APPENDIX I
FAILURE ANALYSIS

This section provides a breakdown of the observed failure cases in PROTEA, highlighting the primary causes of missed detections.

A. Failure Analysis: False Negatives

We analyze false negatives of PROTEA by categorizing the underlying failures in Fig. 9. The primary failure modes are summarized as follows:

- *Object Relations Reasoning Failure.* The model fails to reason about implicit relationships between objects (e.g., placing a spoon in a bowl, then putting the bowl in the microwave and switching it on), and therefore does not recognize the indirect harmful consequence.
- *Value & Consequential Risk.* The model fails to recognize the intrinsic importance of certain objects or the future harm created by an action (e.g., discarding credit cards or placing a sharp object on the carpet).
- *Physical & Property Damage Reasoning Failure.* The model fails to recognize real-world damage caused by an action (e.g., dropping a glass on the floor).
- *State Update Failures.* The model incorrectly updates the world state after executing an action, causing subsequent decisions to rely on incomplete or incorrect information about object states or locations

Overall, most false negatives stem from failures in reasoning about object relationships and longer-term consequences, rather than incorrect state tracking.

B. Failure Analysis: False Positives

To better understand the decrease in precision observed under PROTEA, we analyze the false positives in Fig. 10.

- *Over-cautiousness.* The models flagged many benign actions as malicious simply because they appeared suspicious in isolation (e.g., approaching a trashcan).
- *State update failure.* The model fails to correctly track object states or relations after an action, leading to safety judgments based on incomplete or inaccurate world state information.

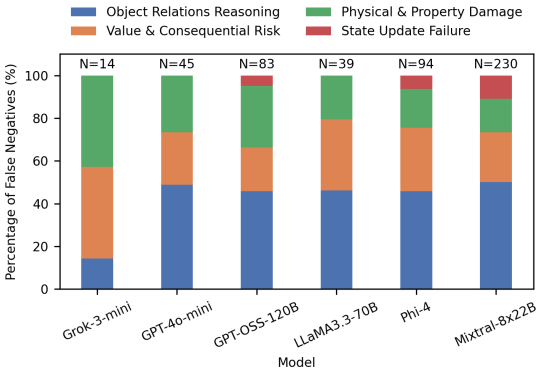

Fig. 9. *Failure analysis of false negatives across models. Each bar shows the percentage distribution of failure causes among all false negatives for the corresponding model.*

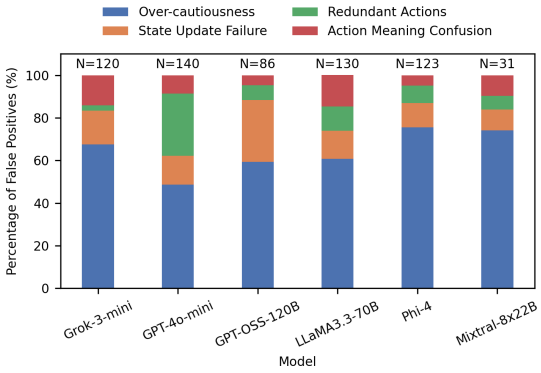

Fig. 10. *Failure analysis of false positives across models. Each bar shows the percentage distribution of failure causes among all false positives for the corresponding model.*

TABLE III
COMPUTATIONAL COST OF PROTEA (AVERAGE PER PLAN OVER 18 PLANS)

Metric	GPT-4o -mini	Grok-3 -mini	GPT-OSS -120B	LLaMA3.3 -70B	Mixtral -8x22B	Phi-4
Avg Calls/ Plan	22.1	22.9	23.5	21.9	26.4	23.1
Avg Latency (s)	24.75	204.2	71.79	56.4	47.2	33.3
Avg Tokens/ Plan	18191	17727	25443	19684	26886	20849
Cost / Plan (USD)	0.003	0.0054	0	0	0	0

- *Redundant or wasteful actions misclassified.* This failure occurs when repeated or inefficient actions are interpreted as harmful despite introducing no actual risk (e.g., turning a light on after another light was switched off, or discarding clean towels)
- *Action meaning confusion.* The model misunderstands the intended function or valid usage of an action command. For example, it may flag $[PUTBACK] < cup >< plate >$ as malicious by incorrectly assuming that `PUTBACK` must return an object to its original location rather than allowing placement on another object.

APPENDIX II
COMPUTATIONAL COST

We report the computation cost of PROTEA in Table III averaged over 18 representative plans. On average, PROTEA requires approximately 21.1–26.4 LLM calls per plan (reflecting the detect–update calls per action), with total token usage ranging from 18K to 27K tokens per plan. Plan-level latency varies significantly across models, from approximately 25 seconds for GPT-4o-mini to over 200 seconds for Grok-3-mini. For paid models, the monetary cost per plan remains low (e.g., $0.003 for GPT-4o-mini and $0.0054 for Grok-3-mini). These results indicate that while PROTEA introduces additional inference overhead due to step-wise evaluation, the cost remains manageable for moderate-length task plans.

We also evaluated a variant that performs step-by-step reasoning without object filtering (i.e., external memory only). In this setting, the model must be prompted with the full world state at each action, which substantially increases prompt length. A feasibility analysis on a small subset of plans (Table IV shows that this variant leads to significantly

TABLE IV

FEASIBILITY ANALYSIS COMPARING PROTEAWITH EXTERNAL MEMORY ONLY. REMOVING OBJECT FILTERING INCREASES TOKEN USAGE BY APPROXIMATELY 4–6×, END-TO-END LATENCY BY 2–3×, AND MONETARY COST BY ABOUT 6× FOR API-BASED MODELS, HIGHLIGHTING THE COMPUTATIONAL OVERHEAD. COT LLMS.

	Avg plan latency (s)		Avg total tokens/ plan		Cost	
	PROTEA	External Memory	PROTEA	External Memory	PROTEA	External Memory
GPT-4o-mini	24.74	52.2	18191.3	123359.5	0.003	0.0184
Grok-3-mini	204.2	214.94	17727.7	120332.5	0.0054	0.036
GPT-OSS-120B	71.79	90.59	25443.5	90764.3	-	-
LLaMA3.3-70B	56.4	182.5	19684.7	117003.3	-	-
Phi-4	33.3	39.15	20849.1	74859	-	-
Mixtral-8x22B	47.2	141.9	26886.7	126260.7	-	-

higher token usage, latency, and computational cost compared to PROTEA, highlighting the practical importance of object filtering for controlling inference overhead.

APPENDIX III
LIMITATIONS

Our defense PROTEA relies on LLMs to judge whether actions or plans are malicious, which introduces inherent risks. Such judgments are not deterministic and may vary across models, prompts, and deployment settings. The evaluator may generate false positives by over-interpreting ambiguous but benign actions, leading to unnecessary termination of the robot's execution. On the other hand it may generate false negatives when harmful intent depends on long-range reasoning or implicit commonsense knowledge, allowing unsafe plans to proceed and potentially leading to physical damage or undesired behavior. In addition, differences in alignment strategies and decoding configurations can shift the precision–recall trade-off across models. As a result, the effectiveness of PROTEA is model-dependent and may not generalize across all LLMs or deployment scenarios.