WHERE MLLMS FAIL AND WHY: CAUSAL TASK DECOMPOSITION FOR CAPABILITY FAILURE DIAGNOSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

End-to-end accuracy on compositional tasks records how often MLLMs fail, but cannot distinguish whether a failure reflects an intrinsic deficit in the targeted capability or a cascading error from an upstream prerequisite. We propose a causal decomposition framework that isolates these two failure modes through controlled interventions on the prerequisite dependencies of each task. Our capability metrics (NC, IC, RC) score each task under unassisted, correct, or incorrect prerequisites to diagnose where failures arise; contribution metrics (N-Score, S-Score), adapted from probabilities of causation, quantify each prerequisite's necessity and sufficiency to determine why. We instantiate the framework in CADET, a diagnostic benchmark of 10 composite tasks decomposed into 46 unit tasks with over 33,000 human-annotated questions spanning perception, spatial, temporal, and cognitive categories. Diagnosing frontier MLLMs with our framework uncovers systematic patterns that end-to-end accuracy obscures. Capability-wise, supplying correct prerequisites eliminates 54% of errors on cognitive tasks, lifting them from weakest to above spatial and temporal. Prerequisite-wise, causal contributions are concentrated in a few critical prerequisites, and supplying the single most important one alone captures 84% of the gain from supplying all prerequisites.

1 INTRODUCTION

Multimodal large language models (MLLMs) are increasingly evaluated on compositional tasks requiring multiple interdependent capabilities, from perceiving objects and estimating spatial layouts to tracking over time and reasoning (Yue et al., 2024; Yang et al., 2025). Such tasks are typically scored by a single end-to-end accuracy, which records how often a model fails but not where its failures originate. The score conflates an intrinsic deficit, where the model lacks the targeted capability, with a cascading error, where it possesses that capability but fails on an upstream prerequisite. In Fig 1, a model failing on multi-view sequential may lack temporal reasoning or merely misjudge the camera viewpoint on which ordering depends, yielding the same error but opposite capability diagnoses. More broadly, low accuracy on high-level cognitive tasks is naturally read as a reasoning deficit, a reading that holds only if the perceptual and spatial steps beneath it are sound, and growing evidence suggests they often are not (Fu et al., 2024; Tong et al., 2024).

A common response examines the constituent steps directly, by decomposing the task into separately scored sub-questions (Press et al., 2023; Zhou et al., 2025a) or inspecting the reasoning trace for the first erroneous step (Lightman et al., 2024). Both localize failure more finely than an aggregate score, yet observing untreated generation alone cannot disentangle the two failure modes. First, a model's generated reasoning trace or answers to isolated sub-questions need not reflect what it relies on to solve the full task (Turpin et al., 2023; Chen et al., 2025b). Second, even a faithful readout reveals only co-occurrence: a prerequisite failing alongside the final task neither shows how much final failure is attributable to it, nor answers the counterfactual had it been solved correctly. Indeed, the outcome need not hinge on the weakest prerequisite: in Fig 1, standalone accuracy is lowest for task-5, yet final performance turns on task-2. Input-level interventions do move beyond observation, supplying ground-truth intermediates (Lu et al., 2024) or injecting false premises (Guan et al., 2024), but each tests a single condition, without a dependency structure linking sub-tasks or a measure of each prerequisite's contribution. What is missing is an evaluation setting every prerequisite's state within a shared, explicit dependency structure, so that the same controlled contrasts apply across capabilities and models, and each prerequisite's contribution is measured rather than inferred.

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.

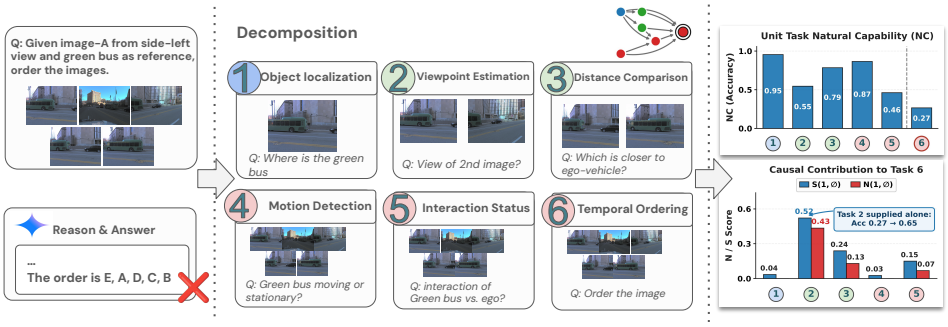

Figure 1: An example CADET task(*left*) decomposed into 6 unit tasks formalized as an SCM(*middle*). *Right*: natural capability(NC) of each unit task(*top*), and their causal contribution($N/S(1,\varnothing)$; *bottom*) to final task-6, where supplying unit task-2 alone raises final accuracy from 0.27 to 0.65.

We propose a causal decomposition framework that turns MLLM evaluation from observation into a controlled experiment. We decompose each task into unit tasks with explicit prerequisite dependencies, and intervene on each by supplying a correct answer, an incorrect answer, or nothing. To diagnose where failures arise, we define *natural*, *intrinsic*, and *resilience* capability (NC, IC, RC) under these three prerequisite states, isolating a model's performance on a target task from the influence of its modeled prerequisites. To explore why the overall outcome fails, we introduce the N-Score and S-Score, which measure each prerequisite's causal necessity and sufficiency for the final outcome: the fraction of success lost when it alone is degraded, and the fraction of failure recovered when it alone is supplied. These metrics adapt classical probabilities of causation (Tian & Pearl, 2000; Pearl, 2022) to the LLM setting without assuming monotonicity, providing valid lower bounds on the probabilities of necessity and sufficiency.

To instantiate this framework, we introduce CADET, a multimodal diagnostic benchmark in which each complex task is decomposed via an explicit structural causal model (SCM) into unit tasks that each target a single capability, and every unit-task question is annotated with both a verified correct answer and a plausible incorrect answer to enable three-state interventions. The benchmark spans 10 tasks, yielding 46 unit tasks and over 33,000 unit-task questions across four capability categories: perception, spatial, temporal, and cognitive. All questions and annotations are newly created by human annotators to mitigate data contamination and assure benchmark quality.

Evaluation on 12 frontier MLLMs exposes diagnostic patterns that end-to-end accuracy obscures. At capability level, cognitive tasks scores lowest naturally (NC$=0.41$) yet rises to second under correct prerequisites (IC$=0.72$), eliminating 54% of its natural errors; whereas spatial and temporal still retain $72-75\%$ of their error, locating the more persistent capability gaps there. Injecting incorrect prerequisites degrades all models on perception and spatial (RC$<$NC) yet improves cognitive in 10 of 12 models. Turning to causal contributions, N- and S-Scores reveal effective bottleneck sparsity, where a few critical prerequisites drive each task, and that a corrupted prerequisite costs far more than an omitted one. The single highest-S prerequisite alone captures 84% full-prerequisite gain, and rarely coincides with the lowest-NC prerequisite. Model-wise exhibits distinct diagnostic profiles, especially, the top four achieve close intrinsic capabilities (IC) despite differing in NC, and pair lower degradation loss (N) with higher supply recovery (S) than others. Together, our work moves MLLM evaluation beyond scoring how often models fail to diagnosing where and why they fail.

2 Related Work

Multimodal Evaluation and Task Decomposition. Existing benchmarks evaluate MLLMs across broad capability taxonomies (Yue et al., 2024; Liu et al., 2024; Zhou et al., 2025b), while specialized suites target individual skills such as perception and spatial (Fu et al., 2024; Yang et al., 2025) or vary visual and textual inputs to test modality reliance (Tong et al., 2024; Zhang et al., 2024). Beyond evaluating tasks as a whole, one line of work decomposes complex problems into sub-questions and compares sub-task accuracy with overall performance (Press et al., 2023; Zhou et al., 2025a). Another inspects a model's generated reasoning steps to locate where errors first occur (Lightman

et al., 2024; Zhang et al., 2025), though recent studies show that stated chains-of-thought do not always reflect the computation behind the final answer (Turpin et al., 2023; Chen et al., 2025b). Both read out the model's unassisted behavior, linking sub-task and final correctness by co-occurrence; we instead set each prerequisite to a known correct or incorrect state.

Causal Interventions for Model Diagnosis. Causal interventions are used to diagnose MLLMs and LLMs by modifying specific components at inference time to measure their effect on outputs, such as patching internal activations (Meng et al., 2022; Wu et al., 2023) or perturbing visual and textual modalities (Li et al., 2025; Hasic et al., 2026). For multi-step reasoning, recent studies apply step-level interventions to execution traces, editing intermediate outputs or re-executing trajectories to identify failure-inducing steps (Ma et al., 2026; Wang et al., 2026; Bonagiri et al., 2026; Shah, 2026). At task-input level, prior studies provide ground-truth intermediate results to isolate downstream reasoning (Lu et al., 2024; Shao et al., 2024) or inject false premises to test robustness under misleading context (Guan et al., 2024; Ming et al., 2025). These interventions either operate on model internals or single-run trajectories, or test one input condition in isolation; ours act on a task-defined causal graph shared across models.

Quantifying Prerequisite Contributions. Attributing MLLM behavior to individual components often involves distinguishing whether a factor is necessary to prevent failure or sufficient to drive success, a distinction formalized by counterfactual necessity and sufficiency (Pearl, 2022; Tian & Pearl, 2000) and adopted in explainable AI for feature-level attribution (Watson et al., 2021; Galhotra et al., 2021). In LLMs, recent studies adapt these causal concepts to test whether responses on synthetic logic problems conform to the ground-truth rules of the queries (González & Nori, 2024), or to add missing and prune redundant chain-of-thought steps (Yu et al., 2026). Both operate on the model's own outputs; we instead attribute final-task outcomes to prerequisite via necessity and sufficiency.

3 Causal Decomposition Framework

3.1 Task Decomposition via Structural Causal Models

Each evaluation task is decomposed into unit tasks testing a single constituent capability, with directed dependencies where some tasks rely on the outcomes of others. We represent these dependencies as a structural causal model (SCM) (Pearl, 2009; Peters et al., 2017), leveraging the established framework of causal interventions for the analysis below.

Definition 3.1 (Task Decomposition SCM)**.** *Consider an evaluation task decomposed into n constituent capabilities. A* task decomposition *is a structural causal model (SCM) over unit task variables $X_1, X_2 \ldots, X_n$, where X_i denotes the outcome of unit task i, given by the structural assignment:*

$$X_i \;:=\; f_i\big(\mathrm{Pa}(X_i),\; N_i\big), \quad i = 1, \ldots, n, \tag{1}$$

where f_i is the structural function determining X_i, $\mathrm{Pa}(X_i) \subseteq \{X_1, \ldots, X_n\}$ is the parents set of X_i, and N_i is a noise variable capturing unmodeled factors such as the model's sampling stochasticity.

Root nodes (those with $\mathrm{Pa}(X_i)=\varnothing$) represent foundational capabilities that can be assessed independently, while the final node $Y:=X_n$ corresponds to the overall task outcome.

3.2 Causal Intervention Design

To disentangle intrinsic capability from cascading upstream influence, and quantify each capability's causal role, we adopt the interventionist framework of causal inference (Pearl, 2022; 2009; Rubin, 1974) and introduce a ternary intervention scheme tailored to LLM evaluation.

Definition 3.2 (Ternary Intervention)**.** *For unit task X_i, we define an intervention $A_i{=}a$, $a{\in}\{1, 0, \varnothing\}$: (i) $A_i{=}1$ denotes injecting the correct answer for X_i into the prompt; (ii) $A_i{=}0$ denotes injecting an incorrect answer for X_i; (iii) $A_i{=}\varnothing$ denotes the natural state where X_i is not injected.*

Notation. All probabilities are induced by a fixed model $\mathcal{M}$, whose subscript we suppress. We write $\mathcal{A}{:=}\{1, 0, \varnothing\}$ for the ternary intervention space. For a unit task set S and $a{\in}\mathcal{A}$, we write $do(\mathbf{A}_S{=}a)$ for the homogeneous intervention placing every $X_j{\in}S$ in state a, and abbreviate $\mathbf{A}_{-i}{:=}\mathbf{A}_{\mathrm{Pa}(Y)\setminus\{X_i\}}$.

Notably, the distinction between $A_i{=}\varnothing$, $A_i{=}0$ is critical: the former preserves the model's natural execution path by evaluating it unassisted, where the latter imposes an active counterfactual intervention

testing reasoning under a false premise. Since LLM evaluation operates through input manipulation rather than passive observation, this ternary formulation emerges as the natural space of interventions, allowing us to systematically evaluate model behavior across each pairwise contrast of states.

Assumption 3.3 (Exogeneity under Intervention-Only Identification)**.** *In LLM evaluation setting, the state of a unit task $X_i \in \{1, 0\}$ admits no faithful observational readout and is identified only through the intervention $A_i{=}a$. Consequently, probabilities conditioning on $X_i{=}x$ for $x \in \{1, 0\}$ are identified with the corresponding interventional probabilities:*

$$P(\cdot \mid X_i = x) \;=\; P(\cdot \mid \mathrm{do}(A_i = x)). \tag{2}$$

Remark 3.4 (Justification). Assumption 3.3 follows from the LLM evaluation setting: the state of X_i admits no faithful observational readout (Turpin et al., 2023; Chen et al., 2025b), thus a reliable mechanism to place X_i in a determinate state $x{\in}\{1, 0\}$ is to inject the corresponding answer into the prompt, making conditioning on $X_i{=}x$ and intervening $\mathrm{do}(A_i = x)$ operationally identical. This condition coincides with the *exogeneity* assumption $P(y_x){=}P(y \mid x)$ of Pearl (2022); Tian & Pearl (2000), but with an inverted justification: classical exogeneity invokes the absence of confounders to treat observations as interventions, whereas ours follows from the absence of a faithful observational channel making intervention the reliable state-assignment mechanism.

3.3 Capability Metrics

With the ternary intervention mechanism established, we can now place the prerequisites $\mathrm{Pa}(X_i)$ of any target task X_i into controlled states and measure how the model's performance on X_i responds. By additionally evaluating under $\mathrm{do}(\mathbf{A}_{\mathrm{Pa}(X_i)}{=}1)$ and $\mathrm{do}(\mathbf{A}_{\mathrm{Pa}(X_i)}{=}0)$, we isolate the model's capability on X_i when its declared prerequisites are correctly provided versus actively corrupted.

Definition 3.5 (Causal Capability)**.** *Let X_i be any task node in the SCM with prerequisite dependencies $\mathrm{Pa}(X_i)$. For an intervention state $a \in \mathcal{A}$, the **causal capability** of $\mathcal{M}$ on X_i is*

$$\mathrm{Cap}(X_i; a) := P\big(X_i{=}1 \mid do(\mathbf{A}_{\mathrm{Pa}(X_i)}{=}a)\big), \tag{3}$$

*i.e., the probability that $\mathcal{M}$ solves X_i when all of its prerequisites are placed in state a. This yields three complementary metrics: (i) **natural capability** $\mathrm{NC}(X_i) := \mathrm{Cap}(X_i; \varnothing)$, the natural performance where no prerequisite is injected; (ii) **intrinsic capability** $\mathrm{IC}(X_i) := \mathrm{Cap}(X_i; 1)$, the performance isolated from declared prerequisites; and (iii) **resilience capability** $\mathrm{RC}(X_i) := \mathrm{Cap}(X_i; 0)$, the robustness under corrupted prerequisite context.*

For root nodes ($\mathrm{Pa}(X_i){=}\emptyset$), no upstream intervention applies and only $\mathrm{NC}(X_i)$ is defined. For non-root nodes, the three metrics disentangle a model's proficiency on X_i from the influence of modeled prerequisites. Specifically, comparing $\mathrm{IC}(X_i)$ with $\mathrm{NC}(X_i)$ separates two failure modes: a large gap $\mathrm{IC}(X_i){-}\mathrm{NC}(X_i)$ indicates that the model can solve X_i given correct prerequisites, pointing to upstream cascading errors as the primary bottleneck; a small gap with low $\mathrm{IC}(X_i)$ suggests an intrinsic deficit on X_i even when all modeled prerequisites are satisfied. Finally, comparing $\mathrm{RC}(X_i)$ with $\mathrm{NC}(X_i)$ shows whether false prerequisites mislead the model below its natural baseline, or still help guide its reasoning despite being factually wrong.

3.4 Contribution Metrics: N-Score and S-Score

Building on node-level capability diagnostics, understanding how upstream unit tasks across the SCM affect the final outcome Y requires quantifying their causal contributions. To formalize this causal attribution, we draw upon the *probabilities of causation* framework (Pearl, 2022; Tian & Pearl, 2000), which evaluates counterfactual causal influence through the Probability of Necessity (PN) and Probability of Sufficiency (PS). We do not assume monotonicity: in LLM reasoning, injecting incorrect answer ($A_i{=}0$) can help more than no injection ($A_i{=}\varnothing$), so RC may exceed NC (Sec 5.2).

Property 3.6 (Absence of Monotonicity). Under the LLM evaluation setting, the causal effect of intervening on a unit task X_i via A_i on downstream outcome task Y is not assumed to satisfy monotonicity. That is, the natural ordering below is not required to hold in general:

$$P(Y{=}1 \mid do(A_i{=}0)) \le P(Y{=}1) \le P(Y{=}1 \mid do(A_i{=}1)). \tag{4}$$

Without monotonicity, exact PN and PS values cannot be determined from interventional probabilities alone, so we introduce the **N-Score** and **S-Score** as observable lower bounds to capture the causal contribution of each prerequisite $X_i \in \mathrm{Pa}(Y)$ to the final outcome Y:

Table 1: Operational semantics of N-Score, S-Score across 4 ordered state contrasts (a, a').

Contrast	**N-Score** (Success drop from $\cdots$)	**S-Score** (Failure recovery from $\cdots$)
$(1, 0)$	corrupting X_i into a wrong answer in oracle env.	fixing X_i into the correct answer in all-wrong env.
$(1, \varnothing)$	leaving X_i unassisted in oracle env.	providing the correct X_i alone in natural env.
$(\varnothing, 0)$	injecting a wrong answer for X_i in natural env.	removing a wrong answer for X_i in all-wrong env.
$(0, \varnothing)$	removing X_i's wrong answer in all-wrong env.	injecting a wrong X_i as a scaffold in natural env.

Definition 3.7 (N-Score)**.** *Let Y be a target outcome task with prerequisite dependencies $\mathrm{Pa}(Y)$, let $X_i \in \mathrm{Pa}(Y)$, and let (a, a') be a pair of distinct states in $\mathcal{A}$. The **N-Score** of X_i on Y is*

$$\mathrm{N}_{(a,a')}(X_i, Y) := \max\left\{0,\ \frac{P\big(Y{=}1 \mid do(\mathbf{A}_{\mathrm{Pa}(Y)}{=}a)\big) - P\big(Y{=}1 \mid do(A_i{=}a', \mathbf{A}_{-i}{=}a)\big)}{P\big(Y{=}1 \mid do(\mathbf{A}_{\mathrm{Pa}(Y)}{=}a)\big)}\right\}, \quad (5)$$

measuring the fraction of Y's success under state a harmed by degrading X_i alone from a to a'.

Definition 3.8 (S-Score)**.** *Under the setting of Definition 3.7, the **S-Score** of X_i on Y is*

$$\mathrm{S}_{(a,a')}(X_i, Y) := \max\left\{0,\ \frac{P\big(Y{=}1 \mid do(A_i{=}a, \mathbf{A}_{-i}{=}a')\big) - P\big(Y{=}1 \mid do(\mathbf{A}_{\mathrm{Pa}(Y)}{=}a')\big)}{1 - P\big(Y{=}1 \mid do(\mathbf{A}_{\mathrm{Pa}(Y)}{=}a')\big)}\right\}, \quad (6)$$

measuring the fraction of Y's failure under state a' recovered by upgrading X_i alone from a' to a.

Theorem 3.9. *Under Assumption 3.3 and Property 3.6, N-Score and S-Score provide valid lower bounds on the Probability of Necessity (*PN*) and Probability of Sufficiency (*PS*).*

Specifically, N-Score quantifies *causal necessity* by measuring how much of the model's success on Y depends on X_i. A high N-Score pinpoints critical performance bottlenecks where one degraded task breaks overall success even when all other prerequisites remain at a. Conversely, S-Score quantifies *causal sufficiency* by measuring how much of the model's failure on Y can be overcome through X_i. A high S-Score identifies high-leverage prerequisites where supplying a single task rescues overall success even when other prerequisites remain at a'. Setting the other prerequisites $\mathbf{A}_{-i}$ asymmetrically to a for N-Score and a' for S-Score aligns with realistic evaluation by testing single-task vulnerability under favorable prerequisite conditions and single-task rescue under degraded ones, making the two metrics complementary rather than directly convertible (Pearl, 2022). We provide the general definitions of N-Score, S-Score under arbitrary $\mathbf{A}_{-i}$ and theoretical discussions in Appendix.

In practice, we evaluate N-Score and S-Score across four ordered state contrasts (a, a') over $\mathcal{A}{=}\{1, 0, \varnothing\}$: $(1, 0)$, $(1, \varnothing)$, $(\varnothing, 0)$, and $(0, \varnothing)$. As summarized in Tab 1, the first two contrasts measure both degradation from prerequisite corruption or omission and recovery through targeted correction or repair, whereas the latter two evaluate both directions between $\varnothing$ and 0 to capture either misinformation harm or structural scaffolding under Property 3.6. Together, our node-level capability metrics (NC, IC, RC) and edge-level contribution metrics (N-Score, S-Score) form a causal diagnostic framework for complex multimodal tasks, which we instantiate in Sec 4.

4 BENCHMARK

To apply the causal decomposition framework to frontier MLLMs and uncover where and why they fail, we construct CADET, a multimodal benchmark where complex tasks are decomposed into unit tasks connected via SCM and every prerequisite admits controlled interventions (details in Appendix).

Benchmark Overview. CADET comprises 10 multimodal evaluation tasks across over 1,100 instances, decomposed into 46 atomic unit tasks and over 33,000 unit-task questions, with both overall and their constituent unit tasks categorized into four core capabilities: perception, spatial, temporal, and cognitive. To operationalize our causal framework, each overall task is paired with an explicit SCM defining its prerequisite structure, and every unit task is annotated with both a correct and an incorrect answer, along with visual annotations on the images where applicable, to support the ternary intervention scheme ($A_i{\in}\{1, 0, \varnothing\}$) at every node. The multimodal inputs span single, multi-images and videos, while the outputs use structured formats (multiple-choice, integer, or list) for 43 of the 46 tasks and free-form text for the remaining three.

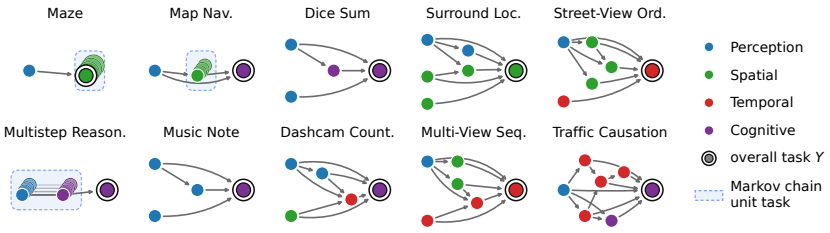

Figure 2: Overview of SCMs corresponding to 10 tasks in CADET.

Task Design and SCM Structure. Tasks are drawn primarily from real-world scenarios and designed as compositional problems requiring interdependent capabilities on which frontier MLLMs are far from saturation, so that diagnosis targets gaps these models still face. We further cover diverse input modalities, application domains, and capability compositions, from foundational perception to high-level cognitive reasoning. For each task, we formalize the prerequisite dependencies among its unit tasks into an SCM (Fig 2). These structures range from shallow aggregations (e.g., Dice Sum) to deep multi-path cascades (e.g., Traffic Causation). In decomposition, we identify unit tasks that involve iterative solving of structurally recurring sub-problems, where each step depends on the previous step's outcome and reasoning depth itself becomes a variable; we define these as **Markov chain unit tasks**. In the task-level SCM, a Markov chain unit task participates as a single node; for finer-grained analysis, we unfold it into a step-wise chain to track how capabilities evolve with reasoning depth. Tasks whose primary structure is a Markov chain are analyzed separately in Sec 5.4.

Data Construction and Quality Assurance. All evaluation content, including questions, answers, and unit-task annotations, is newly created by an experienced rater pool to mitigate data contamination; source images and videos are drawn from publicly available datasets with permissive licenses (e.g., Wilson et al. (2023)). The benchmark is constructed in two stages: raters first generate overall task questions and ground-truth answers under predefined task definitions. Based on each task's SCM, we decompose its unit tasks into standardized atomic annotation tasks, such as bounding box localization and semantic labeling. Raters provide the corresponding visual annotations and ground-truth labels, which are then converted into unit-task questions and correct answers via structured templates, ensuring consistent alignment between all evaluation items and the underlying causal structure. Incorrect answers are randomly sampled from the valid answer space rather than manually crafted, providing a standardized, model-independent perturbation. To ensure benchmark quality, each annotation across both stages is independently verified by a quality-check rater who returns deficient items for iterative revision until acceptance. Beyond per-item verification, instances flagged as ambiguous or un-annotatable are removed during final assembly to ensure reliable causal diagnosis.

5 Experiments and Findings

5.1 Experiment Setting

Evaluated Models. We evaluate 12 frontier MLLMs across six model families, covering both proprietary series: Gemini (Gemini Team, 2026), GPT (OpenAI, 2026), Grok (xAI, 2025); and open-weight series: Gemma (DeepMind, 2026), Qwen (Qwen Team, 2026a), Kimi (Team et al., 2026). This selection encompasses different model families at varying capability scales, enabling both cross-family and within-family diagnostic comparisons. All models are queried through public APIs with default parameter setting. Thinking mode is set to high or left at its default if not-configurable.

Intervention Implementation. To operationalize the ternary intervention, for each prerequisite unit task X_i, we inject its intervened state ($A_i \in \{1, 0\}$) into prompt as a structured JSON question-answer pair. For $A_i=1$, alongside the correct textual answer, we render corresponding visual annotations (e.g., bounding box, path) whenever the prerequisite's ground truth directly represents spatial information in the scene. For $A_i=0$, incorrect answer is injected via text without modifying the image, to prevent shifting the visual referent of the query and ensure we test reasoning under a false premise about the intended target. Under natural state $A_i=\varnothing$, no prerequisite information is provided for X_i.

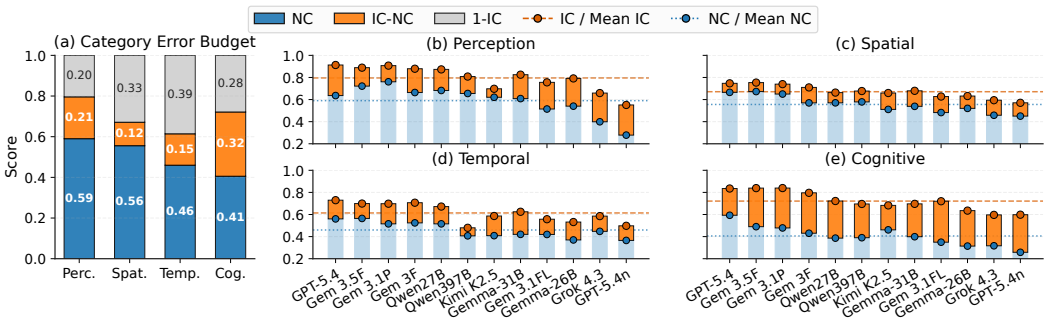

Figure 3: Natural(NC) and intrinsic(IC) capabilities on non-root unit tasks: *(a)* category-level NC and IC averaged over 12 models; *(b–e)* per-model NC and IC within each category.

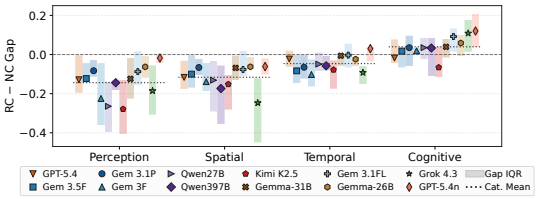

Figure 4: Per-model RC−NC gap across categories on non-root unit tasks.

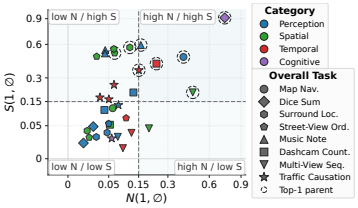

Figure 5: Prerequisite causal contributions $N/S(1, \varnothing)$ to overall task Y.

Evaluation Protocol. We report the capability metrics (NC, IC, RC) for each unit task, and the contribution metrics (N-Score, S-Score) on the overall task across four state contrasts: $(1, 0)$, $(1, \varnothing)$, $(\varnothing, 0)$, and $(0, \varnothing)$. Model predictions are evaluated via exact match for structured outputs, with Gemini 3.6 Flash serving as LLM judger (Zheng et al., 2023) for free-form text tasks and format mismatches. Full experimental settings are provided in Appendix.

5.2 Capability Diagnostics

Cascading errors reorder the apparent capability ranking. Under end-to-end evaluation (NC), accuracy falls across perception, spatial, temporal, and cognitive tasks (0.59, 0.56, 0.46, 0.41; Fig 3(a)), a pattern that would attribute the largest deficits to high-level reasoning. Supplying correct prerequisites (IC) reorders the categories: cognitive reasoning moves to second (0.72), above spatial (0.67) and temporal (0.61), behind only perception (0.80). Decomposing the total error into a cascading component IC−NC and a residual component 1−IC accounts for this shift, as cascading errors constitute 54% of cognitive failures (0.32 of 0.59). Spatial and temporal reasoning show the opposite composition, with residual errors (0.33, 0.39) covering 75% and 72% of their failures and small gains from correct prerequisites (0.12, 0.15), locating the more persistent intrinsic capability gaps at these two categories rather than at cognition.

Frontier models share similar intrinsic ceilings but differ in cascading errors (Fig 3(b-e)). Supplying correct prerequisites (IC) brings the top 4 SOTA models to nearly identical ceilings across categories (0.88-0.91 in perception, 0.80-0.83 in cognition), tracing their end-to-end (NC) gaps to cascading errors (IC−NC): GPT-5.4 and Gemini family trade places, with larger cascading loss for GPT-5.4 in perception (0.28 vs. 0.15-0.22) and for Gemini in cognition (0.35-0.37 vs. 0.24). Across all 12 models, perception exhibits the widest spread in residual errors (1−IC: 0.09–0.45), spatial shows uniformly high residual errors (0.25-0.43), and cognitive carries uniformly large cascading errors (IC−NC: 0.22-0.37). Model-wise, close NC scores mask divergent error compositions: Qwen3.5 397B and Kimi K2.5 match neighboring models in NC yet carry substantially higher residual errors (1−IC) in different categories.

Incorrect prerequisites harm perception and spatial but mildly aid cognition. The RC−NC gap rises across the hierarchy (Fig 4): all 12 models fall below zero in perception and spatial, while 10 of 12 turn positive in cognition. The gap also varies widely across models within each category

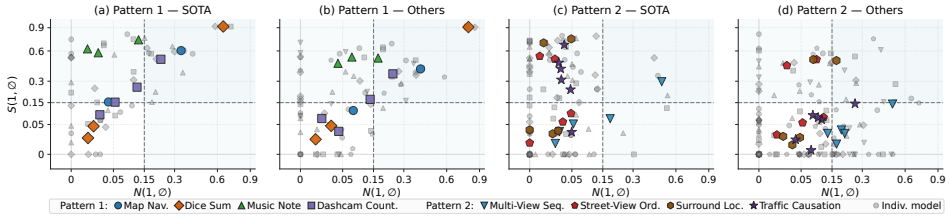

Figure 6: $N/S(1,\varnothing)$ across Pattern 1(*a,b*) and Pattern 2 (*c,d*) tasks for top-4 SOTA (*a,c*) and remaining 8 Others (*b,d*) models ($sqrt(\cdot)$-scaled axes, similar in Fig. 5).

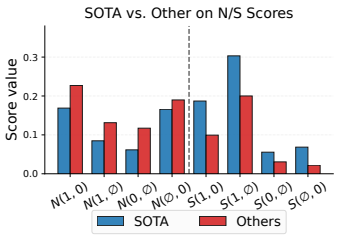

Figure 7: N/S-Scores across the four state contrasts for SOTA vs. Others models.

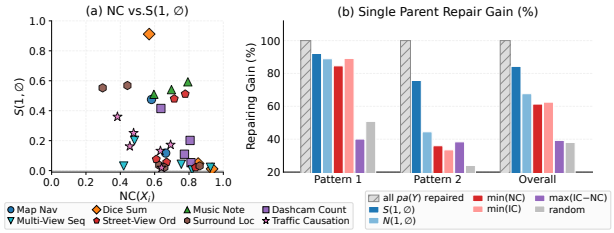

Figure 8: *(a)* $\mathrm{NC}(X_i)$ vs. $S(1,\varnothing)$; *(b)* percentage of full all-prerequisite gain(IC−NC) recovered by single-parent supply across selection criteria.

(-0.28 to $+0.12$). The positive cognitive gaps directly instantiate the non-monotonicity formalized in Property 3.6: incorrect prerequisites do not uniformly degrade performance.

5.3 Causal Contribution Analysis

Causal contributions exhibit effective bottleneck sparsity. Evaluating prerequisite contributions via $N(1,\varnothing)$ and $S(1,\varnothing)$ (Fig 5) across 8 non-Markov tasks shows that causal influence is highly uneven across parents, with each task driven by one or a few critical prerequisites. These critical parents act either as core bottlenecks that cause large oracle drops when omitted and strong natural recovery when provided (high N/high S), or as compensable levers whose omission is offset by other oracle prerequisites yet still rescue natural failures alone (low N/high S). By contrast, the remaining prerequisites have individually small effects (low N/low S), while conjunctive bottlenecks that hurt oracle performance without aiding natural recovery are nearly absent (high N/low S).

Disaggregating by model reveals two distinct criticality patterns. Expanding to individual models (Fig 6) reveals two task patterns and a broad split between top-4 (SOTA) and remaining (Others) models. Under Pattern 1 (Fig 6a,b), both groups exhibit critical prerequisites with high single-point recovery capacity (high S), with Others shifting moderately toward higher N and lower S. Under Pattern 2 (Fig 6c,d), SOTA keeps prerequisites at high S with low oracle omission sensitivity (low N), whereas Others shift substantially toward higher N and lower S, losing effective single-point recoverability while suffering larger oracle drops from single-task omission. Across both patterns, SOTA maintains low oracle vulnerability and often has multiple high-S prerequisites per task that each could rescue natural failures alone; while Others drift rightward and downward, moderately in Pattern 1 and severely in Pattern 2, reflecting higher oracle sensitivity and weaker single-task recovery. Structurally, the prerequisite count $|\mathrm{Pa}(Y)|$ averages 3.0 in Pattern 1 and 5.25 in Pattern 2.

A single supplied prerequisites recovers most full-prerequisite gains. Supplying only the single most sufficient prerequisite ($\max S(1,\varnothing)$) recovers 84% of the full all-prerequisite gain (IC−NC) overall (Fig 8b), showing that one targeted intervention already accounts for most of the attainable gain. This leverage is not readable from standalone accuracy: $S(1,\varnothing)$ does not track $\mathrm{NC}(X_i)$ across prerequisites (Fig 8a). To test whether cheaper node-level signals could substitute for $S(1,\varnothing)$ in identify the same prerequisite, we repeat single-parent repair under alternative criteria (including the weakest prerequisite $\min(\mathrm{NC})$; Fig 8b), none of which recovers this single-parent optimum overall. The shortfall of NC stays around 8 pp in Pattern 1 tasks, where models share a single dominant

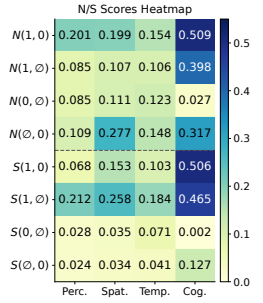

Figure 9: Category-level prerequisite N/S-Scores across 4 state contrasts.

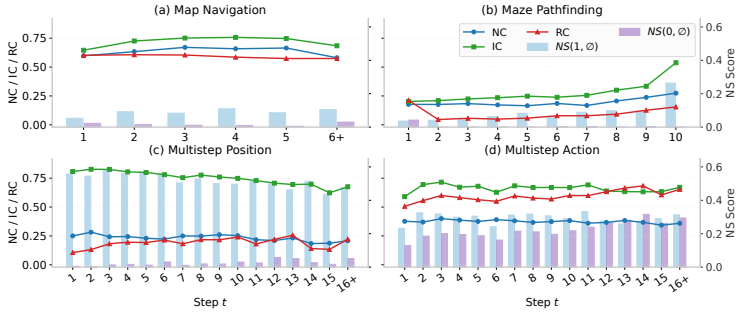

Figure 10: Step-wise capabilities (NC, IC, RC) and causal contributions ($NS(1,\varnothing)$, $NS(0,\varnothing)$) along step t across Markov chain unit tasks.

bottleneck; but widens sharply in Pattern 2 tasks with denser multi-parent dependencies, where $S(1,\varnothing)$-chosen parent sustains 76% of the full gain while NC-based only 36%.

Incorrect prerequisites hurt far more than omission. Extending to full ternary contrast space (Fig 7,9), injecting incorrect prerequisites is consistently more destructive than leaving them unassisted ($N(1,0)>N(1,\varnothing)$ across all categories), with minor gains from incorrect answers on temporal ($S(0,\varnothing)$=0.071). Unlike cognitive target tasks benefiting from incorrect prerequisites (Sec 5.2), the two cognitive prerequisites are highly sensitive to correctness: corrupting or omitting them causes the steepest overall task drops ($N(1,0), N(1,\varnothing)$) with no gain from incorrect answers ($S(0,\varnothing)$), while fixing or removing them drives strong recovery ($S(1,0), S(\varnothing,0)$). SOTA models (Fig 7) pair uniformly lower N with higher S than Others across all four contrasts, showing stronger robustness to single-prerequisite degradation with superior single-point recovery even under corruption.

5.4 Markov Chain Unit Task Analysis

We analyze the Markov chain unit tasks (Sec 4), in which each X_t depends solely on its predecessor X_{t-1}. Under this single-parent structure, N, S-Score converge into a single NS-Score ($NS(a,a')$=$N(a,a')\cdot\mathrm{Cap}(X_t;a)$; Appendix). Fig 10 reports NC, IC, RC and NS score along step t. First, capability degrades nonlinearly with a single-step effective horizon: in goal-directed tasks, IC and $NS(1,\varnothing)$ remain flat until one step from the exit (Fig 10b, t=10); symmetrically, in process-cumulative tasks where NC collapses to a flat floor immediately at t=1, IC and $NS(1,\varnothing)$ peak at the first step (Fig 10c) and gradually decline with depth. Second, alternating Position (Fig 10c) and Action (Fig. 10d) nodes within the same chain respond divergently to incorrect prerequisites: $NS(0,\varnothing)\approx 0$ for spatial tracking versus substantial for action planning, consistent with the finding in Sec 5.2 that incorrect prerequisites benefit cognitive reasoning but not perception. Third, as depth t grows, Position continues to require factual correctness ($NS(0,\varnothing) \leq 0.06 \ll \mathrm{NS}(1,\varnothing)$), whereas in cognitive Action, $NS(0,\varnothing)$ steadily rises from 0.12 (t=1) to 0.30 ($t\geq$14) to match $\mathrm{NS}(1,\varnothing)$ (RC≈IC), showing that explicit prerequisite conditioning, regardless of correctness, increasingly accounts for cognitive recovery at deeper horizons.

6 Conclusion

We introduced a causal decomposition framework and the CADET benchmark, applying controlled ternary interventions over explicit prerequisite structures to diagnose where and why MLLMs fail. Our capability metrics (NC, IC, RC) separate cascading influence from intrinsic deficits, showing that supplying correct prerequisites substantially lifts cognitive tasks while spatial and temporal reasoning retain persistent intrinsic gaps, and that incorrect prerequisites often still aid cognition. The N-, S-Scores reveal effective bottleneck sparsity, where a few critical prerequisites drive each task, and supplying the single most sufficient one recovers most full-prerequisite gain yet rarely coincides with the least accurate prerequisite. Across models, top ones exhibit lower degradation alongside stronger single-point recovery. More broadly, our diagnostic framework can inform future improvements, from inference-time CoT guidance, query rewriting to training and multi-step system design.

AI USE STATEMENT

The research ideas, the causal decomposition framework, and all theoretical content of this paper (Section 3; Appendix A), including the definitions of the capability metrics, the N-Score and S-Score, the assumptions, and the statements and derivations of all theorems, were conceived and derived by hand by the authors. We used generative AI tools to transcribe photographs of these handwritten derivations into LaTeX, to add explanatory prose between derivation steps, and to suggest presentational refinements to notation. The mathematical content of every step comes from the authors' handwriting derivations, all of which the authors checked against the originals. Generative AI was not used for brainstorming, formulating hypotheses, or designing experiments. All final benchmark questions, answers, and visual annotations were written and verified by human raters (Section 4; Appendix B.3). At an early stage of annotation, raters were offered a small pool of AI-generated example questions as optional starting points, for raters to choose from (optionally) and rewrite; incorrect answers used in interventions are selected from each task's answer set rather than generated by AI. The core experimental logic code, including the implementation of the N-Score and S-Score, the dependency logic, was written by the authors and validated against manual spreadsheet calculations. Generative AI assisted with result-processing and plotting code, which the authors reviewed, and was used as an aid in examining and interpreting experimental results; every reported finding was verified by the authors against the raw results. The illustrative figure (Figure 1) in the paper was drawn by the authors without AI. Gemini 3.6 Flash serves as an LLM judge for those text-format tasks and for format mismatches; all other outputs are scored by exact match (Section 5.1). We also used generative AI tools to search for and summarize related literature, with every cited work read and verified by the authors; and to draft, polish, and condense passages from the authors' own content to improve readability and meet the page limit. We take full responsibility for the final content of this paper, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To support reproducibility, we document our framework, benchmark, and evaluation. The causal decomposition framework, including the SCM formulation, the intervention design, the capability metrics, and the N-Score and S-Score, is formally defined in Section 3. All assumptions are stated explicitly, and complete proofs are given in Appendix A, including a step-by-step instantiation of the derivation for our multi-parent setting (Appendix A.2). All reported metrics are computed directly from model accuracies under the specified intervention conditions, using the formulas in Section 3.4 and Appendix A.1. The benchmark's composition, per-task SCMs, and statistics are described in Section 4 and Appendix B.1, and its design rationale in Appendix B.2. The full annotation, quality-control, and filtering protocol, including how the injected correct and incorrect prerequisite answers are constructed, is given in Appendix B.3. All source media are drawn from publicly available datasets with permissive licenses, listed in Table 5. Appendix C specifies all evaluated models, their access routes and inference settings, and the LLM-judge protocol, and Figures 11 and 12 provide the complete prompt templates for interventions and judging. We will publicly release the full benchmark (questions, answers, visual annotations, SCMs, and intervention inputs, with source media redistributed under their original licenses), together with the complete evaluation and metric-computation code.

REFERENCES

Alexander Balke and Judea Pearl. Counterfactual probabilities: Computational methods, bounds and applications. In *Uncertainty in artificial intelligence*, pp. 46–54. Elsevier, 1994.

Akash Bonagiri, Devang Borkar, Gerard Janno Anderias, Setareh Rafatirad, and Houman Homayoun. Causalflow: Causal attribution and counterfactual repair for llm agent failures. *arXiv preprint arXiv:2605.25338*, 2026.

Jiacheng Chen, Tianhao Liang, Sherman Siu, Zhengqing Wang, Kai Wang, Yubo Wang, Yuansheng Ni, Ziyan Jiang, Wang Zhu, Bohan Lyu, et al. Mega-bench: Scaling multimodal evaluation to over 500 real-world tasks. In *International Conference on Learning Representations*, volume 2025, pp. 28384–28494, 2025a.

Liang Chen, Weichu Xie, Yiyan Liang, Hongfeng He, Hans Zhao, Zhibo Yang, Zhiqi Huang, Haoning Wu, Haoyu Lu, Yiping Bao, et al. Babyvision: Visual reasoning beyond language. *arXiv preprint arXiv:2601.06521*, 2026.

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, et al. Reasoning models don't always say what they think. *arXiv preprint arXiv:2505.05410*, 2025b.

Google DeepMind. Gemma 4: Our most intelligent open models to date. https://blog.google/innovation-and-ai/technology/developers-tools/gemma-4/, 2026. Accessed: 2026-05-01.

Maurice Fréchet. Généralisation du théoreme des probabilités totales. *Fundamenta mathematicae*, 25:379–387, 1935.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pp. 148–166. Springer, 2024.

Sainyam Galhotra, Romila Pradhan, and Babak Salimi. Explaining black-box algorithms using probabilistic contrastive counterfactuals. In *Proceedings of the 2021 international conference on management of data*, pp. 577–590, 2021.

Gemini Team. Gemini 3.1 pro: A smarter model for your most complex tasks. 2026. URL https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/.

Gemini Team and Google DeepMind. Gemini 3 flash: Frontier intelligence built for speed. Technical blog / model release, Google, 12 2025. URL https://blog.google/products-and-platforms/products/gemini/gemini-3-flash/.

Gemini Team and Google DeepMind. Gemini 3.1 flash-lite: Built for intelligence at scale. Technical blog / model release, Google, 3 2026. URL https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-flash-lite/.

Anurag Ghosh, Shen Zheng, Robert Tamburo, Khiem Vuong, Juan Alvarez-Padilla, Hailiang Zhu, Michael Cardei, Nicholas Dunn, Christoph Mertz, and Srinivasa G Narasimhan. Roadwork: A dataset and benchmark for learning to recognize, observe, analyze and drive through work zones. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6132–6142. IEEE, 2025.

Javier González and Aditya V Nori. Does reasoning emerge? examining the probabilities of causation in large language models. *Advances in Neural Information Processing Systems*, 37:117737–117761, 2024.

Google DeepMind. Gemini 3.5 flash: Model card, 2026a. URL https://deepmind.google/models/model-cards/gemini-3-5-flash/. Accessed: 2026-08-18.

Google DeepMind. Gemini 3.6 flash: Model card, 2026b. URL https://deepmind.google/models/model-cards/gemini-3-6-flash/.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14375–14385. IEEE, 2024.

Yunzhuo Hao, Jiawei Gu, Huichen Will Wang, Linjie Li, Zhengyuan Yang, Lijuan Wang, and Yu Cheng. Can mllms reason in multimodality? emma: An enhanced multimodal reasoning benchmark. *arXiv preprint arXiv:2501.05444*, 2025.

Vahidin Hasic, Chao Wang, Luis C Garcia-Peraza-Herrera, David Watson, and Senka Krivic. Which modality decides? counterfactual modality attribution for multimodal llms. *arXiv preprint arXiv:2608.00076*, 2026.

Li Li, Jiashu Qu, Linxin Song, Yuxiao Zhou, Yuehan Qin, Tiankai Yang, and Yue Zhao. Treble counterfactual vlms: A causal approach to hallucination. Association for Computational Linguistics, 2025.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations*, volume 2024, pp. 23439–23554, 2024.

Ming Ma, Jue Zhang, Fangkai Yang, Yu Kang, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. Dover: Intervention-driven auto debugging for llm multi-agent systems. In *International Conference on Learning Representations*, volume 2026, pp. 28059–28087, 2026.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

Yifei Ming, Senthil Purushwalkam, Shrey Pandit, Zixuan Ke, Xuan-Phi Nguyen, Caiming Xiong, and Shafiq Joty. Faitheval: Can your language model stay faithful to context, even if" the moon is made of marshmallows". In *International Conference on Learning Representations*, volume 2025, pp. 29430–29456, 2025.

OpenAI. Introducing GPT-5.4, 2026. URL https://openai.com/index/introducing-gpt-5-4/.

Judea Pearl. Causality: Models, reasoning, and inference. 2009.

Judea Pearl. Probabilities of causation: three counterfactual interpretations and their identification. In *Probabilistic and causal inference: the works of Judea Pearl*, pp. 317–372. 2022.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT press, 2017.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, 2023.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026a. URL https://qwen.ai/blog?id=qwen3.5.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, April 2026b. URL https://qwen.ai/blog?id=qwen3.6-27b.

Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

Jaineet Shah. Causal agent replay: Counterfactual attribution for llm-agent failures. *arXiv preprint arXiv:2606.08275*, 2026.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 37:8612–8642, 2024.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, et al. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026.

Jin Tian and Judea Pearl. Probabilities of causation: Bounds and identification. *Annals of Mathematics and Artificial Intelligence*, 28(1):287–313, 2000.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9568–9578. IEEE, 2024.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965, 2023.

Yawen Wang, Wenjie Wu, Junjie Wang, and Qing Wang. From flat logs to causal graphs: Hierarchical failure attribution for llm-based multi-agent systems. *arXiv preprint arXiv:2602.23701*, 2026.

David S Watson, Limor Gultchin, Ankur Taly, and Luciano Floridi. Local explanations via necessity and sufficiency: Unifying theory and practice. In *Uncertainty in Artificial Intelligence*, pp. 1382–1392. PMLR, 2021.

Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting, 2023. URL https://arxiv.org/abs/2301.00493.

Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah Goodman. Interpretability at scale: Identifying causal mechanisms in alpaca. *Advances in neural information processing systems*, 36:78205–78226, 2023.

xAI. Grok-4 model card. 2025. URL https://x.ai/news/grok-4.

Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10632–10643. IEEE, 2025.

Xiangning Yu, Zhuohan Wang, Linyi Yang, Haoxuan Li, Anjie Liu, Xiao Xue, Jun Wang, and Mengyue Yang. Causal sufficiency and necessity improves chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 38:126109–126141, 2026.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9556–9567, 2024.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pp. 169–186. Springer, 2024.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, et al. Which agent causes task failures and when? on automated failure attribution of llm multi-agent systems. *arXiv preprint arXiv:2505.00212*, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Qiji Zhou, YiFan Gong, Guangsheng Bao, Hongjie Qiu, Jinqiang Li, Xiangrong Zhu, Huajian Zhang, and Yue Zhang. Reasoning is all you need for video generalization: A counterfactual benchmark with sub-question evaluation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 2939–2957, 2025a.

Zihao Zhou, Shudong Liu, Maizhen Ning, Wei Liu, Jindong Wang, Derek Wong, Xiaowei Huang, Qiufeng Wang, and Kaizhu Huang. Is your model really a good math reasoner? evaluating mathematical reasoning with checklist. In *International Conference on Learning Representations*, volume 2025, pp. 34238–34281, 2025b.

A Details and Proofs of Causal Decomposition Framework

In this section, we provide the general definitions of N-Score and S-Score under arbitrary prerequisite states, followed by the theoretical proofs and properties of our causal decomposition framework.

The do-operator. We adopt the standard do-operator (Section 3.2; Pearl (2009)). For $a \in \{1, 0\}$, $do(A_i{=}a)$ replaces the structural assignment of X_i in equation 1 with $X_i := a$ while keeping all other assignments unchanged, and $do(A_i{=}\varnothing)$ leaves the assignment of X_i intact; interventions on multiple unit tasks apply these operations jointly. $P(\cdot \mid do(\cdot))$ denotes the distribution induced by the resulting SCM.

A.1 General Definition of N-Score, S-Score

In Section 3.4 (Definitions 3.7 and 3.8), when quantifying the causal contribution of a prerequisite unit task $X_i \in \mathrm{Pa}(Y)$ to the target outcome Y across an ordered contrast (a, a'), we specifically set the remaining prerequisites $\mathbf{A}_{-i} := \mathbf{A}_{\mathrm{Pa}(Y)\setminus\{X_i\}}$ to the homogeneous state a for N-Score and a' for S-Score. Here, we generalize this formulation by allowing the remaining prerequisites to take any arbitrary intervention state configuration.

Specifically, let $\mathbf{c} \in \mathcal{A}^{|\mathrm{Pa}(Y)\setminus\{X_i\}|}$ denote an arbitrary intervention state configuration assigned to $\mathbf{A}_{-i}$, denoted as $do(\mathbf{A}_{-i} = \mathbf{c})$, which sets the intervention state of each prerequisite $X_j \in \mathrm{Pa}(Y)\setminus\{X_i\}$ to $A_j = c_j$ with $c_j \in \mathcal{A}$. Following the notation in Section 3.2, when all unit tasks in $\mathbf{A}_{-i}$ are placed in a single homogeneous state $c \in \mathcal{A}$ (i.e., $c_j = c$ for all $X_j \in \mathrm{Pa}(Y)\setminus\{X_i\}$), we write $do(\mathbf{A}_{-i} = c)$ without boldface. Based on this general state configuration $\mathbf{c}$ for the remaining prerequisites, we define the general N-Score and S-Score as follows.

Definition A.1 (General N-Score)**.** *Let Y be a target outcome task with prerequisite dependencies $\mathrm{Pa}(Y)$, let $X_i \in \mathrm{Pa}(Y)$, and let (a, a') be an ordered pair of distinct states in $\mathcal{A}$. For any intervention state configuration $\mathbf{c} \in \mathcal{A}^{|\mathrm{Pa}(Y)\setminus\{X_i\}|}$ assigned to the remaining prerequisites $\mathbf{A}_{-i}$, the general N-Score of X_i on Y is defined as:*

$$\mathrm{N}_{(a,a')}(X_i, Y \mid \mathbf{c}) := \max\left\{0,\ \frac{P\big(Y{=}1 \mid do(A_i{=}a, \mathbf{A}_{-i}{=}\mathbf{c})\big) - P\big(Y{=}1 \mid do(A_i{=}a', \mathbf{A}_{-i}{=}\mathbf{c})\big)}{P\big(Y{=}1 \mid do(A_i{=}a, \mathbf{A}_{-i}{=}\mathbf{c})\big)}\right\}, \quad (7)$$

measuring the fraction of Y's success under $do(A_i{=}a, \mathbf{A}_{-i}{=}\mathbf{c})$ that is lost when X_i alone is degraded from a to a'.

Definition A.2 (General S-Score)**.** *Under the setting of Definition A.1, for any intervention state configuration $\mathbf{c} \in \mathcal{A}^{|\mathrm{Pa}(Y)\setminus\{X_i\}|}$ assigned to $\mathbf{A}_{-i}$, the general S-Score of X_i on Y is defined as:*

$$\mathrm{S}_{(a,a')}(X_i, Y \mid \mathbf{c}) := \max\left\{0,\ \frac{P\big(Y{=}1 \mid do(A_i{=}a, \mathbf{A}_{-i}{=}\mathbf{c})\big) - P\big(Y{=}1 \mid do(A_i{=}a', \mathbf{A}_{-i}{=}\mathbf{c})\big)}{1 - P\big(Y{=}1 \mid do(A_i{=}a', \mathbf{A}_{-i}{=}\mathbf{c})\big)}\right\}, \quad (8)$$

measuring the fraction of Y's failure under $do(A_i{=}a', \mathbf{A}_{-i}{=}\mathbf{c})$ that is recovered when X_i alone is upgraded from a' to a.

Under these general definitions, the N-Score (Definition 3.7) and S-Score (Definition 3.8) introduced in Section 3.4 correspond to the special cases where every element of $\mathbf{c}$ is set to the homogeneous state $c_j = a$ (i.e., $do(\mathbf{A}_{-i} = a)$) for N-Score and $c_j = a'$ (i.e., $do(\mathbf{A}_{-i} = a')$) for S-Score, respectively.

A.2 Lower Bound Guarantee of N-Score and S-Score

This section provides the formal statement and proof of Theorem 3.9 in Section 3.4. We establish the lower-bound guarantee for the general N-Score and S-Score (Definitions A.1 and A.2) under an arbitrary prerequisite configuration $\mathbf{c}$ (Theorems A.3 and A.4), where Theorem 3.9 corresponds to the special cases $\mathbf{c} = a$ for N-Score and $\mathbf{c} = a'$ for S-Score.

Theorem A.3. *Under Assumption 3.3 (Exogeneity), for any prerequisite unit task $X_i \in \mathrm{Pa}(Y)$, any ordered contrast (a, a') of distinct states in $\mathcal{A}$, and any intervention state configuration $\mathbf{c} \in \mathcal{A}^{|\mathrm{Pa}(Y)\setminus\{X_i\}|}$ assigned to the remaining prerequisites $\mathbf{A}_{-i}$, the general N-Score (Definition A.1) provides a valid lower bound on the Probability of Necessity (*PN*) under prerequisite context $do(\mathbf{A}_{-i}{=}\mathbf{c})$:*

$$\mathrm{PN}_{(a,a')}(X_i, Y \mid \mathbf{c}) \geq \mathrm{N}_{(a,a')}(X_i, Y \mid \mathbf{c}) \quad (9)$$

In particular, setting every element of $\mathbf{c}$ *to the homogeneous state* a *(i.e.,* $do(\mathbf{A}_{-i}{=}a)$*) directly establishes the N-Score lower bound in Theorem 3.9.*

Proof. Note that under exogeneity and non-monotonicity, the mathematical technique of bounding probabilities of causation via Fréchet inequalities (Fréchet, 1935) is established in the foundational works of Pearl (2022); Tian & Pearl (2000); Pearl (2009); Balke & Pearl (1994). Here, we adapt and instantiate their derivation step-by-step to demonstrate how it applies to our multi-parent LLM evaluation setting under prerequisite configuration $do(\mathbf{A}_{-i}{=}\mathbf{c})$, Assumption 3.3, and Property 3.6.

For notational conciseness under the prerequisite intervention $do(\mathbf{A}_{-i}{=}\mathbf{c})$, in below proof, let $P_{\mathbf{c}}(\cdot) := P(\cdot \mid do(\mathbf{A}_{-i}{=}\mathbf{c}))$ denote the probability measure in the induced submodel $\mathcal{M}_{\mathbf{c}}$ (Pearl, 2009), where Y_a and $Y_{a'}$ denote the potential outcomes of Y under further regime interventions $do(A_i{=}a)$ and $do(A_i{=}a')$ within $\mathcal{M}_{\mathbf{c}}$. Following Pearl (2022), the Probability of Necessity (PN) across contrast (a, a') under context $\mathbf{c}$ is defined as the counterfactual probability that Y fails under $do(A_i{=}a')$, given that Y succeeded under regime $A_i{=}a$:

$$\mathrm{PN}_{(a,a')}(X_i, Y \mid \mathbf{c}) := P_{\mathbf{c}}(Y_{a'}{=}0 \mid A_i{=}a,\, Y{=}1)$$

If $P_{\mathbf{c}}(A_i{=}a,\, Y{=}1) = 0$, the success baseline is zero and the bound holds trivially by convention ($\mathrm{N}_{(a,a')} = 0$). Thus, assuming $P_{\mathbf{c}}(A_i{=}a,\, Y{=}1) > 0$, we first apply the definition of conditional probability:

$$P_{\mathbf{c}}(Y_{a'}{=}0 \mid A_i{=}a,\, Y{=}1) = \frac{P_{\mathbf{c}}(Y_{a'}{=}0,\, A_i{=}a,\, Y{=}1)}{P_{\mathbf{c}}(A_i{=}a,\, Y{=}1)}$$

By the consistency property of structural causal models, the observed outcome under regime $A_i{=}a$ coincides with the potential outcome Y_a. Substituting $Y = Y_a$ into the numerator yields:

$$P_{\mathbf{c}}(Y_{a'}{=}0 \mid A_i{=}a,\, Y{=}1) = \frac{P_{\mathbf{c}}(Y_{a'}{=}0,\, A_i{=}a,\, Y_a{=}1)}{P_{\mathbf{c}}(A_i{=}a,\, Y{=}1)}$$

Next, we factor out the marginal regime probability $P_{\mathbf{c}}(A_i{=}a)$ in the numerator and divide both the numerator and the denominator by it:

$$\begin{aligned} P_{\mathbf{c}}(Y_{a'}{=}0 \mid A_i{=}a,\, Y{=}1) &= \frac{P_{\mathbf{c}}(Y_{a'}{=}0,\, Y_a{=}1 \mid A_i{=}a)\, P_{\mathbf{c}}(A_i{=}a)}{P_{\mathbf{c}}(A_i{=}a,\, Y{=}1)} \\ &= \frac{P_{\mathbf{c}}(Y_{a'}{=}0,\, Y_a{=}1 \mid A_i{=}a)}{P_{\mathbf{c}}(Y{=}1 \mid A_i{=}a)} \end{aligned}$$

Under the exogeneity of the regime assignment (Assumption 3.3), the potential outcomes $(Y_a, Y_{a'})$ are independent of A_i, and conditioning on A_i is operationally identical to intervening on it. This removes the conditioning event in both the numerator and the denominator:

$$P_{\mathbf{c}}(Y_{a'}{=}0 \mid A_i{=}a,\, Y{=}1) = \frac{P_{\mathbf{c}}(Y_{a'}{=}0,\, Y_a{=}1)}{P_{\mathbf{c}}(Y_a{=}1)}$$

Since we do not assume monotonicity (Property 3.6) in LLM reasoning, the joint counterfactual probability $P_{\mathbf{c}}(Y_{a'}{=}0,\, Y_a{=}1)$ cannot be point-identified from marginal probabilities alone. Applying the classical Fréchet lower bound $P_{\mathbf{c}}(E_1, E_2) \geq \max\{0,\, P_{\mathbf{c}}(E_1) + P_{\mathbf{c}}(E_2) - 1\}$ (Pearl, 2022) to the numerator gives:

$$\mathrm{PN}_{(a,a')}(X_i, Y \mid \mathbf{c}) \geq \max\left\{0,\, \frac{P_{\mathbf{c}}(Y_{a'}{=}0) + P_{\mathbf{c}}(Y_a{=}1) - 1}{P_{\mathbf{c}}(Y_a{=}1)}\right\}$$

Using the complement rule $P_{\mathbf{c}}(Y_{a'}{=}0) = 1 - P_{\mathbf{c}}(Y_{a'}{=}1)$ gives:

$$\mathrm{PN}_{(a,a')}(X_i, Y \mid \mathbf{c}) \geq \max\left\{0,\, \frac{P_{\mathbf{c}}(Y_a{=}1) - P_{\mathbf{c}}(Y_{a'}{=}1)}{P_{\mathbf{c}}(Y_a{=}1)}\right\}$$

Finally, unpacking the submodel notation back into full interventional probabilities via $P_{\mathbf{c}}(Y_a{=}1) = P(Y{=}1 \mid do(A_i{=}a,\, \mathbf{A}_{-i}{=}\mathbf{c}))$ and $P_{\mathbf{c}}(Y_{a'}{=}1) = P(Y{=}1 \mid do(A_i{=}a',\, \mathbf{A}_{-i}{=}\mathbf{c}))$ yields:

$$\mathrm{PN}_{(a,a')}(X_i, Y \mid \mathbf{c}) \geq \max\left\{0,\, \frac{P(Y{=}1 \mid do(A_i{=}a,\, \mathbf{A}_{-i}{=}\mathbf{c})) - P(Y{=}1 \mid do(A_i{=}a',\, \mathbf{A}_{-i}{=}\mathbf{c}))}{P(Y{=}1 \mid do(A_i{=}a,\, \mathbf{A}_{-i}{=}\mathbf{c}))}\right\}$$

Thus

$$\mathrm{PN}_{(a,a')}(X_i, Y \mid \mathbf{c}) \geq \mathrm{N}_{(a,a')}(X_i, Y \mid \mathbf{c}).$$

□

Theorem A.4. *Under Assumption 3.3 (Exogeneity), for any prerequisite unit task $X_i \in \mathrm{Pa}(Y)$, any ordered contrast (a, a') of distinct states in $\mathcal{A}$, and any intervention state configuration $\mathbf{c} \in \mathcal{A}^{|\mathrm{Pa}(Y)\setminus\{X_i\}|}$ assigned to the remaining prerequisites $\mathbf{A}_{-i}$, the general S-Score (Definition A.2) provides a valid lower bound on the Probability of Sufficiency (*PS*) under prerequisite context $do(\mathbf{A}_{-i}{=}\mathbf{c})$:*

$$\mathrm{PS}_{(a,a')}(X_i, Y \mid \mathbf{c}) \geq \mathrm{S}_{(a,a')}(X_i, Y \mid \mathbf{c}) \tag{10}$$

In particular, setting every element of $\mathbf{c}$ to the homogeneous state a' (i.e., $do(\mathbf{A}_{-i}{=}a')$) directly establishes the S-Score lower bound in Theorem 3.9.

Proof. Similar to the proof of Theorem A.3, let $P_{\mathbf{c}}(\cdot) := P(\cdot \mid do(\mathbf{A}_{-i}{=}\mathbf{c}))$ denote the probability measure in the submodel $\mathcal{M}_{\mathbf{c}}$, where Y_a and $Y_{a'}$ represent the potential outcomes of Y under $do(A_i{=}a)$ and $do(A_i{=}a')$. Following Pearl (2022), the Probability of Sufficiency (PS) across contrast (a, a') under context $\mathbf{c}$ measures the counterfactual probability that upgrading X_i to state a would rescue Y to success, given that Y failed under regime $A_i{=}a'$:

$$\mathrm{PS}_{(a,a')}(X_i, Y \mid \mathbf{c}) := P_{\mathbf{c}}(Y_a{=}1 \mid A_i{=}a',\, Y{=}0)$$

If $P_{\mathbf{c}}(A_i{=}a',\, Y{=}0) = 0$, the failure baseline is zero and the bound holds trivially by convention ($\mathrm{S}_{(a,a')} = 0$). Thus, assuming $P_{\mathbf{c}}(A_i{=}a',\, Y{=}0) > 0$, we expand this conditional counterfactual probability using the definition of conditional probability:

$$P_{\mathbf{c}}(Y_a{=}1 \mid A_i{=}a',\, Y{=}0) = \frac{P_{\mathbf{c}}(Y_a{=}1,\, A_i{=}a',\, Y{=}0)}{P_{\mathbf{c}}(A_i{=}a',\, Y{=}0)}$$

By consistency, the observed outcome under regime $A_i{=}a'$ coincides with the potential outcome $Y_{a'}$, so we replace the observed failure $Y{=}0$ in the numerator with $Y_{a'}{=}0$:

$$P_{\mathbf{c}}(Y_a{=}1 \mid A_i{=}a',\, Y{=}0) = \frac{P_{\mathbf{c}}(Y_a{=}1,\, A_i{=}a',\, Y_{a'}{=}0)}{P_{\mathbf{c}}(A_i{=}a',\, Y{=}0)}$$

Factoring out $P_{\mathbf{c}}(A_i{=}a')$ in the numerator and dividing both the numerator and the denominator by it yields:

$$\begin{aligned} P_{\mathbf{c}}(Y_a{=}1 \mid A_i{=}a',\, Y{=}0) &= \frac{P_{\mathbf{c}}(Y_a{=}1,\, Y_{a'}{=}0 \mid A_i{=}a')\, P_{\mathbf{c}}(A_i{=}a')}{P_{\mathbf{c}}(A_i{=}a',\, Y{=}0)} \\ &= \frac{P_{\mathbf{c}}(Y_a{=}1,\, Y_{a'}{=}0 \mid A_i{=}a')}{P_{\mathbf{c}}(Y{=}0 \mid A_i{=}a')} \end{aligned}$$

Under the exogeneity of the regime assignment (Assumption 3.3), the potential outcomes $(Y_a, Y_{a'})$ are independent of A_i, and conditioning on A_i is operationally identical to intervening on it. This removes the conditioning event in both the numerator and the denominator:

$$P_{\mathbf{c}}(Y_a{=}1 \mid A_i{=}a',\, Y{=}0) = \frac{P_{\mathbf{c}}(Y_a{=}1,\, Y_{a'}{=}0)}{1 - P_{\mathbf{c}}(Y_{a'}{=}1)}$$

Since monotonicity is not assumed (Property 3.6), the joint counterfactual probability $P_{\mathbf{c}}(Y_a{=}1,\, Y_{a'}{=}0)$ is again not point-identified. Applying the Fréchet lower bound $P_{\mathbf{c}}(Y_a{=}1,\, Y_{a'}{=}0) \geq \max\{0,\, P_{\mathbf{c}}(Y_a{=}1) + P_{\mathbf{c}}(Y_{a'}{=}0) - 1\}$ (Pearl, 2022) to the numerator yields:

$$\mathrm{PS}_{(a,a')}(X_i, Y \mid \mathbf{c}) \geq \max\left\{0,\, \frac{P_{\mathbf{c}}(Y_a{=}1) + P_{\mathbf{c}}(Y_{a'}{=}0) - 1}{1 - P_{\mathbf{c}}(Y_{a'}{=}1)}\right\}$$

Substituting $P_{\mathbf{c}}(Y_{a'}{=}0) = 1 - P_{\mathbf{c}}(Y_{a'}{=}1)$ into the numerator simplifies the fraction to:

$$\mathrm{PS}_{(a,a')}(X_i, Y \mid \mathbf{c}) \geq \max\left\{0,\, \frac{P_{\mathbf{c}}(Y_a{=}1) - P_{\mathbf{c}}(Y_{a'}{=}1)}{1 - P_{\mathbf{c}}(Y_{a'}{=}1)}\right\}$$

Finally, unpacking the submodel notation $P_{\mathbf{c}}(Y_a{=}1) = P(Y{=}1 \mid do(A_i{=}a,\, \mathbf{A}_{-i}{=}\mathbf{c}))$ and $P_{\mathbf{c}}(Y_{a'}{=}1) = P(Y{=}1 \mid do(A_i{=}a',\, \mathbf{A}_{-i}{=}\mathbf{c}))$ establishes the general S-Score lower bound:

$$\mathrm{PS}_{(a,a')}(X_i, Y \mid \mathbf{c}) \geq \max\left\{0,\, \frac{P(Y{=}1 \mid do(A_i{=}a,\, \mathbf{A}_{-i}{=}\mathbf{c})) - P(Y{=}1 \mid do(A_i{=}a',\, \mathbf{A}_{-i}{=}\mathbf{c}))}{1 - P(Y{=}1 \mid do(A_i{=}a',\, \mathbf{A}_{-i}{=}\mathbf{c}))}\right\}$$

Thus

$$\mathrm{PS}_{(a,a')}(X_i, Y \mid \mathbf{c}) \geq \mathrm{S}_{(a,a')}(X_i, Y \mid \mathbf{c}).$$

□

A.3 THEORETICAL PROPERTIES

In this section, we discuss two key theoretical properties of our contribution metrics that directly underpin our experimental design and diagnostic analysis in the main text: (i) the exact identification of PN and PS under monotonicity versus our conservative lower bounds under LLM non-monotonicity, and (ii) the rationale for our asymmetric prerequisite control scheme over $\mathbf{A}_{-i}$.

A.3.1 EXACT IDENTIFICATION UNDER MONOTONICITY

In the probabilities of causation framework (Pearl, 2022; Tian & Pearl, 2000), when the monotonicity assumption holds, the Fréchet lower bounds in Theorems A.3 and A.4 tighten to exact equality.

Proposition A.5 (Exact Identification under Monotonicity; Tian & Pearl, 2000)**.** *Under the setting of Theorems A.3 and A.4, if Y is monotonic with respect to X_i across contrast (a, a') under context $\mathbf{c}$ (i.e., $P_{\mathbf{c}}(Y_a = 0, Y_{a'} = 1) = 0$), then N-Score and S-Score equal PN and PS.*

For the detailed proof, we refer the reader to Tian & Pearl (2000). In LLM evaluation, however, we do not impose the monotonicity assumption (Property 3.6), as we discussed in Section 3. Meanwhile, as our experiments in Section 5.2 show, injecting an incorrect answer ($A_i = 0$) can provide a reasoning scaffold that guides downstream deduction better than providing no information ($A_i = \varnothing$), causing $\mathrm{RC}(X_i)$ to exceed $\mathrm{NC}(X_i)$.

A.3.2 REMAINING PREREQUISITE $\mathbf{A}_{-i}$ CONTROLLING

Controlling for Remaining Prerequisites $\mathbf{A}_{-i}$. In a multi-parent SCM, the causal effect of a unit task X_i on the final outcome Y is also influenced by the states of the remaining prerequisites $\mathbf{A}_{-i}{:=}\mathbf{A}_{\mathrm{Pa}(Y)\setminus\{X_i\}}$. When $\mathbf{A}_{-i}$ is left un-intervened in the natural state $\varnothing$, the actual realizations of these prerequisites remain unobserved and depend on each model's upstream proficiency, which conflates X_i's causal effect with model-specific upstream variations. To isolate X_i's contribution under controlled conditions, our general definitions (Definitions A.1 and A.2) allow $\mathbf{A}_{-i}$ to take any intervention state configuration $\mathbf{c} \in \mathcal{A}^{|\mathrm{Pa}(Y)|-1}$. Because each prerequisite takes one of three states in $\mathcal{A}{=}\{1, 0, \varnothing\}$, the full configuration space contains $3^{|\mathrm{Pa}(Y)|-1}$ possible combinations. To evaluate X_i across representative prerequisite contexts without enumerating this exponential space, we restrict $\mathbf{A}_{-i}$ to homogeneous configurations where all remaining prerequisites share the same state $c \in \{1, 0, \varnothing\}$. This reduction yields three interpretable background environments: the oracle environment ($\mathbf{A}_{-i}{=}1$), the corrupted environment ($\mathbf{A}_{-i}{=}0$), and the natural environment ($\mathbf{A}_{-i}{=}\varnothing$).

Asymmetric Prerequisite Control in N-Score and S-Score. To capture complementary diagnostic properties across these homogeneous environments, our main-text definitions of N-Score and S-Score (Definitions 3.7 and 3.8) assign asymmetric background states to $\mathbf{A}_{-i}$ for any ordered contrast (a, a'). Specifically, N-Score sets $\mathbf{A}_{-i}{=}a$, whereas S-Score sets $\mathbf{A}_{-i}{=}a'$. This design aligns each metric with its intended diagnostic role in LLM evaluation. For N-Score, conditioning on $\mathbf{A}_{-i}{=}a$ establishes a baseline environment where all prerequisites begin at state a, allowing the metric to measure single-task vulnerability when X_i alone degrades to a'. For S-Score, conditioning on $\mathbf{A}_{-i}{=}a'$ establishes a baseline environment where all prerequisites begin at state a', allowing the metric to measure single-task recovery when X_i alone upgrades to a. Furthermore, combining this asymmetric control with our ordered contrasts $(1, 0)$, $(1, \varnothing)$, $(\varnothing, 0)$, and $(0, \varnothing)$ systematically covers all three homogeneous environments $\mathbf{A}_{-i} \in \{1, 0, \varnothing\}$ and assigns each contrast a distinct operational meaning. For instance, under contrast $(1, \varnothing)$, $\mathrm{S}_{(1,\varnothing)}$ evaluates how providing the correct answer for X_i alone improves overall task accuracy in the natural environment ($\mathbf{A}_{-i}{=}\varnothing$), directly identifying high-leverage targets for single-task repair. Under the same contrast, $\mathrm{N}_{(1,\varnothing)}$ measures how leaving X_i unassisted reduces accuracy in the oracle environment ($\mathbf{A}_{-i}{=}1$). Similarly, contrasts $(1, 0)$, $(\varnothing, 0)$, and $(0, \varnothing)$ quantify the impact of prerequisite corruption and structural scaffolding across oracle, natural, and corrupted environments.

Algebraic Coupling under Identical Prerequisite Control. Under our general definitions of N-Score and S-Score (Definitions A.1 and A.2), when both metrics are evaluated under the same prerequisite configuration $\mathbf{c} \in \mathcal{A}^{|\mathrm{Pa}(Y)|-1}$, they share a common interventional background $\mathrm{do}(\mathbf{A}_{-i}{=}\mathbf{c})$. Within this shared environment, necessity and sufficiency are algebraically coupled through the Probability of Necessity and Sufficiency (PNS) (Pearl, 2022; Tian & Pearl, 2000).

Proposition A.6 (Algebraic Coupling under Identical Context)**.** *For any prerequisite unit task $X_i \in \mathrm{Pa}(Y)$, any ordered contrast (a, a') of distinct states in $\mathcal{A}$, and any fixed prerequisite configuration $\mathbf{c} \in \mathcal{A}^{|\mathrm{Pa}(Y)|-1}$, let $p_{a|\mathbf{c}} := P(Y{=}1 \mid \mathrm{do}(A_i{=}a, \mathbf{A}_{-i}{=}\mathbf{c}))$ and $p_{a'|\mathbf{c}} := P(Y{=}1 \mid \mathrm{do}(A_i{=}a', \mathbf{A}_{-i}{=}\mathbf{c}))$. Denote the conditional Probability of Necessity and Sufficiency under context $\mathbf{c}$ as $\mathrm{PNS}_{(a,a')}(X_i, Y \mid \mathbf{c}) := P(Y_a{=}1, Y_{a'}{=}0 \mid \mathrm{do}(\mathbf{A}_{-i}{=}\mathbf{c}))$. Under exogeneity assumption (Assumption 3.3), the exact counterfactual probabilities satisfy:*

$$\mathrm{PN}_{(a,a')}(X_i, Y \mid \mathbf{c}) \cdot p_{a|\mathbf{c}} = \mathrm{PS}_{(a,a')}(X_i, Y \mid \mathbf{c}) \cdot (1 - p_{a'|\mathbf{c}}) = \mathrm{PNS}_{(a,a')}(X_i, Y \mid \mathbf{c}), \tag{11}$$

and the general N-Score and S-Score satisfy the parallel lower-bound coupling:

$$\mathrm{N}_{(a,a')}(X_i, Y \mid \mathbf{c}) \cdot p_{a|\mathbf{c}} = \mathrm{S}_{(a,a')}(X_i, Y \mid \mathbf{c}) \cdot (1 - p_{a'|\mathbf{c}}) = \max\left\{0,\ p_{a|\mathbf{c}} - p_{a'|\mathbf{c}}\right\} \le \mathrm{PNS}_{(a,a')}(X_i, Y \mid \mathbf{c}). \tag{12}$$

Proof. Let $P_{\mathbf{c}}(\cdot) := P(\cdot \mid \mathrm{do}(\mathbf{A}_{-i}{=}\mathbf{c}))$ denote the probability measure in the submodel induced by $\mathrm{do}(\mathbf{A}_{-i}{=}\mathbf{c})$ (Pearl, 2009). Under exogeneity ssumption, conditioning on A_i within this submodel is operationally identical to intervening on it, so $P_{\mathbf{c}}(Y_a{=}1) = p_{a|\mathbf{c}}$ and $P_{\mathbf{c}}(Y_{a'}{=}0) = 1 - p_{a'|\mathbf{c}}$. By consistency and the definition of conditional probability, $\mathrm{PN}_{(a,a')}(X_i, Y \mid \mathbf{c}) \cdot p_{a|\mathbf{c}} = P_{\mathbf{c}}(Y_{a'}{=}0 \mid A_i{=}a, Y{=}1) \cdot P_{\mathbf{c}}(Y{=}1 \mid A_i{=}a) = P_{\mathbf{c}}(Y_a{=}1, Y_{a'}{=}0) = \mathrm{PNS}_{(a,a')}(X_i, Y \mid \mathbf{c})$. An analogous expansion yields $\mathrm{PS}_{(a,a')}(X_i, Y \mid \mathbf{c}) \cdot (1 - p_{a'|\mathbf{c}}) = \mathrm{PNS}_{(a,a')}(X_i, Y \mid \mathbf{c})$, establishing Equation equation 11.

Next, multiplying Definition A.1 by $p_{a|\mathbf{c}}$ and Definition A.2 by $1 - p_{a'|\mathbf{c}}$ directly gives $\max\{0, p_{a|\mathbf{c}} - p_{a'|\mathbf{c}}\}$. Applying the classical Fréchet inequality yields $\mathrm{PNS}_{(a,a')}(X_i, Y \mid \mathbf{c}) \ge \max\{0, p_{a|\mathbf{c}} + (1 - p_{a'|\mathbf{c}}) - 1\} = \max\{0, p_{a|\mathbf{c}} - p_{a'|\mathbf{c}}\}$, which establishes Equation 12. □

Metric Independence under Asymmetric Control. Proposition A.6 shows that necessity and sufficiency are algebraically convertible when evaluated within the same background environment $\mathbf{c}$. However, note that in our main-text formulation (Definitions 3.7 and 3.8), N-Score and S-Score evaluate X_i across two distinct background environments: N-Score conditions on $\mathbf{A}_{-i}{=}a$, whereas S-Score conditions on $\mathbf{A}_{-i}{=}a'$. Whenever Y depends on multiple prerequisites ($|\mathrm{Pa}(Y)| > 1$), the contrast states $a \neq a'$ place the remaining prerequisites into two different interventional configurations. Because the two metrics operate under distinct prerequisite states, their numerators correspond to the Fréchet lower bounds of two different conditional PNS quantities, $\mathrm{PNS}_{(a,a')}(X_i, Y \mid a)$ and $\mathrm{PNS}_{(a,a')}(X_i, Y \mid a')$, where in general $p_{a|a} - p_{a'|a} \neq p_{a|a'} - p_{a'|a'}$. Consequently, our main-text N-Score and S-Score cannot be algebraically derived from one another through marginal outcome probabilities. A special case arises when $|\mathrm{Pa}(Y)|{=}1$, representing the single-parent setting where $\mathbf{A}_{-i}$ is empty; in this case, both metrics operate within the same unconditional environment and reduce directly to the classical coupled relationship in Equation 12. For general multi-parent tasks, this LLM-specific decoupling ensures that N-Score and S-Score serve as independent diagnostic dimensions, allowing each paired (a, a') score to provide distinct and practical diagnostic value for pinpointing failure bottlenecks versus high-leverage repair targets.

A.4 Markov Chain Structure

In Section 4, we introduced Markov chain unit tasks to represent sub-problems that require iterative solving across sequential steps. In the task-level SCM (Figure 2), a Markov chain unit task is represented as a single composite node that receives inputs from upstream prerequisites and feeds into downstream tasks. To analyze how model capabilities and error propagation evolve over long reasoning horizons, we unfold this composite node into an explicit sequential chain of unit tasks (Section 5.4). Below, we formalize the Markov chain structure within a task decomposition SCM and derive its step-wise causal properties.

Definition A.7 (Markov Chain Structure in Task Decomposition SCM)**.** *Consider a task decomposition SCM (Definition 3.1). A Markov chain structure is a recursive sub-model region defined over T sequential reasoning steps ($t = 1, \ldots, T$), where each step consists of a recurring cycle of p constituent unit tasks ($p \ge 1$), represented by unit task variables $X_1^{(t)}, X_2^{(t)}, \ldots, X_p^{(t)}$. Each unit task variable $X_i^{(t)}$ is given by the structural assignment:*

$$X_i^{(t)} := f_i^{(t)}\Big(\mathrm{Pa}\big(X_i^{(t)}\big),\ N_i^{(t)}\Big), \quad \textit{for } i = 1, \ldots, p,\ \ t = 1, \ldots, T, \tag{13}$$

Table 2: Overview of the 10 overall evaluation tasks in CADET.

Task Name	Modality	Category	# Instances	Answer Format
Music Note Counting	Image	Perception	114	Integer
Dice Sum	Image	Perception	130	Integer
Map Navigation	Image	Spatial	107	Integer
Maze	Image	Spatial	80	Multiple-Choice
Surround Localization	Multi-Image	Spatial	154	Integer
Multi-View Sequencing	Multi-Image	Temporal	154	Letter List
Street-View Ordering	Multi-Image	Temporal	91	Letter List
Dashcam Counting	Video	Cognitive	79	Integer
Traffic Causation	Video	Cognitive	107	Free-form Text
Multistep Reasoning	Image	Cognitive	128	Free-form Text

where $f_i^{(t)}$ is the structural function determining $X_i^{(t)}$, $N_i^{(t)}$ is the exogenous noise variable, and the parent set $\mathrm{Pa}\big(X_i^{(t)}\big)$ is uniquely defined by:

$$\mathrm{Pa}\big(X_i^{(t)}\big) = \begin{cases} \big\{X_{i-1}^{(t)}\big\}, & \text{if } i \in \{2,\dots,p\}, \\ \big\{X_p^{(t-1)}\big\}, & \text{if } i = 1 \text{ and } t \in \{2,\dots,T\}, \end{cases} \tag{14}$$

with the initial prerequisite set $\mathrm{Pa}\big(X_1^{(1)}\big)$ given by the upstream prerequisites external to the Markov chain region in the SCM (or $\emptyset$ if the chain originates at a root node). Unfolding this recursive dependency across all T steps yields the explicit sequential causal chain:

$$X_1^{(1)} \to X_2^{(1)} \to \cdots \to X_p^{(1)} \;\to\; X_1^{(2)} \to \cdots \to X_p^{(T-1)} \;\to\; X_1^{(T)} \to \cdots \to X_p^{(T)}. \tag{15}$$

Remark A.8. In the unfolded Markov chain structure (Definition A.7), every non-initial unit task variable X_j has a single causal prerequisite, i.e., $|\mathrm{Pa}(X_j)| = 1$. Consequently, for any such unit task X_j and its unique direct parent $X_{\mathrm{prev}} \in \mathrm{Pa}(X_j)$, the set of remaining prerequisites is empty:

$$\mathbf{A}_{-\mathrm{prev}} := \mathbf{A}_{\mathrm{Pa}(X_j)\setminus\{X_{\mathrm{prev}}\}} = \emptyset. \tag{16}$$

Under this single-parent setting, $\mathbf{c} \in \mathcal{A}^{|\mathrm{Pa}(X_j)\setminus\{X_{\mathrm{prev}}\}|}$ in Definitions A.1 and A.2 becomes $\emptyset$, and the N-Score and S-Score simplify to:

$$\begin{aligned} \mathrm{N}_{(a,a')}(X_{\mathrm{prev}}, X_j) &= \max\left\{0,\ \frac{P\big(X_j = 1 \mid do(A_{\mathrm{prev}} = a)\big) - P\big(X_j = 1 \mid do(A_{\mathrm{prev}} = a')\big)}{P\big(X_j = 1 \mid do(A_{\mathrm{prev}} = a)\big)}\right\}, \\ \mathrm{S}_{(a,a')}(X_{\mathrm{prev}}, X_j) &= \max\left\{0,\ \frac{P\big(X_j = 1 \mid do(A_{\mathrm{prev}} = a)\big) - P\big(X_j = 1 \mid do(A_{\mathrm{prev}} = a')\big)}{1 - P\big(X_j = 1 \mid do(A_{\mathrm{prev}} = a')\big)}\right\}. \end{aligned} \tag{17}$$

As an immediate consequence, Proposition A.6 directly holds without requiring external context conditioning:

$$\mathrm{N}_{(a,a')}(X_{\mathrm{prev}}, X_j) \cdot p_a = \mathrm{S}_{(a,a')}(X_{\mathrm{prev}}, X_j) \cdot (1 - p_{a'}) = \max\{0,\ p_a - p_{a'}\}, \tag{18}$$

where $p_a = P\big(X_j = 1 \mid do(A_{\mathrm{prev}} = a)\big)$, $p_{a'} = P\big(X_j = 1 \mid do(A_{\mathrm{prev}} = a')$, and $\max\{0,\ p_a - p_{a'}\}$ is the NS score reported in Sec. 5.4, denoting lower bound of PNS.

In our CADET benchmark, three tasks contain Markov chain structures with cycle lengths $p = 1$ (Maze, Map Navigation) and $p = 2$ (Multistep vision reasoning) (Figure 2), corresponding to the step-wise evaluations in Sec. 5.4 (Figure 10).

B Details of Benchmark

B.1 Benchmark Composition

B.1.1 Task Inventory and Statistics

This section reports the benchmark details following our Section 4. Table 2 lists the 10 overall evaluation tasks, and Table 3 reports the overview and structure of their task-level SCMs, reported depth is the length of the longest directed path terminating at Y.

Table 3: Structure of the task-level SCM of each evaluation task. – on multistep reasoning is due to it start from a Markov chain structure.

Task	#Units	#Roots	Depth	$\lvert\mathrm{Pa}(Y)\rvert$	Category	#Questions	Has Markov
Music Note Counting	4	2	2	3	P/C	1,163	No
Dice Sum	4	2	2	3	P/C	1,919	No
Map Navigation	3	1	2	2	P/S/C	918	Yes
Maze	2	1	1	1	P/S	880	Yes
Surround Localization	6	3	2	5	P/S	4,928	No
Multi-View Sequencing	6	2	3	5	P/S/T	4,158	No
Street-View Ordering	6	2	3	5	P/S/T	3,268	No
Dashcam Counting	5	2	3	4	P/S/T/C	6,895	No
Traffic Causation	7	1	4	6	P/T/C	2,974	No
Multistep Reasoning	3	–	2	2	P/C	6,828	Yes

B.1.2 Capability Categories

We group unit tasks into four categories according to the kind of information the task requires the model to produce. Each unit task is assigned to the category of the capability it primarily targets.

- **Perception**: reading out the presence, identity, category, or image position of an object from the visual signal, without relating separate elements to one another.
- **Spatial**: relating elements to each other or to the observer, covering distance, size, orientation, viewpoint, and navigation.
- **Temporal**: relating observations across frames or views, covering motion, ordering, and localization in time.
- **Cognitive**: operating on already extracted facts, covering counting, arithmetic, causal attribution, and multi-step or goal-directed reasoning.

The overall task of each SCM is labelled with the same four categories (Table 2). Across the 46 unit tasks, the four categories contain 15, 12, 10, and 9 tasks respectively.

B.1.3 Per-Task SCM and Unit Task

Table 4 specifies the unit tasks and SCM structure of every evaluation task, giving for each node its capability, category, answer format, number of questions, and parent set. Nodes are indexed in a topological order of the task-level SCM. As a result, CADET contains over 33,000 unit task level questions.

B.2 Benchmark Design Rationale

Task Selection. The 10 evaluation tasks in CADET are selected based on three considerations. First, we focus on problems where current frontier MLLMs remain far from saturated under end-to-end evaluation, so that the diagnosis concerns capability gaps that current models still struggle with. Second, we select compositional tasks that require multiple interdependent capabilities to reach the final answer, so that each task admits a meaningful decomposition into unit tasks. Third, we prioritize real-world visual scenarios across single-image, multi-image, and video modalities, and the resulting 46 unit tasks cover perception, spatial, temporal, and cognitive capabilities.

Prerequisite Identification and SCM Structure. Each unit task corresponds to a single capability. A capability is included in the decomposition of an evaluation task when it is a required information for answering the overall task question. For example, to tell the number on the top face of a dice, one needs to know where the dice is and which face is its top face; to tell the direction of a pedestrian relative to the ego vehicle, one needs to locate the pedestrian and determine the viewpoint of the camera that observes it. We specify a prerequisite dependency from X_j to X_i when solving X_i requires the information or the capability represented by X_j. The same task structure is used for all evaluated models. We discuss the completeness of the specified SCMs in Appendix E.

Unit-Task Question Granularity. As shown in Tables 2 and 4, the number of questions of a unit task is not necessarily equal to the number of overall task instances. When an overall task is built on

Table 4: Unit task specification of the 10 evaluation tasks in CADET. Y denotes the overall task outcome; ⟳ marks a Markov chain unit task.

Node	Unit Task	Cat.	Answer Format	#Q	Parents
Music Note Counting					
X_1	Object Identification	P	Multiple-choice	821	—
X_2	Object Classification	P	Multiple-choice	114	X_1
X_3	Completeness Verification	P	Multiple-choice	114	—
Y	Object Counting	C	Integer	114	X_1, X_2, X_3
Dice Sum					
X_1	Object Detection	P	Integer	130	—
X_2	3D Surface Recognition	P	Multiple-choice	791	—
X_3	Fine-grained Counting	C	Integer	868	X_2
Y	Arithmetic Summation	C	Integer	130	X_1, X_2, X_3
Map Navigation					
X_1	Object Localization	P	Multiple-choice	443	—
X_2	Sequential Navigation ⟳	S	Multiple-choice	368	X_1
Y	Path-Based Counting	C	Integer	107	X_1, X_2
Maze					
X_1	Object Localization	P	Multiple-choice	80	—
Y	Sequential Navigation ⟳	S	Multiple-choice	800	X_1
Surround Localization					
X_1	Object Identification	P	Multiple-choice	1,078	—
X_2	Front-View Identification	S	Multiple-choice	1,232	—
X_3	Viewpoint Estimation	S	Multiple-choice	2,156	—
X_4	Camera View Identification	S	Multiple-choice	154	X_1, X_2
X_5	Object Localization	P	Multiple-choice	154	X_1
Y	Clock Position Estimation	S	Integer	154	$X_1, \ldots, X_5$
Multi-View Sequencing					
X_1	Object Localization	P	Multiple-choice	770	—
X_2	Viewpoint Estimation	S	Multiple-choice	1,386	X_1
X_3	Distance Comparison	S	Multiple-choice	1,540	X_1
X_4	Motion Detection	T	Multiple-choice	154	—
X_5	Interaction Status Identification	T	Multiple-choice	154	X_1, X_3, X_4
Y	Temporal Ordering	T	Letter list	154	$X_1, \ldots, X_5$
Street-View Ordering					
X_1	Object Localization	P	Multiple-choice	455	—
X_2	Size Comparison	S	Multiple-choice	897	X_1
X_3	Distance Comparison	S	Multiple-choice	910	X_1, X_2
X_4	Horizontal Position Comparison	S	Multiple-choice	824	X_1
X_5	Motion Detection	T	Multiple-choice	91	—
Y	Temporal Ordering	T	Letter list	91	$X_1, \ldots, X_5$
Dashcam Counting					
X_1	Object Classification	P	Multiple-choice	2,860	—
X_2	Spatial Relation	S	Multiple-choice	2,237	—
X_3	Object Re-identification	P	Multiple-choice	1,378	X_1
X_4	Motion Classification	T	Multiple-choice	341	X_1, X_2, X_3
Y	Temporal Object Counting	C	Integer	79	$X_1, \ldots, X_4$
Traffic Causation					
X_1	Object Localization	P	Multiple-choice	2,290	—
X_2	State Change Recognition	T	Multiple-choice	200	X_1
X_3	Reference-Based Action Recognition	T	Multiple-choice	153	X_1
X_4	Temporal Localization	T	Multiple-choice	93	X_2, X_3
X_5	Temporal Sequencing	T	Multiple-choice	105	X_4
X_6	False Correlation Rejection	C	Multiple-choice	26	X_1, X_3
Y	Causal Reasoning	C	Free-form text	107	$X_1, \ldots, X_6$
Multistep Reasoning					
X_1	Position Tracking ⟳	P	Integer list	3,350	X_2
X_2	Action Prediction ⟳	C	Free-form text	3,350	X_1
Y	Multi-step Reasoning	C	Free-form text	128	X_1, X_2

Table 5: Source media used in CADET.

Source	Source Media Used	Reference	License
Argoverse 2	Urban driving scenes: multi-camera ring imagery and ego-view video	Wilson et al. (2023)	CC BY-NC-SA 4.0
ROADWork	Roadwork and construction-zone driving images and videos	Ghosh et al. (2025)	ODC-By v1.0
BabyVision	Language-independent core visual reasoning images	Chen et al. (2026)	MIT
MEGA-Bench	Real-world multimodal task images	Chen et al. (2025a)	Apache-2.0
EMMA	Scientific and mathematical reasoning figures	Hao et al. (2025)	Apache-2.0
Mutopia Project	Engraved sheet-music scores	Mutopia Website	CC BY licensed*
Wikimedia Commons	Sheet-music and general-purpose images	Wikimedia Website	Common License*
Procedurally generated	Synthetic mazes, navigation maps, dice renderings	–	Ours (1P)

Source URLs. Argoverse 2: `https://www.argoverse.org/av2.html`; ROADWork: `https://github.com/anuragxel/roadwork-dataset`; BabyVision: `https://huggingface.co/datasets/UnipatAI/BabyVision`; MEGA-Bench: `https://github.com/TIGER-AI-Lab/MEGA-Bench`; EMMA: `https://github.com/EMMA-Bench/EMMA`; Mutopia: `https://www.mutopiaproject.org/`; Wikimedia Commons: `https://commons.wikimedia.org/`.

* The Mutopia Project and Wikimedia Commons both host media under a mixture of licenses that are declared individually for each file. For these two sources we applied the site-level CC BY license filter when browsing and retained only those images distributed under a CC BY license.

multiple images, video frames, or target objects, the capability of a unit task does not necessarily apply at the same granularity, and querying all of them within a single bundled question would turn the unit task into a multi-target aggregation problem. In such cases, we formulate separate questions at the granularity at which the capability applies. For example, when a task instance contains five images, object identification is asked separately for each image; when a task instance contains several target objects, object localization is asked separately for each object.

B.3 Benchmark Construction

B.3.1 Image/Video Source and License

As described in Section 4, while all evaluation questions, task decompositions, and unit-task annotations in CADET are newly created to mitigate data contamination, the underlying source images and videos are curated from publicly available datasets and open repositories with permissive licenses. Table 5 summarizes all source media (image and video) used in CADET, along with their corresponding descriptions, references, and license terms.

B.3.2 Data Construction and Quality Assurance

Two-Stage Construction and Closed-Form Annotation. We construct CADET in two stages following predefined task structures. In the first stage, for each of the 10 overall tasks, we define the task scope, a question-answer template, and detailed task instructions on the selected source images and videos. Raters follow these instructions and templates to write the overall task question and its ground-truth answer. In the second stage, to decompose each overall task into unit tasks without introducing noise from free-form question writing, we design the decomposition step as closed-form annotation and labeling tasks. Depending on the unit task, raters draw bounding boxes or paths on the images and provide labels that follow strict format specifications. Once these annotations are collected, we extract the labeled information and convert it into unit-task questions, correct answers ($A_i = 1$), and rendered visual annotations through structured templates. For the incorrect answers ($A_i = 0$), structured unit tasks use random sampling over plausible candidates; specifically, for numerical tasks, we sample within a bounded local neighborhood around the ground truth (e.g.,

within ± 2 of a groundtruth value of 5) rather than drawing distant values (e.g., 20) that models would trivially rule out. For the remaining three free-form text tasks where random sampling is not applicable, raters write the incorrect answers following instructions designed to minimize subjective bias.

Two-Rater Iterative Verification. All annotations are conducted by a pool of over 130 experienced multimodal raters. Across both stages, every instance is processed by two raters in sequence: the first rater completes the annotation, and a second quality-check rater independently inspects the result. If the quality-check rater identifies an inaccurate answer, an imprecise bounding box or path, or a label that violates the required format specification, the instance is returned to the first rater for revision. This review cycle repeats until the quality-check rater accepts the annotation.

Instance Revision and Corner-Case Filtering. At every annotation step, raters can flag issues with a candidate instance. When a question requires minor rephrasing or an answer can be refined against clear visual content, raters directly update the item within the review loop; when an instance cannot be resolved or cleanly annotated at the unit-task level, raters are instructed to drop it. Beyond ambiguities in the overall task, drops in the second stage occur when an intermediate unit task cannot be reliably annotated across frames or views. Typical cases include a spatial reference (such as a roadside curb) shifting to the image boundary with insufficient visibility, a target vehicle tracked across views becoming too small or partially occluded in a later frame, or a second vehicle of the same color partially entering the scene and preventing unique target identification. Removing such instances ensures that every unit task in the SCM admits a well-defined intervention state. Across 1,467 candidate instances dispatched for annotation, 323 (22.0%) were dropped under these criteria, yielding the final 1,144 instances and over 33,000 unit-task questions in CADET.

C Experiments Setup

Evaluated Models. We evaluate 12 MLLMs across six model families: Gemini 3.1 Pro (Gemini Team, 2026), Gemini 3.5 Flash (Google DeepMind, 2026a), Gemini 3 Flash (Gemini Team & Google DeepMind, 2025), Gemini 3.1 Flash-Lite (Gemini Team & Google DeepMind, 2026), GPT-5.4 (OpenAI, 2026), GPT-5.4 Nano (OpenAI, 2026), Grok 4.3 (xAI, 2025), Gemma 4 31B (DeepMind, 2026), Gemma 4 26B (DeepMind, 2026), Qwen3.5-397B-A17B (Qwen Team, 2026a), Qwen3.6 27B (Qwen Team, 2026b), and Kimi K2.5 (Team et al., 2026). The Gemini, GPT, Grok and Gemma models are queried through the official APIs, while the remaining open-weight models are queried via OpenRouter. All models are run with their default sampling settings, maximum supported output tokens, and thinking mode set to *high* (or left at the default active setting when not configurable).

LLM-as-a-judge details. Among the 46 unit tasks, 43 use structured output formats (multiple-choice, integer, or coordinate) and are evaluated via deterministic rule-based parsing and exact match, while the remaining 3 free-form text unit tasks are evaluated using Gemini 3.6 Flash (Google DeepMind, 2026b) by comparing model predictions against ground-truth reference answers. Although these 3 unit tasks do not constrain outputs to option letters or numbers, each still pairs with a concise, specific reference answer (e.g., an atomic action such as "turn left"), so the judge only checks whether the predicted phrase conveys the same meaning as the reference. For the 43 structured unit tasks, Gemini 3.6 Flash is invoked strictly as a fallback when a response deviates from the required surface format (e.g., outputting "So the answer is **C**." instead of "C"). Across all evaluation runs on structured unit tasks, deterministic exact match directly resolves 98.03% of predictions, with only 1.97% triggering the LLM-as-a-judge fallback.

Prompts. Figures 11 and 12 present the prompt templates used to assemble intervened prerequisite context and to run the LLM judge, respectively. For active interventions ($A_i \in \{1, 0\}$), the prerequisites $\mathrm{Pa}(X_i)$ are formatted as a structured JSON question–answer block and prepended to the target task query using the same template for both $A_i = 1$ and $A_i = 0$, differing only in the filled prerequisite answer values (along with the corresponding visual annotations rendered on the image when $A_i = 1$ represents spatial ground truth). Under the natural state ($A_i = \varnothing$), the prerequisite context block is omitted while the target question and output-format instructions remain the same.

```
[
  {
    "question": "{unit_task_question}",
    "answer": "{unit_task_answer}",
    "question_image": ["Image-B", "Image-C"]
  },
  ...
]

Above are some fundamental facts to help you analyze and answer the question
below:

**Question**: {question}

Make sure the **REPLY** only contains the correct answer (for example, A,B,C,D,E
for ordering task, letter for multiple-choice task, etc). No other analysis in
the **REPLY** is allowed. Any expression like 'Final Answer', '*Answer' in the
**REPLY** is **not allowed**.
```

Figure 11: Prompt template for our evaluation under active causal interventions ($A_i \in \{1, 0\}$). Under the natural state ($A_i = \varnothing$), only the target question and last paragraph of output format instructions are prompted.

```
You are an expert evaluator. Given a question, its ground truth answer, and a
model's prediction, your task is to determine if the prediction is correct based
on the ground truth.

**Question**: {question}
**Ground Truth**: {ground_truth}

**Prediction**:
{prediction}

Is the prediction correct? Answer strictly with 'Yes' or 'No'.
```

Figure 12: LLM-as-a-judge prompt template.

D More Experiment Results

D.1 More Capability Diagnosis Results (NC, IC, RC)

This section provides additional visualizations and the complete numerical results supporting the capability diagnostics in Section 5.2. First, whereas Section 5.2 focuses on the 30 non-root unit tasks ($\text{Pa}(X_i) \neq \emptyset$) that admit upstream interventions, Figure 13,14 and Table 7,6 incorporate the 16 foundational root unit tasks ($\text{Pa}(X_i) = \emptyset$, for which only natural capability NC is defined), comparing root NC against non-root NC and IC across individual models (Figure 14, Table 7) and capability categories (Figure. 13, Table 6). Second, Figure 15,16 and Table 8,9 expand the compact RC $-$ NC summary in Figure 4 of the main text by plotting absolute NC and RC levels alongside their signed differences at the category level (Figure 15, Table 8) and across all 12 models within each category (Figure 16, Table 9), reporting the full per-group means and standard deviations.

Comparing Non-Root Capabilities (NC, IC) with Foundational Root Capabilities (NC). Because root unit tasks have no modeled upstream prerequisites in the task SCM ($\text{Pa}(X_i) = \emptyset$), their

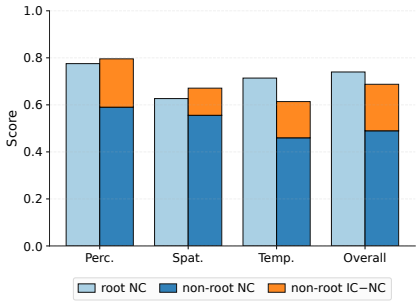

Figure 13: Comparing NC on root unit tasks against NC and IC on non-root unit tasks, per category, averaged over the 12 models. Cognitive is not shown as it has no root unit task. The non-root bar stacks IC−NC on top of NC, so its top is IC.

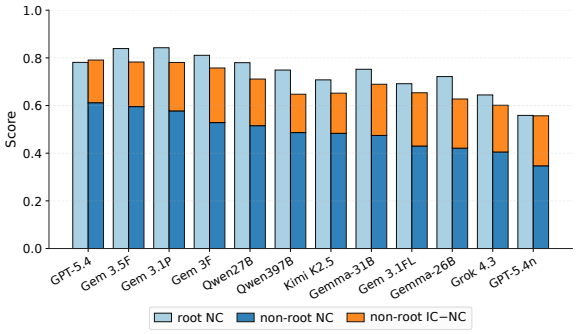

Figure 14: Comparing NC on root unit tasks against NC and IC on non-root unit tasks for each of the 12 models. There are 16 root unit tasks and 30 non-root unit tasks. The non-root bar stacks IC−NC on top of NC, so its top is IC. Models are ordered by overall non-root NC.

natural capability (NC) assesses foundational proficiency independently of upstream prerequisite states. Under unassisted evaluation ($\text{do}(\mathbf{A}_{\text{Pa}(X_i)} = \emptyset)$), models achieve higher accuracy on the 16 root unit tasks than on the 30 downstream non-root unit tasks: as reported in Table 7 and Figure 14, mean NC across the 12 models is 0.740 on root nodes versus 0.490 on non-root nodes (a difference of $+0.250$, ranging from $+0.169$ to $+0.301$ across individual models). A consistent NC difference between root and non-root nodes appears within each category that contains root tasks (Table 6, Figure 13): Perception (0.775 vs. 0.590), Spatial (0.627 vs. 0.556), and Temporal (0.714 vs. 0.460), while Cognitive unit tasks occur exclusively as non-root nodes ($\text{Pa}(X_i) \neq \emptyset$) and score lowest under unassisted evaluation (NC $= 0.406$). When correct prerequisite answers are supplied ($\text{do}(\mathbf{A}_{\text{Pa}(X_i)} = 1)$), mean non-root IC across models rises to 0.688 against the 0.740 root NC baseline (narrowing the overall gap from 0.250 to 0.052, with differences as small as -0.010 for GPT-5.4 and 0.002 for GPT-5.4n). Across categories, this comparison reinforces the capability profile in Section 5.2: Perception maintains high accuracy on both root tasks (NC $= 0.775$) and prerequisite-supplied non-root tasks (IC $= 0.796$), and Cognitive non-root tasks rise from NC $= 0.406$ to IC $= 0.721$, approaching the average root-node baseline (0.740). By contrast, Spatial exhibits the lowest root-node accuracy (NC $= 0.627$) and reaches a similarly bounded ceiling on non-root tasks when prerequisites are supplied (IC $= 0.671$), while Temporal non-root tasks reach IC $= 0.614$ (vs. 0.714 on root tasks), consistent with the persistent residual gaps ($1 -$ IC) in spatial and temporal capabilities identified in the main text.

Expanded Breakdown of Resilience Capability (RC vs. NC). Figure 15 and Table 8 complement Figure 4 by displaying NC and RC on the same absolute capability scale across categories, while Figure 16 and Table 9 provide the disaggregated per-model profiles parallel to Figure 3(b–e). On average across the 12 models (Table 8), injecting incorrect prerequisite answers ($\text{do}(\mathbf{A}_{\text{Pa}(X_i)} = 0)$) yields lower accuracy than unassisted evaluation in Perception ($0.590 \rightarrow 0.446$, RC$-$NC $= -0.144$), Spatial ($0.556 \rightarrow 0.439$, -0.116), and Temporal ($0.460 \rightarrow 0.414$, -0.046), corresponding to an

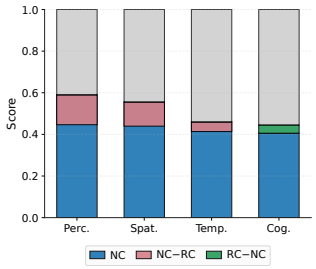

Figure 15: Comparing NC and RC across the four categories on non-root unit tasks, averaged over the 12 models. Each bar shows NC with the signed NC/RC difference stacked out of it, so the boundary of the shaded band is RC.

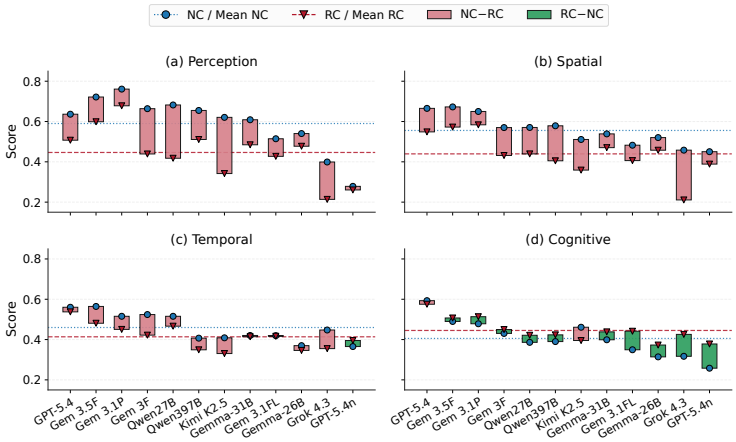

Figure 16: Comparing NC and RC for each of the 12 models within each category on non-root unit tasks. Circles mark NC and triangles mark RC; the shaded band spans the two and is colored by the sign of RC − NC. The horizontal lines are the means over the 12 models.

overall non-root difference of -0.054 ($0.490 \to 0.435$), whereas Cognitive tasks exhibit a modest positive shift ($0.406 \to 0.445$, $+0.040$). The per-model breakdown in Figure 16 and Table 9 shows that these category-level trends hold consistently across model families and scales: all 12 models exhibit RC $<$ NC in Perception and Spatial, 11 of 12 decline in Temporal (with GPT-5.4n showing a $+0.030$ shift), and 10 of 12 show positive RC − NC shifts in Cognitive (all except GPT-5.4 at -0.018 and Kimi K2.5 at -0.066), providing the full per-model evidence for the non-monotonicity discussed in Property 3.6 and Section 5.2.

D.2 More Causal Contribution Diagnosis Results (N, S-score)

In this section, we provide additional experimental results complementing the causal contribution analysis in Section 5.3. Figure 17 reports prerequisite N- and S-Scores under the remaining three state contrasts $(1, 0)$, $(0, \varnothing)$, and $(\varnothing, 0)$ (complementing $(1, \varnothing)$ in Figure 5). Figure 18 disaggregates these three contrasts by task pattern and model group (extending Figure 6). Figure 19 shows the inter-model rank agreement across all eight contribution scores, and Figure 20 provides the per-prerequisite breakdown across the four state contrasts. To prevent small denominators from producing numerically unstable ratios in Eqs. (5)–(6), we set a score to 0 whenever its baseline denominator falls below 0.10. Crucially, this threshold has zero effect on six of the eight metrics and affects only 3.0% ($94/3{,}168$) of all scores, concentrated in $N(0, \varnothing)$ (with a small remainder in $N(\varnothing, 0)$). This is expected by design: $N(0, \varnothing)$ measures the success drop from setting a prerequisite to $\varnothing$ in the all-incorrect (0) environment, where baseline success $\mathrm{RC}(Y)$ is naturally near zero whenever incorrect prerequisites prevent task completion.

Table 6: NC on root unit tasks and NC, IC on non-root unit tasks, per category, with the number of unit tasks in each group. Entries are the unweighted mean over unit tasks, averaged over the 12 models, ± the standard deviation across models.

	#unit tasks		Root	Non-root		
Category	root	non-root	NC	NC	IC	IC − NC
Perception	11	4	0.775 ± 0.077	0.590 ± 0.138	0.796 ± 0.112	+0.206
Spatial	3	9	0.627 ± 0.100	0.556 ± 0.077	0.671 ± 0.059	+0.115
Temporal	2	8	0.714 ± 0.119	0.460 ± 0.072	0.614 ± 0.087	+0.154
Cognitive	0	9	–	0.406 ± 0.092	0.721 ± 0.089	+0.316

Table 7: NC on root unit tasks and NC, IC on non-root unit tasks for each of the 12 models. Entries are the unweighted mean over unit tasks ± the standard deviation across unit tasks (16 root, 30 non-root). The last two columns are differences of the corresponding means.

Model	Root NC	Non-root NC	Non-root IC	NC(root)−NC(non-root)	NC(root)−IC(non-root)
GPT-5.4	0.781 ± 0.108	0.612 ± 0.180	0.791 ± 0.167	+0.169	−0.010
Gem 3.5F	0.839 ± 0.099	0.595 ± 0.217	0.783 ± 0.172	+0.244	+0.056
Gem 3.1P	0.842 ± 0.090	0.577 ± 0.202	0.781 ± 0.173	+0.265	+0.062
Gem 3F	0.811 ± 0.119	0.528 ± 0.206	0.757 ± 0.180	+0.282	+0.053
Qwen27B	0.780 ± 0.132	0.515 ± 0.247	0.711 ± 0.219	+0.264	+0.069
Qwen397B	0.749 ± 0.144	0.487 ± 0.240	0.647 ± 0.226	+0.262	+0.101
Kimi K2.5	0.708 ± 0.200	0.483 ± 0.199	0.652 ± 0.213	+0.224	+0.056
Gemma-31B	0.752 ± 0.161	0.475 ± 0.235	0.689 ± 0.202	+0.278	+0.063
Gem 3.1FL	0.692 ± 0.186	0.430 ± 0.217	0.654 ± 0.209	+0.262	+0.038
Gemma-26B	0.722 ± 0.196	0.421 ± 0.248	0.627 ± 0.218	+0.301	+0.094
Grok 4.3	0.644 ± 0.175	0.405 ± 0.210	0.601 ± 0.205	+0.239	+0.043
GPT-5.4n	0.559 ± 0.239	0.347 ± 0.190	0.557 ± 0.236	+0.212	+0.002

Corrupting a prerequisite reduces success more than leaving it unassisted. $N(1,0)$ and $N(1,\varnothing)$ share the same oracle baseline where all prerequisites are correctly provided, differing only in whether X_i receives an incorrect answer or is left unassisted. Across prerequisites, $N(1,0)$ consistently exceeds $N(1,\varnothing)$ (Figure 17a vs. Figure 5), shifting more prerequisites into the high-N / low-S quadrant and fewer into the low-N / high-S quadrant. For example, on the perception prerequisites of Music Note, $N(1,\varnothing)$ is low (0.04–0.16) and $S(1,\varnothing)$ is high (0.51–0.59): omitting a single prerequisite in the oracle setting incurs minimal loss, while supplying it alone in the natural state recovers over half of the failures. Under $(1,0)$, however, $N(1,0)$ rises to 0.27–0.73 and $S(1,0)$ falls to at most 0.02: a single incorrect prerequisite removes a substantial fraction of oracle success, while correcting one prerequisite when others remain incorrect recovers almost no failures. Similarly, under $(\varnothing,0)$ (Figure 17c), prerequisites shift along $N(\varnothing,0)$ into the high-N / low-S region, whereas under $(0,\varnothing)$ (Figure 17b), most prerequisites remain near zero with positive values concentrated in specific prerequisites examined in Figure 20.

Separation between SOTA and Others under prerequisite corruption. When disaggregated by task pattern and model group (Figure 18), SOTA and Others separate clearly in Pattern 2 across all three contrasts involving incorrect prerequisites: $S(1,0)$, $S(\varnothing,0)$, and $N(1,0)$ (Figure 18c,d,k,l). Starting from the all-incorrect environment, SOTA achieves consistently higher single-prerequisite recovery than Others both when correcting a single prerequisite ($S(1,0)$: 0.143–0.263 for SOTA vs. 0.029–0.105 for Others) and when removing a single incorrect answer ($S(\varnothing,0)$: 0.044–0.152 vs. 0.015–0.028). Symmetrically, starting from the oracle environment, corrupting a single prerequisite causes smaller drops for SOTA ($N(1,0)$: 0.097–0.106) than for Others (0.108–0.295), mirroring the shift toward higher N and lower S for Others under $(1,\varnothing)$ in Figure 6.

Inter-model agreement on high-$S(1,\varnothing)$ prerequisites. Among the eight contribution metrics, $S(1,\varnothing)$ exhibits the strongest inter-model agreement, with a mean pairwise Spearman correlation of 0.72, compared with 0.41 for $N(1,\varnothing)$ (Figure 19). Thus, the high-leverage prerequisites that recover natural failures when supplied alone are largely shared across models, whereas sensitivity to single-prerequisite omission in the oracle setting varies more across models. Agreement on $S(1,\varnothing)$ is especially strong among the four SOTA models (mean $\rho = 0.85$ vs. 0.70 among Others), holding both within the Gemini series (0.87) and across families between GPT-5.4 and Gemini (0.84), and within individual tasks the top-$S(1,\varnothing)$ prerequisite is identical or shared between the same top pair across models on the majority of tasks.

Table 8: NC, IC and RC per category on the 30 non-root unit tasks. Entries are the unweighted mean over unit tasks, averaged over the 12 models, ± the standard deviation across models.

Category	NC	IC	RC	IC − NC	RC − NC
Perception	0.590 ± 0.138	0.796 ± 0.112	0.446 ± 0.131	+0.206	−0.144
Spatial	0.556 ± 0.077	0.671 ± 0.059	0.439 ± 0.102	+0.115	−0.116
Temporal	0.460 ± 0.072	0.614 ± 0.087	0.414 ± 0.063	+0.154	−0.046
Cognitive	0.406 ± 0.092	0.721 ± 0.089	0.445 ± 0.060	+0.316	+0.040

Table 9: NC, IC and RC per model and category on the 30 non-root unit tasks, with *Overall* aggregating all 30. Entries are the unweighted mean over unit tasks ± the standard deviation across unit tasks.

	Perception			Spatial			Temporal			Cognitive			Overall		
Model	NC	IC	RC	NC	IC	RC	NC	IC	RC	NC	IC	RC	NC	IC	RC
GPT-5.4	0.636 *±0.194*	0.913 *±0.087*	0.507 *±0.288*	0.665 *±0.202*	0.747 *±0.214*	0.549 *±0.213*	0.560 *±0.148*	0.730 *±0.181*	0.537 *±0.219*	0.593 *±0.192*	0.835 *±0.094*	0.575 *±0.274*	0.612 *±0.180*	0.791 *±0.167*	0.548 *±0.232*
Gem 3.5F	0.722 *±0.192*	0.889 *±0.042*	0.599 *±0.194*	0.672 *±0.231*	0.754 *±0.213*	0.572 *±0.252*	0.564 *±0.154*	0.699 *±0.202*	0.481 *±0.187*	0.490 *±0.232*	0.839 *±0.091*	0.507 *±0.219*	0.595 *±0.217*	0.783 *±0.172*	0.532 *±0.212*
Gem 3.1P	0.761 *±0.155*	0.907 *±0.027*	0.678 *±0.158*	0.650 *±0.217*	0.740 *±0.230*	0.584 *±0.207*	0.515 *±0.164*	0.698 *±0.175*	0.450 *±0.222*	0.478 *±0.174*	0.839 *±0.088*	0.513 *±0.194*	0.577 *±0.202*	0.781 *±0.173*	0.540 *±0.206*
Gem 3F	0.664 *±0.250*	0.879 *±0.072*	0.439 *±0.304*	0.570 *±0.268*	0.710 *±0.253*	0.431 *±0.240*	0.524 *±0.122*	0.707 *±0.163*	0.422 *±0.205*	0.430 *±0.154*	0.796 *±0.121*	0.450 *±0.223*	0.528 *±0.206*	0.757 *±0.180*	0.435 *±0.222*
Qwen27B	0.682 *±0.167*	0.873 *±0.119*	0.418 *±0.237*	0.571 *±0.308*	0.663 *±0.252*	0.439 *±0.249*	0.515 *±0.248*	0.672 *±0.210*	0.466 *±0.276*	0.386 *±0.159*	0.722 *±0.220*	0.421 *±0.213*	0.515 *±0.247*	0.711 *±0.219*	0.438 *±0.233*
Qwen397B	0.655 *±0.126*	0.808 *±0.167*	0.511 *±0.184*	0.579 *±0.287*	0.677 *±0.269*	0.405 *±0.217*	0.407 *±0.164*	0.480 *±0.150*	0.349 *±0.196*	0.390 *±0.238*	0.695 *±0.194*	0.423 *±0.246*	0.487 *±0.240*	0.647 *±0.226*	0.410 *±0.212*
Kimi K2.5	0.621 *±0.112*	0.698 *±0.127*	0.342 *±0.161*	0.511 *±0.245*	0.659 *±0.300*	0.359 *±0.231*	0.409 *±0.156*	0.586 *±0.175*	0.331 *±0.227*	0.461 *±0.204*	0.682 *±0.187*	0.395 *±0.138*	0.483 *±0.199*	0.652 *±0.213*	0.360 *±0.189*
Gemma-31B	0.609 *±0.291*	0.825 *±0.035*	0.485 *±0.272*	0.539 *±0.275*	0.679 *±0.279*	0.470 *±0.247*	0.420 *±0.182*	0.625 *±0.163*	0.414 *±0.236*	0.399 *±0.200*	0.696 *±0.182*	0.438 *±0.217*	0.475 *±0.235*	0.689 *±0.202*	0.448 *±0.227*
Gem 3.1FL	0.514 *±0.294*	0.756 *±0.117*	0.427 *±0.264*	0.482 *±0.221*	0.627 *±0.260*	0.406 *±0.207*	0.419 *±0.177*	0.557 *±0.230*	0.416 *±0.232*	0.349 *±0.220*	0.720 *±0.141*	0.441 *±0.212*	0.430 *±0.217*	0.654 *±0.209*	0.422 *±0.211*
Gemma-26B	0.540 *±0.351*	0.791 *±0.115*	0.477 *±0.301*	0.520 *±0.287*	0.632 *±0.285*	0.458 *±0.260*	0.370 *±0.171*	0.532 *±0.212*	0.346 *±0.196*	0.314 *±0.186*	0.635 *±0.154*	0.373 *±0.216*	0.421 *±0.248*	0.627 *±0.218*	0.405 *±0.230*
Grok 4.3	0.399 *±0.237*	0.659 *±0.096*	0.214 *±0.223*	0.458 *±0.249*	0.594 *±0.273*	0.211 *±0.161*	0.447 *±0.215*	0.585 *±0.220*	0.355 *±0.218*	0.317 *±0.150*	0.597 *±0.174*	0.426 *±0.215*	0.405 *±0.210*	0.601 *±0.205*	0.314 *±0.213*
GPT-5.4n	0.278 *±0.250*	0.551 *±0.331*	0.260 *±0.234*	0.450 *±0.204*	0.571 *±0.232*	0.389 *±0.202*	0.365 *±0.176*	0.498 *±0.233*	0.395 *±0.222*	0.258 *±0.120*	0.598 *±0.233*	0.378 *±0.166*	0.347 *±0.190*	0.557 *±0.236*	0.370 *±0.196*

Cross-contrast consistency and per-prerequisite breakdown. Across prerequisites, rankings correlate positively between $N(1, \varnothing)$ and $N(1, 0)$ (Spearman $\rho = 0.64$) and between $S(1, \varnothing)$ and $S(1, 0)$ ($\rho = 0.53$; Figure 20), indicating that prerequisites whose omission most reduces oracle performance also tend to be the most destructive when corrupted, and those that best recover natural failures also drive recovery in the all-incorrect setting. For transitions between $\varnothing$ and 0, $(\varnothing, 0)$ and $(0, \varnothing)$ measure opposite directions of the same state pair; under zero-clipping, at most one direction is active for a given model and prerequisite, separating performance degradation from incorrect prerequisites ($N(\varnothing, 0), S(\varnothing, 0)$) from positive recovery when an incorrect prerequisite is provided over no prerequisite ($S(0, \varnothing), N(0, \varnothing)$) under Property 3.6. Consistent with Section 5.2 and Figure 9, positive $S(0, \varnothing)$ values appear primarily on prerequisites feeding into downstream cognitive or spatial-relational integration—such as Temporal Sequencing (0.34), Temporal Localization (0.11), and Object Localization (0.07) in Traffic Causation, and Viewpoint Estimation (0.23) in Surround Loc.—while remaining near zero on foundational perception and spatial prerequisites.

D.3 Detail Explanation of Markov Chain unit task results

Section 5.4 analyzes the three Markov chain unit tasks in CADET (Figure 10): *Maze* and *Multistep Reasoning*, whose primary task structure is a Markov chain, together with the route-planning unit task embedded within *Map Navigation*. While all three share the same Markov sequential dependency structure (Definition A.7), they differ in their concrete step-wise logic, and because this distinction directly manifests in their empirical behavior, we group them into the two regimes discussed in Section 5.4: *goal-directed* and *process-cumulative*. In *Maze*, the trajectory from the start to the exit is partitioned into T=10 steps; because the overall query (identifying which candidate is the exit) cannot be further decomposed into distinct sub-questions, at each step t the model is positioned at the

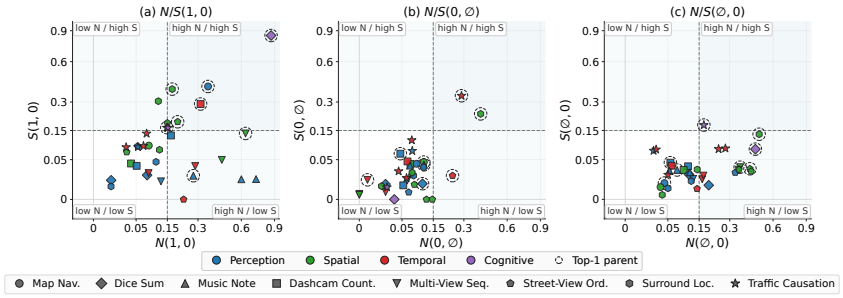

Figure 17: Prerequisite N/S-Scores under state contrasts (a) $(1, 0)$, (b) $(0, \varnothing)$, and (c) $(\varnothing, 0)$, averaged across models. Dashed circles mark the highest-S prerequisite within each task. Axes and quadrant thresholds follow Figure 5.

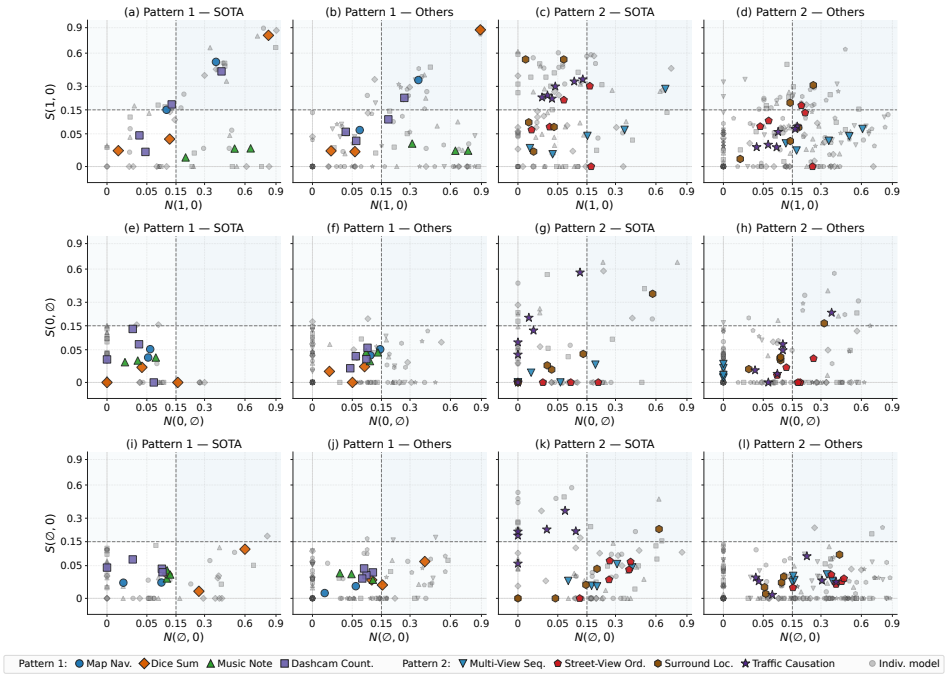

Figure 18: Prerequisite N/S-Scores under $(1, 0)$ (a–d), $(0, \varnothing)$ (e–h), and $(\varnothing, 0)$ (i–l) by task pattern and model group. Large markers denote group means; grey markers indicate individual models.

t-th waypoint to answer this same exit-identification question. Similarly, in the Markov chain node of *Map Navigation*, the model stands at each intermediate step along the route and looks toward a given destination to decide the next move. Both tasks are therefore goal-directed: conditioned on the current step t, solving the task depends only on the remaining path toward the goal rather than the history already traversed. By contrast, *Multistep Reasoning* is process-cumulative: starting from an initial state without a pre-specified destination, the model follows movement rules where each step alternates between identifying the current *Position* and determining the current *Action*. Because the terminal state is unknown in advance, accurately resolving either node at step t requires tracking the cumulative execution process from the starting point up to step t.

To operationalize the ternary intervention at step t along an unfolded chain, we provide the full preceding trajectory context from step 1 to $t-1$ in the prompt. Under $A{=}1$, we inject the ground-truth questions, answers, and corresponding visual annotations for all preceding steps. A special case arises in *Maze*: because every step asks the same exit-identification question and shares the identical ground-truth answer, injecting the textual answer of any preceding step would directly reveal the answer at step t. For $A{=}1$ in *Maze*, we therefore inject the preceding step questions without their

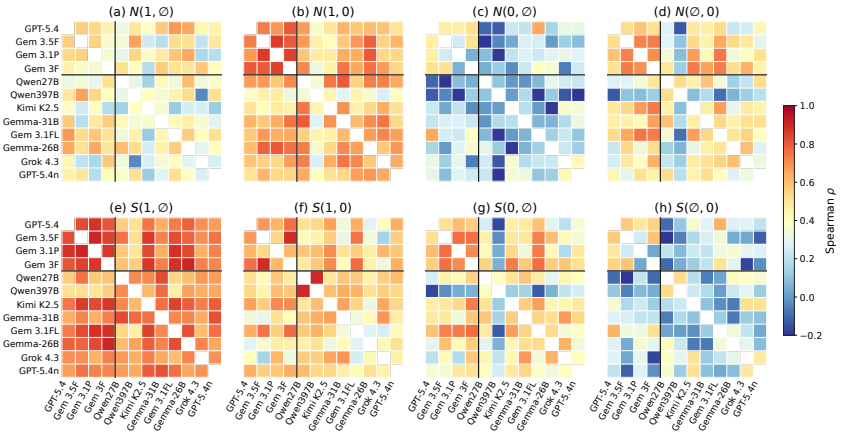

Figure 19: Pairwise Spearman rank correlation between models on N-Scores (a–d) and S-Scores (e–h) across prerequisites under the four state contrasts. Lines separate SOTA from Others.

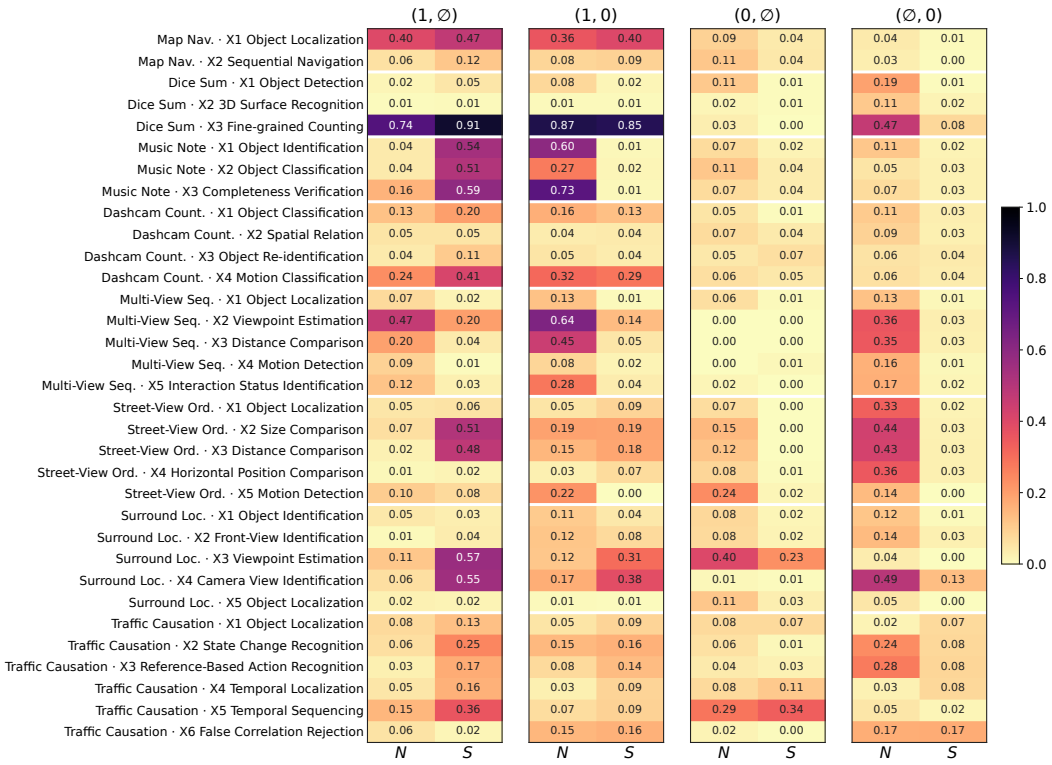

Figure 20: Prerequisite N/S-Scores under the four state contrasts, averaged over evaluated models.

textual answers, accompanied by the visual annotation on the image marking the current waypoint and the human-annotated path from the start up to step $t-1$. Under $A=0$, each preceding step from 1 to $t-1$ is independently assigned a randomly sampled incorrect answer from the valid answer space via text only (Section 5.1), while $A=\varnothing$ leaves the preceding trajectory unassisted.

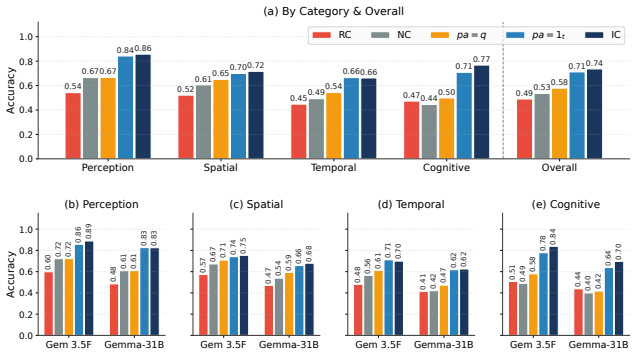

Figure 21: Capability on non-root unit tasks under five prerequisite states: RC, NC , question-only (q), text-answer-only (1_t), and IC. (a) Category-level and overall averages over 30 non-root unit tasks and two models; (b–e) per-model results within each category.

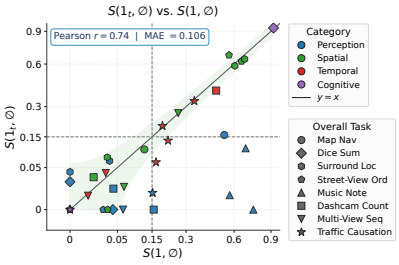

Figure 22: $S(1_t, \varnothing)$ vs. $S(1, \varnothing)$ for each prerequisite of an overall task, averaged over two models. Color denotes the prerequisite's capability category and marker shape the overall task. The solid line denotes $y = x$ and the shaded band $|y - x| \leq 0.06$; dashed lines mark the 0.15 threshold as in Figure 5 ($\mathrm{sqrt}(\cdot)$-scaled axes).

D.4 Additional Experiments and Findings

D.4.1 Ablation on Intervention Modality

In the main experiments, the correct intervention ($A_i = 1$) injects both a textual answer and, where applicable, visual annotations (e.g., bounding boxes, trajectories) onto the image. To examine the role of visual annotations in this intervention, we also evaluate a text-answer-only condition $A_i = 1_t$, which provides the same textual answer as $A_i = 1$ but keeps the original image unmodified. We run this condition on two models, Gemini 3.5 Flash from the top-4 (SOTA) group and Gemma-4-31B from the remaining (Others) group, across all 30 non-root unit tasks.

Figure 21(a) shows that 1_t closely approaches IC across all four capability categories. In temporal, the two conditions are virtually identical (both 0.66); in perception and spatial, they differ by only 0.02 (0.84 vs. 0.86; 0.70 vs. 0.72). The largest gap appears in cognitive (0.71 vs. 0.77), where 1_t nonetheless remains far above NC (0.44) and RC (0.47). Overall, 1_t reaches 0.71 against an IC of 0.74. The per-model results (Figure 21(b–e)) are consistent: for both models, 1_t tracks IC in every category, with a gap of at most 0.06 in any model–category pair. These results indicate that the textual answer accounts for the large majority of the capability gain under correct prerequisites, while the remaining difference between 1_t and IC reflects the additional contribution of visual annotations.

The same pattern holds for the causal contribution metrics. Figure 22 compares $S(1_t, \varnothing)$ with 1_t in place of state 1, against $S(1, \varnothing)$ for each prerequisite. Spatial, temporal, and cognitive prerequisites lie along the diagonal, indicating that the magnitude and ranking of S-Scores are largely preserved without visual annotations. The systematic departures below the diagonal correspond to perception prerequisites, whose $S(1_t, \varnothing)$ falls well below $S(1, \varnothing)$. The corresponding N-Score comparison shows a similar agreement (Pearson $r = 0.87$ between $N(1_t, \varnothing)$ and $N(1, \varnothing)$ across all prerequisites).

D.4.2 Ablation on Question-Only Injection

In Section 5.2, incorrect prerequisites improve cognitive tasks for most models (RC > NC), and in Section 5.3, $S(0, \varnothing)$ is nonzero for some prerequisite categories, indicating that even incorrect answers occasionally help. Since the incorrect intervention ($A_i = 0$) injects both a sub-question and an incorrect answer, a natural follow-up is to ask how much of the prerequisite effect comes from the sub-question itself. We therefore evaluate a question-only condition $A_i = q$, which injects each prerequisite's sub-question without any answer, on the same two models. As shown in Figure 21(a), q yields a modest improvement over NC (overall 0.58 vs. 0.53) but remains far below 1_t (0.71) and IC (0.74). Across categories, q exceeds NC by at most 0.06, whereas 1_t exceeds NC by 0.09 (spatial) to 0.26 (cognitive). The causal contribution metrics show the same trend: $S(q, \varnothing)$ averages 0.05 across

prerequisites, compared with 0.26 for $S(1, \varnothing)$ on the same two models, and $S(q, \varnothing)$ does not exceed 0.09 in any prerequisite category. Thus, providing the sub-question without answer does improve performance, meanwhile the improvement is limited and remains far smaller than that from providing its answer.

E DISCUSSION

On the Scope and Completeness of Task Decompositions. The SCM of each task declares, for every target task, which unit tasks are placed under controlled intervention. It is not intended as an exhaustive characterization of the task's causal structure or of the model's internal computation, and we do not claim that it is complete: the set of factors that may influence a model's answer on a real-world task is open-ended. A unit task is declared as a prerequisite of a target when solving the target requires the information it represents (Appendix. B.2), and the same declared structure is used for all evaluated models. Any factor outside the declared parent sets belongs to the unmodeled factors captured by N_i (Definition 3.1); it is not separately intervened on and remains in its natural regime.

Completeness is not required for the validity of the metrics. Every quantity entering the capability and contribution metrics is an interventional probability, estimated by directly running the model under the corresponding regime, including the natural regime $\varnothing$, rather than inferred from co-occurrences between unit-task outcomes. The lower-bound guarantees (Theorems A.3 and A.4) hold for any configuration $\mathbf{c}$ of the remaining declared prerequisites and rest on prerequisite regimes being assigned through injection (Assumption 3.3), not on the declared parent sets capturing every factor that influences the outcome. N- and S-Scores therefore remain valid lower bounds on PN and PS for the context actually realized, in which the target's declared prerequisites are set as specified and all other factors are left in their natural regime.

What the decomposition determines is the reference set against which each metric is read (Section 3.3): IC measures performance on X_i when its declared prerequisites are supplied correctly, IC $-$ NC is the net gain from supplying them (the cascading component in Section 5.2), and $1 -$ IC is the residual error that persists under correct prerequisites. The category-level findings in Section 5.2 are statements of this form, namely which share of the natural error is resolved by supplying the declared prerequisites and which share persists, obtained under the same inclusion criterion for every category.

Finally, declared dependencies are task-defined, whereas the extent to which a model relies on each of them is measured rather than assumed. Averaged within each category, supplying the declared prerequisites improves performance for every model (IC $>$ NC in all 48 model-category cells; Tab. 9). Individually, a declared prerequisite may have small N- and S-Scores under $(1, \varnothing)$ (Section 5.3), meaning that neither withholding it in the oracle environment nor supplying it alone in the natural environment substantially changes the outcome. This can occur for a genuine prerequisite: both scores can be small when the model already resolves it correctly within the full task, N when its omission is compensated by the other supplied prerequisites, and S when the outcome remains limited by the other, unsupplied ones. Small scores therefore characterize a model's reliance under the given contrast.

Scope and Limitations.

Input-level interventions. In our framework, prerequisite states are controlled by injecting answers into the prompt, so all metrics characterize how a model's output responds to prerequisite information provided in its input, rather than how the prerequisite is represented inside the model. Operating at the input level allows the same controlled states to be applied across models, including closed-weight models accessible only through APIs, without relying on unverified readouts of intermediate states. How a model processes the injected prerequisite (whether adopting it, checking it against the visual input, or overriding it—is) therefore part of the behavior measured by the metrics rather than an external assumption: IC measures task performance when correct prerequisite answers are available in the context, and RC measures performance when incorrect ones are supplied. Empirically, models consistently respond to the injected content: under the same prompt template, supplying correct answers improves performance in all 48 model-category cells, whereas supplying incorrect answers degrades all 12 models in perception and spatial tasks (Section 5.2). Investigating the internal mechanisms by which models attend to and integrate injected prerequisites is an important direction for future work.

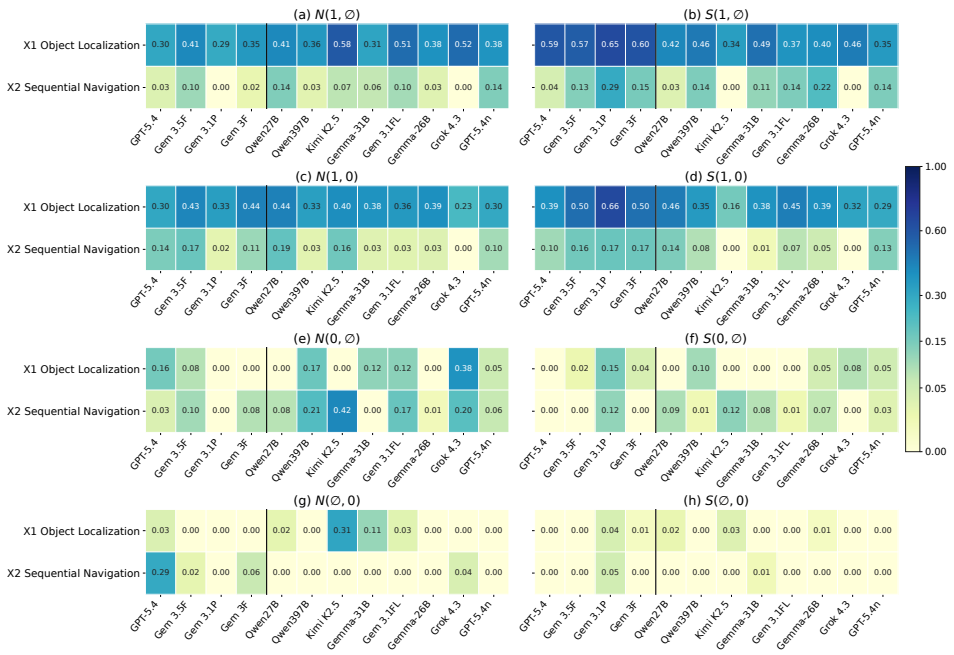

Figure 23: Per-model N- and S-Scores for the parents of Map Nav. under the four state contrasts. The vertical line separates SOTA from Others.

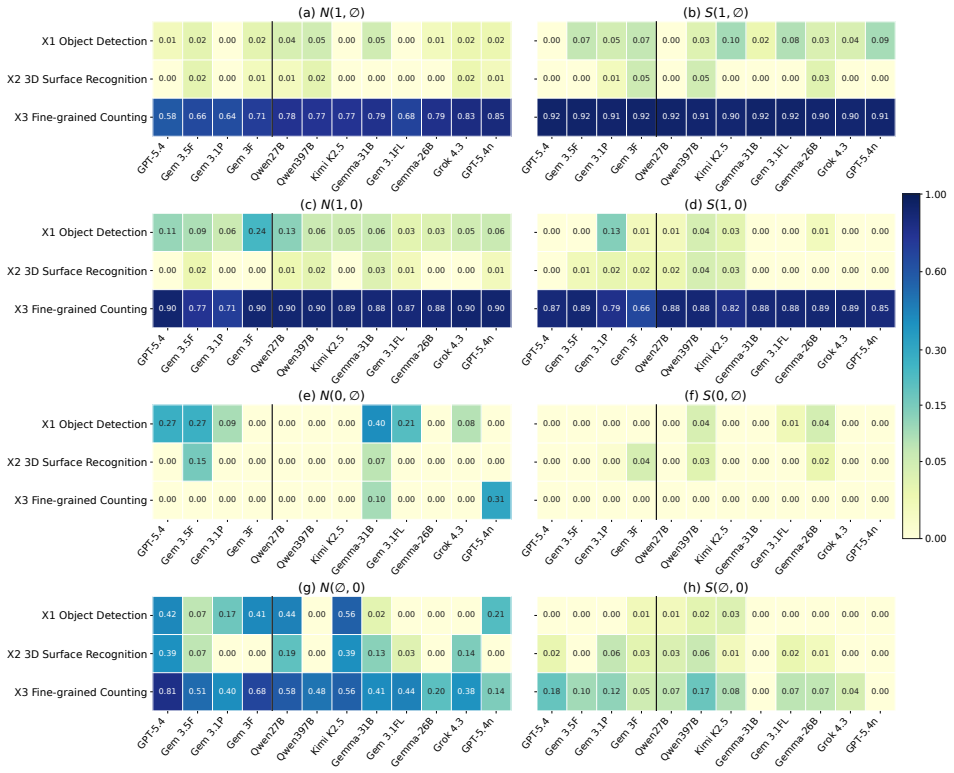

Figure 24: Same as Figure 23, for Dice Sum.

Design of the incorrect intervention. For the incorrect state $A_i = 0$, we inject an answer randomly sampled from the valid answer space of unit task X_i (Section 4). This provides a standardized, model-independent perturbation that is identical across all evaluated models, ensuring that RC and the contribution scores involving state 0 are directly comparable across models. It is intended to test how models behave under an explicitly false premise, rather than to simulate the natural errors that a specific model would make on its own, which are already captured under the unassisted state $A_i = \varnothing$.

Evaluation cost. Evaluating a target task Y with $k = |\mathrm{Pa}(Y)|$ prerequisites across the full ternary state space would in principle require 3^k prerequisite configurations. By setting the remaining prerequisites A_{-i} to homogeneous states (Appendix Section A.3.2), our framework reduces the required runs per instance to linear in k: for each prerequisite X_i, pairing $A_i \in \{1, 0, \varnothing\}$ with $A_{-i} \in \{1, 0, \varnothing\}$ yields nine state pairs, where the three all-same states (NC, IC, RC) are shared across all prerequisites, resulting in $3+6k$ runs per instance across all four contrasts. In this paper, we evaluate all four contrasts to provide a complete characterization of model behavior across the ternary intervention space. In practical applications, the $(1, \varnothing)$ contrast is the most directly informative, as it diagnoses how much each prerequisite limits and can improve performance relative to the model's natural state; evaluating $(1, \varnothing)$ alone requires only $2 + 2k$ runs per instance (the standard end-to-end evaluation plus $1 + 2k$ interventional runs). Exploring automated task decomposition is also a promising direction for scaling this diagnostic pipeline to broader task domains.

Scope and future directions. This work focuses on diagnostic evaluation, establishing a causal framework and benchmark to identify where and why MLLMs fail on compositional tasks. Developing training or inference-time methods to address the diagnosed bottlenecks is a natural next step that we hope our work will inform. More broadly, our diagnostic framework can be extended to several related areas in future work, including localizing failure bottlenecks in multi-step agentic workflows, guiding targeted capability improvement in post-training and reinforcement learning, and complementing chain-of-thought reasoning analysis.

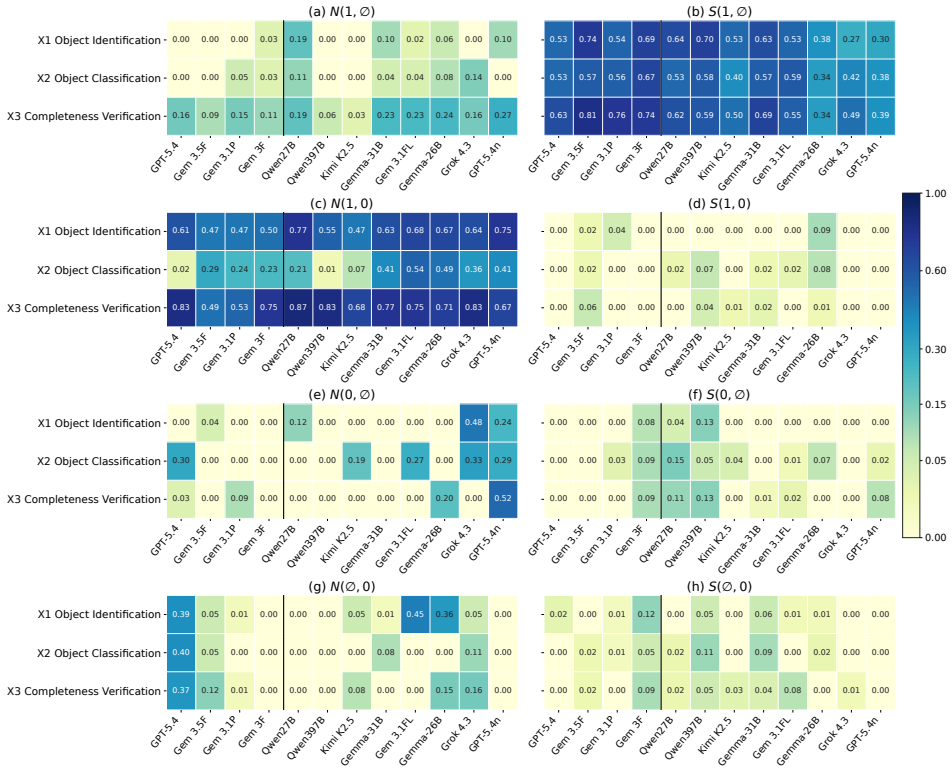

Figure 25: Same as Figure 23, for Music Note.

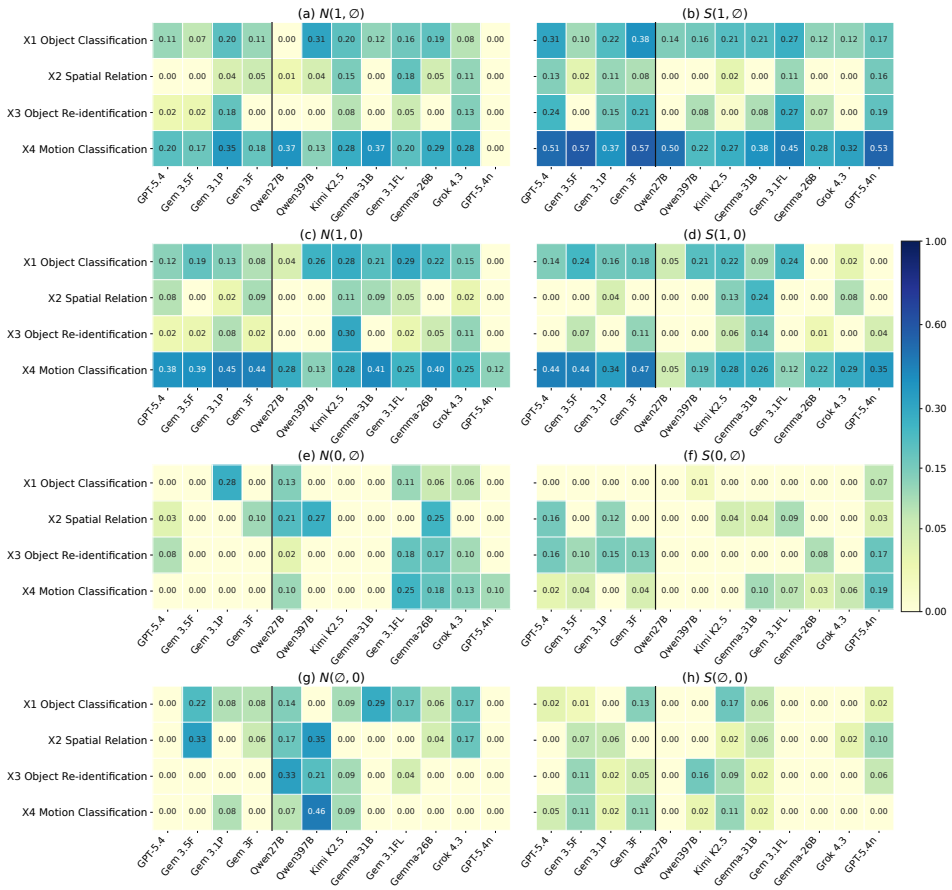

Figure 26: Same as Figure 23, for Dashcam Count..

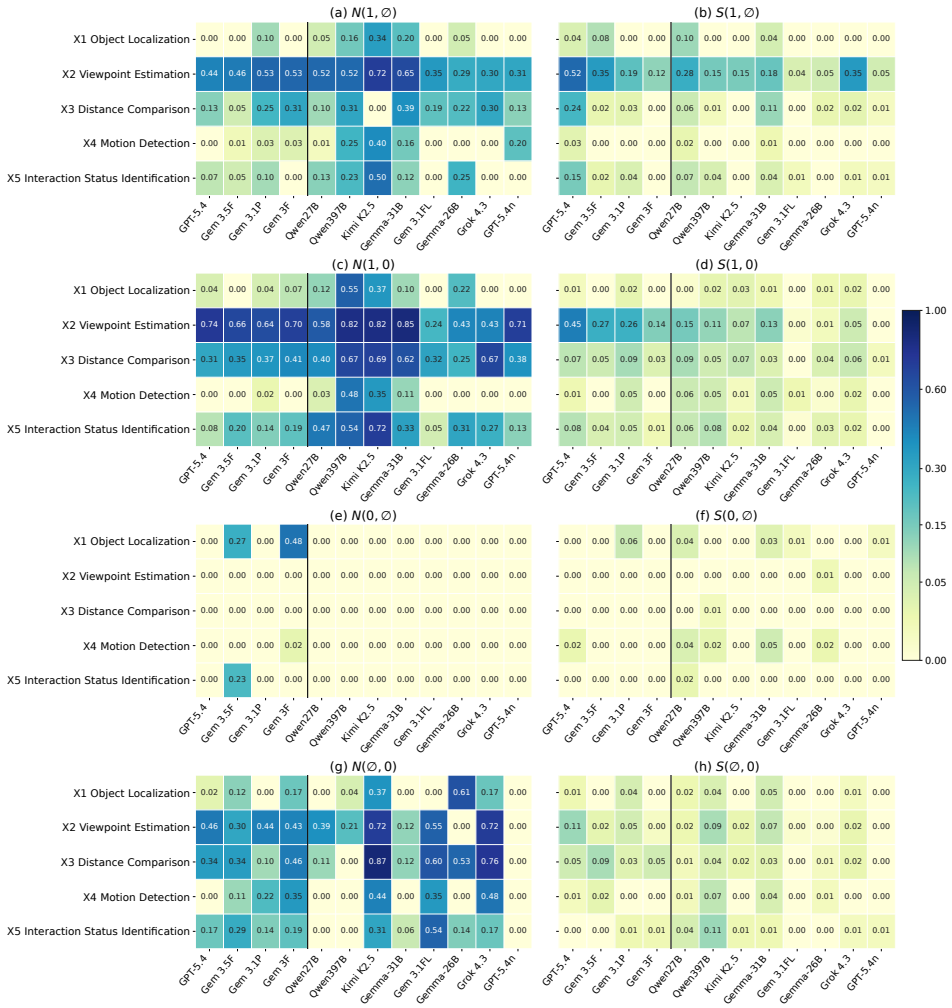

Figure 27: Same as Figure 23, for Multi-View Seq..

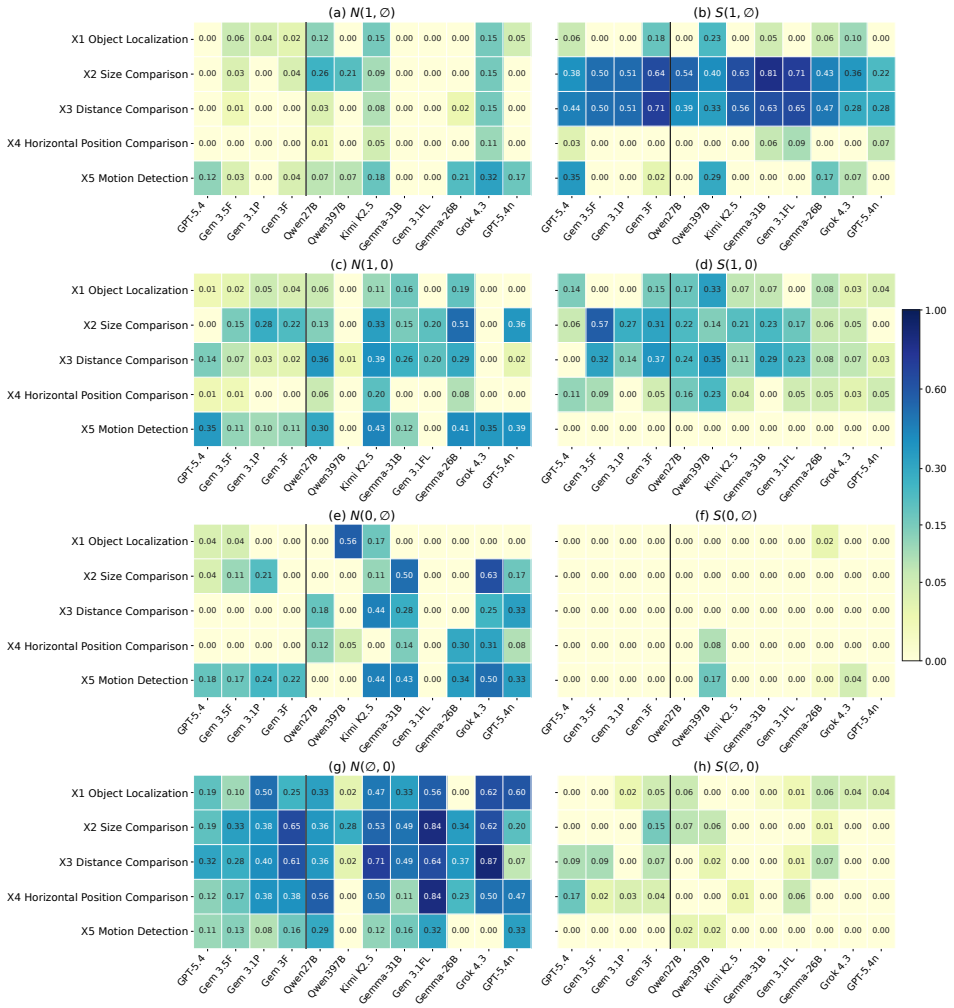

Figure 28: Same as Figure 23, for Street-View Ord..

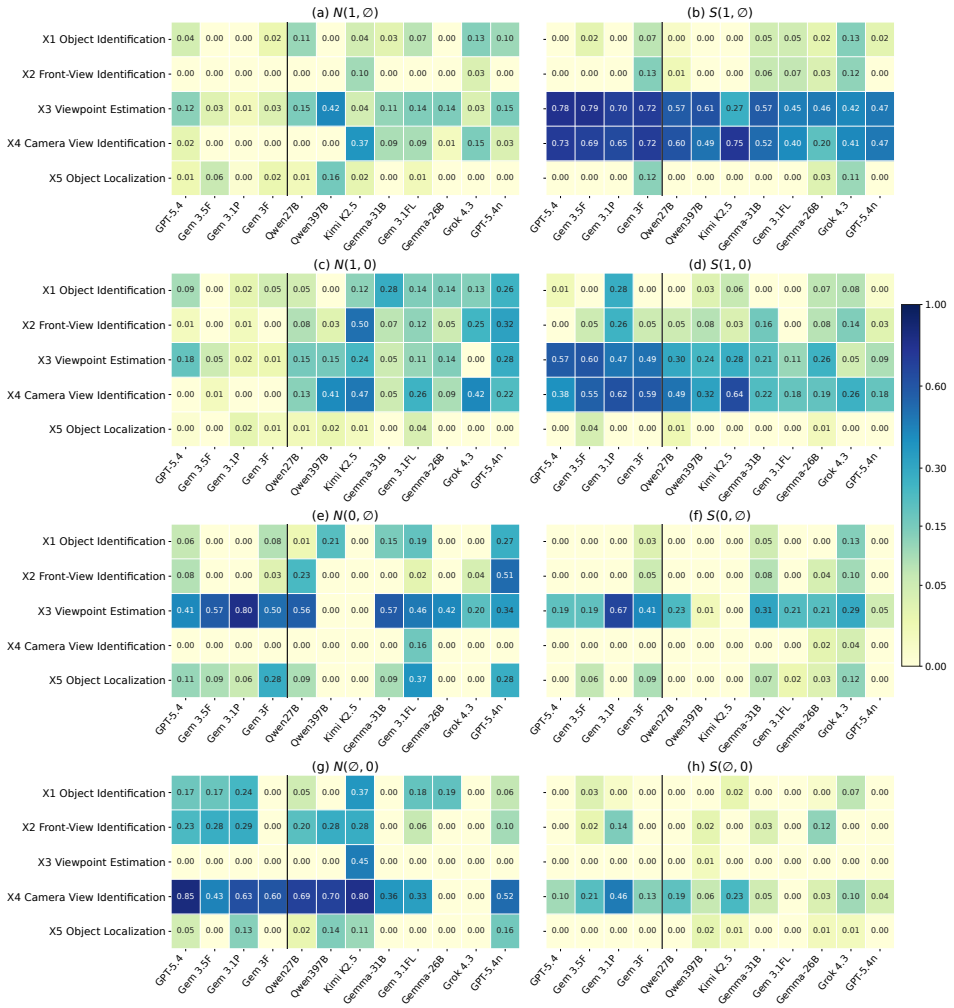

Figure 29: Same as Figure 23, for Surround Loc..

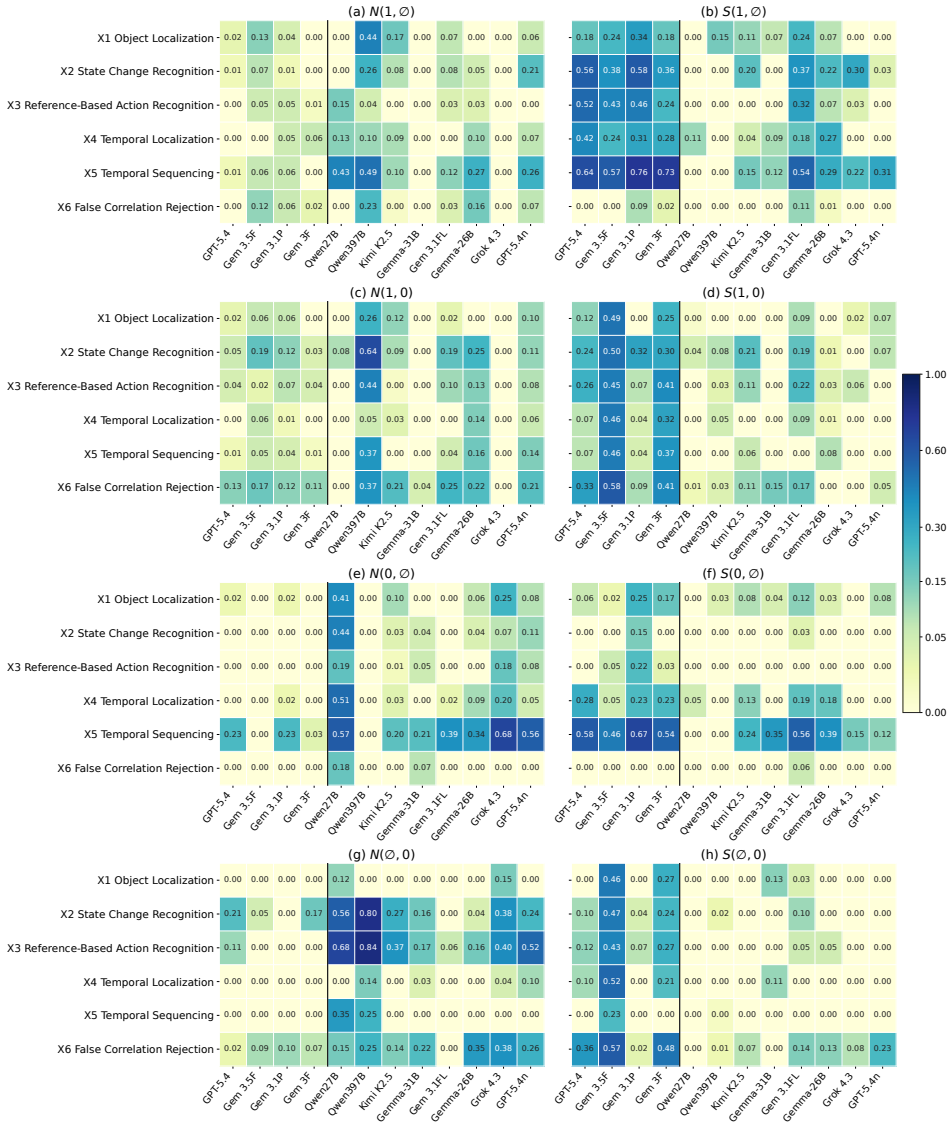

Figure 30: Same as Figure 23, for Traffic Causation.