AURA: Affordance-Understanding and Risk-aware Alignment Technique for Large Language Models

Sayantan Adak[1], **Pratyush Chatterjee**[1], **Somnath Banerjee**[1,2], **Rima Hazra**[3], **Somak Aditya**[1], **Animesh Mukherjee**[1]

[1]Indian Institute of Technology Kharagpur, India
[2] Cisco Systems
[3] Eindhoven University of Technology, Netherlands

Abstract

Present day LLMs face the challenge of managing affordance-based safety risks—situations where outputs inadvertently facilitate harmful actions due to overlooked logical implications. Traditional safety solutions, such as scalar outcome-based reward models, parameter tuning, or heuristic decoding strategies, lack the granularity and proactive nature needed to reliably detect and intervene during subtle yet crucial reasoning steps. Addressing this fundamental gap, we introduce **AURA**, an innovative, multi-layered framework centered around Process Reward Models (PRMs), providing comprehensive, step level evaluations across logical coherence and safety-awareness. Our framework seamlessly combines introspective self-critique, fine-grained PRM assessments, and adaptive safety-aware decoding to dynamically and proactively guide models toward safer reasoning trajectories. Empirical evidence clearly demonstrates that this approach significantly surpasses existing methods, significantly improving the logical integrity and affordance-sensitive safety of model outputs. This research represents a pivotal step toward safer, more responsible, and contextually aware AI, setting a new benchmark for alignment-sensitive applications.

Extended version — https://arxiv.org/pdf/2508.06124

Introduction

In the physical world, certain situations may arise where performing certain actions (or *affordances*) may incidentally cause physical or materialistic harm to humans (or materials) involved in the situation. Imagine a situation where John is driving a car, steering along a busy road while simultaneously checking his phone to reply to a friend's text. In this scenario, while John, the *actor*, may physically *afford* to drive and type simultaneously; the action of *typing* (or messaging) may compromise his safety (or that of other passengers) by drawing his concentration away from the critical action (or affordance) of *driving* a car. Such hypothetical risks may arise even while John's action is being influenced by the suggestions of an LLM-based personal assistant. For example, if the LLM guiding John's AI assistant fails to recognize the implicit risk associated with responding immediately, *suggesting John reply right away*, it inadvertently facilitates a hazardous situation. We define this as an affordance-based safety risk, i.e., *situations in which model outputs implicitly encourage harmful actions due to overlooked contextual possibilities and their logical consequences* (Birr et al. 2024).

While LLMs continue to be very effective across a range of tasks, they increasingly expose critical vulnerabilities, particularly in their inability to adequately recognize and proactively address affordance-based safety risks alongside maintaining logical coherence (Son et al. 2025; Zhou et al. 2025). The intersection of coherence and affordance-sensitive safety becomes particularly crucial in nuanced, real-world contexts, where discerning potential harm demands meticulous, stepwise understanding of implicit logical implications. Affordances represent potential actions implicitly available in a given context, and affordance-based safety pertains to an LLM's capability to anticipate and manage scenarios where its outputs may inadvertently enable harm. Closely intertwined with coherence, the principle of maintaining logical consistency throughout the reasoning steps, ensuring each inference logically proceeds from the preceding one (Wang et al. 2025b). In high-stakes applications, such as healthcare, finance, automated decision-making, and social interactions (Zhai et al. 2025), overlooking affordance-based risks or failing to maintain coherence could propagate misinformation, confusion, or tangible harm (Son et al. 2025). Consequently, developing robust mechanisms to identify and mitigate these risks is not merely beneficial but essential (Zhang et al. 2025a).

Current approaches to managing these risks primarily involve retrospective corrections like flagging problematic outputs after their generation or scalar optimization strategies like Outcome-based Reward Models (ORMs) (Lyu et al. 2025).ORMs fall short in intricate logical reasoning tasks, where overlooking critical intermediate steps limits proactive intervention. Similarly, parameter tuning (Hazra et al. 2024; Banerjee et al. 2025a) and decoding heuristics (Banerjee et al. 2025b) face substantial limitations due to their rigidity and context insensitivity. To address these fundamental shortcomings, we advocate a paradigm shift toward detailed, stepwise logical reasoning assessment. Process Reward Models (PRMs) (Zhang et al. 2025b) have emerged as promising tools capable of delivering granular, multi-dimensional evaluations, covering logical coherence,

affordance-sensitive safety awareness, and proactive intervention opportunities. Our proposed system explicitly leverages PRMs through introspective chain-of-thought (CoT) refinement, granular intermediate reasoning assessments, and dynamic safety-aware decoding. These components collectively steer the model away from potentially hazardous or incoherent reasoning paths, embedding coherence and safety deeply into the model's reasoning processes. Our main contributions include the following.

① To the best of our knowledge, we are the first to introduce **AURA** , a unique affordance-aware PRM based alignment specifically tailored for fine-grained, step level evaluation of coherence and safety within complex, context-rich logical reasoning scenarios. Unlike prior PRM approaches, which predominantly target structured, mathematically-defined reasoning domains (Zhang et al. 2025b; Pala et al. 2025), **AURA** uniquely addresses nuanced affordance-sensitive risks in ambiguous, real-world complex situations, enabling proactive intervention in unsafe or incoherent reasoning trajectories.

② We curate a robust step-annotated dataset **SituationAfford**, comprising over $2,550$ unique situations, $7,506$ harm-intent queries, and $15,011$ annotated reasoning steps, constructed from realistic affordance contexts.

③ **First**, through extensive experiments, we demonstrate that **AURA** achieves state-of-the-art performance in step level safety and coherence classification. **Second**, the safety rate for our PRM-guided response generation is notably better compared to the base model generation across multiple models. **Finally**, **AURA** generalizes effectively to downstream generation tasks, reducing the relative Attack Success Rate (ASR) by up to **50%** on two different multi-turn jailbreak benchmarks—validated via both automatic and human evaluation.

Related Work

Safety-sensitive reasoning and coherence in LLMs: Recent work underscores the importance of coherent multi-step reasoning and safety in high-stakes LLM deployments. The *chain-of-thought monitorability* framework reveals CoT trace exposure as both essential and fragile for identifying unsafe reasoning (Korbak and Balesni 2025), while also highlighting trade-offs between failure detectability and language drift. Jiang et al. (2025) quantify safety risks in extended reasoning (e.g., math/code) and show that post-hoc classifiers often miss context-sensitive affordance violations. Broader safety reviews echo persistent vulnerabilities like prompt injection, misuse, and latent reasoning errors. In planning, affordances, *implicit action possibilities* are often overlooked, leading to unsafe outputs when models lack affordance-awareness (Zhang et al. 2025b; Choudhury 2025). Together, these insights advocate for real-time, inference-level safety interventions over reactive output filtering.

Reward modeling for process-aware reasoning: Alignment efforts have shifted from scalar Outcome Reward Models (ORMs) to Process Reward Models (PRMs) that assess reasoning steps for correctness, coherence, and safety (Zhang et al. 2025b). PRMs expose step level failures and outperform best-of-N baselines (Zhao et al. 2025). GenPRM enhances CoT with symbolic verification, surpassing GPT-4 on math tasks (Zhao et al. 2025); Athena-PRM extends PRMs to multimodal reasoning (Wang et al. 2025a), while R-PRM achieves parity with large models using only 15% of training data (She et al. 2025). DG-PRM applies Pareto-dominant reward trees for improved generalization (Yin et al. 2025), RetrievalPRM mitigates distribution shifts via trace similarity (Zhu et al. 2025), and SP-PRM combines process and outcome signals for 3.6–10.3% human-eval gains over ORM-only methods (Xie et al. 2025).

The Overall Architecture of AURA

We introduce **AURA**, a structured framework for affordance-based, risk-aware alignment in LLMs. **AURA** mitigates unsafe completions by intervening during the reasoning process itself, targeting stepwise errors arising from misaligned affordances or incoherent logic. Instead of relying on post-hoc filtering, **AURA** takes a two-staged approach: (i) a self-critique-guided reasoning loop that revises candidate responses based on safety-oriented feedback, and (ii) a reward-based trajectory selection mechanism that ranks reasoning paths using a specialised process reward model – AFFORDRANKER. This integrated approach enables the model to generate trajectories that are both contextually coherent and aligned with situational affordances. We describe the overall methodology, the construction of the **SituationAfford** dataset, and the training process in the following sections.

Preliminaries

We conceptualize the reasoning process in **AURA** as a structured decision-making task, where the language model operates as a policy over an abstract environment defined by natural language contexts. Formally, we define the state space $\mathcal{S}$ such that each state $s \in \mathcal{S}$ is a tuple $(S, Q, \mathcal{H})$, comprising a textual situation S, a query Q, and a reasoning history $\mathcal{H} = \{r_1, \ldots, r_{j-1}\}$ of prior steps. The action space $\mathcal{A}$ consists of atomic reasoning steps r_j, where each step represents a single proposition that advances the reasoning toward answering the query. The reward space is captured by a structured function $\mathcal{R}_p(s, a_j) = (E_{pc}, E_{av})$, where E_{pc} denotes the procedural coherence score—reflecting logical consistency with earlier steps—and E_{av} denotes the affordance validation score—indicating whether the step respects contextual safety constraints. The policy model $\mathcal{M}_p$, instantiated as an instruction-tuned LLM, takes a state s and stochastically generates the next action $a_j \in \mathcal{A}$, corresponding to a reasoning step r_j, conditioned on the current situation, query, and step history. A complete reasoning trajectory $R = \{r_1, \ldots, r_t\}$ is thus a sequence of such actions, which we evaluate using a Process Reward Model (PRM)—referred to as AFFORDRANKER. This model computes per-step error vectors $(\mathbf{ev}_j)$ and aggregates them to produce a cumulative score $\mathcal{RW}(R) = \frac{1}{t}\sum_{j=1}^{t}(E_{\text{pc}}^{(j)} + E_{\text{av}}^{(j)})$, which is used to rank and select the most coherent and safe trajectory.

Overall Framework

AURA follows a two-stage alignment framework. First, it produces self-critique-conditioned reasoning trajectories via $\mathcal{M}_p$ that reflect safety feedback from itself (self-critique). Then, it ranks and selects the most reliable response using a reward model trained to capture stepwise affordance coherence. Each stage is detailed below.

Self-critique-conditioned reasoning: We construct self-critique (Wen et al. 2025; Valmeekam, Marquez, and Kambhampati 2023; Gou et al. 2024) conditioned reasoning trajectories by prompting the policy model $\mathcal{M}_p$, which we define as the language model responsible for generating stepwise reasoning given a situation and a query, with its own safety feedback. Instead of applying iterative corrections, we embed the critique as a conditioning signal within the input prompt. Given a situation S and a query Q, the policy model M_p generates two initial reasoning trajectories R_1^0 and R_2^0, each comprising a stepwise explanation and a final answer. These responses serve as first-pass attempts, which the model then critiques to identify potential flaws in reasoning and affordance violations. It produces a critique rationale $\mathcal{RS}$ and a refined answer A, which we append to the original inputs to form an augmented prompt $\mathcal{P}_{aug}$. This prompt encodes safety-aware preferences and guides the generation of improved candidate responses in the subsequent reward-based selection phase. This phase is shown in Algorithm 1 (see Step 1).

Reward-based trajectory selection: Once we have the augmented prompt $\mathcal{P}_{aug}$ through self-critique, we sample a set of N candidate reasoning trajectories $R_1^1, R_2^1, \ldots, R_N^1$ from the policy model $\mathcal{M}_p$. Each trajectory R_i^1 consists of a sequence of t reasoning steps that attempt to answer the given query based on the provided situation. To evaluate and select the most reliable trajectory, we use AFFORDRANKER, which performs step level reward assessment focused on reasoning quality and affordance alignment. AFFORDRANKER takes each reasoning step r_j from a trajectory R_i^1 and produces two scalar values: a procedural coherence score E_{pc} that measures the logical consistency of r_j with respect to the prior reasoning steps, and an affordance validation score E_{av} that quantifies how well r_j aligns with the contextual constraints and affordances present in the situation. We provide detailed definitions and modeling of *procedural coherence score* and *affordance validation score* in the next subsection. For each trajectory, we compute the cumulative reward $\mathcal{RW}(R_i^1)$ as the average of the total stepwise scores (see line number 17-21 in Algorithm 1).

The Design of The AFFORDRANKER

A Process Reward Model (PRM) evaluates multi-step reasoning chain of a response by assigning rewards at the level of individual reasoning steps (see Step 2 of Algorithm 1). Unlike conventional reward models that assess the final output in isolation, a PRM operates over the full trajectory, scoring each intermediate step based on its contribution to coherent, goal-directed reasoning. Given a query Q and a sequence of reasoning steps $r_1, r_2, \ldots, r_t$ of a response R generated by a policy model, the PRM computes a reward

Algorithm 1: Overall framework of **AURA**

Input: Situation S, Query Q, policy model $\mathcal{M}_p$
Output: $\boldsymbol{R}_{\text{best}}$

1: Step 1: Self-critique-conditioned reasoning.
2: // Generate initial independent responses
3: $R_1^0, R_2^0 \leftarrow \mathcal{M}_p(S, Q)$
4: // Obtain critique rationale $\mathcal{RS}$ and answer
5: $\mathcal{RS}, A \leftarrow \mathcal{M}_p(S, Q, R_1^0, R_2^0)$
6: // Augment base prompt with self critique
7: $\mathcal{P}_{aug} \leftarrow \text{concat}(S, Q, R_1^0, R_2^0, \mathcal{RS}, A)$
8: Train AFFORDRANKER
9: $\mathcal{M}_{aff} \leftarrow train\_PRM\_model()$;
10: Step 2: Reward based trajectory selection
11: // Generate N number of independent responses
12: $\{R_1^1, R_2^1, \cdots R_N^1\} \leftarrow \mathcal{M}_p(\mathcal{P}_{aug})$
13: // Obtain the rewards for the responses
14: **for** $i = 1$ to N **do**
15: $E_{PC} = 0, E_{AV} = 0$
16: **for** $j = 1$ to t **do**
17: // Rewards for each step r_j of response R_i^1
18: $E_{pc}, E_{av} \leftarrow \mathcal{M}_{aff}(r_j)$
19: E_{PC} += E_{pc}, E_{AV} += E_{av}
20: **end for**
21: $\mathcal{RW}(R_i^1) = \frac{1}{t}(E_{PC} + E_{AV})$
22: **end for**
23: // Rank the responses $\{R_1^1, R_2^1, \ldots, R_N^1\}$ by descending final reward score $\mathcal{RW}(R_i^1)$
24: $R_{\text{best}} \leftarrow \arg\max_i \mathcal{RW}(R_i^1)$
25: return $\boldsymbol{R}_{\text{best}}$

for step r_j as

$$R_j = \text{PRM}(Q, r_1, \ldots, r_t) \quad (1)$$

Here, PRM denotes the process reward model, and R_j reflects how well the j-th step maintains logical consistency with previous steps and supports progress toward answering the query. By providing step level supervision, PRMs enable fine-grained control over the reasoning process and facilitate more robust alignment than output-only evaluation schemes.

While traditional PRMs combine multiple reasoning signals into a single scalar, we explicitly disentangle reward types to capture distinct dimensions of reasoning quality. This separation allows us to diagnose both the nature of coherence errors and their implications for safety. Specifically, we define two primary error categories, each associated with a corresponding reward component: – (a) *procedural coherence errors* (E_{pc}): This error reflects a breakdown in the progression of reasoning steps. It arises when a step deviates from the expected inferential path—by skipping intermediate steps, introducing unsupported conclusions, or failing to preserve continuity with prior context; (b) *affordance violation errors* (E_{av}): A reasoning step that introduces or relies on an affordance likely to result in unsafe or harmful outcomes. This includes assumptions about actions or entities that, while possible, violate implicit safety constraints of the environment or task.

By assigning rewards along these two axes, our model provides a structured and interpretable evaluation of each

reasoning step. We train the AFFORDRANKER using a carefully curated dataset **SituationAfford**, which we describe next.

The SituationAfford Dataset

We construct the **SituationAfford** dataset with fine-grained, step level annotations to train AFFORDRANKER (denoted by $\mathcal{M}_{aff}$) used in Step 2 of our framework. Each reasoning step r_j in a trajectory is annotated with a binary error vector $\mathbf{ev}_j = (E_{pc}, E_{av})$, where $E_{pc}, E_{av} \in \{0, 1\}$ indicate whether the step violates procedural coherence ($E_{pc} = 1$) or safety affordance ($E_{av} = 1$), respectively. The overall dataset creation process involves: (i) textual situation generation, (ii) query generation and categorisation, (iii) step level **SituationAfford** dataset annotation, and targeted data augmentation. These annotated trajectories enable $\mathcal{M}_{aff}$ to learn step level reward signals that are later aggregated into the final trajectory score $\mathcal{RW}(R_i^1)$, used for ranking in the inference stage.

(a) Situation generation: To construct the **SituationAfford** dataset, we leverage two existing resources: **(a)** MSSBench (Zhou et al. 2024) and **(b)** Text2Afford (Adak et al. 2024). From MSSBench, we extract 186 unique unsafe contexts originally designed for multimodal safety evaluations. Complementarily, we incorporate 2,369 natural language descriptions from Text2Afford, each encoding an object-centric affordance context suitable for generating diverse situational prompts. To enrich these contexts with realistic dynamics, we design a generation prompt that expands each input into a 250-word scene narrative[1] having two human activities. This setup encourages affordance conflict situations where the action of one agent implicitly limits or contradicts the other's. Such interactions naturally surface violations of implicit safety norms and expose weaknesses in step-wise reasoning. These generated narratives, combined with curated unsafe contexts, form the basis of the procedural and affordance-related errors E_{pc} and E_{av} at the step level, as noted in the previous section.

(b) Query generation and categorization: For each generated situation S, we construct prompts to elicit sensitive queries spanning three harm intent categories, adapted from MSSBench (Zhou et al. 2024): (i) *goal-based*, (ii) *property damage*, and (iii) *physical harm*. We define an affordance violation as any case where the model implicitly assumes that an unsafe or infeasible action is valid within the given physical or situational context. The intent categories can be briefly described as follows –*goal-based* queries are generic and task-oriented without explicit harmful intent, *property damage* queries imply intentions to harm objects or surroundings, while *physical harm* queries encourage actions that could endanger individuals, even if the risk is only implicit. This structured categorization supports targeted evaluation of affordance failures across varying levels of risk. For each situation, we generate three queries per harm category using a prompt that includes a single in-context example from the corresponding MSSBench category. We iteratively refine these prompts through controlled manual tuning to ensure semantic coherence and category alignment, consistent with standard few-shot prompting practices (Le Scao and Rush 2021; Liu et al. 2021; Zhao et al. 2021). To ensure quality, we apply automated filtering using GPT-4o (OpenAI and Team 2024) to remove non-harmful or trivial outputs, followed by manual verification to validate their alignment with the intended harm categories.

(c) Reasoning trajectory generation: Given a situation-query (S-Q) pair, we construct prompts to elicit two alternate reasoning trajectories from the policy model $\mathcal{M}_p$. Each trajectory consists of up to ***seven*** steps[2], where each step expresses a single, concise fact or action that is logically consistent with prior steps and advances the response toward answering the query. Steps must incorporate both explicit and implicit elements from the context, avoid redundancy, and reflect a distinct human activity. We generate the entire sequence in a single pass using a structured prompt.

(d) Stepwise label annotation and data augmentation: For each generated trajectory, we annotate every reasoning step with two binary labels: E_{pc} for procedural coherence errors and E_{av} for affordance violation errors. We use GPT-4o to generate these annotations synthetically. To label a specific step r_j, we construct an input prompt that includes the situation, the query, and all preceding steps $r_1, \ldots, r_{j-1}$. We iteratively refine the prompt design to ensure that the annotations are both consistent and aligned with our reward signal definitions. To further ensure label reliability, we apply a secondary verification step using an LLM-as-a-judge framework, where we discard the instances having at least two incorrect step level judgements. Samples with label vectors that conflict with the judge model's decision are discarded to maintain overall dataset quality.

The final **SituationAfford** dataset comprises 2550 unique situations, 7506 harm-intent queries, and 15011 annotated reasoning trajectories, resulting in a total of 208,862 step level annotations. After the annotation was completed, we found an imbalance between positive ($E_{pc} = 1/E_{av} = 1$) and negative ($E_{pc} = 0/E_{av} = 0$) instances across both error dimensions. To mitigate this, we perform label balancing by downsampling overrepresented positive samples and augmenting the underrepresented negative class. For augmentation, we perturb selected valid reasoning steps to introduce incoherence or affordance-inconsistent behavior, preserving grammaticality while injecting subtle logical or contextual flaws. This augmentation strategy improves the model's ability to detect nuanced failures in reasoning, especially within affordance-sensitive scenarios. We partition the dataset in an 8:1:1 split for training, validation, and test sets, respectively.

Manual validation: To assess the reliability of our synthetic annotations, we conduct a human validation study using the Prolific platform. A total of 33 qualified annotators participated in the process. We randomly sampled 50 situation-query-response instances from each category of

[1]We use 250-word situations to provide sufficient context for multi-step safety reasoning and prevent refusal from LLM

[2]Each trajectory contains *seven* reasoning steps to ensure consistent supervision and capture evolving safety dynamics without excessive annotation overhead.

queries from our dataset, and got them independently annotated by three annotators. Annotators had to first determine whether the query was harmful given the situation, and then evaluate the response across four dimensions: *safe and helpful*, *unsafe but helpful*, *safe but not helpful*, and *unsafe and not helpful*. We observe high agreement on query harmfulness (Fleiss' $\kappa = 0.83$) and moderately high agreement on response evaluation (Fleiss' $\kappa = 0.62$), with the query identified as harmful for 86% of the cases and the most frequent response label being *safe and helpful*. In 87% of cases, annotators marked the response as helpful (safe + unsafe). These results validate the integrity of our dataset and confirm that our affordance-sensitive annotations reflect human-aligned safety and helpfulness judgments. Instruction to the annotators is provided in Section D in the supplementary material.

Training Procedure

We implement our AFFORDRANKER ($\mathcal{M}_{aff}$) using Qwen-2.5-7B-instruct (Qwen et al. 2025) as the base model, chosen for its strong performance on tasks involving multi-step, safety-aware reasoning. Unlike prior approaches that replace the language modeling head with a scalar regression head (Zhang et al. 2025b; Xia et al. 2025; Tan et al. 2025), we preserve the model's original architecture to maintain its generative flexibility. To support step level supervision, we extend the tokenizer with two additional control tokens, $\langle + \rangle$ and $\langle - \rangle$, used to mark positive and negative labels for individual steps during training. To train $\mathcal{M}_{aff}$, we convert each annotated reasoning trajectory into a set of supervised instances. For a trajectory with t steps, we extract $(t-1)$ training instances by iterating over each step r_j for $j = 2$ to t. Each instance includes the situation S, the query Q, the sequence of prior steps $r_1, \ldots, r_{j-1}$, and the current step r_j. The model is trained to generate a label vector (E_{pc}, E_{av}) for r_j based on this context. We apply this formulation consistently across all harm categories (i.e., *goal-based*, *property damage*, and *physical harm* scenarios).

Evaluation

We evaluate the effectiveness of **AURA** in guiding safer and more coherent reasoning using both our curated dataset and external multi-turn safety benchmarks. Our evaluation comprises three components: (i) step level prediction accuracy of AFFORDRANKER, (ii) quality of reward-guided safe response generation, and (iii) the defense capability against multi-turn jailbreak attacks.

Evaluation of AFFORDRANKER: We first assess the performance of AFFORDRANKER using the **SituationAfford** dataset. We construct two evaluation setups as follows – *balanced setting*: This setup contains an equal number of positive and negative test instances (16,422 annotated steps) for each reasoning dimension (*coherence* and *safety*) matching the class distribution used during training, and *imbalanced setting*: This setup reflects real-world conditions without explicit balancing, where naturally occurring class imbalance is preserved (20,848 annotated steps). We use F1-score as the metric for each label category – *coherence* and *safety* – under both balanced and imbalanced evaluation settings.

Reward-guided safe response generation: Given a situation and query, we use AFFORDRANKER to score multiple reasoning trajectories and select the one with the highest cumulative reward (using the approach mentioned in line 24 of the Algorithm 1), enabling search-based decoding aligned with coherence and safety. We evaluate *safety rate* across varying depths (e.g., **AURA**@2, @4, @8), and compare against baseline generation, and **AURA** without self-critique. Safety rate is computed by comparing a model's generated response against the two annotated 7-step trajectories from **SituationAfford**. For base generations, we directly compare to the ground truth; for AFFORDRANKER-guided search, we select the top-ranked response (as described in Algorithm 1). An *LLM-as-judge* (`DeepSeek-R1-Distill-Qwen-32B`) is used to determine whether the output is equally safe or safer than ground truth, defining safety rate as

$$\textit{safety rate} = \frac{\text{\#responses judged as safe or safer}}{\text{\#total number of instances}} \tag{2}$$

Defense capabilities: To assess the generalisation and real-world robustness, we apply our reward-guided inference to two benchmarks datasets – **CoSafe** (Yu et al. 2024) and **STREAM** (Kuo et al. 2025). **CoSafe** does not include step level annotations and so we adapt it for our evaluation by treating each dialogue-query pair as input and measure the ASR[3]. **STREAM** offers more diverse and compositional attack types across varying dialogue lengths. Here again, we evaluate ASR under different response generation strategies.

For both these datasets, in order to measure ASR, we score each response by GPT-4o on a scale from 1 (extremely unsafe) to 10 (completely safe). We count an attack as successful if the score is below 5. Thus, we have

$$\text{ASR} = \frac{\text{\# responses with score} < 5}{\text{Total number of responses}}. \tag{3}$$

This metric quantifies how often a model fails to resist adversarial or harmful queries.

Experimental Setup

We conduct extensive evaluations using a diverse set of baselines and model configurations to benchmark the performance of **AURA** across both step level safety classification and reward-guided safer response generation tasks.

Baselines for step level evaluation: We compare **AURA**'s AFFORDRANKER for coherence and safety classification, against multiple strong discriminative PRMs as follows – `PathFinder-PRM-7B`, (Pala et al. 2025), `RLHFlow-Mistral-8B`, `RLHFlow-DeepSeek-8B`, and `ReasonEval-7B` (Xia et al. 2025). We follow a thresholding approach to convert predicted reward scores into binary labels, consistent with prior work such as (Pala et al. 2025). None of these models are explicitly optimized for safety-sensitive affordance reasoning. To compare against models tuned explicitly for

[3]Safety rate is not applicable here due to the absence of annotated $\mathbf{ev}_j$ vectors.

Category	Model	Imbalanced		Balanced		Overall
		Coherence	Safety	Coherence	Safety	
Discriminative process reward models						
	ReasonEval-7B	0.62	0.32	0.64	0.36	0.48
	RLHFlow-Mistral-8B	0.54	0.31	0.58	0.36	0.45
	RLHFlow-DeepSeek-8B	0.67	0.33	0.65	0.32	0.49
	PathFinder-PRM-7B	0.56	0.35	0.68	0.37	0.49
Safety aligned reward models						
	Beaver-7B-v1.0-cost	–	0.55	–	0.59	0.57
	Beaver-7B-v3.0-cost	–	0.64	–	0.65	0.65
	AURA (Ours)	0.83*	0.81*	0.88*	0.82*	0.83*

Table 1: Step level performance (F1 score) comparison of models under balanced and imbalanced settings. Best results are highlighted. * indicates statistically significant improvement from the best baseline using *Mann-Whitney U test* with $p < 0.05$

Policy model	Base generation	AURA$_{\setminus \text{Self-critique}}$	AURA@2	AURA@4	AURA@8
qwen2.5(7b)-inst	0.28	0.34*	0.52*	0.67*	0.71*
llama-3.1(8b)-inst	0.11	0.15*	0.36*	0.56*	0.69*
mistral(7b)-v0.3-inst	0.18	0.23*	0.57*	0.65*	0.67*
internlm3(8b)-inst	0.42	0.45*	0.73*	0.78*	0.80*
gemma-2-9b-it	0.11	0.14*	0.26*	0.36*	0.41*

Table 2: Safety rate for reward-guided safer response generation. Higher is better. Best results are highlighted. * indicates statistically significant improvement from base response.

safety, we consider Beaver-7B-v1.0-cost and Beaver-7B-v3.0-cost (Dai et al. 2024), trained on human preference data emphasizing safe response generation.

Policy models for reward-guided generation: We use five medium-sized (7B–9B) instruction-tuned LLMs as base policy models for the reward-guided safer response generation task – mistral(7b)-v0.3-inst, gemma2(9b)-it, internlm3(8b)-inst, qwen2.5(7b)-inst, and llama3.1(8b)-inst. We set the decoding parameters as follows: temperature = 0.7, top-p = 0.95, and maximum tokens = 512. For self-critique generation, we reuse the policy model itself. For PRM-guided decoding, we generate $k \in \{2, 4, 8\}$ response trajectories and select the one with the highest average PRM score as the final output.

Defense capability: For the ASR evaluation, we use the same five base models and generation settings as above. For AFFORDRANKER-guided inference, we sample 8 response candidates per query and apply step level reward scoring to rank and select the most coherent and safe response.

Results

Step level evaluation: Table 1 presents a comparative analysis of various models on step level coherence and safety classification under both imbalanced and balanced settings. The proposed **AURA** model achieves the best overall performance, substantially outperforming both discriminative baselines and safety-aligned models, with F1 scores of 0.88 for coherence and 0.82 for safety in the balanced setting. Crucially, **AURA** maintains consistently high performance even in the *imbalanced* evaluation scenario—closely mirroring real-world distributions—where it still achieves F1-scores of 0.83 (coherence) and 0.81 (safety) respectively. This highlights **AURA**'s strong generalization ability and resilience to label skew, a key requirement for deployment in safety-critical settings. Among discriminative PRMs, models such as PathFinder-PRM-7B and RLHFlow-DeepSeek-8B show moderate effectiveness, with overall F1-scores below 0.70, indicating limited capacity to capture nuanced affordance-sensitive violations. Safety-aligned models like Beaver-7B-v3.0-cost perform reasonably well on safety detection (0.65), but do not support coherence assessment, as they are not trained for multi-step reasoning. **Reward-guided response generation**: Table 2 reports the safety rate of model responses under different generation strategies across five diverse policy models. We observe a consistent and substantial improvement in safety when integrating both self-critique and PRM-guided search over the base generation. The base models, when used alone, yield relatively low safety rates (e.g., 0.11 for llama-3.1(8b)-inst and gemma-2(9b)-it), highlighting their vulnerability to unsafe completions despite instruction tuning. Incorporating self-critique offers modest gains across all models (e.g., +0.19 for qwen2.5(7b)-inst, +0.23 for internlm3(8b)-inst), but this effect is significantly amplified when coupled with AFFORDRANKER. Specifically, using $k = 8$ yields the highest safety rates across all models, with internlm3(8b)-inst reaching 0.80 (+0.38) and qwen2.5(7b)-inst reaching 0.71 (+0.43).

Model	CoSafe			STREAM			SituationAfford		
	Base	AURA$_{\text{\Self-critique}}$	AURA	Base	AURA$_{\text{\Self-critique}}$	AURA	Base	AURA$_{\text{\Self-critique}}$	AURA
`qwen2.5(7b)-inst`	0.15	0.12*	0.08*	0.2	0.14*	0.12*	0.39	0.33*	0.22*
`llama3.1(8b)-inst`	0.16	0.12*	0.1*	0.16	0.12*	0.09*	0.42	0.35*	0.21*
`mistral(7b)-v0.3-inst`	0.18	0.15*	0.13*	0.19	0.15*	0.11*	0.45	0.39*	0.24*
`internlm3(8b)-inst`	0.14	0.12*	0.09*	0.13	0.1*	0.06*	0.37	0.29*	0.18*
`gemma-2(9b)-it`	0.24	0.14*	0.08*	0.23	0.19*	0.12*	0.46	0.42*	0.36*

Table 3: Average Attack Success Rate (ASR) across models using **CoSafe**, **STREAM** benchmarks, and SituationAfford. Lower is better. Best results are highlighted. * indicates statistically significant improvement from base response.

These results confirm the effectiveness of the AURA framework in navigating toward safer reasoning trajectories through reward-guided decoding. Finally, the steady upward trend from AURA@2 to AURA@8 further supports the hypothesis that deeper sampling coupled with step level scoring leads to more reliable safety alignment.

General defense capabilities: Table 3 reports the ASR across three benchmarks: **CoSafe**, **STREAM**, and the **SituationAfford** dataset. Across all models and datasets, we observe a consistent reduction in ASR when incorporating our PRM, with the full AURA pipeline achieving the lowest ASR in every setting. Improvements are particularly pronounced on the **SituationAfford** benchmark, where AURA reduces ASR by up to 50% relative to the base model, underscoring its effectiveness in affordance-sensitive safety scenarios. The performance gap between only PRM and AURA further highlights the synergistic benefit of integrating reward-guided decoding with self-critique.

Manual evaluation: To complement the automated safety rate analysis, we conduct a manual evaluation using a random sample of total 100 response pairs across all policy models, comparing base generation and AURA@8 outputs. 5 human annotators with prior experience in LLM safety assessment, recruited via *Prolific*, independently assess which response in each pair is safer, based on the context and query. Aggregated results show that responses guided by AFFORDRANKER are judged safer in **81%** of the comparisons, with substantial agreement among annotators (Fleiss' $\kappa = 0.72$). This supports the reliability of our reward-guided decoding strategy in producing safer, more aligned completions.

Similarly, to validate the reliability of automated ASR scoring, for each dataset, we randomly sample 50 instances—25 generated by the base model and 25 by AURA—and present them in randomized, blinded pairs to 10 independent annotators. Annotators are asked to select the response that appears safer in each pair. Aggregated results show that AURA-generated responses are preferred in 82% of the comparisons, closely aligning with the automated ASR trends.

Discussion: We conduct a detailed analysis of AFFORDRANKER's behavior across reasoning steps and error modes. Stepwise performance reveals a U-shaped trend, with the highest safety prediction accuracy at the beginning and end of reasoning chains, and a dip in intermediate steps—likely due to evolving context and affordance ambiguity. A category-wise error breakdown shows lowest misclassification in goal-based queries, and higher errors in property damage and physical harm scenarios, suggesting challenges in recognizing subtle or implicit risks. Qualitative inspection highlights recurring failure modes, including difficulty with latent hazards, multitasking affordance conflicts, and fluency of the surface-level text.

Runtime analysis: AURA adds minimal overhead to standard LLM inference. Given L layers, hidden size as d, sequence length as N, single-sample inference scales as $\mathcal{O}(L \cdot N^2 \cdot d)$. For self-critique, we perform three short generations, costing roughly $3 \times \mathcal{O}(L \cdot N^2 \cdot d)$. Next, we generate K full trajectories in a single batched forward pass, scaling as $K \times \mathcal{O}(L \cdot N^2 \cdot d)$, but with GPU parallelism, the latency remains close to a single decode. The reward scoring is lightweight and adds a small constant-time selection overhead. Overall, AURA's total cost is approximately $(3 + K) \times \mathcal{O}(L \cdot N^2 \cdot d)$, with practical latency dominated by one batched decode. For small K (typically ≤ 8), this achieves a favourable safety-efficiency trade-off.

General utilities: As AURA operates purely at inference time without modifying model weights, it is expected to preserve the LLM's general utility. We confirm this by evaluating `LLaMA-3.1-8B` (5-shot) on MMLU and ARC-Challenge. Across 3 runs, the baseline achieved accuracies of 65.6–66.2 (MMLU) and 79.2–79.7 (ARC), while AURA yielded accuracies of 65.2–65.9 and 79.1–79.5, respectively—showing no significant degradation.

Conclusion

We introduced AURA, an affordance-aware, risk-sensitive alignment framework leveraging process-level supervision via Process Reward Models (PRMs) for LLM reasoning. By integrating introspective self-critique, fine-grained PRM assessments, and adaptive safety-aware decoding, AURA dynamically steers reasoning toward safer trajectories, outperforming traditional scalar outcome-based reward models and heuristic approaches. We developed the **SituationAfford** dataset, comprising fine-grained step annotations across diverse situational contexts, demonstrating state-of-the-art performance in step-level safety and coherence. At inference, our PRM-guided approach significantly improved safe response rates and reduced attack success across multi-turn safety benchmarks. Future directions include explicit modeling of latent and multi-agent risks via affordance graphs, expanding multimodal integration to further enhance reliability in safety-critical applications.

References

Adak, S.; Agrawal, D.; Mukherjee, A.; and Aditya, S. 2024. Text2Afford: Probing Object Affordance Prediction abilities of Language Models solely from Text. In *Proceedings of the 28th Conference on Computational Natural Language Learning*, 342–364.

Banerjee, S.; Layek, S.; Chatterjee, P.; Mukherjee, A.; and Hazra, R. 2025a. Soteria: Language-Specific Functional Parameter Steering for Multilingual Safety Alignment. arXiv:2502.11244.

Banerjee, S.; Layek, S.; Tripathy, S.; Kumar, S.; Mukherjee, A.; and Hazra, R. 2025b. SafeInfer: Context Adaptive Decoding Time Safety Alignment for Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(26): 27188–27196.

Birr, T.; Pohl, C.; Younes, A.; and Asfour, T. 2024. AutoGPT+P: Affordance-based Task Planning using Large Language Models. In *Robotics: Science and Systems XX*, RSS2024. Robotics: Science and Systems Foundation.

Choudhury, S. 2025. Process Reward Models for LLM Agents: Practical Framework and Directions. arXiv:2502.10325.

Dai, J.; Pan, X.; Sun, R.; Ji, J.; Xu, X.; Liu, M.; Wang, Y.; and Yang, Y. 2024. Safe RLHF: Safe Reinforcement Learning from Human Feedback. In *The Twelfth International Conference on Learning Representations*.

Gou, Z.; Shao, Z.; Gong, Y.; Shen, Y.; Yang, Y.; Duan, N.; and Chen, W. 2024. CRITIC: Large Language Models Can Self-Correct with Tool-Interactive Critiquing. In *ICLR*.

Hazra, R.; Layek, S.; Banerjee, S.; and Poria, S. 2024. Safety Arithmetic: A Framework for Test-time Safety Alignment of Language Models by Steering Parameters and Activations. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 21759–21776. Miami, Florida, USA: Association for Computational Linguistics.

Jiang, F.; Xu, Z.; Li, Y.; Niu, L.; Xiang, Z.; Li, B.; Lin, B. Y.; and Poovendran, R. 2025. SafeChain: Safety of Language Models with Long Chain-of-Thought Reasoning Capabilities. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Findings of the Association for Computational Linguistics: ACL 2025*, 23303–23320. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-256-5.

Korbak, T.; and Balesni, M. 2025. Chain of Thought Monitorability: A New and Fragile Opportunity for AI Safety. arXiv:2507.11473.

Kuo, M.; Zhang, J.; Ding, A.; DiValentin, L.; Hass, A.; Morris, B. F.; Jacobson, I.; Linderman, R.; Kiessling, J.; Ramos, N.; et al. 2025. SafeTy Reasoning Elicitation Alignment for Multi-Turn Dialogues. *arXiv preprint arXiv:2506.00668*.

Le Scao, T.; and Rush, A. 2021. How many data points is a prompt worth? In Toutanova, K.; Rumshisky, A.; Zettlemoyer, L.; Hakkani-Tur, D.; Beltagy, I.; Bethard, S.; Cotterell, R.; Chakraborty, T.; and Zhou, Y., eds., *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2627–2636. Online: Association for Computational Linguistics.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2021. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. arXiv:2107.13586.

Lyu, C.; Gao, S.; Gu, Y.; Zhang, W.; Gao, J.; Liu, K.; Wang, Z.; Li, S.; Zhao, Q.; Huang, H.; Cao, W.; Liu, J.; Liu, H.; Liu, J.; Zhang, S.; Lin, D.; and Chen, K. 2025. Exploring the Limit of Outcome Reward for Learning Mathematical Reasoning. arXiv:2502.06781.

OpenAI; and Team. 2024. GPT-4o System Card. arXiv:2410.21276.

Pala, T. D.; Sharma, P.; Zadeh, A.; Li, C.; and Poria, S. 2025. Error Typing for Smarter Rewards: Improving Process Reward Models with Error-Aware Hierarchical Supervision. arXiv:2505.19706.

Qwen; :; Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; Lin, H.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Lin, J.; Dang, K.; Lu, K.; Bao, K.; Yang, K.; Yu, L.; Li, M.; Xue, M.; Zhang, P.; Zhu, Q.; Men, R.; Lin, R.; Li, T.; Tang, T.; Xia, T.; Ren, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; and Qiu, Z. 2025. Qwen2.5 Technical Report. arXiv:2412.15115.

She, S.; Liu, J.; Liu, Y.; Chen, J.; Huang, X.; and Huang, S. 2025. R-PRM: Reasoning-Driven Process Reward Modeling. arXiv:2503.21295.

Son, Y.; Kim, M.; Kim, S.; Han, S.; Kim, J.; Jang, D.; Yu, Y.; and Park, C. 2025. Subtle Risks, Critical Failures: A Framework for Diagnosing Physical Safety of LLMs for Embodied Decision Making. arXiv:2505.19933.

Tan, X.; Yao, T.; Qu, C.; Li, B.; Yang, M.; Lu, D.; Wang, H.; Qiu, X.; Chu, W.; Xu, Y.; and Qi, Y. 2025. AURORA:Automated Training Framework of Universal Process Reward Models via Ensemble Prompting and Reverse Verification. arXiv:2502.11520.

Valmeekam, K.; Marquez, M.; and Kambhampati, S. 2023. Investigating the Effectiveness of Self-critiquing in LLMs solving Planning Tasks. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*.

Wang, S.; Liu, Z.; Wei, J.; Yin, X.; Li, D.; and Barsoum, E. 2025a. Athena: Enhancing Multimodal Reasoning with Data-efficient Process Reward Models. arXiv:2506.09532.

Wang, T.; Jiang, Z.; He, Z.; Tong, S.; Yang, W.; Zheng, Y.; Li, Z.; He, Z.; and Gong, H. 2025b. Towards Hierarchical Multi-Step Reward Models for Enhanced Reasoning in Large Language Models. arXiv:2503.13551.

Wen, X.; Lou, J.; Lu, X.; Yang, J.; Liu, Y.; Lu, Y.; Zhang, D.; and Yu, X. 2025. Scalable Oversight for Superhuman AI via Recursive Self-Critiquing. arXiv:2502.04675.

Xia, S.; Li, X.; Liu, Y.; Wu, T.; and Liu, P. 2025. Evaluating Mathematical Reasoning Beyond Accuracy. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(26): 27723–27730.

Xie, B.; Xu, B.; Yuan, Y.; Zhu, S.; and Shen, H. 2025. From Outcomes to Processes: Guiding PRM Learning from ORM for Inference-Time Alignment. arXiv:2506.12446.

Yin, Z.; Sun, Q.; Zeng, Z.; Cheng, Q.; Qiu, X.; and Huang, X. 2025. Dynamic and Generalizable Process Reward Modeling. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 4203–4233. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-251-0.

Yu, E.; Li, J.; Liao, M.; Wang, S.; Zuchen, G.; Mi, F.; and Hong, L. 2024. CoSafe: Evaluating Large Language Model Safety in Multi-Turn Dialogue Coreference. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 17494–17508. Miami, Florida, USA: Association for Computational Linguistics.

Zhai, W.; Liao, J.; Chen, Z.; Su, B.; and Zhao, X. 2025. A Survey of Task Planning with Large Language Models. *Intelligent Computing*, 4: 0124.

Zhang, J.; Elgohary, A.; Magooda, A.; Khashabi, D.; and Durme, B. V. 2025a. Controllable Safety Alignment: Inference-Time Adaptation to Diverse Safety Requirements. In *The Thirteenth International Conference on Learning Representations*.

Zhang, Z.; Zheng, C.; Wu, Y.; Zhang, B.; Lin, R.; Yu, B.; Liu, D.; Zhou, J.; and Lin, J. 2025b. The Lessons of Developing Process Reward Models in Mathematical Reasoning. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Findings of the Association for Computational Linguistics: ACL 2025*, 10495–10516. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-256-5.

Zhao, J.; Liu, R.; Zhang, K.; Zhou, Z.; Gao, J.; Li, D.; Lyu, J.; Qian, Z.; Qi, B.; Li, X.; and Zhou, B. 2025. GenPRM: Scaling Test-Time Compute of Process Reward Models via Generative Reasoning. arXiv:2504.00891.

Zhao, T. Z.; Wallace, E.; Feng, S.; Klein, D.; and Singh, S. 2021. Calibrate Before Use: Improving Few-Shot Performance of Language Models. arXiv:2102.09690.

Zhou, K.; Liu, C.; Zhao, X.; Compalas, A.; Song, D.; and Wang, X. E. 2024. Multimodal Situational Safety. arXiv:2410.06172.

Zhou, K.; Liu, C.; Zhao, X.; Compalas, A.; Song, D.; and Wang, X. E. 2025. Multimodal Situational Safety. In *The Thirteenth International Conference on Learning Representations*.

Zhu, J.; Zheng, C.; Lin, J.; Du, K.; Wen, Y.; Yu, Y.; Wang, J.; and Zhang, W. 2025. Retrieval-Augmented Process Reward Model for Generalizable Mathematical Reasoning. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Findings of the Association for Computational Linguistics: ACL 2025*, 8453–8468. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-256-5.