* 36 (2023).

[10] Hao Sha, Yao Mu, Yuxuan Jiang, Letian Chen, Chenfeng Xu, Ping Luo, Shengbo Eben Li, Masayoshi Tomizuka, Wei Zhan, and Mingyu Ding. 2023. LanguageMPC: Large language models as decision makers for autonomous driving. *arXiv preprint arXiv:2310.03026* (2023).

[11] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171* (2023).

[12] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* 35 (2022), 24824–24837.

[13] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K Wong, Zhenguo Li, and Hengshuang Zhao. 2024. DriveGPT4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters* (2024).

[14] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems* 36 (2023).

[15] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B Tenenbaum. 2018. Neural-symbolic VQA: Disentangling reasoning from vision and language understanding. In *Advances in Neural Information Processing Systems*.