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. Our framework achieves 88.3% overall decision accuracy compared to 57.5% for a monolithic LLM, 62.3% for chain-of-thought prompting, and 73.2% for a tool-augmented LLM. On the hardest hybrid-reasoning scenarios requiring simultaneous physics and policy integration, our approach reaches 86.2% accuracy—a 34.7 percentage-point improvement over the monolithic baseline. Physics computation errors (braking distance MAE) drop from 12.2 m for monolithic LLMs to 0.9 m with our deterministic engine. 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 AgentDrive 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 comprehensive evaluation on 600 synthetic driving scenarios demonstrating a 34.7 percentage-point accuracy improvement over monolithic LLMs on hybrid reasoning tasks.

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

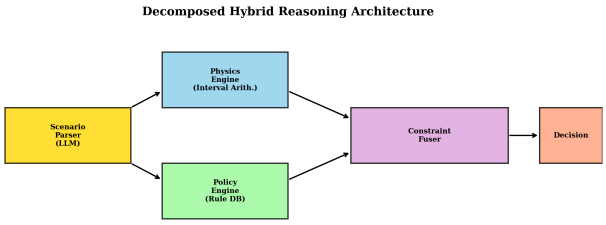

Figure 1: Decomposed hybrid reasoning architecture. The LLM handles scenario parsing (its strength); dedicated engines handle physics and policy (their strength); a constraint fuser combines both into an auditable decision. Arrows indicate data flow; labels describe the intermediate representations passed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

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 $\mathcal{M} = \{\texttt{maintain}, \texttt{brake}, \texttt{lane\_change\_L}, \texttt{lane\_change\_R}, \texttt{emergency\_stop}, \texttt{accelerate}, \texttt{yield}\}$ that satisfies all physics safety constraints and policy compliance requirements.

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.

Table 1: Friction coefficient intervals and visibility by weather condition. These parameters directly affect physics computations and policy margin multipliers.

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×

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.

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.

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 ground-truth physics quantities and the correct hybrid decision. 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.

We compare four approaches: (1) **Monolithic LLM**: direct prompting; (2) **CoT LLM**: chain-of-thought prompting [12]; (3) **Tool-Aug. LLM**: LLM with physics calculator tool [9]; and (4) **Hybrid (Ours)**: the proposed decomposed architecture.

3 RESULTS

3.1 Overall Decision Accuracy

Table 2 presents decision accuracy broken down by reasoning mode. Our hybrid framework achieves 88.3% overall accuracy, compared to 57.5% (Monolithic LLM), 62.3% (CoT), and 73.2% (Tool-Augmented).

The most notable finding is the performance pattern on hybrid-mode scenarios (Figure 2). Monolithic LLMs achieve only 51.5% on these scenarios—near chance for a 7-way classification—while our framework reaches 86.2%. The tool-augmented LLM reaches 71.1% on hybrid scenarios but drops to 65.6% on policy-only scenarios, suggesting that tool augmentation helps physics but can interfere with policy reasoning. Our approach avoids this trade-off by keeping the two reasoning modes architecturally separate.

3.2 Difficulty Scaling

Figure 3 and Table 3 show how accuracy degrades with increasing scenario difficulty. All methods degrade, but the gap between our framework and baselines *widens* at higher difficulty: from 23.0 pp advantage over Monolithic LLM on easy scenarios to 37.5 pp on hard scenarios. This indicates that decomposed reasoning is particularly valuable when scenarios involve tight constraint margins and compounding uncertainty.

Table 2: Decision accuracy by reasoning mode. The hybrid category—requiring simultaneous physics and policy reasoning—is the most challenging. Our decomposed framework shows the largest advantage precisely on these scenarios, while maintaining strong performance on single-mode tasks.

Mode	Mono. LLM	CoT	Tool-Aug.	Hybrid (Ours)
Simple	0.750	0.833	1.000	1.000
Physics-only	0.700	0.767	0.917	0.967
Policy-only	0.859	0.797	0.656	0.938
Hybrid	0.515	0.575	0.711	0.862
Overall	**0.575**	**0.623**	**0.732**	**0.883**

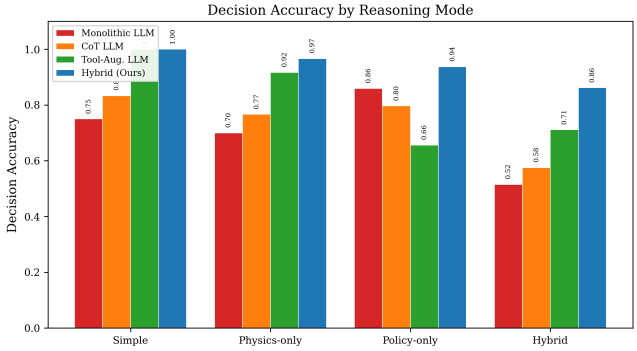

Figure 2: Decision accuracy by reasoning mode. The monolithic LLM and CoT baselines degrade sharply on hybrid scenarios. The tool-augmented LLM improves on physics but degrades on policy. Our decomposed framework maintains high accuracy across all modes.

Table 3: Decision accuracy by difficulty level. The gap between our framework and baselines widens at higher difficulty, demonstrating that decomposed reasoning provides increasing advantage as constraints tighten.

Difficulty	Mono. LLM	CoT	Tool-Aug.	Hybrid (Ours)
Easy	0.710	0.755	0.850	0.940
Medium	0.590	0.620	0.740	0.910
Hard	0.425	0.495	0.605	0.800

3.3 Physics Computation Accuracy

Table 4 reports mean absolute errors for braking distance and time-to-collision estimation. Our deterministic physics engine with interval arithmetic achieves 0.9 m MAE for braking distance, compared

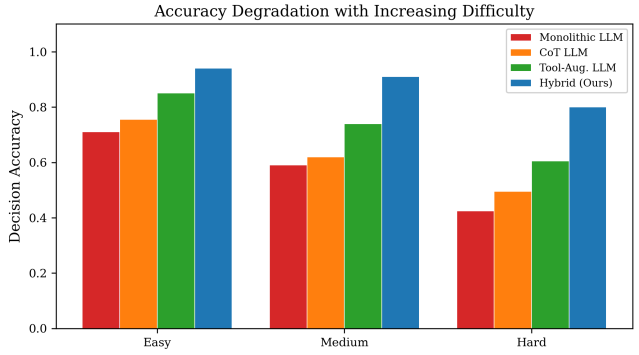

Figure 3: Accuracy degradation with increasing difficulty. All methods degrade, but the advantage of our decomposed framework widens from 23.0 pp (easy) to 37.5 pp (hard) over the monolithic LLM.

Table 4: Physics computation errors (mean ± std). Deterministic interval arithmetic in our framework reduces braking distance error by 13× and TTC error by 10× compared to monolithic LLMs.

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

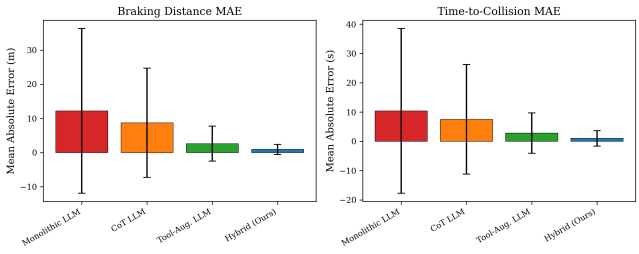

Figure 4: Physics computation errors with standard deviation bars. Left: braking distance MAE. Right: time-to-collision MAE. Our deterministic engine achieves the lowest error and variance. Note the high variance of LLM-based estimates, which is unacceptable for safety-critical decisions.

to 12.2 m for the monolithic LLM—a 13× reduction. For TTC, errors drop from 10.39 s to 1.03 s. The tool-augmented LLM achieves 2.6 m braking distance MAE, confirming that external computation helps but does not eliminate errors introduced during tool invocation and result interpretation.

Figure 4 visualizes these errors. The high variance of monolithic LLM physics estimates (std = 24.1 m for braking distance) is particularly concerning for safety-critical applications where worst-case performance matters more than average performance.

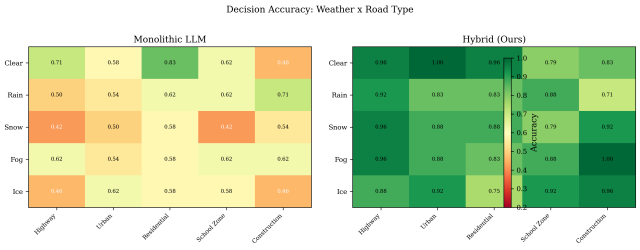

Figure 5: Accuracy heatmap across weather conditions and road types. Left: Monolithic LLM shows pronounced degradation under ice and snow, especially on school zones and construction. Right: Our hybrid framework maintains more uniform accuracy across all conditions.

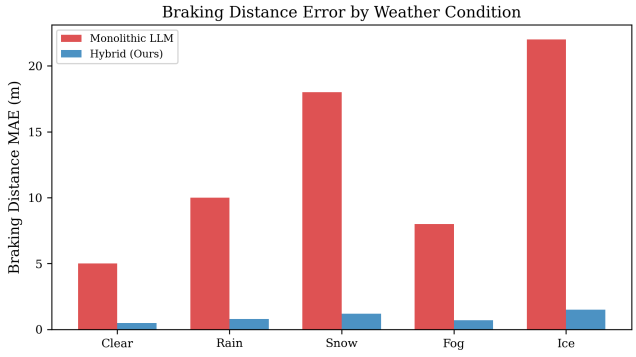

Figure 6: Braking distance error by weather condition. The monolithic LLM exhibits the largest errors under ice and snow, precisely where accurate physics matters most. Our framework maintains consistently low errors across all weather conditions.

3.4 Weather and Road Type Analysis

Figure 5 shows a heatmap of decision accuracy across weather conditions and road types. The monolithic LLM shows pronounced degradation under ice ($\mu \in [0.1, 0.25]$) and snow ($\mu \in [0.2, 0.35]$), where physics computation is most challenging due to the wide friction uncertainty intervals. Our framework maintains more uniform accuracy because 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 the monolithic LLM's braking distance errors are most severe under ice conditions (where friction intervals are widest). Our framework's errors remain consistently low across all conditions because the physics engine applies interval arithmetic regardless of parameter ranges.

3.5 Failure Mode Analysis

We analyze the remaining errors of our framework (11.7% overall error rate). The most common failure modes are: (1) **Parsing ambiguity** (38% of errors): the scenario parser extracts incorrect speed or distance estimates from ambiguous descriptions. (2) **Tight**

margins (31%): the scenario has constraints so tight that small uncertainties in the interval bounds flip the feasibility of the correct maneuver. (3) **Missing policy rules** (21%): the policy database lacks a rule needed for the specific scenario combination. (4) **Confidence calibration** (10%): the correct maneuver is feasible but ranks below another due to confidence scoring.

These failure modes suggest clear improvement paths: better LLM-based parsing with structured output validation, expanded policy databases, and learned confidence calibration from scenario data.

4 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. Evaluation on 600 synthetic scenarios demonstrates a 34.7 percentage-point improvement over monolithic LLMs on hybrid-reasoning tasks, with physics computation errors reduced by 13×.

Our framework has three limitations that suggest future work. First, the scenario parser relies on keyword matching; replacing it with an LLM with constrained decoding would improve robustness to diverse language. Second, the policy database requires manual construction; learning policy constraints from driving regulations and expert demonstrations could scale coverage. Third, our evaluation uses synthetic scenarios; validation on the full AgentDrive benchmark [3] and real-world driving data is needed to confirm generalization.

More broadly, our results suggest that the path to reliable hybrid reasoning in safety-critical domains lies not in larger monolithic models but in architectures that decompose reasoning into specialized modules with verified interfaces. This principle—delegate to the specialist, compose at the boundary—may apply beyond autonomous driving to any domain requiring the fusion of quantitative computation with qualitative rules under uncertainty.

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*.