INFERENCE-TIME GUARDRAILS: REDUCING HALLUCINATIONS VIA STRUCTURED INVARIANTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) and vision-language models (VLMs) often produce outputs that are internally inconsistent: a high-level hypothesis H conflicts with finer-grained fields that should determine it under explicit domain policy. Such failures persist under chain-of-thought prompting and under tool- or solver-mediated scaffolding when the model must itself nondeterministically generate fragile code or specifications at inference time.

We propose *Inference-Time Guardrails* (ITG), a post-inference scaffolding framework that requires no weight updates. Given that LLM/VLM performs a single structured generation that jointly predicts the answer to a high-level hypothesis H and answers A to a fixed set of auxiliary probes, lightweight, human-authored predicates $I(H,A)$ then check consistency, and a deterministic correction $C(H,A)$ repairs violations when possible. This decouples probabilistic extraction from auditable policy enforcement.

We implement ITG across a diverse set of domains. ITG is compared with structured baselines without guardrails, hypothesis-only structured outputs, unconstrained chain-of-thought, and with related work (when applicable) such as program-style (PAL (Gao et al., 2023)) and solver-style (SatLM (Ye et al., 2023)) scaffolding. The evaluation is performed across multiple generalist and specialist LLMs and VLMs. ITG often yields large accuracy gains over these baselines. Moreover, for smaller models, ITG sometimes closes the performance gap with larger models.

1 INTRODUCTION

Generative AI is increasingly deployed in high-stakes environments, from autonomous vehicle perception systems navigating complex traffic (Waymo AI Team, 2026; Seff et al., 2023) to legal assistants reviewing contracts (Narendra et al., 2024; Bloomberg Law, 2026). In these critical applications, the primary goal is not merely fluent text generation but reliable, fact-based decision-making where errors can lead to significant safety or liability consequences. Reliability/liability in these domains is non-negotiable: a missed clause in a contract or a misidentified hazard on the road substantially reduces a system's value regardless of its conversational fluency.

Despite significant advancements in alignment techniques such as Reinforcement Learning from Human Feedback (RLHF) (Bai et al., 2022) and architectural improvements that have reduced the frequency of egregious errors (Zhang et al., 2024), generative models fundamentally remain probabilistic engines. They often function as "System 1" thinkers (Waymo AI Team, 2026), producing intuitive answers based on statistical correlations rather than rigorous logic. Consequently, hallucinations persist, particularly in reasoning gaps where a model might correctly identify specific facts (e.g., "The car is 3 meters away") but fail to reason about them into a logically consistent conclusion (e.g., "It is safe to accelerate").

Current mitigation strategies primarily focus on prompting or retrieval. Techniques like Chain-of-Thought (CoT) prompting encourage models to verbalize reasoning steps in the context window or a latent scratchpad space (Wei et al., 2022), but they increase inference latency and often produce hallucinated reasoning when the logic chain itself is flawed (Zheng et al., 2025). Retrieval-Augmented Generation (RAG) grounds answers in external documents (Lewis et al., 2020), yet models can still misinterpret the retrieved context (Wang et al., 2025a). Self-correction frameworks attempt to have models check

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

their own work (Zhang et al., 2024), but without external ground truth, models often double down on initial errors or lack the specific domain logic required to identify subtle inconsistencies (Wang et al., 2024).

The flaws of self-correction expose the method's failure to properly mitigate hallucinations. However, with grounded logic to guide the self-correction process, we can overcome these limitations. This leads us to our research question:

> **Research question.** In safety/liability-critical domains, can *deterministic* logical checks serve as a post-inference scaffolding method that detects and corrects confident hallucinations that probabilistic or model-generated scaffolding (e.g., free-text CoT, tool-generated code) often fails to eliminate?

To address this question, we propose **Inference-Time Guardrails** (ITG), a scaffolding methodology that enforces logical consistency without model retraining. The model fills a structured schema that includes both a target hypothesis H and answers A to a fixed set of auxiliary probes. We then apply pre-specified, deterministic guardrail predicates $I(H,A)$ that test whether H is consistent with A under explicit domain policy. When a violation is detected, a deterministic correction $C(H,A)$ reconciles H with A. The predicates are written once, audited offline, and reused at inference time, unlike methods where the model synthesizes its own programs or formalisms on every query (Gao et al., 2023; Pan et al., 2023).

This approach decouples *extraction* (what the model reports about the input) from *policy enforcement* (what must follow under human-specified rules). Unlike CoT alone, ITG provides a hard consistency check; unlike RAG alone, it can validate structured interpretations of the input. The procedure of ITG is reusable across domains. The schemas, auxiliary probes, and predicate code are domain-specific by design, which supports auditability in regulated settings.

2 Related Work

Prompting and Internal Reasoning. CoT and its variants improve multi-step reasoning but remain stochastic. In fact, reasoning text can be inconsistent with the final answer (Wei et al., 2022; Zheng et al., 2025). Additionally, latent thinking and/or long CoT models increase the cost of token output and are harder to audit than explicit structured fields.

Program- and Solver-Aided Scaffolding. PAL (Gao et al., 2023) and related "program of thought" approaches output programs for solving problems in a non-deterministic manner, therefore delegating calculation to an interpreter. LOGIC-LM / SatLM-style pipelines (Pan et al., 2023; Ye et al., 2023) translate natural language into formal problems for external solvers. These methods expose powerful machinery but rely on model-generated code or logic at the time of inference, which can introduce syntax errors, specification mistakes, or spurious "proofs." Furthermore, the programs they create are not held to a specific, pre-determined standard, making the actual problem/proof they solve non-deterministic.

ITG instead applies fixed human-authored predicates and corrections and applies them to structured extractions, trading flexibility for deterministic auditability and stable runtime cost. ITG targets policy-governed domains where clearly defined regulations or safety rules exist *a priori*. We leave out open-domain reasoning benchmarks that have no pre-definable symbolic invariants, such as StrategyQA-style questions like "Did Al Capone own an iPhone?". A rule for that kind of question would have to encode the answer directly, which is label leakage and tells us nothing about policy enforcement.

System One Models. A recent paradigm shift in AI deployment is the emergence of "System One" models, such as Jev (TypeSafe AI, 2026), which replace autoregressive token generation with parallel-sampled, schema-constrained probabilistic decisions. Rather than producing free-form natural text, these models ingest unstructured state and return typed values (such as categorical choices, scores, or yes/no probabilities) mapped directly to predefined schemas, similar to the objective of this work. While models such as Jev eliminate hallucinations that lead to malformed responses and string-level parsing errors, they do not inherently guarantee global logical consistency between outputs. Because a System One model evaluates independent fields or auxiliary prompts in parallel, its outputs can still contain inconsistencies. Thus, ITG is complementary to System One models, providing the deterministic validation and correction scaffolding necessary to bridge the gap between type-safe generation and rigorous, policy-compliant logic.

Neuro-Symbolic Frameworks. While the decoupling of probabilistic extraction from deterministic logic is a cornerstone of neuro-symbolic AI (Mao et al., 2019), ITG differs conceptually from

existing frameworks in its enforcement of pre-defined, exhaustive invariants. Frameworks like NeMo Guardrails (Rebedea et al., 2023) typically focus on semantic steering, using canonical forms or vector embeddings to map user intent to safe dialogue flows. These systems often rely on the model to categorize its own input into these forms at runtime, introducing a layer of semantic ambiguity. In contrast, ITG utilizes a fixed, schema-driven approach, where the invariants $I(H,A)$ are hard-coded requirements of the domain policy instead of being synthesized or retrieved based on similarity.

Furthermore, whereas traditional neuro-symbolic pipelines often treat the symbolic solver as a black box for computation (*e.g.*, DeepProbLog (Manhaeve et al., 2018)), ITG functions as a post-hoc scaffolding layer specifically designed for error reconciliation, using the structured relationship between the hypothesis H and answers A to auxiliary probes to apply deterministic corrections $C(H,A)$. This transitions the framework from a simple pass/fail gatekeeper to an active consistency restorer. By embracing this fixed-logic constraint, ITG trades the open-ended flexibility of general neuro-symbolic reasoners for the high-integrity auditability required in regulated, high-stakes environments where the rules of logic are known *a priori* and must be enforced without exception.

Agentic AI Tool-use frameworks, such as ReAct (Yao et al., 2023), interleave generation with actions to dynamically navigate evolving environments. These agentic prompting strategies focus on sequential information gathering and dynamic adaptation through an iterative "act-and-observe" loop. While powerful for open-ended exploration and recursive context decomposition, this paradigm addresses fundamentally different problems than ITG.

Unlike agentic frameworks that rely on continuous, probabilistic model interventions, ITG is designed for tasks governed by complete, deterministic specifications. By operating as a post-inference scaffolding layer, ITG enforces rigid invariants on structured outputs without requiring iterative feedback from the underlying LLM. However, ITG and agentic frameworks are highly complementary. For instance, within a ReAct loop, ITG can be deployed to deterministically authenticate the preconditions of a tool invocation or validate its resulting output before the agent proceeds to its next step.

Uncertainty and Guardrails. Sampling (Zhang et al., 2024) and uncertainty scores (Bakman et al., 2025) help when models are unsure of their mistakes, but models can still be confidently wrong with their reasoning (Mei et al., 2026). ITG targets symbolic inconsistency between structured fields, complementary to calibration-based uncertainty filtering (we compare empirically in Appendix E), as it can capture mistakes that models are confident about. As such, it should be noted that in practice, uncertainty estimation and guardrail frameworks are orthogonal to each other: neither relies on the other for success, but in combination they can make the system more reliable.

3 METHODOLOGY

3.1 THE ITG FRAMEWORK

We define a reasoning task as a tuple (X,H), where X is the input (text and, for VLMs, an image) and H is the target hypothesis. In our experiments H is often boolean or numeric, but in general, it may be a label, count, or structured field.

Let $\mathcal{P}=\{p_1,...,p_n\}$ denote the fixed auxiliary probes, i.e., the schema keys and their natural-language specifications in the prompt. Let A denote the model's answers to those probes (booleans, numbers, short strings, etc.). Both H and A are produced by the model in a single structured generation step, with examples of the prompts in Appendix H.

3.2 ITG PREDICATES AND CORRECTION

We validate outputs with deterministic predicates $I(H,A)\to\{0,1\}$. We refer to these as **invariants** in the sense that they implement fixed domain rules that must hold for an output to be consistent. Thus, they encode policy chosen for the deployment (e.g., safety margins, contract clauses). An output is consistent with the invariant if $I(H,A)=1$. Here, I checks that H follows from A under the policy.

Depending on the result of $I(H,A)$, we apply

$$C(H,A)=\begin{cases} H & \text{if } I(H,A)=1 \\ f_{\text{derive}}(H,A) & \text{if } I(H,A)=0 \end{cases}, \tag{1}$$

where f_{derive} is a deterministic mapping from A to an updated H.

While this work explores the consistency of hypotheses based on extracted answers (*i.e.*, that H follows from A), one may also examine the internal consistency of the auxiliary facts themselves, *e.g.*, $I_{\text{int}}(A) \to \{0,1\}$. An example of internal consistency would be ordering constraints on extracted times or distances. If $I_{\text{int}}(A) = 0$ and the deployment requires caution, the pipeline can instead abstain or flag the instance rather than force a correction, as the internal consistency violation reveals an issue with the extracted answers.

3.3 Design and Engineering Process

In this work, schemas and invariants are authored from dataset semantics and domain heuristics. The difficulty of designing a suitable set of invariants varies by domain complexity. For example, a more complicated domain like road scene analysis requires substantial effort, while a simple, structured domain like bar chart analysis requires less effort. In practice, this effort is a specification cost, analogous to compliance checklists or unit tests, not an end-to-end learned component. Consequently, if the specification changes, then the invariants would also require adaptation. From a human-factor perspective, the design of such specifications should undergo multiple rigorous iterations from a group of domain experts, such as what occurs for international standards groups (International Organization for Standardization and International Electrotechnical Commission, 2024). The goal is to obtain group consensus on proper definitions, which—based on conclusions from prior literature on using domain-experts for labeling efforts (Chang et al., 2017) and group consensus (Dalkey and Helmer, 1963)—should translate to improving the robustness against model hallucinations using ITG.

Without these design principles, if the responsibility of creating a robust set of rules is left to a single domain expert, then ITG may result in breaking more cases than it corrects. While the case studies examined in this paper use tasks simple enough to implement without needing such rigorous practices, real-world usage of ITG would encounter cases with increased complexity.gio Furthermore, to reduce bias, we design the invariant rules on the training data and freeze the rules before evaluation on held-out test data.

4 Experimental Setup

4.1 Datasets and Evaluation Protocol

We experimentally evaluate ITG on a diverse set of four datasets. We select domains where generalist LLMs/VLMs show mixed reliability. Pre-processing details appear in each case study's section. Our evaluation uses the following datasets:

ContractNLI (Koreeda and Manning, 2021). Non-Disclosure Agreement review. Using the test split, we evaluate five ContractNLI hypotheses (*NDA-5, 7, 16, 19, 20*) for which we could specify auditable predicates with our level of legal expertise. Therefore, we do not claim coverage of all 17 labels present in the dataset. We only consider samples where the target hypothesis for all five hypotheses is not "Contradiction". Following this selection process, we have 123 samples.

nuScenes (Caesar et al., 2020). Autonomous driving scene analysis. We use the train split since labels required for the evaluation are not included in the testing dataset. We derive the labels ourselves from geometry, velocity, and nuScenes attributes, thus they are not present in VLM training data. Moreover, we note that these labels are not used to inform the design of the auxiliary probes used by ITG. We select one frame per scene for visual diversity, amounting to 850 samples.

GSM8K (Cobbe et al., 2021). Grade school math word problems. Using the test set, we extract the numeric answer from the solution field and evaluate. There are 1,319 samples.

FigureQA (Kahou et al., 2017). Synthetic chart reasoning. For our evaluation, we use a balanced subset of bar-chart items from the test set. We sample 120 charts, targeting approximately a 50/50 split of True/False labels per hypothesis.

The results for ContractNLI and FigureQA can be found in Section 5 and Section 6, and, due to space limits and because the conclusions are similar, the results for nuScenes and GSM8K are in Appendix B and Appendix A, respectively. Furthermore, as stated in Section 3.3, the auxiliary probes for each

domain were determined using the training split for each dataset; however, in the case of nuScenes where we used an augmentation of the training dataset, we held out 50 random samples for the design of the auxiliary probes.

4.2 Implementation

We use Pydantic (Colvin et al., 2025) schemas for structured outputs. Predicates are deterministic Python functions over the parsed JSON. If parsing fails schema validation, we re-prompt up to a maximum of 20 retries before being considered a failure. In evaluation, failures are counted as incorrect. Pseudo-code for the implementation is in Appendix C. Source code is available at *[withheld for review, included in supplementary materials]*.

4.3 Baselines and Models

We evaluate generalist models (Llama-3.1 (Grattafiori et al., 2024), Llama-4-scout (Meta, 2025), Gemma-3 (Team et al., 2025), GPT-4o (Hurst et al., 2024), gpt-oss-120b (Agarwal et al., 2025), Qwen3 (Yang et al., 2025)) and, where noted, domain-focused models (e.g., DeepSeek-Coder-V2-Lite (Zhu et al., 2024)). Vision tasks use VLM variants. We compare:

1. **Base:** Structured JSON for H only (no auxiliary probes in the schema).
2. **Base+CoT:** Unconstrained natural-language reasoning with a final answer in delimiters without A.
3. **Base+A:** Same structured JSON call as ITG for H and A, but no deterministic correction, isolating structured extraction without guardrails.
4. **ITG (our method):** Structured H and A with $I(H,A)$ and $C(H,A)$.

We provide bar charts in each case study to summarize the class-balanced accuracy across all H. We additionally report the performance of PAL (Gao et al., 2023) on FigureQA and GSM8K and SatLM (Ye et al., 2023) on ContractNLI and GSM8K as a comparison with the related work. Fine-grained, per-hypothesis tables of results (in Appendix F) report on each of these cases with variance over $n{=}10$ runs. All experiments utilized top-p of 0.9, a temperature of 0.2, and model default top-k settings.

4.4 Recorded Metrics

Accuracy. Unless noted otherwise, accuracy is the fraction of examples for which the model's final hypothesis H matches the dataset label. For GSM8K (numeric exact match), macro bar charts report Acc since the answer is numeric and not boolean.

Class-Balanced Accuracy (Bal-Acc). For binary hypotheses we report the bal-acc, which is the recall of that average of per-class recalls. The datasets used in our experiments can experience substantial class imbalance, so presenting bal-acc mitigates bias in the measurement of success.

Token counts. We log token usage and plot median output tokens per model.

False-correction rate (FCR). FCR measures how often ITG changes a correct prediction into an incorrect one, reported as a percentage:

$$\mathrm{FCR}(\%) = 100 \times \frac{\#\{\text{Pre-ITG correct} \wedge \text{Post-ITG incorrect}\}}{\text{Total samples}}. \tag{2}$$

Fix-to-break ratio (FBR). The FBR compares helpful corrections to harmful ones:

$$\mathrm{FBR} = \frac{\#\{\text{Pre-ITG wrong} \to \text{Post-ITG correct}\}}{\#\{\text{Pre-ITG correct} \to \text{Post-ITG wrong}\}}. \tag{3}$$

5 Case Study 1: ContractNLI

Legal contract review often hinges on detecting permissions that are implied by definitions or liability structures rather than explicitly stated. Our analysis reveals that older, smaller language models have a severe reasoning gap in this domain. While they can often extract the relevant text chunks, they fail to capture the logical implications. For example, a model might correctly identify a clause defining

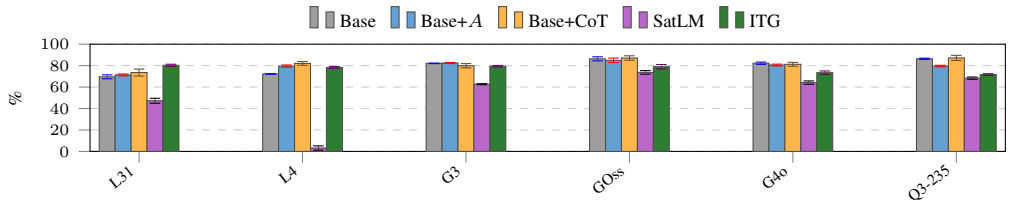

Figure 1: ContractNLI: mean bal-acc (%) over the evaluated NDA hypotheses. Bars compare structured base decoding, Base+A, Base+CoT, a logic-solver baseline (SatLM), and ITG. Error bars show run-to-run variability.

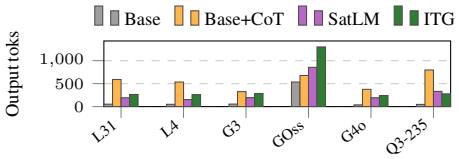

Figure 2: Median model-generated output tokens per example.

Model	NDA-5	NDA-7	NDA-16	NDA-19	NDA-20	Overall	FBR
L31	4.07	0.81	7.40	2.44	0.81	3.11	5.30
L4	8.78	5.69	5.12	0.89	0.00	4.10	2.94
G3	6.75	4.07	5.53	2.44	1.63	4.08	2.26
GOss	8.78	5.61	4.80	1.95	0.00	4.23	1.88
G4o	7.97	4.15	8.13	1.71	0.00	4.39	1.67
Q3-235	8.05	4.55	6.50	1.95	0.81	4.37	1.44

Figure 3: FCR (%) by NDA hypothesis and overall, with FBR in the last column. Cell heatmap shading highlights higher FCR (scaled from 0 to table's maximum).

"representatives" to include consultants, yet simultaneously predict that sharing with third-party consultants is forbidden. This results in many false negatives for sharing permissibility labels, as models default to forbidding sharing in the absence of a direct, affirmative statement.

5.1 Dataset Processing

For ContractNLI, we adapt the task for binary consistency checking on a subset of five hypotheses (*NDA-5, 7, 16, 19, 20*) chosen for auditable predicate design given our legal-domain expertise.

5.2 Invariants Applied

The model fills boolean fields and quote snippets in one JSON object. Below, following from the fixed auxiliary probe set $\mathcal{P}$ as input to each LLM, b_i are the boolean-valued auxiliary answers in A, Q_i are the extracted text snippets from the contract text in A, and $H = \{H_{\text{emp}}, H_{\text{3rd}}, H_{\text{ret}}, H_{\text{surv}}, H_{\text{keep}}\}$ are the hypotheses corresponding to NDA-5, NDA-7, NDA-16, NDA-19, and NDA-20.

- **Hierarchy of Trust:** Sharing with third parties implies permission to share with employees:
$$H_{\text{3rd}} = \text{True} \implies H_{\text{emp}} = \text{True}. \tag{4}$$
- **Implied Permissions:** We extract auxiliary facts about specific roles which imply a valid 3rd party:
$$b_{\text{attorney}} \vee b_{\text{whistleblower}} \vee b_{\text{advisor}} \vee b_{\text{subcontractor}} \vee b_{\text{vendors}} \vee b_{\text{exceptions confidential}} \implies H_{\text{3rd}} = \text{True} \tag{5}$$
$$b_{\text{employees}} \vee b_{\text{need to know}} \vee b_{\text{liable for employees}} \implies H_{\text{emp}} = \text{True} \tag{6}$$
- **Definition Grounding:** We extract the definition of access groups, Q_{def}, and apply string matching. If the extracted quote explicitly lists a role (e.g., "consultant") for a specific access level (employee or 3rd party), the permission is forced to True:
$$\{\text{"consultant"},\text{"advisor"},\text{"agent"},\text{"contractor"},\text{"accountant"},\ldots\} \in Q_{\text{def}} \implies H_{\text{3rd}} = \text{True} \tag{7}$$
$$\{\text{"employee"},\text{"officer"},\text{"director"},\text{"partner"},\text{"personnel"}\} \in Q_{\text{def}} \implies H_{\text{emp}} = \text{True} \tag{8}$$
- **Obligation Strictness:** A return or destroy obligation (H_{ret}) must be supported by a mandatory quote. If the quote Q_{ret} implies the action is conditional or permissive (e.g., "may") we falsify the hypothesis.
$$\{\text{"may"},\text{"can"},\text{"option"},\text{"discretion"}\} \in Q_{\text{ret}} \implies H_{\text{ret}} = \text{False} \tag{9}$$
- **Retention Dependency Logic:** If retention of data (H_{keep}) is permitted (e.g., for b_{backup} or b_{counsel}), the confidentiality obligation must survive (H_{surv}) to protect that data:
$$b_{\text{backup}} \vee b_{\text{counsel copy}} \implies H_{keep} = \text{True}, \tag{10}$$
$$H_{\text{keep}} \vee b_{\text{indefinite}} \implies H_{\text{surv}} = \text{True}. \tag{11}$$

5.3 RESULTS

Figure 1 summarizes overall Bal-Acc following the experimental methodology described in Section 4.

Models Evaluated. We evaluated on Llama-3.1-8B (L31) (Grattafiori et al., 2024), Llama-4-scout (L4) (Meta, 2025), Gemma-3-27b (G3) (Team et al., 2025), GPT-oss-120b (GOss) (Agarwal et al., 2025), GPT-4o (G4o) (Hurst et al., 2024), and Qwen3-235B-Instruct-FP8 (Q3-235) (Yang et al., 2025). For ContractNLI, we attempted to evaluate on legal-domain specialist models, including SaulLM-7B (Colombo et al., 2024a) and SaulLM-54B (Colombo et al., 2024b), but they were not able to reliably follow the strict JSON schema of ITG.

Baseline Comparison. On the test set, Bal-Acc gains from ITG are concentrated in weaker models with low baseline Bal-Acc, e.g., Llama-3.1-8B rises from 69.6 to 80.3 from ITG. On the other hand, Llama-4-scout with ITG improves over Base ($72.3 \rightarrow 78.3$), a smaller gain.

The improvement from ITG is largely dependent on H. In the case of **NDA-16**, ITG improves Bal-Acc over both Base for all 6 models ($+1.7$ to $+8.1$). For **NDA-7**, models with a weak baseline show improvements (L31 $+20.5$, L4 $+18.1$, G3 $+1.4$), while the stronger models do not. In the case of **NDA-19** and **NDA-20** most models already have a high bal-acc, so ITG gains remain small since there is not room for much opportunity. In contrast, SatLM remains well below the baselines on Bal-Acc

Base+CoT Comparison. Base+CoT is the strongest macro baseline for five of six models (all except L31, where ITG leads). Its drawback is token cost: ITG uses far fewer output tokens (Figure 2) while remaining competitive when Base has headroom. Deployment trade-offs therefore combine accuracy with predictable, auditable structure and generation cost, not accuracy alone. The exception is GOss, which has very high token utilization compared to the other models. Unlike the other models, GOss uses a latent-space reasoning scratchpad, which makes it much "chattier." The large auxiliary probe set in ITG appears to amplify that behavior.

FCR. Figure 3 overviews the FCRs and domain-wide FBRs. In the table, FBR is > 1 for every model, meaning that helpful corrections outnumber false corrections. Together with ITG's deterministic, predicate-governed updates on extracted JSON fields, this supports the claim that ITG mitigates the impact of hallucination when the baseline is weak.

6 CASE STUDY 2: FIGUREQA

VLMs often struggle with precise arithmetic reasoning on visual data. While they can easily identify visual extrema such as the Min or Max, they fail significantly at aggregation tasks like calculating the Median, which requires sorting and indexing a set of values. Furthermore, models often hallucinate relationships (e.g., claiming

Bar X $>$ Bar Y) when the visual height difference is subtle.

6.1 DATASET PROCESSING

We evaluate on a balanced subset of the official FigureQA test split[1] for bar chart data. FigureQA provides ground-truth annotations for chart elements (bar heights, labels, colors). We use a VLM to parse them to obtain reference minima, maxima, medians, and pairwise bar relations. Following the FigureQA definition for even-sized sets, low median and high median refer to the lower and upper middle order statistics (not the typically used arithmetic mean of the two middle data).

6.2 INVARIANTS APPLIED

The model extracts numerical values and labels and includes them in one JSON object. Below, following from the fixed auxiliary probe set $\mathcal{P}$ as input to each VLM, d_i are the numeric auxiliary answers in A (bar heights and other numeric auxiliary facts) and $H = \{H_{\text{min}}, H_{\text{max}}, H_{\text{LoMed}}, H_{\text{HiMed}}, H_{>}, H_{<}\}$ are the hypotheses corresponding to each label in FigureQA. The hypotheses are questions generally ask whether specific bars in the chart are the min, max, low/high median, or greater/less than another bar.

[1]The original 2017 FigureQA dataset was not released with labels for the test split, but a 2025 release does include these labels: `https://github.com/microsoft/FigureQAComplete`

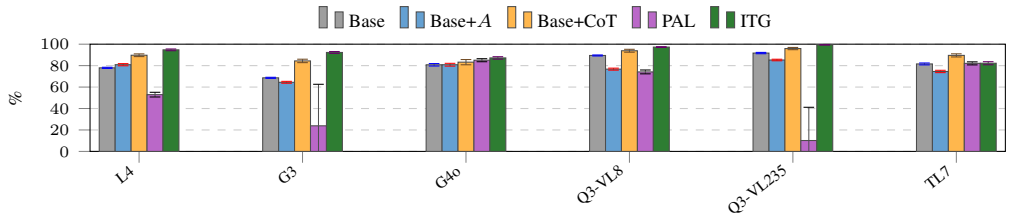

Figure 4: FigureQA: mean balanced accuracy (%) over chart-reading hypotheses for structured base decoding, Base+A, Base+CoT, a program-execution baseline (PAL), and ITG. Error bars show run-to-run variability.

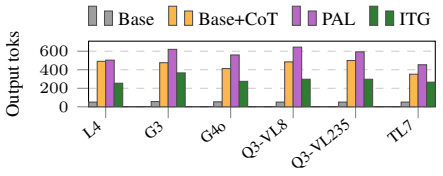

Figure 5: Median model-generated output tokens per example.

Model	Min	Max	LoMed	HiMed	<	>	Overall	FBR
L4	0.83	4.33	3.42	0.42	0.08	0.00	1.51	10.15
G3	3.75	2.08	3.33	2.58	2.42	1.08	2.54	11.95
G4o	2.83	2.33	9.83	8.42	0.00	0.08	3.92	2.62
Q3-VL8	0.83	2.58	0.00	0.83	1.17	2.83	1.38	15.97
Q3-VL235	0.00	0.00	0.00	0.00	1.92	0.00	0.32	45.30
TL7	2.50	6.83	0.67	0.25	11.00	9.75	5.17	2.48

Figure 6: FCR (%) by chart-reasoning hypothesis and overall, with FBR in the last column. Cell heatmap shading highlights higher FCR (scaled from 0 to table's maximum).

- **Verify Extracted Data Against Function:** Given f is one of $\min$, $\max$, med_{hi}, or med_{lo}, and H is respectively $H_{\min}$, $H_{\max}$, H_{LoMed}, or H_{HiMed}. The bar X is given in the question, asking whether it is the min, max, or low/high median.

$$d_{\text{bar X}} = f(d_{\text{bar 1}}, \ldots, d_{\text{bar n}}) \implies H = \text{True} \tag{12}$$

$$d_{\text{bar X}} \neq f(d_{\text{bar 1}}, \ldots, d_{\text{bar n}}) \implies H = \text{False} \tag{13}$$

- **Verify Extracted Data Against Itself:** The following is for checking the extracted data for checking if one bar's value is greater/less than another bar's value. The bar X is given in the question, asking whether it is larger/smaller than bar Y, which also given in the question.

$$d_{\text{bar X}} > d_{\text{bar Y}} \implies H_{>} = \text{True} \tag{14}$$

$$d_{\text{bar X}} < d_{\text{bar Y}} \implies H_{<} = \text{True} \tag{15}$$

6.3 Results

Figure 4 summarizes macro-averaged Bal-Acc following the experimental methodology.

Models Evaluated. We evaluated on Llama-4-scout (L4) (Meta, 2025), Gemma-3-27b (G3) (Team et al., 2025), GPT-4o (G4o) (Hurst et al., 2024), Qwen3-VL-8B-Instruct (Q3-VL8) (Bai et al., 2025), and Qwen3-VL-235B-Instruct (Q3-VL235) (Bai et al., 2025). We additionally evaluated on the domain-specific model ThinkLite-VL-7B (TL7) (Wang et al., 2025b).

Macro Bal-Acc. Relative to Base and Base+A, ITG raises macro Bal-Acc for all 6 models (Figure 4). Gains are large for Llama-4-Scout and Gemma-3-27B and still clear for GPT-4o and both Qwen3-VL models. ThinkLite-VL-7B is the main exception to the broader pattern, where ITG barely exceeds Base ($+0.7$) and comes short compared with Base+CoT. ITG also beats the PAL program-execution baseline for every model, and PAL is performs especially poor for Gemma-3-27B and Qwen3-VL-235B.

Per-hypothesis Bal-Acc. On Q1 (`is minimum?`), ITG improves over Base for all 6 models and over Base+A for 5/6. On Q2 (`is maximum?`), where Base is already near 100% for several models, ITG improves over Base for only 2/6 and often slightly reduces Bal-Acc when corrections overwrite a correct H (e.g., ThinkLite-VL-7B -7.7). Comparison questions (Q5/Q6) have mixed gains, but ITG is able to help on Q5/Q6 for Qwen3-VL-8B and Qwen3-VL-235B on Q6. For median questions (Q3/Q4), ITG has the clearest benefit. Llama-4-Scout and Gemma-3-27B are $\sim$48–57% Bal-Acc as Base but reach the high 80s–90% under ITG. GPT-4o and both Qwen3-VL models also see double-digit improvements.

Domain	ITG (ms)	SatLM (ms)	PAL (ms)
ContractNLI	<0.001	4.8	—
GSM8K	<0.001	5.1	15
NuScenes VLM	<0.001	—	—
FigureQA	<0.001	—	24

Table 1: Median runtime for each post-processing framework for each domain. SatLM and PAL report solver and program-execution overhead where that scaffolding is used.

7 Discussion

Applying ITG to Other Domains. Across four domains, ITG improves reliability when structured extraction is strong enough to support deterministic policy checks, and when the base model is not too close to 100%. In practice, deployment of ITG depends on the application and the ability of a model without post-inference checks. Thus, users of ITG need to make an informed decision, fixing their compute requirements, ablate candidate models on the target hypotheses, and apply ITG only where there is clear indications it will improve performance. We discuss the design and engineering process, which can be applied to other domains outside of those explored in this work, in Section 3.3. Additionally, since the design and selection of A is important to understanding the general application of ITG, we discuss the improvements from probe richness and asking more questions in Appendix D..

ITG Token Use versus Base+CoT. Structured ITG emits compact JSON fields, while CoT outputs variable and chatty responses. The per-domain token bars (*c.f.,* Figures 2, 5, 8 and 11) summarize median output lengths for each baseline. Note that, Base+A and ITG share the same generated tokens before correction. The general observation across most domains is that Base+CoT utilizes many more tokens than ITG. However, when too many auxiliary probes are asked, the token usage of ITG can exceed that of even Base+CoT. There is an inherent trade-off depending on the domain, the specific hypotheses answered, and the verbosity required from the auxiliary probes.

Computation Overhead. The computation overhead of ITG is negligible. As summarized in Table 1, the latency overhead is usually on the order of microseconds. Compared with other scaffolding frameworks, such as SatLM and PAL, ITG is several orders of magnitude more efficient. SatLM and PAL take several milliseconds by comparison.

Dependence on Extraction of A. If auxiliary answers in A are wrong, corrections may cause additional errors. When evaluating ITG on a novel domain, the FBR must be closely monitored. If it is <1, then ITG is causing more problems than it is helpful. When it is >1, then it becomes important to understand the impact of the cases which cause it to break to assess whether the positive FBR is actually useful.

There are a few ways to detect when the extraction of answers A to auxiliary probes fails and/or is unreliable. First, if the performance of ITG does not change compared to Base+A, then the auxiliary probes are not eliciting anything meaningful from the model. This is further corroborated by a substantial drop in performance compared to just Base. Second, if the FBR is <1, then the auxiliary answers should be individually investigated to understand which is causing the false corrections. Third, before designing ITG for a domain, it should be decided on an FCR that aligns with acceptable levels of failure for the domain. The level of acceptability may vary from hypothesis to hypothesis within a given domain.

Human Specification Cost. Results depend on auxiliary probe and invariant quality. Since these are designed by domain experts, there is an inherent cost that comes from creating a specification that ITG must follow. In practice, complex domains will require more iteration for several experts to arrive at a consensus. This mirrors the prompt- and template-engineering cost of other scaffolding methods, but substitutes auditable code for opaque chain text. While a full human-factors study is outside our scope, one can follow the procedures used for the design of international standards, which utilize the Delphi method (Dalkey and Helmer, 1963).

Policy vs. label-generation. A fair worry is that our invariants simply encode each benchmark's labels. Across domains, overlap between the label generation policy and the auxiliary probes is possible (e.g., the human labeler looked for similar keywords when deciding labels for ContractNLI). When it occurs, it may mean the labeling policy follows the same policy encoded in the auxiliary probes, and in structured environments this kind of overlap is an intended consequence of encoding expert decision rules.

AI USE STATEMENT

In this work, we used generative AI tools for providing initial drafts of a paper outline and revising writing. We additionally used it for launching experiments and basic data processing. We have not used generative AI tools for writing the main technical components of ITG or idea formulation. Use of generative AI for mathematical claims or synthetic dataset generation are not applicable to this work. We have reviewed all AI-assisted work, manually investigating any related work it brought up during paper editing, and verifying LLM-generated experiment launching and data processing. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

As stated in Section 4.2, source code is available at *[withheld for review, included in supplementary materials]*. Details are included for rerunning the experiments and utilizing the datasets in this work. All models used are included in the experimental sections, but the exact configurations used are also included in the source code. Scripts for processing the datasets used in this work are included in the source code as well.

REFERENCES

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. PAL: Program-aided language models. In *International conference on machine learning*, pages 10764–10799, 2023.

Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. SatLM: Satisfiability-aided language models using declarative prompting. *Advances in Neural Information Processing Systems*, 36:45548–45580, 2023.

Waymo AI Team. Demonstrably safe ai for autonomous driving. `https://waymo.com/blog/2025/12/demonstrably-safe-ai-for-autonomous-driving`, 2026. [Accessed: 2026-01-27].

Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S. Refaat, Rami Al-Rfou, and Benjamin Sapp. MotionLM: Multi-agent motion forecasting as language modeling. *arXiv preprint arXiv:2309.16534*, 2023.

Savinay Narendra, Kaushal Shetty, and Adwait Ratnaparkhi. Enhancing contract negotiations with LLM-based legal document comparison. In *Proceedings of the Natural Legal Language Processing Workshop 2024*, pages 143–153, November 2024.

Bloomberg Law. Bloomberg law platform: Ai-driven legal research and tools. `https://pro.bloomberglaw.com/products/ai-and-bloomberg-law`, 2026. [Accessed: 2026-01-27].

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Yunxiang Zhang, Muhammad Khalifa, Lajanugen Logeswaran, Jaekyeom Kim, Moontae Lee, Honglak Lee, and Lu Wang. Small language models need strong verifiers to self-correct reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15637–15653, August 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

Tianshi Zheng, Yixiang Chen, Chengxi Li, Chunyang Li, Qing Zong, Haochen Shi, Baixuan Xu, Yangqiu Song, Ginny Y Wong, and Simon See. The curse of CoT: On the limitations of chain-of-thought in in-context learning. *arXiv preprint arXiv:2504.05081*, 2025.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33: 9459–9474, 2020.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Retrieval-augmented generation with conflicting evidence. *arXiv preprint arXiv:2504.13079*, 2025a.

Wenxuan Wang, Juluan Shi, Zhaopeng Tu, Youliang Yuan, Jen-tse Huang, Wenxiang Jiao, and Michael R Lyu. The earth is flat? unveiling factual errors in large language models. *arXiv preprint arXiv:2401.00761*, 2024.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. LOGIC-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824, 2023.

TypeSafe AI. Introducing system one models & jev. https://typesafe.ai/blog/introducing-system-one-models-and-jev, 2026. URL `https://typesafe.ai/blog/introducing-system-one-models-and-jev`.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision. In *International Conference on Learning Representations*, 2019.

Traian Rebedea, Razvan Dinu, Makesh Narsimhan Sreedhar, Christopher Parisien, and Jonathan Cohen. Nemo guardrails: A toolkit for controllable and safe llm applications with programmable rails. In *Proceedings of the 2023 conference on empirical methods in natural language processing: system demonstrations*, pages 431–445, 2023.

Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Deepproblog: Neural probabilistic logic programming. *Advances in neural information processing systems*, 31, 2018.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Yavuz Faruk Bakman, Duygu Nur Yaldiz, Sungmin Kang, Tuo Zhang, Baturalp Buyukates, Salman Avestimehr, and Sai Praneeth Karimireddy. Reconsidering llm uncertainty estimation methods in the wild. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29531–29556, 2025.

Zhiting Mei, Christina Zhang, Tenny Yin, Justin Lidard, Ola Sho, and Anirudha Majumdar. Reasoning about uncertainty: Do reasoning models know when they don't know? In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 3408–3458, 2026.

International Organization for Standardization and International Electrotechnical Commission. Iso/iec directives, part 1: Procedures for the technical work. Technical report, ISO/IEC, Geneva, Switzerland, 2024. URL `https://www.iso.org/directives-and-policies.html`.

Joseph Chee Chang, Saleema Amershi, and Ece Kamar. Revolt: Collaborative crowdsourcing for labeling machine learning datasets. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pages 2334–2346, 2017.

Norman Dalkey and Olaf Helmer. An experimental application of the delphi method to the use of experts. *Management science*, 9(3):458–467, 1963.

Yuta Koreeda and Christopher Manning. ContractNLI: A dataset for document-level natural language inference for contracts. In *Findings of the Association for Computational Linguistics (EMNLP '21)*, pages 1907–1919, November 2021.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR '20)*, June 2020.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. FigureQA: An annotated figure dataset for visual reasoning. *arXiv preprint arXiv:1710.07300*, 2017.

Samuel Colvin et al. Pydantic, 2025. URL `https://docs.pydantic.dev/`.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Meta. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation. `https://ai.meta.com/blog/llama-4-multimodal-intelligence/`, 2025. [Accessed: 2026-01-27].

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Qihao Zhu, Daya Guo, Zhihong Shao, Dejian Yang, Peiyi Wang, Runxin Xu, Y Wu, Yukun Li, Huazuo Gao, Shirong Ma, et al. Deepseek-coder-v2: Breaking the barrier of closed-source models in code intelligence. *arXiv preprint arXiv:2406.11931*, 2024.

Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, Andre FT Martins, Fabrizio Esposito, Vera Lúcia Raposo, Sofia Morgado, et al. SaulLM-7B: A pioneering large language model for law. *arXiv preprint arXiv:2403.03883*, 2024a.

Pierre Colombo, Telmo Pires, Malik Boudiaf, Rui Melo, Dominic Culver, Etienne Malaboeuf, Gabriel Hautreux, Johanne Charpentier, and Michael Desa. SaulLM-54B & SaulLM-141B: Scaling up domain adaptation for the legal domain. *Advances in Neural Information Processing Systems*, 37: 129672–129695, 2024b.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Xiyao Wang, Zhengyuan Yang, Chao Feng, Hongjin Lu, Linjie Li, Chung-Ching Lin, Kevin Lin, Furong Huang, and Lijuan Wang. Sota with less: Mcts-guided sample selection for data-efficient visual reasoning self-improvement. *arXiv preprint arXiv:2504.07934*, 2025b.

Ayesha Ishaq, Jean Lahoud, Ketan More, Omkar Thawakar, Ritesh Thawkar, Dinura Dissanayake, Noor Ahsan, Yuhao Li, Fahad Shahbaz Khan, Hisham Cholakkal, et al. Drivelmm-o1: A step-by-step reasoning dataset and large multimodal model for driving scenario understanding. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 20501–20508. IEEE, 2025.

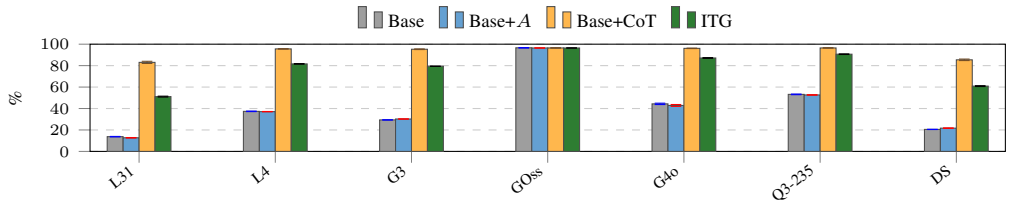

Figure 7: GSM8K: test-set accuracy (%) for structured base decoding, Base+A, Base+CoT, and ITG. Error bars show run-to-run variability.

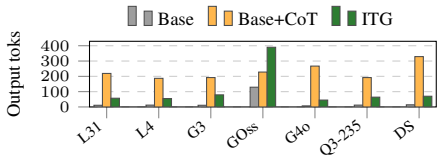

Figure 8: Median model-generated output tokens per example.

Model	Final Answer	FBR
L31	2.62	15.79
L4	2.21	21.32
G3	2.26	22.79
GOss	0.00	N/A
G4o	1.69	27.36
Q3-235	1.68	23.75
DS	3.84	11.25

Figure 9: FCR (%) for the final numeric answer, with FBR in the last column. Cell heatmap shading highlights higher FCR (scaled from 0 to table's maximum).

A Case Study 3: GSM8K

Language models often struggle at reliable arithmetic: they may outline correct steps yet produce a final number that does not follow from those steps. The GSM8K dataset prompts the model with a word problem and then requests a final numeric answer. For our evaluation, the model's proposed final numeric answer is the hypothesis, $H = H_{\text{ans}}$.

A.1 Dataset Processing

For the GSM8K dataset, the ground-truth answers are included with reasoning logic. Since we only wish to evaluate the correctness of the final answer, we parsed the answers by extracting the numeric value following the '####' delimiter in the dataset's solution field.

A.2 Invariants Applied

The model fills a boolean field and arithmetic string in one JSON object. Below, following from the auxiliary probes, $A = \{b_{\text{sign}}, a_{\text{expr}}\}$, where b_{sign} is the boolean-valued auxiliary fact extracting whether the answer should be positive based on the context and a_{expr} is a single safe arithmetic expression parsed from the word problem. $H = H_{\text{ans}}$ is the hypothesis corresponding to model's proposed final numeric answer.

- **Calculator Check:** Let $v = \text{eval}(a_{\text{expr}})$, evaluated as a Python expression. If $|v - H_{\text{ans}}| < \epsilon$, the pair is consistent; otherwise, we set $H_{\text{ans}} = v$. We set $\epsilon = 0.01$ for our experiments.

$$|v - H_{\text{ans}}| \geq 0.01 \implies H_{\text{ans}} = v. \tag{16}$$

- **Sign:** For quantities that must be non-negative (e.g., counts), if $H_{\text{ans}} < 0$, we correct to $|H_{\text{ans}}|$.

$$b_{\text{sign}} \implies H_{\text{ans}} = |H_{\text{ans}}|. \tag{17}$$

A.3 Results

Figure 7 summarizes macro-averaged accuracy following the experimental methodology.

Models Evaluated. We evaluated on Llama-3.1-8B (L31) (Grattafiori et al., 2024), Llama-4-scout (L4) (Meta, 2025), Gemma-3-27b (G3) (Team et al., 2025), GPT-oss-120b (GOss) (Agarwal et al., 2025), GPT-4o (G4o) (Hurst et al., 2024), and Qwen3-235B-Instruct-FP8 (Q3-235) (Yang et al., 2025).

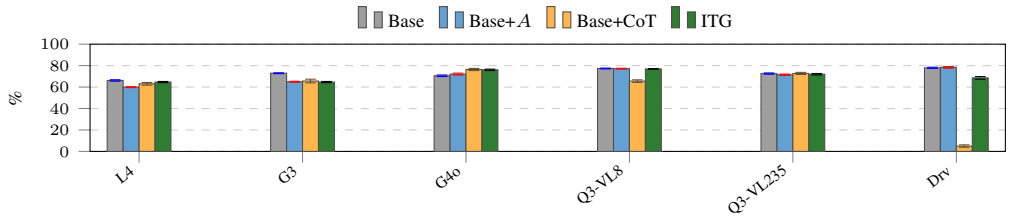

Figure 10: NuScenes: mean balanced accuracy (%) over the two evaluated driving hypotheses. Bars compare structured base decoding, Base+A, Base+CoT, and ITG. Error bars show run-to-run variability.

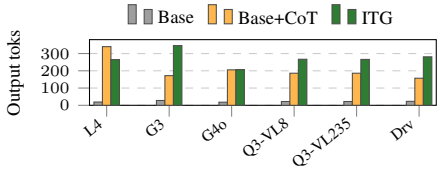

Figure 11: Median model-generated output tokens per example.

Model	Safe accel.	Lead stopped	Overall	FBR
L4	2.20	0.00	1.10	9.64
G3	8.18	1.09	4.63	0.59
G4o	0.29	1.76	1.02	18.26
Q3-VL8	4.14	0.76	2.45	2.91
Q3-VL235	1.49	1.30	1.39	5.81
Drv	1.90	5.33	3.61	5.02

Figure 12: FCR (%) by driving hypothesis and overall, with FBR in the last column. Cell heatmap shading highlights higher FCR (scaled from 0 to table's maximum).

We additionally evaluated on the domain-specific model DeepSeek-Coder-V2-Lite (DS) (Zhu et al., 2024).

Impact of Symbolic Execution. The results show large gains across almost all models relative to Base, with Gemma-3-27b seeing a nearly +50 point increase. This confirms that many models struggle to perform arithmetic internally while remaining highly capable of formulating the correct logic to solve the problem. Offloading the computation to the ITG layer unlocks that reasoning potential without retraining.

CoT Performance vs. Token Utilization While Base+CoT demonstrates the best performance on each model, there is a tradeoff between token utilization and accuracy. ITG is capable of substantially boosting the accuracy of the task and in some cases closely competing with Base+CoT while utilizing a fraction of the tokens.

Token Utilization of GPT-OSS As seen in Figure 8, there is a notable increase in token utilization for the GPT-oss model experiments. This is because this model has a "thinking" mode, where it does latent space reasoning and meticulously thinking through its answer for each auxiliary probe before providing its final output. Unfortunately, the thinking mode of GPT-OSS cannot be officially[2] disabled, so the token savings of ITG are completely eliminated for such models which have mandatory latent space reasoning.

B Case Study 4: nuScenes

VLMs often hallucinate attributes about dynamic objects in a monocular ego camera, failing to account for objects that are physically close but appear as the background to the model, or conversely, flagging safe objects in adjacent lanes as hazards. Moreover, due to the monocular lens, determining the distance of an object can be difficult without considering further context (e.g., comparing variable size objects against objects with a universal standard size). Models often default to an overly conservative stance, flagging navigable scenarios as unsafe due to the mere presence of traffic, even when the context of the traffic would indicate otherwise.

[2]There is an unofficial way to disable thinking mode: `https://huggingface.co/openai/gpt-oss-20b/discussions/86`

B.1 Dataset Processing

We use nuScenes v1.0 with train split frames, as the public test set lacks the 3D annotations needed to derive our semantic labels. From the train split, we algorithmically label frames based on heuristics, and provide ground truth for our hypothesis labels. The hypotheses we provide labels assert whether it is safe to accelerate and whether the lead vehicle (if present) is stopped. From each video available in the dataset, we sample a single keyframe. We generate the label for "safe to accelerate" via a geometric corridor in the ego lane with a ± 1.5 m lateral width, where it is unsafe if a pedestrian is within 15 m, a vehicle within 10 m, or construction obstacles within 12 m. We generate the label for "lead vehicle stopped" based on the nearest in-lane object's velocity and nuScenes attributes. Because these labels are generated ourselves, there is no label leakage in the evaluated models' training. They are also built from geometry, velocity, and nuScenes attributes, not from the visual auxiliary probes we later refined for ITG (brake lights, impede, ego-lane, and related ones).

B.2 Invariants Applied

We define the set of hypotheses $H = \{H_{\text{safe acc}}, H_{\text{lead stopped}}\}$ corresponding to the labels we derived for "safe to accelerate" and "lead vehicle stopped". Using the auxiliary probe set $\mathcal{P}$ as input to each LLM, we extract answers A as auxiliary boolean facts b_i and distance values d_i.

- **Nearest Object Safety:** Checking the location ($a_{\text{near front}} \in A$) and distance ($d_{\text{nearest}} \in D$) of the nearest vehicle in order to determine whether it breaks our assumptions about a vehicle that is unsafe to accelerate behind:
$$d_{\text{nearest}} < 8.0 \wedge a_{\text{near front}} \implies H_{\text{safe acc}} = \text{False} \tag{18}$$
or if there is a pedestrian (d_{ped}) or road cone (d_{cone}):
$$d_{\text{ped}} < 12.0 \vee d_{\text{cone}} < 12.0 \implies H_{\text{safe acc}} = \text{False}. \tag{19}$$
- **Road Rules:** On the contrary, if the vehicle does not have nearby objects of concern ($a_{\text{no ped}}$) and the road rules indicate the car does not need to stop ($a_{\text{no red light}}$), then we can accelerate:
$$a_{\text{no ped}} \wedge a_{\text{no red light}} \implies H_{\text{safe acc}} = \text{True} \tag{20}$$
- **Evidence of Lead Vehicle Stopped:** We can also look for evidence of a stopped lead vehicle using attributes of a stopped car like no brake lights (a_{brake}), not in the same lane ($a_{\text{near front}}$), or not in our path at an intersection ($a_{\text{lead impedes}}$):
$$\neg a_{\text{brake}} \vee \neg a_{\text{near front}} \vee \neg a_{\text{lead impedes}} \implies H_{\text{lead stopped}} = \text{False}$$

B.3 Results

Figure 10 summarizes macro-averaged Bal-Acc following the experimental methodology.

Models Evaluated. We evaluated on Llama-4-scout (L4) (Meta, 2025), Gemma-3-27b (G3) (Team et al., 2025), GPT-4o (G4o) (Hurst et al., 2024), Qwen3-VL-8B-Instruct (Q3-VL8) (Bai et al., 2025), and Qwen3-VL-235B-Instruct-FP8 (Q3-VL235) (Bai et al., 2025). We additionally evaluated on the domain-specific model DriveLMM-o1 (Drv) (Ishaq et al., 2025).

Limitation of Token Usage For the implementation of ITG for these hypotheses of the nuScenes dataset, there were many auxiliary probes used. A full list is seen in the prompt in Appendix H. Several of these auxiliary probes request descriptions of the scene, as these descriptions were deemed necessary in order to extract useful information from the distance answers. Thus, there is a trade-off in this case since the accuracy improvements do come at an increase in the number of tokens used.

Limitation of Dataset Processing. It is important to acknowledge that the ground truth for these experiments was algorithmically derived from 3D bounding box annotations rather than manual human labeling. While we employed heuristics such as a safety corridor and velocity estimation to infer semantic states, this approach introduces inherent uncertainty. For instance, a vehicle cutting in might be geometrically safe by our distance metric but semantically unsafe to a human driver. Despite this limitation, the relative improvements on $H_{\text{safe acc}}$ suggest that ITG can enforce consistency with the physical reality of the scene, even when that reality is approximated by geometric rules. On $H_{\text{lead stopped}}$, the ITG auxiliary probes we refined (brake lights, ego-lane, impede, parked) are not the inputs used to build the auto-generated label (attribute `vehicle.stopped` or speed < 1 m/s).

Stage	Base+A			ITG		
	Acc	Bal-Acc	F_1	Acc	Bal-Acc	F_1
(1) Hypothesis only	14.0	50.1	0.4	14.0	50.1	0.4
(2) Internal disclosures	13.8	50.0	0.0	60.8	77.2	70.5
(3) Edge / pattern match	13.8	50.0	0.0	64.6	79.5	74.1
(4) All auxiliary probes	13.8	50.0	0.0	87.4	77.8	92.5

Table 2: NDA-7 auxiliary probe richness ablation on Llama-3.1-8B-Instruct (n=10).

C ITG Implementation Pseudo-code

```python
class ExampleGuardrailSchema():
    # Auxiliary probe answers (filled by the model into schema fields)
    fact: bool
    extracted_number: float
    extracted_text: str

    # Target Hypothesis (H)
    prediction: bool

    def correct_inconsistencies(self):
        # Example Invariant: If fact implies NOT prediction
        if self.fact and self.prediction:
             self.prediction = False
        return

def run_inference(input_data, model):
    # 1. Construct Prompt with Inputs
    message = build_message(input_data)

    # 2. Generate Structured Output
    completion = llm_completion(
        model=model,
        messages=message,
        schema=ExampleGuardrailSchema
    )

    # 3. Apply Logic Guardrails
    completion.correct_inconsistencies()

    return completion
```

Listing 1: Pseudocode of the ITG Pipeline

D Ablating Auxiliary Probe Richness

A natural worry is that ITG gains come from asking more questions, not from the deterministic check. We test that on NDA-7 with Llama-3.1-8B by growing the auxiliary probe set in four nested stages (Table 2): 1) hypothesis only, 2) common internal disclosures, 3) edge-case patterns, and 4) the full set with definition quotes. At each stage we compare Base+A to ITG, separating whether corrections from C in ITG improve performance over the case where no correction is applied with just Base+A. Without correction provided by C, we find that the longer prompts do almost nothing moving from (1) to (4). Base+A stays near Acc of 13.8, Bal-Acc of 50, and F_1 of 0 from the smallest probe set through the richest one. With only the internal-disclosure probes, applying C raises Acc to 60.8, Bal-Acc to 77.2, and F_1 to 70.5. Adding further auxiliary probes raises Acc and F_1 further as those answers support additional correction rules. Across stages (2)–(4), C adds about $+27$ to $+30$ Bal-Acc.

E CALIBRATION-BASED UNCERTAINTY FILTERING

This section details the empirical evaluation of uncertainty estimation (UE) for the ContractNLI task using Llama-3.1-8B-Instruct. We define the effectiveness of an uncertainty filter by its ability to identify and abstain from confident hallucinations. These are instances where the model produces a high-confidence probabilistic signal (p) for a logically inconsistent or factually incorrect hypothesis (H). Summaries of these experimental results are presented in Tables 3 and 4.

H	Accuracy	Mean p	BS $\downarrow$	ECE $\downarrow$	$\mathbf{AUROC_{wrong}}$
NDA-5	0.8615 ± 0.016	0.2155 ± 0.010	0.6346 ± 0.0027	0.6512 ± 0.013	0.6131 ± 0.028
NDA-7	0.3923 ± 0.019	0.7705 ± 0.026	0.6705 ± 0.0065	0.6798 ± 0.005	0.3122 ± 0.028
NDA-16	0.1846 ± 0.000	0.0148 ± 0.000	0.1693 ± 0.0000	0.1698 ± 0.000	0.5417 ± 0.000
NDA-19	0.2077 ± 0.011	0.0586 ± 0.010	0.1405 ± 0.0005	0.1491 ± 0.002	0.6658 ± 0.018
NDA-20	0.6923 ± 0.019	0.3917 ± 0.024	0.2947 ± 0.0018	0.3022 ± 0.017	0.7825 ± 0.009
Pooled	$\mathbf{0.4677 \pm 0.007}$	$\mathbf{0.2902 \pm 0.009}$	$\mathbf{0.3819 \pm 0.0011}$	$\mathbf{0.3724 \pm 0.004}$	$\mathbf{0.5881 \pm 0.004}$

Table 3: Verbalized uncertainty metrics for ContractNLI with Llama-3.1-8B. $n = 10$ runs with metrics reported $\mu \pm \sigma$.

H	Accuracy	Mean p	BS $\downarrow$	ECE $\downarrow$	$\mathbf{AUROC_{wrong}}$
NDA-5	0.9600 ± 0.0108	0.8673 ± 0.0042	0.0487 ± 0.0019	0.0999 ± 0.0053	0.7800 ± 0.0556
NDA-7	0.3431 ± 0.0104	0.9192 ± 0.0026	0.5954 ± 0.0023	0.5835 ± 0.0082	0.3799 ± 0.0248
NDA-16	0.5108 ± 0.0141	0.8037 ± 0.0047	0.3232 ± 0.0041	0.3040 ± 0.0113	0.6786 ± 0.0342
NDA-19	0.2938 ± 0.0184	0.8699 ± 0.0022	0.5665 ± 0.0044	0.5923 ± 0.0109	0.5138 ± 0.0434
NDA-20	0.7508 ± 0.0215	0.8571 ± 0.0040	0.1712 ± 0.0054	0.1294 ± 0.0113	0.7868 ± 0.0324
Pooled	$\mathbf{0.5717 \pm 0.0058}$	$\mathbf{0.8635 \pm 0.0013}$	$\mathbf{0.3410 \pm 0.0016}$	$\mathbf{0.2961 \pm 0.0059}$	$\mathbf{0.5254 \pm 0.0104}$

Table 4: Logit-based uncertainty metrics for ContractNLI with Llama-3.1-8B. $n = 10$ runs with metrics reported $\mu \pm \sigma$.

Metrics for Uncertainty Evaluation. We evaluate the quality of the uncertainty signals using five standard metrics.

- **Accuracy**: The fraction of hypothesis slots where the model's base prediction matches the ground truth.
- **Mean p**: The average confidence score assigned by the model (derived from raw logits or verbalized self-reports).
- **Brier Score (BS)**: The mean squared difference between the predicted probability and the actual outcome $y \in \{0,1\}$. A score of 0.25 represents a non-informative 50/50 guess. Scores above this indicate poor calibration.
- **Expected Calibration Error (ECE)**: The weighted average gap between mean confidence and accuracy across binned predictions.
- $\mathbf{AUROC_{wrong}}$: The area under the receiver operating characteristic curve treating incorrect predictions as the positive class and $(1-p)$ as the ranking score. It represents the probability that a correctly predicted instance is assigned higher confidence than an incorrect one.

The Calibration Paradox. Our results reveal a significant discrepancy between the model's internal probability and its actual performance. The logit UE exhibits systemic overconfidence, with a pooled mean p of 0.8635 against a true accuracy of only 0.5717. Conversely, the verbal UE (self-reported confidence via zero-shot prompting) demonstrates extreme underconfidence, with a pooled mean p of 0.2902. This suggests that neither internal weights nor verbalized probability effectively map to the model's performance in high-stakes legal document analysis. In other words, the model knows the right answer more often than it says it does, but it also claims to be certain about answers that are logically flawed.

Reliability Metrics. We quantify the calibration gap using ECE. The logit UE achieved a pooled ECE of 0.2961, outperforming the verbal UE's ECE of 0.3724. While raw logits remain a more reliable signal for filtering than subjective self-assessment, both exhibit high error rates that ITG is designed to mitigate by enforcing domain-specific invariants, $I(H,A)$.

Failure Mode Analysis. We measure the utility of uncertainty scores for automated filtering using the $AUROC_{wrong}$ metric, which represents the probability that a correctly predicted instance is assigned a higher confidence score than an incorrect one. For the logit signal, the pooled $AUROC_{wrong}$ is 0.5254. This value, being only marginally better than a random baseline (0.50), confirms that simple threshold-based filtering is insufficient for legal document analysis tasks where models remain confidently wrong.

ITG as a Robust Alternative to UE. While probabilistic signals failed to distinguish errors, the ITG framework provides a superior filtering signal by targeting symbolic inconsistencies. In our evaluation, 59.56% of incorrect slots occurred in instances where the model maintained high logit confidence ($p > 0.85$) while simultaneously violating ITG constraints. By using deterministic predicates $I(H,A)$ to identify these logic-probability conflicts, ITG successfully repairs violations that traditional uncertainty estimation would overlook.

F PER-HYPOTHESIS DETAILED RESULTS

The following tables repeat Base, Base+A, and ITG with n=10 seed variance for readers who want the full breakdown. Macro-averaged summaries and bar charts appear in the case studies (Figures 1, 4, 7 and 10).

NDA-5: Share with employees	Accuracy ($\mu\pm\sigma\%$ $n=10$)					F1-Score ($\mu\pm\sigma\%$ $n=10$)				
Model	**Base**	**Base+A**	**Base+CoT**	**SatLM**	**ITG**	**Base**	**Base+A**	**Base+CoT**	**SatLM**	**ITG**
Llama-3.1-8B	40.3 ±2.2	49.0 ±2.1	79.6 ±3.7	68.5 ±1.7	**85.8 ±0.4**	35.3 ±3.9	49.5 ±3.1	84.9 ±3.1	80.5 ±1.0	**90.6 ±0.3**
Llama-4-scout	69.3 ±0.6	86.2 ±1.3	**93.0 ±1.5**	2.1 ±1.2	84.5 ±0.0	74.7 ±0.6	90.1 ±1.0	**95.3 ±1.0**	3.1 ±0.8	90.8 ±0.0
Gemma-3-27b-it	91.5 ±0.0	91.7 ±0.4	**93.1 ±1.7**	58.7 ±0.7	81.4 ±0.9	94.3 ±0.0	94.5 ±0.3	**95.5 ±1.1**	69.1 ±0.7	88.7 ±0.6
GPT-oss-120b	92.7 ±1.5	94.8 ±1.2	**95.1 ±1.0**	88.6 ±1.8	83.1 ±1.1	95.0 ±1.0	96.5 ±0.8	**96.7 ±0.7**	92.9 ±1.2	90.0 ±0.7
GPT-4o	94.5 ±0.4	91.3 ±0.9	**95.8 ±1.1**	81.0 ±1.2	78.7 ±1.2	96.5 ±0.3	94.5 ±0.6	**97.3 ±0.7**	88.9 ±0.6	87.7 ±0.6
Qwen3-235B-Instruct-FP8	96.9 ±0.9	93.4 ±1.0	**97.5 ±1.3**	85.6 ±0.6	79.4 ±0.7	98.0 ±0.6	95.8 ±0.6	**98.3 ±0.8**	91.4 ±0.3	88.1 ±0.4

NDA-7: Share with third-party	Accuracy ($\mu\pm\sigma\%$ $n=10$)					F1-Score ($\mu\pm\sigma\%$ $n=10$)				
Model	**Base**	**Base+A**	**Base+CoT**	**SatLM**	**ITG**	**Base**	**Base+A**	**Base+CoT**	**SatLM**	**ITG**
Llama-3.1-8B	40.1 ±2.3	33.9 ±0.4	44.9 ±3.2	22.3 ±1.7	**75.6 ±1.9**	39.1 ±3.7	28.0 ±0.8	47.2 ±4.9	27.7 ±2.4	**82.1 ±1.6**
Llama-4-scout	33.2 ±0.7	53.0 ±1.2	42.7 ±2.3	1.7 ±0.6	**88.7 ±0.0**	26.6 ±1.4	57.5 ±1.5	42.8 ±3.7	2.2 ±1.2	**93.2 ±0.0**
Gemma-3-27b-it	70.6 ±0.4	60.6 ±0.9	64.9 ±3.5	54.8 ±0.4	**88.2 ±0.7**	78.0 ±0.4	67.4 ±1.0	73.4 ±2.9	65.8 ±0.5	**92.7 ±0.5**
GPT-oss-120b	83.5 ±2.3	77.6 ±3.6	79.3 ±2.7	74.8 ±2.0	**86.8 ±1.0**	88.9 ±1.7	84.0 ±2.9	85.3 ±2.2	84.0 ±1.4	**92.2 ±0.6**
GPT-4o	75.9 ±1.4	76.2 ±1.2	67.0 ±3.9	66.2 ±1.9	**84.4 ±1.2**	83.0 ±1.2	84.2 ±0.9	74.5 ±3.6	77.3 ±1.5	**91.0 ±0.6**
Qwen3-235B-Instruct-FP8	86.5 ±0.7	82.1 ±1.0	**87.5 ±2.3**	73.0 ±1.1	83.7 ±0.7	91.0 ±0.5	88.6 ±0.7	**91.8 ±1.6**	83.0 ±0.7	90.6 ±0.4

NDA-16: Return/destroy doc	Accuracy ($\mu\pm\sigma\%$ $n=10$)					F1-Score ($\mu\pm\sigma\%$ $n=10$)				
Model	**Base**	**Base+A**	**Base+CoT**	**SatLM**	**ITG**	**Base**	**Base+A**	**Base+CoT**	**SatLM**	**ITG**
Llama-3.1-8B	44.4 ±1.2	47.7 ±1.0	50.7 ±1.8	23.8 ±0.8	**74.6 ±1.5**	47.1 ±1.4	47.8 ±0.5	**48.7 ±1.9**	36.9 ±0.8	46.8 ±2.1
Llama-4-scout	46.5 ±0.0	42.5 ±0.6	52.7 ±1.2	0.8 ±1.4	**74.5 ±1.2**	48.6 ±0.0	46.9 ±0.3	51.6 ±1.0	1.6 ±2.6	**56.4 ±2.5**
Gemma-3-27b-it	45.1 ±0.0	46.5 ±0.0	46.5 ±0.0	35.2 ±0.0	**73.7 ±1.2**	48.0 ±0.0	48.6 ±0.0	48.6 ±0.0	42.5 ±0.0	**54.5 ±2.7**
GPT-oss-120b	61.3 ±2.6	47.7 ±2.1	73.4 ±3.5	40.7 ±1.0	**80.7 ±1.5**	54.9 ±2.4	48.0 ±2.1	**62.2 ±3.8**	45.4 ±1.2	62.0 ±3.8
GPT-4o	46.5 ±0.0	46.5 ±0.0	46.9 ±2.1	36.2 ±1.0	**76.3 ±0.6**	48.6 ±0.0	48.6 ±0.0	47.3 ±1.6	44.3 ±0.4	**48.8 ±0.6**
Qwen3-235B-Instruct-FP8	46.5 ±0.0	46.5 ±0.0	**82.0 ±3.6**	38.0 ±0.7	80.4 ±0.4	48.6 ±0.0	48.6 ±0.0	**67.6 ±5.3**	44.9 ±0.7	59.0 ±0.6

NDA-19: Survival obligation	Accuracy ($\mu\pm\sigma\%$ $n=10$)					F1-Score ($\mu\pm\sigma\%$ $n=10$)				
Model	**Base**	**Base+A**	**Base+CoT**	**SatLM**	**ITG**	**Base**	**Base+A**	**Base+CoT**	**SatLM**	**ITG**
Llama-3.1-8B	70.3 ±0.4	72.0 ±0.8	74.2 ±3.5	62.8 ±2.1	**83.7 ±1.0**	70.2 ±0.6	73.1 ±1.0	82.5 ±2.1	70.4 ±1.9	**86.9 ±0.9**
Llama-4-scout	73.2 ±0.0	83.1 ±1.5	**86.3 ±3.3**	5.2 ±1.6	85.8 ±1.4	82.2 ±0.0	87.5 ±1.0	89.1 ±2.7	9.9 ±2.9	**89.9 ±0.9**
Gemma-3-27b-it	**89.2 ±0.7**	85.9 ±0.9	85.1 ±1.7	76.9 ±0.7	88.7 ±0.0	91.0 ±0.5	88.1 ±0.9	89.7 ±1.0	84.2 ±0.6	**91.3 ±0.0**
GPT-oss-120b	85.1 ±2.1	86.9 ±2.2	82.3 ±2.2	81.3 ±2.1	**88.7 ±1.9**	88.2 ±1.8	89.7 ±1.7	86.1 ±1.8	86.9 ±1.5	**91.7 ±1.3**
GPT-4o	87.2 ±1.8	**88.2 ±1.4**	83.2 ±1.0	74.2 ±1.2	88.0 ±1.2	90.9 ±1.2	91.1 ±1.0	88.3 ±0.7	83.4 ±0.6	**91.4 ±0.8**
Qwen3-235B-Instruct-FP8	**89.2 ±1.0**	85.2 ±0.7	81.4 ±2.6	74.2 ±1.0	81.8 ±0.4	**92.1 ±0.6**	89.6 ±0.5	86.2 ±2.0	83.3 ±0.6	87.5 ±0.3

NDA-20: Permissible retention	Accuracy ($\mu\pm\sigma\%$ $n=10$)					F1-Score ($\mu\pm\sigma\%$ $n=10$)				
Model	**Base**	**Base+A**	**Base+CoT**	**SatLM**	**ITG**	**Base**	**Base+A**	**Base+CoT**	**SatLM**	**ITG**
Llama-3.1-8B	90.1 ±2.3	**93.8 ±1.2**	89.9 ±3.1	58.6 ±2.8	92.4 ±1.2	83.6 ±4.4	**90.3 ±1.9**	85.9 ±4.4	39.5 ±2.9	88.3 ±1.9
Llama-4-scout	97.2 ±0.0	96.1 ±0.6	**99.3 ±1.0**	5.1 ±1.4	96.1 ±0.6	95.7 ±0.0	94.3 ±0.8	**99.0 ±1.4**	5.9 ±0.9	94.3 ±0.8
Gemma-3-27b-it	91.5 ±0.0	**98.6 ±0.0**	96.9 ±0.9	78.3 ±0.7	95.8 ±0.0	87.0 ±0.0	**97.9 ±0.0**	95.6 ±1.2	70.9 ±1.3	93.9 ±0.0
GPT-oss-120b	99.3 ±0.7	**99.4 ±0.7**	99.0 ±0.7	91.0 ±0.7	**99.4 ±0.7**	98.9 ±1.1	**99.2 ±1.1**	98.5 ±1.0	88.2 ±0.8	**99.2 ±1.1**
GPT-4o	95.8 ±0.9	98.5 ±0.4	**98.9 ±0.6**	77.6 ±2.6	98.5 ±0.4	94.1 ±1.2	97.8 ±0.6	**98.4 ±0.9**	75.2 ±2.2	97.8 ±0.6
Qwen3-235B-Instruct-FP8	**100.0 ±0.0**	91.5 ±0.0	98.5 ±0.4	88.7 ±0.7	90.1 ±0.0	**100.0 ±0.0**	88.9 ±0.0	97.7 ±0.7	85.7 ±0.7	87.3 ±0.0

Table 5: Mean and standard deviation ($n=10$) on ContractNLI. Per row, **bold** marks the highest mean. Macro summary in Figure 1.

Is Final Answer Correct?	Accuracy ($\mu\pm\sigma\%$ $n=10$)			
Model	**Base**	**Base+A**	**Base+CoT**	**ITG**
Llama-3.1-8B	13.8 ±0.2	12.7 ±0.2	**83.0 ±1.0**	51.1 ±0.4
Llama-4-scout	37.4 ±0.2	37.1 ±0.2	**95.6 ±0.2**	81.6 ±0.3
Gemma-3-27b-it	29.4 ±0.2	30.2 ±0.1	**95.3 ±0.3**	79.4 ±0.2
GPT-oss-120b	**96.6 ±0.3**	96.5 ±0.1	96.5 ±0.2	96.5 ±0.1
GPT-4o	44.3 ±0.7	42.9 ±0.9	**96.1 ±0.2**	87.1 ±0.3
Qwen3-235B-Instruct-FP8	53.2 ±0.3	52.7 ±0.3	**96.5 ±0.2**	90.7 ±0.3
DeepSeek-Coder-V2-Lite	20.5 ±0.1	21.7 ±0.2	**85.4 ±0.7**	60.9 ±0.4

Table 6: Mean and standard deviation ($n=10$) on GSM8K. Per row, **bold** marks the highest mean. Macro summary in Figure 7.

Is Safe to Accelerate?	Accuracy ($\mu \pm \sigma\%$ $n=10$)				F1-Score ($\mu \pm \sigma\%$ $n=10$)			
Model	**Base**	**Base+A**	**Base+CoT**	**ITG**	**Base**	**Base+A**	**Base+CoT**	**ITG**
Llama-4-scout	71.2 ±0.3	61.9 ±0.3	**80.4 ±1.0**	79.1 ±0.4	80.5 ±0.2	72.6 ±0.3	**88.3 ±0.7**	86.7 ±0.3
Gemma-3-27B-it	75.4 ±0.2	**78.2 ±0.2**	53.4 ±1.3	73.7 ±0.4	84.3 ±0.1	**86.5 ±0.1**	63.5 ±1.4	82.6 ±0.3
GPT-4o	42.7 ±0.6	45.8 ±0.6	**65.3 ±0.6**	60.4 ±0.5	50.2 ±0.7	54.1 ±0.7	**75.4 ±0.5**	70.2 ±0.5
Qwen3-VL-8B-Instruct	64.5 ±0.2	67.4 ±0.2	45.3 ±0.8	**72.0 ±0.3**	74.4 ±0.2	77.2 ±0.2	53.9 ±0.9	**81.0 ±0.3**
Qwen3-VL-235B-Instruct-FP8	57.2 ±0.2	68.0 ±0.3	59.5 ±0.7	**77.5 ±0.3**	67.3 ±0.2	77.8 ±0.2	70.2 ±0.7	**85.3 ±0.2**
DriveLMM-o1	70.9 ±0.4	74.4 ±0.4	2.7 ±0.3	**77.2 ±0.4**	79.9 ±0.3	82.9 ±0.3	3.3 ±0.7	**85.2 ±0.3**
Is Lead Vehicle Stopped?	**Accuracy** ($\mu \pm \sigma\%$ $n=10$)				**F1-Score** ($\mu \pm \sigma\%$ $n=10$)			
Model	**Base**	**Base+A**	**Base+CoT**	**ITG**	**Base**	**Base+A**	**Base+CoT**	**ITG**
Llama-4-scout	79.3 ±0.4	91.1 ±0.1	89.3 ±0.5	**92.8 ±0.1**	18.1 ±0.9	6.8 ±1.3	**28.9 ±2.9**	8.3 ±1.6
Gemma-3-27B-it	80.0 ±0.3	91.9 ±0.1	86.6 ±0.8	**92.6 ±0.1**	**34.1 ±0.5**	29.8 ±1.2	28.9 ±3.5	14.5 ±2.2
GPT-4o	62.7 ±0.9	71.0 ±1.1	74.9 ±0.8	**91.8 ±0.3**	26.8 ±0.5	30.1 ±1.1	33.5 ±1.1	**52.4 ±1.0**
Qwen3-VL-8B-Instruct	78.5 ±0.2	87.8 ±0.3	83.2 ±0.6	**92.6 ±0.2**	36.0 ±0.5	45.7 ±0.6	29.5 ±1.7	**54.2 ±0.6**
Qwen3-VL-235B-Instruct-FP8	84.3 ±0.3	88.5 ±0.3	79.1 ±0.6	**92.4 ±0.1**	37.0 ±1.0	**37.2 ±1.2**	34.0 ±1.0	35.8 ±1.9
DriveLMM-o1	65.7 ±0.5	67.2 ±0.6	6.6 ±0.5	**93.5 ±0.3**	27.9 ±0.4	29.1 ±0.6	0.6 ±0.3	**29.9 ±4.4**

Table 7: Mean and standard deviation ($n=10$) on NuScenes. Per row, **bold** marks the highest mean. Macro summary in Figure 10.

Is Minimum?	Accuracy ($\mu \pm \sigma\%$ $n=10$)					F1-Score ($\mu \pm \sigma\%$ $n=10$)				
Model	**Base**	**Base+A**	**Base+CoT**	**PAL**	**ITG**	**Base**	**Base+A**	**Base+CoT**	**PAL**	**ITG**
Llama-4-scout	92.1 ±0.4	91.0 ±0.7	**97.6 ±1.0**	57.4 ±1.1	97.3 ±0.7	91.7 ±0.5	91.1 ±0.7	**97.5 ±1.0**	60.5 ±1.0	97.3 ±0.7
Gemma-3-27B-it	80.2 ±0.6	83.7 ±0.8	90.1 ±1.8	24.1 ±39.1	**91.8 ±1.1**	77.7 ±0.8	83.5 ±0.8	90.0 ±1.9	24.0 ±39.1	**91.6 ±1.2**
GPT-4o	87.4 ±1.0	**91.7 ±1.4**	87.0 ±2.9	90.1 ±1.3	90.5 ±1.3	87.2 ±1.1	**91.6 ±1.5**	86.8 ±3.0	90.6 ±1.2	90.7 ±1.2
Qwen3-VL-8B-Instruct	97.5 ±0.0	87.6 ±1.0	96.9 ±1.3	74.8 ±1.9	**98.7 ±0.4**	97.4 ±0.0	87.3 ±1.2	96.9 ±1.3	71.4 ±2.1	**98.7 ±0.4**
Qwen3-VL-235B-Instruct-FP8	95.4 ±0.6	93.2 ±0.4	99.1 ±0.3	10.2 ±31.3	**99.2 ±0.0**	95.6 ±0.5	93.6 ±0.3	99.1 ±0.3	10.3 ±31.3	**99.2 ±0.0**
ThinkLite-VL-7B	89.7 ±0.6	71.9 ±0.9	94.2 ±1.6	91.1 ±1.1	**94.7 ±1.3**	88.5 ±0.7	61.5 ±1.7	94.3 ±1.5	90.4 ±1.3	**94.7 ±1.2**
Is Maximum?	**Accuracy** ($\mu \pm \sigma\%$ $n=10$)					**F1-Score** ($\mu \pm \sigma\%$ $n=10$)				
Model	**Base**	**Base+A**	**Base+CoT**	**PAL**	**ITG**	**Base**	**Base+A**	**Base+CoT**	**PAL**	**ITG**
Llama-4-scout	96.7 ±0.0	96.3 ±0.8	**97.8 ±0.9**	59.3 ±2.0	93.0 ±0.7	96.6 ±0.0	96.4 ±0.8	**97.8 ±0.9**	63.9 ±1.8	93.5 ±0.6
Gemma-3-27B-it	89.8 ±0.4	92.2 ±0.6	**94.7 ±0.4**	24.6 ±39.8	92.0 ±0.9	88.6 ±0.5	91.6 ±0.7	**94.4 ±0.5**	24.3 ±39.4	91.4 ±1.0
GPT-4o	**94.4 ±0.7**	91.8 ±0.8	90.0 ±1.8	91.2 ±0.7	89.8 ±0.5	**94.2 ±0.8**	91.4 ±0.8	89.3 ±1.9	90.8 ±0.8	89.4 ±0.5
Qwen3-VL-8B-Instruct	94.8 ±0.4	87.1 ±0.4	**98.2 ±0.8**	75.7 ±1.6	92.2 ±0.6	95.1 ±0.3	88.4 ±0.3	**98.2 ±0.8**	72.5 ±2.0	92.7 ±0.5
Qwen3-VL-235B-Instruct-FP8	99.2 ±0.0	99.2 ±0.0	99.9 ±0.3	10.2 ±31.3	**100.0 ±0.0**	99.2 ±0.0	99.2 ±0.0	99.9 ±0.3	10.2 ±31.3	**100.0 ±0.0**
ThinkLite-VL-7B	**99.2 ±0.0**	96.5 ±0.8	97.0 ±0.9	86.5 ±1.0	91.5 ±1.4	**99.2 ±0.0**	96.6 ±0.7	97.1 ±0.9	86.9 ±1.2	92.2 ±1.2
Is Low Median?	**Accuracy** ($\mu \pm \sigma\%$ $n=10$)					**F1-Score** ($\mu \pm \sigma\%$ $n=10$)				
Model	**Base**	**Base+A**	**Base+CoT**	**PAL**	**ITG**	**Base**	**Base+A**	**Base+CoT**	**PAL**	**ITG**
Llama-4-scout	57.1 ±1.0	58.9 ±1.2	79.8 ±2.7	44.9 ±2.5	**87.7 ±1.2**	48.7 ±1.0	47.6 ±2.1	78.4 ±3.1	47.2 ±3.0	**87.0 ±1.4**
Gemma-3-27B-it	52.2 ±0.8	45.5 ±1.2	74.2 ±3.6	20.9 ±34.0	**90.1 ±0.7**	18.7 ±2.1	19.5 ±0.8	72.6 ±3.7	19.8 ±32.3	**89.7 ±0.8**
GPT-4o	66.5 ±1.6	64.2 ±1.6	69.9 ±2.2	70.3 ±2.1	**80.2 ±1.4**	66.2 ±1.7	68.6 ±1.4	63.9 ±3.1	63.5 ±3.1	**78.7 ±1.5**
Qwen3-VL-8B-Instruct	83.7 ±1.0	76.5 ±1.2	87.6 ±2.2	68.7 ±2.1	**99.5 ±0.4**	81.6 ±1.3	72.3 ±1.5	86.8 ±2.3	65.1 ±2.7	**99.5 ±0.4**
Qwen3-VL-235B-Instruct-FP8	83.7 ±0.8	81.6 ±1.3	93.7 ±2.6	10.2 ±31.0	**100.0 ±0.0**	85.6 ±0.7	84.5 ±0.9	93.3 ±2.8	10.2 ±31.0	**100.0 ±0.0**
ThinkLite-VL-7B	72.2 ±1.5	61.8 ±1.5	**87.4 ±1.3**	85.9 ±1.1	66.3 ±1.9	63.2 ±2.9	39.8 ±3.6	**87.0 ±1.4**	85.4 ±1.3	50.0 ±3.9
Is High Median?	**Accuracy** ($\mu \pm \sigma\%$ $n=10$)					**F1-Score** ($\mu \pm \sigma\%$ $n=10$)				
Model	**Base**	**Base+A**	**Base+CoT**	**PAL**	**ITG**	**Base**	**Base+A**	**Base+CoT**	**PAL**	**ITG**
Llama-4-scout	48.2 ±0.6	51.8 ±0.7	65.9 ±1.9	42.2 ±3.2	**96.2 ±1.0**	15.3 ±1.3	32.1 ±1.8	53.3 ±2.7	43.5 ±4.0	**96.2 ±1.0**
Gemma-3-27B-it	51.9 ±0.4	49.0 ±1.2	68.2 ±2.4	22.5 ±36.5	**91.7 ±0.7**	13.0 ±1.4	14.5 ±2.4	61.0 ±3.1	22.0 ±35.8	**91.3 ±0.7**
GPT-4o	57.8 ±1.5	57.1 ±1.6	76.2 ±3.1	80.0 ±2.7	**82.3 ±1.3**	49.7 ±2.1	56.2 ±1.8	73.2 ±4.2	78.0 ±3.2	**80.8 ±1.6**
Qwen3-VL-8B-Instruct	68.2 ±0.7	65.2 ±1.6	84.2 ±2.2	68.2 ±2.5	**99.2 ±0.0**	58.7 ±1.3	57.1 ±2.4	82.3 ±2.4	63.2 ±3.2	**99.2 ±0.0**
Qwen3-VL-235B-Instruct-FP8	75.2 ±0.9	80.0 ±0.7	84.5 ±3.2	10.3 ±31.5	**100.0 ±0.0**	73.1 ±1.0	81.3 ±0.7	82.5 ±3.8	10.3 ±31.5	**100.0 ±0.0**
ThinkLite-VL-7B	65.5 ±1.5	56.7 ±1.4	**68.8 ±2.5**	59.2 ±1.3	66.8 ±2.0	54.1 ±2.8	35.3 ±3.1	**63.2 ±3.0**	49.8 ±1.9	51.6 ±4.1
Is Less Than?	**Accuracy** ($\mu \pm \sigma\%$ $n=10$)					**F1-Score** ($\mu \pm \sigma\%$ $n=10$)				
Model	**Base**	**Base+A**	**Base+CoT**	**PAL**	**ITG**	**Base**	**Base+A**	**Base+CoT**	**PAL**	**ITG**
Llama-4-scout	93.2 ±0.4	91.8 ±1.5	**98.4 ±0.6**	57.8 ±2.5	97.0 ±0.6	92.9 ±0.4	92.1 ±1.5	**98.4 ±0.6**	61.3 ±1.9	97.0 ±0.6
Gemma-3-27B-it	78.0 ±0.4	67.9 ±0.9	89.5 ±0.7	26.2 ±42.4	**94.1 ±0.9**	78.8 ±0.3	71.8 ±0.8	90.2 ±0.6	26.2 ±42.4	**94.0 ±0.9**
GPT-4o	**93.4 ±0.6**	90.2 ±1.2	88.1 ±2.3	89.3 ±0.9	90.2 ±1.2	**93.3 ±0.6**	90.4 ±1.0	88.5 ±2.0	89.9 ±0.7	90.4 ±1.0
Qwen3-VL-8B-Instruct	93.4 ±0.7	87.1 ±0.6	**98.1 ±0.8**	77.9 ±1.5	97.2 ±0.4	93.4 ±0.7	87.3 ±0.6	**98.1 ±0.8**	75.6 ±1.9	97.1 ±0.5
Qwen3-VL-235B-Instruct-FP8	**99.3 ±0.5**	97.5 ±1.0	99.2 ±0.0	9.8 ±30.0	98.1 ±0.7	**99.3 ±0.5**	97.6 ±0.9	99.2 ±0.0	9.8 ±30.0	98.1 ±0.7
ThinkLite-VL-7B	73.2 ±1.0	73.5 ±1.3	**93.7 ±1.2**	79.9 ±1.8	84.9 ±1.3	72.8 ±1.2	73.8 ±1.3	**93.6 ±1.2**	80.5 ±1.6	86.3 ±1.1
Is Greater Than?	**Accuracy** ($\mu \pm \sigma\%$ $n=10$)					**F1-Score** ($\mu \pm \sigma\%$ $n=10$)				
Model	**Base**	**Base+A**	**Base+CoT**	**PAL**	**ITG**	**Base**	**Base+A**	**Base+CoT**	**PAL**	**ITG**
Llama-4-scout	80.4 ±0.8	95.5 ±1.1	**98.6 ±0.4**	55.9 ±1.9	97.1 ±0.7	79.0 ±0.9	95.4 ±1.1	**98.6 ±0.4**	59.3 ±2.0	97.0 ±0.7
Gemma-3-27B-it	59.7 ±0.4	48.5 ±0.7	89.6 ±0.7	25.0 ±40.5	**94.1 ±0.9**	70.6 ±0.2	62.8 ±0.7	90.3 ±0.6	25.1 ±40.7	**94.0 ±0.9**
GPT-4o	85.2 ±1.5	90.1 ±1.2	88.2 ±2.3	89.3 ±0.9	**90.2 ±1.2**	86.4 ±1.3	90.3 ±1.1	88.6 ±2.0	89.9 ±0.7	**90.4 ±1.0**
Qwen3-VL-8B-Instruct	**99.2 ±0.0**	56.9 ±0.8	98.2 ±0.9	79.7 ±1.1	97.2 ±0.4	**99.2 ±0.0**	69.9 ±0.4	98.1 ±0.9	77.3 ±1.6	97.1 ±0.5
Qwen3-VL-235B-Instruct-FP8	97.5 ±0.9	60.1 ±1.1	**99.2 ±0.0**	10.2 ±31.0	**99.2 ±0.0**	97.5 ±0.8	69.6 ±0.7	**99.2 ±0.0**	10.2 ±31.0	**99.2 ±0.0**
ThinkLite-VL-7B	89.8 ±0.8	87.0 ±0.7	**96.2 ±1.3**	90.0 ±2.1	89.2 ±0.8	88.6 ±1.0	85.4 ±0.8	**96.1 ±1.3**	89.8 ±2.2	89.0 ±0.7

Table 8: Mean and standard deviation ($n=10$) on FigureQA. Per row, **bold** marks the highest mean. Macro summary in Figure 4.

G ABSOLUTE FIX AND BREAK COUNTS

In addition to FCR (%) and FBR, we report absolute mean counts per run (n=10): examples correct/wrong after ITG, plus wrong→right fixes and right→wrong breaks. Σ columns sum fixes/breaks over 10 seeds. Counts use the same balanced FigureQA split as the main paper (N=120).

Table 9: ContractNLI absolute counts per run (mean±std over seeds).

Model	Correct	Wrong	Fix (w→r)	Break (r→w)	ΣFix / ΣBreak
Llama-3.1-8B	262.6±3.0	62.4±3.0	101.2±3.2	19.1±0.3	1012 / 191
Llama-4-Scout	281.1±0.9	43.9±0.9	74.0±1.2	19.1±0.9	740 / 191
Gemma-3-27B	278.7±1.3	46.3±1.3	56.7±0.8	22.7±1.1	567 / 227
GPT-4o	282.0±1.2	43.0±1.2	45.0±1.2	21.6±0.5	450 / 216
Qwen3-235B	273.1±0.3	51.9±0.3	38.8±0.4	21.4±0.5	388 / 214

Table 10: FigureQA (bar) absolute counts per run.

Model	Correct	Wrong	Fix (w→r)	Break (r→w)	ΣFix / ΣBreak
Llama-4-Scout	682.0±2.5	38.0±2.5	110.6±2.5	10.9±1.6	1106 / 109
Gemma-3-27B	664.5±3.8	55.5±3.8	218.7±3.4	18.3±2.1	2187 / 183
GPT-4o	627.8±3.6	92.2±3.6	73.9±5.0	28.2±1.7	739 / 282
Qwen3-VL-8B	700.6±1.8	19.4±1.8	158.1±3.2	9.9±1.2	1581 / 99
Qwen3-VL-235B	715.7±0.8	4.3±0.8	104.2±2.0	2.3±0.8	1042 / 23
ThinkLite-VL-7B	592.1±6.8	127.9±6.8	92.4±5.1	37.2±3.4	924 / 372

Table 11: GSM8K absolute counts per run (corrected evaluation).

Model	Correct	Wrong	Fix (w→r)	Break (r→w)	ΣFix / ΣBreak
Llama-3.1-8B	674.1±5.6	644.9±5.6	546.5±6.4	34.6±2.2	5465 / 346
Llama-4-Scout	1076.4±3.8	242.6±3.8	622.6±4.4	29.2±1.4	6226 / 292
Gemma-3-27B	1047.5±2.6	271.5±2.6	679.1±2.8	29.8±1.0	6791 / 298
GPT-oss-120B	1272.4±1.8	46.6±1.8	0.0±0.0	0.0±0.0	0 / 0
GPT-4o	1149.3±4.6	169.7±4.6	610.2±11.4	22.3±2.8	6102 / 223
Qwen3-235B	1195.9±3.9	123.1±3.9	524.8±4.8	22.1±1.7	5248 / 221
DeepSeek-Coder-V2-Lite	803.4±5.4	515.6±5.4	570.6±5.8	50.7±2.6	5706 / 507

Table 12: nuScenes absolute counts per run.

Model	Correct	Wrong	Fix (w→r)	Break (r→w)	ΣFix / ΣBreak
Llama-4-Scout	1375.3±3.1	224.7±3.1	169.6±3.0	17.6±2.2	1696 / 176
Gemma-3-27B	1330.5±2.9	269.5±2.9	43.6±1.1	74.1±2.8	436 / 741
GPT-4o	1217.3±4.8	382.7±4.8	299.4±8.2	16.4±1.8	2994 / 164
Qwen3-VL-8B	1316.5±3.1	283.5±3.1	113.9±2.8	39.2±3.5	1139 / 392
Qwen3-VL-235B	1359.4±2.4	240.6±2.4	129.6±3.0	22.3±2.0	1296 / 223
DriveLMMo1	1365.6±4.5	234.4±4.5	290.3±5.2	57.8±2.7	2903 / 578

H Full Prompts Used in Experiments

For transparency, we provide the prompts used for our ITG experiments.

H.1 ContractNLI

Inference-Time Guardrail Prompt

You are an expert Legal Contract Analyst. Your task is to review Non-Disclosure Agreements (NDAs) and extract specific logical clauses into a structured JSON format.

Analyze the contract text provided. You must determine whether the following specific conditions are Explicitly True or False based on the text.

Return a JSON object with the following fields:

- "share_employees": True if Receiving Party may share some Confidential Information with some of Receiving Party's employees; False otherwise.
- "def_representatives_includes_employees": True if the definition of 'Representatives' (or 'Authorized Persons') explicitly list 'employees', 'officers', or 'directors'; False otherwise.
- "permitted_need_to_know": True if the contract state that information may be disclosed to persons with a 'need to know'; False otherwise.
- "liable_for_employees": True if the contract states that the Receiving Party is responsible/liable for any breach committed by its 'Representatives' or employees; False otherwise.
- "share_third_parties": True if Receiving Party may share some Confidential Information with some third-parties (consultants, agents, professional advisors); False otherwise.
- "permitted_advisors": True if the contract explicitly permits disclosure to professional advisors (lawyers, accountants, financial advisors); False otherwise.
- "permitted_financing_subcontractors": True if the contract explicitly permits disclosure to 'financing sources', 'lenders', 'co-investors', or 'subcontractors'; False otherwise.
- "permitted_vendors": True if the contract explicitly permits disclosure to 'vendors', 'service providers', 'auditors', or 'data processors'?
- "has_whistleblower_clause": True if the contract contains a 'Defend Trade Secrets Act' (DTSA) notice or Whistleblower immunity clause allowing disclosure to an attorney; False otherwise.
- "has_exceptions_to_confidentiality": True if the contract indicates that disclosure to certain third-parties is allowed under specific exceptions (e.g., legal requirement, prior knowledge); False otherwise.
- "definition_of_access_groups_quote": Copy the exact text defining 'Representatives', 'Authorized Users', 'Authorized Persons', or the clause stating exactly which groups 'will be given access' (e.g., 'Only employees and consultants...'). Remove any characters such as \" or \' which may interfere with parsing. Be careful not to repeat yourself.
- "return_or_destroy": True if Receiving Party shall destroy or return some Confidential Information upon the termination of Agreement; False otherwise.
- "return_or_destroy_quote": Copy the exact text snippet requiring return or destruction. Remove any characters such as \" or \' which may interfere with parsing. Be careful not to repeat yourself.
- "survival_obligations": True if some obligations of Agreement may survive termination of Agreement; False otherwise.
- "duration_is_indefinite": True if the contract states that the confidentiality obligation is perpetual, indefinite, or never expires; False otherwise.
- "permissible_retention": True if Receiving Party may retain some Confidential Information even after the return or destruction (e.g., for legal/archival purposes); False otherwise.
- "retention_includes_backups": True if the contract states that electronic copies stored in automatic backup/archive systems do not need to be destroyed immediately; False otherwise.
- "retention_counsel_copy": True if the Receiving Party's legal counsel is permitted to keep a copy for their records; False otherwise.

Rules for Extraction:

If the text is silent on a topic, set the value to False.

Do not guess. Only mark True if the text explicitly permits or requires it.

<INPUT CONTRACT DATA>

Base Prompt

You are an expert Legal Contract Analyst. Your task is to review Non-Disclosure Agreements (NDAs) and extract specific logical clauses into a structured JSON format.

Analyze the contract text provided. You must determine whether the following specific conditions are Explicitly True or False based on the text.

Return a JSON object with the following boolean fields:

- "share_employees": True if Receiving Party may share some Confidential Information with some of Receiving Party's employees; False otherwise.
- "share_third_parties": True if Receiving Party may share some Confidential Information with some third-parties (consultants, agents, professional advisors); False otherwise.
- "return_or_destroy": True if Receiving Party shall destroy or return some Confidential Information upon the termination of Agreement; False otherwise.
- "survival_obligations": True if some obligations of Agreement may survive termination of Agreement; False otherwise.
- "permissible_retention": True if Receiving Party may retain some Confidential Information even after the return or destruction (e.g., for legal/archival purposes); False otherwise.

Rules for Extraction:

If the text is silent on a topic, set the value to False.

Do not guess. Only mark True if the text explicitly permits or requires it.

<INPUT CONTRACT DATA>

Base+CoT Prompt

You are an expert Legal Contract Analyst. Your task is to review Non-Disclosure Agreements (NDAs) and decide whether specific conditions are explicitly supported by the contract text.

Please think step-by-step to answer the following. Provide your reasoning in plain text (not JSON).

For each condition below, determine if it is Explicitly True or False based only on the contract. If the text is silent on a topic, treat it as False. Do not guess; only mark True if the text explicitly permits or requires it.

Conditions to evaluate (same meanings as in structured evaluation):

- share_employees: May the Receiving Party share some Confidential Information with some of the Receiving Party's employees?
- share_third_parties: May the Receiving Party share some Confidential Information with third parties (consultants, agents, professional advisors)?
- return_or_destroy: Shall the Receiving Party destroy or return some Confidential Information upon termination of the Agreement?
- survival_obligations: May some obligations survive termination of the Agreement?
- permissible_retention: May the Receiving Party retain some Confidential Information after return or destruction (e.g., legal/archival purposes)?

After your reasoning, conclude with your final structured answer wrapped exactly in <FINAL_ANSWER> and </FINAL_ANSWER> tags. Inside the tags, output a single JSON object with only these five boolean keys: "share_employees", "share_third_parties", "return_or_destroy", "survival_obligations", "permissible_retention". Example shape: <FINAL_ANSWER>"share_employees": true, "share_third_parties": false, ...</FINAL_ANSWER>
Do not put your step-by-step reasoning inside the JSON or inside the tags—only the JSON object belongs inside <FINAL_ANSWER>.

<INPUT CONTRACT DATA>

H.2 NUSCENES

Inference-Time Guardrail Prompt

<INPUT IMAGE DATA>

You are an expert Traffic Scene Analyst and Autonomous Driving Assistant. Your task is to analyze dashboard camera images and extract specific semantic and physical details into a structured JSON format.

Analyze the image provided. You must determine whether the following specific conditions are Explicitly True or False based on the visual evidence.

Return a JSON object with the following fields:

- "is_safe_to_accelerate": True if it is safe for the ego-vehicle to accelerate immediately; false otherwise.
- "lead_vehicle_stopped": True if the vehicle directly in front of the ego-vehicle is stopped; false otherwise or if there is no lead vehicle.
- "lead_vehicle_at_intersection": True if the lead vehicle is positioned at an intersection, crosswalk, or stop line; False otherwise.
- "vehicle_count": As an integer, count ALL visible vehicles (cars, trucks, buses, motorcycles) in the image.
- "lead_vehicle_brake_lights_on": True if the brake lights or taillights of the lead vehicle are illuminated; False otherwise.
- "lead_vehicle_parked_off_road": True if the lead vehicle is parked off the road or in a parking lot; false otherwise.
- "lead_vehicle_in_distance": True if the lead vehicle is in the distance, not near the ego vehicle; false otherwise.
- "lead_vehicle_is_parked": True if the lead vehicle appears to be parked (e.g., on the shoulder, in a parking spot, empty driver seat); false otherwise.
- "lead_vehicle_impedes_ego": True if the lead vehicle is directly in the ego-vehicle's path, forcing the ego-vehicle to stop or slow down; false otherwise.
- "lead_vehicle_is_in_ego_lane": True if the lead vehicle is in the same lane as the ego vehicle, or in the lane that the ego vehicle is turning into via a lane change or at an intersection; false otherwise.
- "ego_vehicle_first_at_intersection": True if the ego vehicle is the first at an intersection with a red light or stop sign. This may be the case if there are cars driving in front of the ego vehicle's view from other parts of the intersection. There should be no cars also waiting in front of the ego vehicle. False otherwise.
- "nearest_vehicle_distance_lane": As a real number, estimate the distance (in meters) to the nearest vehicle in the ego-lane or directly ahead. Return 999.0 if no vehicle ahead.
- "nearest_vehicle_distance": DIFFERENT FROM THE PREVIOUS QUESTION. As a real number, estimate the distance (in meters) to the nearest vehicle in the whole scene. Return 999.0 if no vehicle at all.
- "nearest_vehicle_is_in_ego_lane": True if the nearest vehicle is in the same lane as the ego-vehicle; false otherwise.
- "pedestrians_crossing_road": True if there are pedestrians actively crossing the road ahead; false otherwise.
- "pedestrian_count": As an integer, count ALL visible pedestrians.
- "nearest_pedestrian_distance_lane": As a real number, estimate the distance (in meters) to the nearest pedestrian in the ego-lane or directly ahead. Return 999.0 if no pedestrians ahead.
- "nearest_pedestrian_distance": DIFFERENT FROM THE PREVIOUS QUESTION. As a real number, estimate the distance (in meters) to the nearest pedestrian in the whole scene. Return 999.0 if no pedestrians at all.
- "traffic_light_visible": True if there is a traffic light visible facing the ego-vehicle; false otherwise.
- "is_traffic_light_red": True if there is a traffic light is visible and it is RED; false otherwise.
- "stop_sign_visible": True if a Stop Sign is visible facing the ego-lane; false otherwise.
- "blocked_by_cones_barrier_distance_lane": As a real number, estimate the distance (in meters) to the nearest cone or barrier in the ego-lane or directly ahead. Return 999.0 if no cone or barrier ahead.
- "visible_sky_is_blue": True if the sky is blue; false otherwise.
- "sun_glare_visible": True if there is visible sun glare or direct sunlight shadows; false otherwise.

Rules for Extraction:

"Ego Lane" means the specific lane you are currently driving in. Cars in adjacent lanes do not block acceleration unless cutting in.

"Safe to accelerate" requires: No cars <10m in ego-lane, No pedestrians <15m, No Red Lights, No Lane Blockages, and No Stopped Lead Vehicle nearby.

Be precise with distance estimation.

Base Prompt

<INPUT IMAGE DATA>

You are an expert Traffic Scene Analyst and Autonomous Driving Assistant. Your task is to analyze dashboard camera images and extract specific semantic and physical details into a structured JSON format.

Analyze the image provided. You must determine whether the following specific conditions are Explicitly True or False based on the visual evidence.

Return a JSON object with the following fields:

- "is_safe_to_accelerate": True if it is safe for the ego-vehicle to accelerate immediately; false otherwise.
- "lead_vehicle_stopped": True if the vehicle directly in front of the ego-vehicle is stopped; false otherwise or if there is no lead vehicle.

Rules for Extraction:

"Safe to accelerate" requires: No cars <10m in ego-lane, No pedestrians <15m, No Red Lights, No Lane Blockages, and No Stopped Lead Vehicle nearby.

Base+CoT Prompt

<INPUT IMAGE DATA>

You are an expert Traffic Scene Analyst and Autonomous Driving Assistant. Your task is to analyze the dashboard camera image and decide whether specific conditions are explicitly true or false from the visual evidence.

Please think step-by-step. Provide your reasoning in plain text (not JSON).

Evaluate these conditions:

- is_safe_to_accelerate: Is it safe for the ego-vehicle to accelerate immediately? (Consider: no cars within about 10m in the ego lane, no pedestrians within about 15m, no red lights, no lane blockages, and no stopped lead vehicle posing a hazard.)
- lead_vehicle_stopped: Is the vehicle directly in front of the ego vehicle stopped? If there is no lead vehicle, false.

After your reasoning, conclude with your final answer wrapped exactly in <FINAL_ANSWER> and </FINAL_ANSWER> tags. Inside the tags, output a single JSON object with only these two boolean keys: "is_safe_to_accelerate", "lead_vehicle_stopped".
Do not put your step-by-step reasoning inside the JSON or inside the tags—only the JSON object belongs inside <FINAL_ANSWER>.

H.3 GSM8K

Inference-Time Guardrail Prompt

You are an expert Math Logic Translator and Verifier. Your task is to analyze word problems and extract the mathematical logic required to solve them into a structured JSON format.

Analyze the math problem provided. You must separate the raw data (numbers) from the reasoning (logic expression) and the final conclusion.

Return a JSON object with the following fields:

- "final_answer": The final numerical answer of the problem you believe is correct.
- "extracted_numbers": A list of all numerical values explicitly stated in the problem text. Do not include constants like '2' or '100' unless they appear in the text.
- "python_expression": A valid, single-line Python arithmetic expression that calculates the answer. Use the numbers from the text or standard constants (e.g., 60 minutes in an hour). Do not use variables (e.g., 'x = 5'). Just the math (e.g., '(48 / 2) + 10').
- "should_be_positive": True if the answer represents a real-world count, price, dimension, or time that physically cannot be negative. False if negative values are plausible (e.g., temperature change, debt).

Rules for Extraction:

Do not compute the answer mentally if possible; rely on constructing the correct formula in python_expression.

The python_expression is the "Ground Truth Logic." If it evaluates to a different number than final_answer, it indicates a calculation error.

<INPUT MATH PROMPT DATA>

Base Prompt

You are an expert Math Logic Translator and Verifier. Your task is to analyze word problems and extract the mathematical logic required to solve them into a structured JSON format.

Analyze the math problem provided. You must separate the raw data (numbers) from the reasoning (logic expression) and the final conclusion.

Return a JSON object with only the following field:

- "final_answer": The final numerical answer of the problem you believe is correct.

<INPUT MATH PROBLEM DATA>

Base+CoT Prompt

You are an expert at solving grade-school math word problems.

Please think step-by-step to solve the problem. Show your reasoning in plain text (not JSON). Perform any unit conversions needed so times become numbers when required (e.g., 5:30 as 5.5).

After your reasoning, give the single numerical answer the problem asks for. Wrap that final answer exactly in <FINAL_ANSWER> and </FINAL_ANSWER> tags. Inside the tags you may either output a bare number (e.g., <FINAL_ANSWER>18</FINAL_ANSWER>) or a tiny JSON object with one key: "final_answer": <number>.

Do not put your step-by-step work inside the tags—only the final numeric answer (or the one-key JSON) belongs inside <FINAL_ANSWER>.

Problem:

<INPUT MATH PROBLEM DATA>

H.4 FIGUREQA

Inference-Time Guardrail Prompt

<INPUT IMAGE DATA>

You are an expert Graph Analyst. Your task is to review graphs and answer questions into a structured JSON format.

Analyze the graph provided. You must determine whether the following specific conditions are Explicitly True or False based on the graph.

Note that X and Y are per question and are not necessarily the same across all questions.

Return a JSON object with the following boolean fields:

- "chart_data": A list of objects representing every bar in the chart.
- "label": The name of the bar.
- "value": The numerical value (height) of the bar.
- "y_axis_min": The minimum value displayed on the Y-axis scale.
- "y_axis_max": The maximum value displayed on the Y-axis scale.
- "q1_value_x": The numerical value of bar X mentioned in Question 1.
- "q2_value_x": The numerical value of bar X mentioned in Question 2.
- "q3_value_x": The numerical value of bar X mentioned in Question 3.
- "q4_value_x": The numerical value of bar X mentioned in Question 4.
- "q5_value_x": The numerical value of bar X mentioned in Question 5.
- "q5_value_y": The numerical value of bar Y mentioned in Question 5.
- "q6_value_x": The numerical value of bar X mentioned in Question 6.
- "q6_value_y": The numerical value of bar Y mentioned in Question 6.
- "question_1": Is X the minimum? True if yes; False if no.
- "question_2": Is X the maximum? True if yes; False if no.
- "question_3": Is X the low median? True if yes; False if no.
- "question_4": Is X the high median? True if yes; False if no.
- "question_5": Is X less than Y? True if yes; False if no.
- "question_6": Is X greater than Y? True if yes; False if no.

Rules for Extraction:

- Extract numerical values as precisely as possible from the chart.
- If X or Y refers to a bar, provide the estimated value of that bar.=
- For 'chart_data', ensure you list ALL bars visible in the chart.

NOTICE: You will use the questions below for determining X and Y.

<INPUT TEXT DATA DEFINING X AND Y>

Base Prompt

<INPUT IMAGE DATA>

You are an expert Graph Analyst. Your task is to review graphs and answer questions into a structured JSON format.

Analyze the graph provided. You must determine whether the following specific conditions are Explicitly True or False based on the graph.

Return a JSON object with the following boolean fields:

- "question_1": Is X the minimum? True if yes; False if no.
- "question_2": Is X the maximum? True if yes; False if no.
- "question_3": Is X the low median? True if yes; False if no.
- "question_4": Is X the high median? True if yes; False if no.
- "question_5": Is X less than Y? True if yes; False if no.
- "question_6": Is X greater than Y? True if yes; False if no.

<INPUT TEXT DATA DEFINING X AND Y>

Base+CoT Prompt

<INPUT IMAGE DATA>

You are an expert Graph Analyst. Your task is to read the bar chart image and answer the yes/no questions below.

Please think step-by-step. Provide your reasoning in plain text (not JSON). X and Y refer to the bars named in each question and may differ per question.

Answer all six questions:

- question_1: Is X the minimum?
- question_2: Is X the maximum?
- question_3: Is X the low median?
- question_4: Is X the high median?
- question_5: Is X less than Y?
- question_6: Is X greater than Y?

After your reasoning, conclude with your final answer wrapped exactly in <FINAL_ANSWER> and </FINAL_ANSWER> tags. Inside the tags, output a single JSON object with only these six boolean keys: "question_1", "question_2", "question_3", "question_4", "question_5", "question_6".

Do not put your step-by-step reasoning inside the JSON or inside the tags—only the JSON object belongs inside <FINAL_ANSWER>.

The specific X/Y questions for this chart follow.

<INPUT TEXT DATA DEFINING X AND Y>

PAL Prompt

You are an expert Graph Analyst. Review the graph and answer in structured JSON.

X and Y are per question. Answer all questions.

You must return:

- "chart_data": Every bar as "label": str, "value": float.
- "question_1" . . . "question_6": booleans (overwritten after PAL runs).
- "python_program": A complete Python 3 script.

PAL: Before your script runs, the runner sets 'CHART_DATA' from your 'chart_data'. Your script must use 'CHART_DATA', then end with a single 'print(json.dumps(...))' containing exactly keys 'question_1' . . . 'question_6' (all booleans). 'json' is already imported. Do not overwrite 'CHART_DATA'.

<INPUT TEXT DATA DEFINING X AND Y>