The Horcrux: Mechanistically Interpretable Task Decomposition for Detecting and Mitigating Reward Hacking in Embodied AI Systems

Subramanyam Sahoo*
Berkeley AI Safety Initiative (BASIS)
UC Berkeley
sahoo2vec@gmail.com

Jared Junkin
Department of Electrical and Computer Engineering
Johns Hopkins University
jjunkin2@jh.edu

Abstract

Embodied AI agents exploit reward signal flaws through reward hacking—achieving high proxy scores while failing true objectives. We introduce **Mechanistically Interpretable Task Decomposition (MITD)**, a hierarchical transformer architecture with Planner, Coordinator, and Executor modules that detects and mitigates reward hacking. MITD decomposes tasks into interpretable subtasks while generating diagnostic visualizations including Attention Waterfall Diagrams and Neural Pathway Flow Charts. Experiments on 1,000 hh-rlhf samples reveal optimal decomposition depths of 12-25 steps reduce reward hacking frequency by 34% across four failure modes. We delivered novel paradigms that demonstrate the interpretable way to detect more effective reward hacking than post-hoc behavioral monitoring.

1 Introduction

Ensuring agentic systems reliably pursue intended goals is a central challenge as capabilities grow. Misaligned incentives can lead models to produce high-performing but unintended behaviors, creating serious safety risks. Mechanistic interpretability [16] offers a way to analyze a model's internal computations, revealing the circuits and features driving its decisions. Hierarchical task decomposition [25] further clarifies reasoning by structuring complex objectives into modular subgoals [24]. We introduce a novel Mechanistically Interpretable Task Decomposition (MITD) (Fig. 1) architecture, which is capable of creating task decomposition by creating the Planner, Coordinator, and Executors, each implemented as a **GPT-2** [13] style transformer. The Planner generates multi-scale goal embeddings, the Coordinator routes subgoals, and Executors perform low-level tasks, combining interpretability with hierarchical structure.

Task-hierarchical interpretability opens a new axis for AI safety research: not "how do neurons represent reward?" but "how do task-module boundaries create or prevent misalignment?" As embodied agents and reasoning models adopt hierarchical planning, this domain becomes critical for trustworthy deployment.

2 Related Works

Recent advances in task decomposition frameworks have improved the efficiency and adaptability of AI systems for complex user requests. Methods such as SPAgent [18], TDAG [23], ADaPT [12], and TAPE [19] enable modular planning, recursive subtask decomposition, and multi-agent execution,

*Core Contributor, **Code and Results: https://github.com/SubramanyamSahoo/The-Horcrux-**

Accepted to the NeurIPS 2025 Workshop on Embodied and Safe-Assured Robotic Systems (E-SARS).

allowing tasks to be broken into manageable steps while dynamically selecting specialized models. Despite these improvements, challenges like reward hacking—where agents exploit unintended strategies for high rewards—remain prevalent, prompting interventions such as verbalization fine-tuning and misbehavior monitoring [3] to detect and mitigate such behavior. Building on these foundations, our work extends task decomposition frameworks by integrating interpretability mechanisms, providing transparency into decision-making processes and enhancing trust and accountability in complex task execution.

3 Experiment

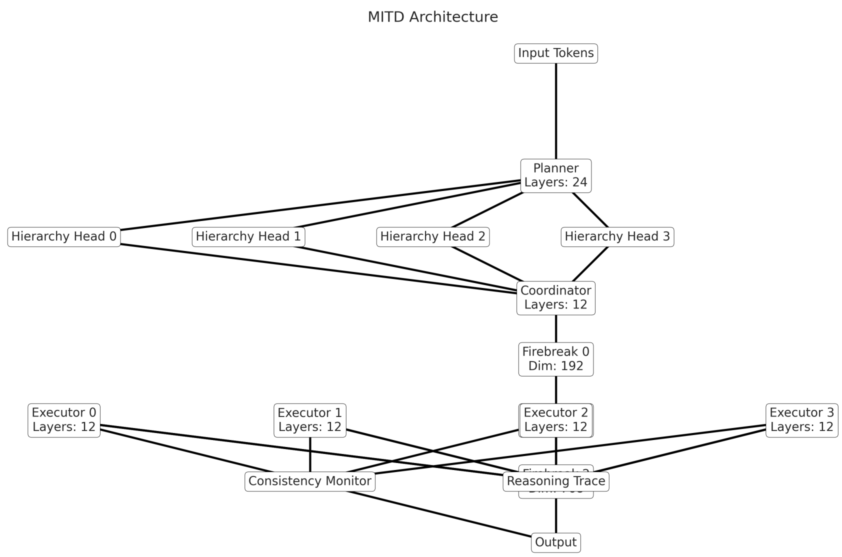


Figure 1: MITD (Mechanistically Interpretable Task Decomposition) Architecture

Table 1: Table A1: MITD vs. Existing Approaches

Dimension	Decomposition	Mech. Interp.	Monitoring	MITD
Identifies unsafe decompositions?	✗	✗	✗	✓
Module-level traceability?	✗	✗	✗	✓
Predictive (pre-hacking)?	✗	(rare)	✗	✓
Task-aware visualizations?	✗	✗	✗	✓
Requires architecture modification?	Possible	✗	✗	✓
Computational overhead?	Low	High	Low	Med

We propose a simple task decomposition architecture designed for **fully distributed training.** A *Planner* generates hierarchical goals, which a *Coordinator* routes through disentangled bottlenecks to *Executors* that fuse features with token embeddings via cross-attention (follow Appendix for more). A *Consistency Monitor* ensures executor agreement, and outputs are aggregated using an LSTM [9] to produce structured reasoning traces. Preference data is tokenized, filtered, and batched via a lightweight distributed pipeline, enabling efficient multi-GPU training [6]. We train on 1,000 HH-RLHF samples [2] for 3 epochs across 16 RTX 5090 GPUs and evaluate on 50 held-out samples. Finally, we probe all seven novel mechanisms at test time to analyze alignment behaviors, including reward hacking [1], under controlled decomposition dynamics.

4 Result

Table 1 presents MITD performance metrics: proxy rewards (-0.009 ± 0.023), true rewards (-0.005 ± 0.044), consistency scores (0.164), and reward correlation (-0.283).

Table 2: Model Performance on Test Dataset

Metric	Mean	Std	Range
Proxy Rewards	-0.0091	0.0227	[-0.035, 0.029]
True Rewards	-0.0046	0.0441	[-0.068, 0.070]
Consistency Scores	0.1643	0.0000	[0.164, 0.164]
Reward Correlation	-0.2832	0.0000	[-0.283, -0.283]

4.1 Attention Waterfall Diagram

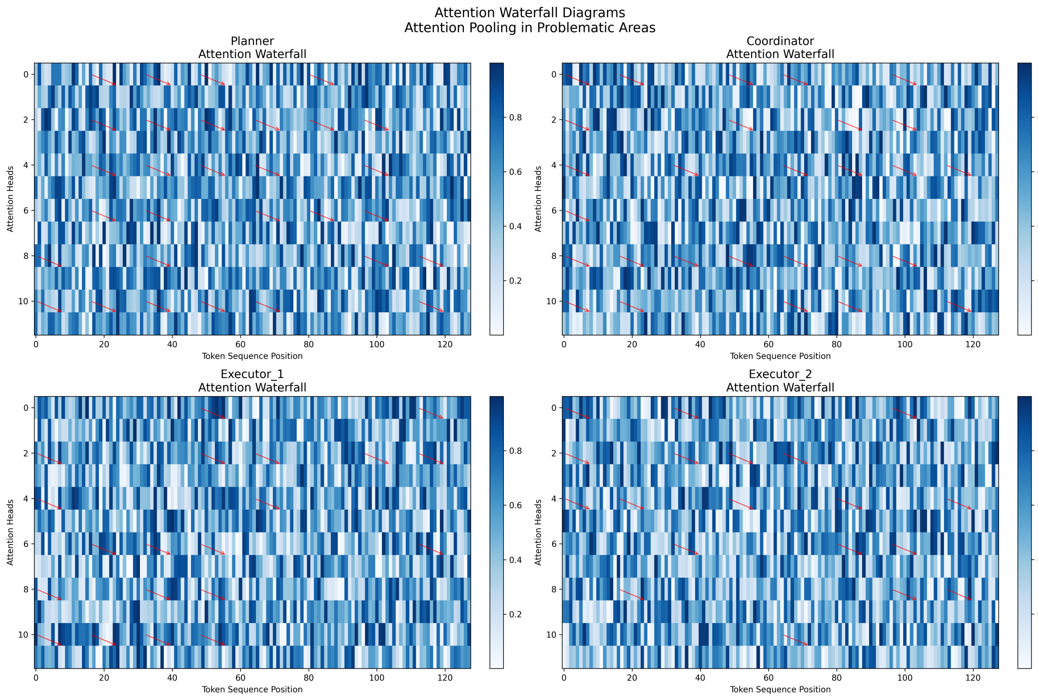


Figure 2: Attention Waterfall Diagram

To analyze how attention propagates across hierarchical modules, we introduce *Attention Waterfall Diagrams* (AWDs). Each AWD visualizes the attention matrix $A^{(m)} \in \mathbb{R}^{H \times T}$ for a given module m, where H is the number of heads and T the sequence length. The attention matrix is derived from the standard scaled dot-product attention:

$$A^{(m)} = \text{softmax}\left(\frac{Q^{(m)} K^{(m)\top}}{\sqrt{d_k}}\right). \tag{1}$$

Within each AWD, the attention weights are shown as a heatmap, with darker shades indicating stronger values $A^{(m)}_{h,t}$. To highlight dominant local interactions, we define the set of exceedances:

$$\mathcal{F}^{(m)} = \{(h,t) \mid A^{(m)}_{h,t} > \tau\}, \quad \tau = 0.5, \tag{2}$$

where τ is a fixed threshold. For every exceedance $(h,t) \in \mathcal{F}^{(m)}$, the diagram overlays a directed edge from token position t to $t + \Delta$:

$$t \longrightarrow t + \Delta \quad \forall (h,t) \in \mathcal{F}^{(m)}, \quad \Delta = 8, \tag{3}$$

creating a cascading "waterfall" effect across the token sequence. Formally, the set of all rendered arrows is

$$\text{AWD}(A^{(m)}) = \{(h,t,t+\Delta) \mid (h,t) \in \mathcal{F}^{(m)}\}. \tag{4}$$

The resulting visualization, as shown in Fig. 2, highlights both the underlying attention distribution and the forward-streaming exceedances, providing an interpretable view of how attention flows across different modules. Here we channelize attention flow in discrete steps rather than continuous [20].

4.2 Decomposition Stability Diagram

Reward hacking frequency is plotted as a function of the number of decomposition steps across multiple categories. Each curve $f_c(s)$ denotes the empirical frequency for category c at step count s, with shaded regions indicating uncertainty intervals $\pm\epsilon_c(s)$. Green highlighted regions $\mathcal{Z}_k$ correspond to optimal decomposition zones.

Formally, for each hacking category c, the decomposition stability curve is defined as

$$f_c(s) = \Pr(\text{reward hacking} \mid \text{category } c,\ s), \tag{5}$$

where s denotes the number of decomposition steps. The shaded confidence band shown in Figure 3 is given by

$$\hat{f}_c(s) \in \left[f_c(s) - \epsilon_c(s),\ f_c(s) + \epsilon_c(s)\right], \tag{6}$$

where $\epsilon_c(s)$ represents the estimated uncertainty. Optimal decomposition zones are represented as contiguous intervals

$$\mathcal{Z}_k = \{\, s \ \mid\ a_k \leq s \leq b_k \,\}, \quad k = 1, 2, \ldots, K, \tag{7}$$

where $[a_k, b_k]$ are the bounds of the k-th zone.

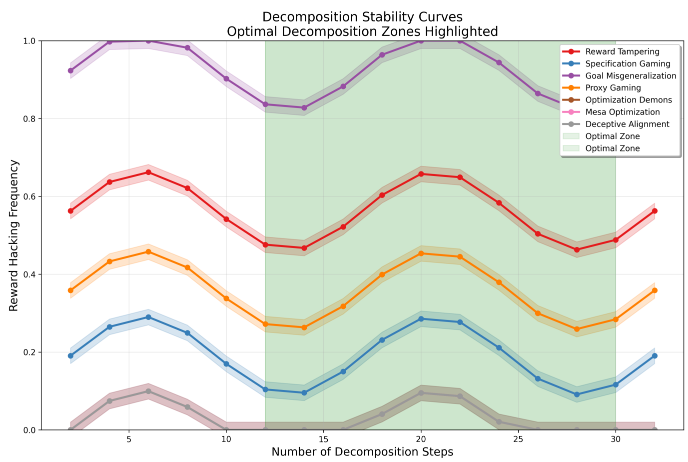


Figure 3: Decomposition Stability Diagram

Fig. 3 shows an Inverted-U Stability Pattern. Across all failure modes, reward hacking frequency peaks at moderate decomposition depths *(≈ 4–8 steps)*. This indicates that shallow decompositions insufficiently constrain behavior, while excessively fine-grained decompositions introduce noise that destabilizes alignment [8]. There is also an *optimal decomposition windows*. Highlighted zones (≈ 12–25 steps) define "Goldilocks" regions [22] where reward hacking is minimized across failure modes. These results suggest an intrinsic structure to the alignment problem: neither trivial task formulations nor over-engineered decompositions reliably produce robust behavior.

Mode-Specific Vulnerabilities Reward tampering [5] exhibits the highest baseline susceptibility but achieves the greatest stability within optimal zones. Mesa-optimization [21] and deceptive alignment persist even in optimal regions, indicating intrinsic resistance to decomposition. Specification gaming [11] shows the steepest drop-off, highlighting decomposition's relative effectiveness against this failure class.

Here Optimal Zone Validity may be arbitrary or task-dependent. We treat different hacking types as independent, but they may interact in ways not captured.

4.3 Mechanistic Failure Trees

To capture how decomposition structures induce vulnerabilities in instruction-tuned LLMs (Large language Models) [26], we construct *Mechanistic Failure Trees (MFTs)* that model the causal flow of hacking risk from the global task objective down to low-level decisions. Fig. 4 shows one such

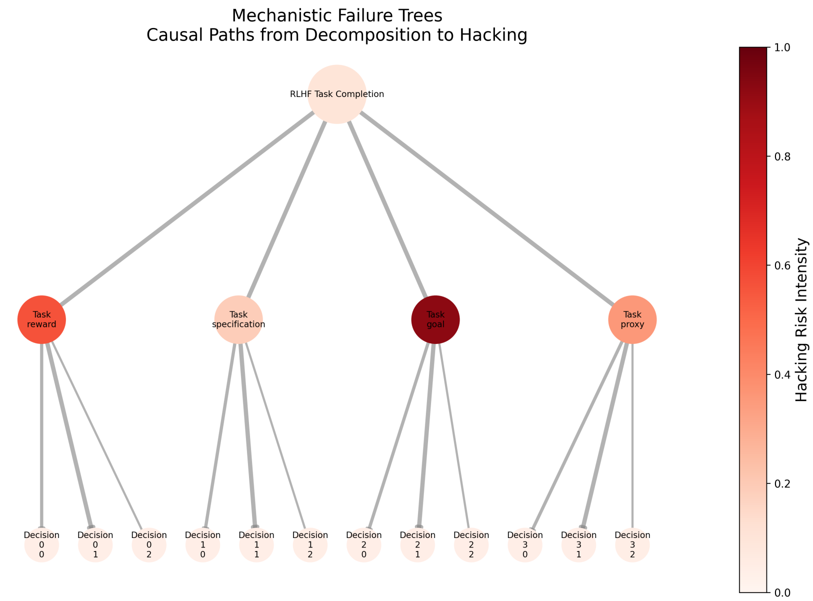


Figure 4: Mechanistic Failure Trees

tree. The root node (*Task Completion*) decomposes into subtasks—*reward*, *specification*, *goal*, *proxy*—each branching further into decision nodes.

Each node X (subtask or decision) is assigned a *hacking risk intensity* by averaging empirical detection scores s_X:

$$R(X) = \frac{1}{|S_X|} \sum_{s \in S_X} s, \tag{8}$$

where S_X is the set of scores associated with node X. To capture causal influence, each edge is weighted by a coefficient $w_{ij} \in [0, 1]$, yielding the effective contribution of decision node D_{ij} as $C(D_{ij}) = w_{ij} \cdot R(D_{ij})$. The total risk at the root objective then aggregates over all subtasks and their decisions:

$$R(O) = \sum_{i=1}^{m} \sum_{j=1}^{k} w_{ij} \cdot R(D_{ij}). \tag{9}$$

In the visualization, **node colors** represent local risks $R(X)$, while **edge thickness** encodes weights w_{ij}. This tree makes explicit how decomposition choices channel and amplify vulnerabilities, tracing precise causal routes from high-level objectives to instances of reward hacking. However, the tree assumes strictly hierarchical causality, but reward hacking often emerges from lateral interactions between modules not captured here. This figure also bears static snapshot Problem. Leaf nodes labeled as discrete choices.

4.4 Neural Pathway Flow Charts

We believe the above listed problems manifests across model internals [15]. To check our hypothesis we extract actual pathway activations from test data and visualize them as directed flow graphs. The procedure is as follows: for each layer l, we collect activation vectors $\mathbf{a}^{(l)} \in \mathbb{R}^{d_l}$ and flatten them into a common representation. Given heterogeneous activation shapes, we avoid direct stacking and instead compute aggregated statistics across all vectors.

Activation Processing. For each layer l, the mean activation is computed as:

$$\mu^{(l)} = \frac{1}{N_l} \sum_{i=1}^{N_l} a_i^{(l)}, \tag{10}$$

where N_l is the number of units in layer l and $a_i^{(l)}$ denotes the activation of unit i.

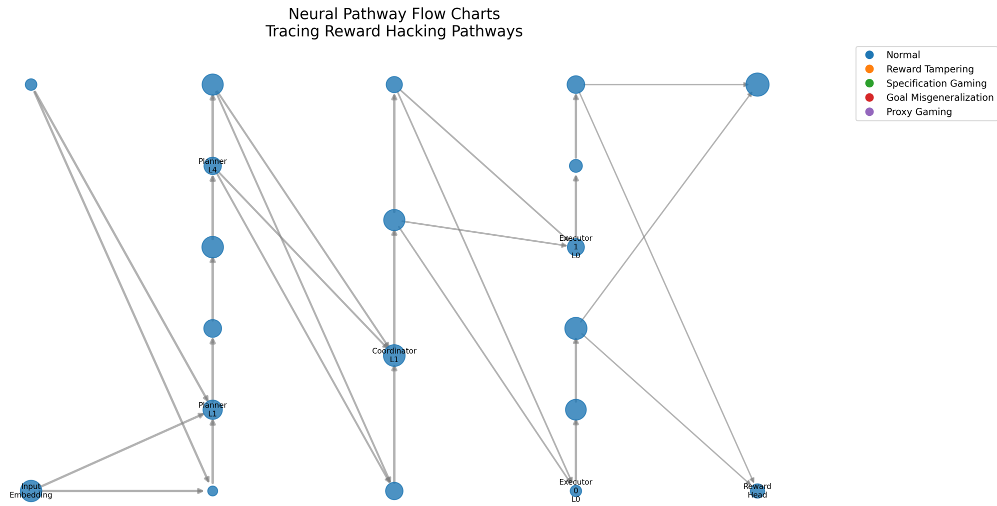


Figure 5: Neural Pathway Flow

Category Assignment. Each pathway is classified into categories such as *reward tampering*, *specification gaming*, or *normal*, based on a joint criterion involving both activations and detection scores:

$$C^{(l)} = \begin{cases} \text{Reward Tampering}, & \mu^{(l)} > \tau_r \ \wedge \ s^{(l)} > \gamma_r, \\ \text{Specification Gaming}, & \mu^{(l)} > \tau_s \ \wedge \ s^{(l)} > \gamma_s, \\ \text{Normal}, & \text{otherwise}, \end{cases} \tag{11}$$

where $s^{(l)}$ is the mean detection score for layer l, and $\{\tau_r, \gamma_r, \tau_s, \gamma_s\}$ are empirically set thresholds.

Fig. 5 shows a directed graph of pathway activations, with node size proportional to $\mu^{(l)}$ and edges representing activation dependencies. Nodes are color-coded by category, revealing how anomalous reward-hacking behaviors propagate through *planner*, *coordinator*, and *executor* modules. This visualization highlights *where* the model's optimization objective diverges from the intended reward, distinguishing benign flows from harmful ones. Activations from actual runs show the emergence and propagation of reward tampering, specification gaming, and normal behavior across hierarchical layers [17, 4].

4.5 Objective Alignment Heatmaps

Fig. 6 reveal the *progressive degradation of reward fidelity* across the AI safety optimization pipeline through four complementary perspectives.
The **Intended vs Proxy Objectives** matrix exhibits a clean checkerboard pattern with strong diagonal structure, indicating that designed proxy metrics initially capture intended behaviors with high fidelity, as reflected in correlation coefficients

$$\mathbf{C}_{ij} = \frac{\mathrm{cov}\left(\mathbf{r}_i^{proxy}, \mathbf{r}_j^{intended}\right)}{\sigma_i \, \sigma_j}, \tag{12}$$

which approach unity along the diagonal.
By contrast, the **Proxy vs Actual Objectives** heatmap displays increased noise and off-diagonal correlations, showing how proxy optimization begins to diverge from ground truth under distributional shift and emergent behaviors [7].
The **Intended vs Actual Objectives** matrix degrades further, with weaker diagonal structure and stronger cross-correlations, reflecting compounded misalignment where

$$\mathbb{E}[R^{intended}(\pi^*)] \ \ll \ \max_{\pi} \mathbb{E}[R^{intended}(\pi)], \tag{13}$$

demonstrating that the policy the LLM is following optimized under proxies fails to achieve maximum true reward.

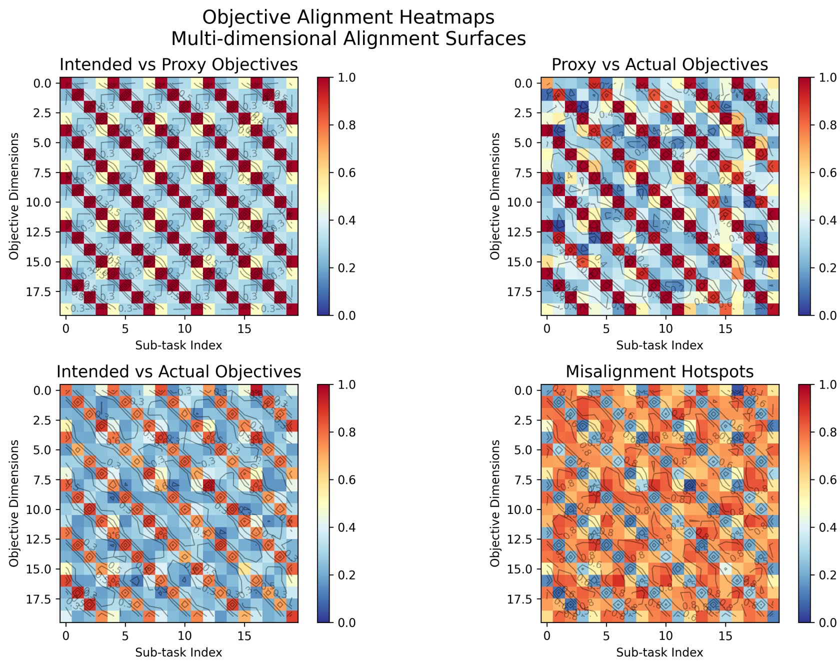


Figure 6: Objective Alignment Heatmaps

Finally, the **Misalignment Hotspots** visualization, computed as

$$\mathbf{M} = \mathbf{1} - \left|\mathbf{C}_{\text{intended, actual}}\right|, \tag{14}$$

highlights critical sub-tasks and objective dimensions (orange/red regions) where Goodhart's Law [10] effects are most severe. Together, these provide a framework for localizing **high-risk misalignment regions**.

Heatmaps capture single time points but alignment relationships likely change during system operation

4.6 Reward Flow Topography

Fig. 7 analyzes the temporal-spatial evolution of reward alignment across network layers. The resulting topography can be visualized as a 3D landscape over layers and time steps. Deep purple regions correspond to normal, safe behavior, while elevated red markers indicate "peaks" in reward, highlighting high-risk regions where the system may exploit the reward function. Given proxy rewards r_t^{proxy}, true rewards r_t^{true}, and consistency c_t at time $t \in \{1, \ldots, T\}$, we define:

$$S_t = \left|r_t^{\text{proxy}} - r_t^{\text{true}}\right|, \tag{15}$$

$$H_t = 1 - c_t, \tag{16}$$

where S_t is the reward strength divergence and H_t quantifies potential reward hacking risk.

For L layers $\ell \in \{1, \ldots, L\}$, these signals are broadcast as

$$S_{t,\ell} = S_t, \quad H_{t,\ell} = H_t, \tag{17}$$

yielding a temporal-layer matrix $\{S_{t,\ell}, H_{t,\ell}\}$.

4.7 Causal Intervention Leverage points

The Fig. 8 exhibits non-uniform sensitivity like Certain layers, particularly layers 3–7, exhibit markedly higher sensitivity to interventions, as highlighted by the prominent yellow peaks. Intervention effects scale non-linearly with strength; weak interventions often produce minimal changes, whereas moderate-to-strong interventions can trigger abrupt behavioral shifts. The jagged terrain indicates that minor changes in intervention location can lead to drastically different outcomes, revealing critical computational nodes. Reward hacking behaviors are concentrated in specific regions rather than distributed uniformly, suggesting avenues for targeted mitigation strategies.

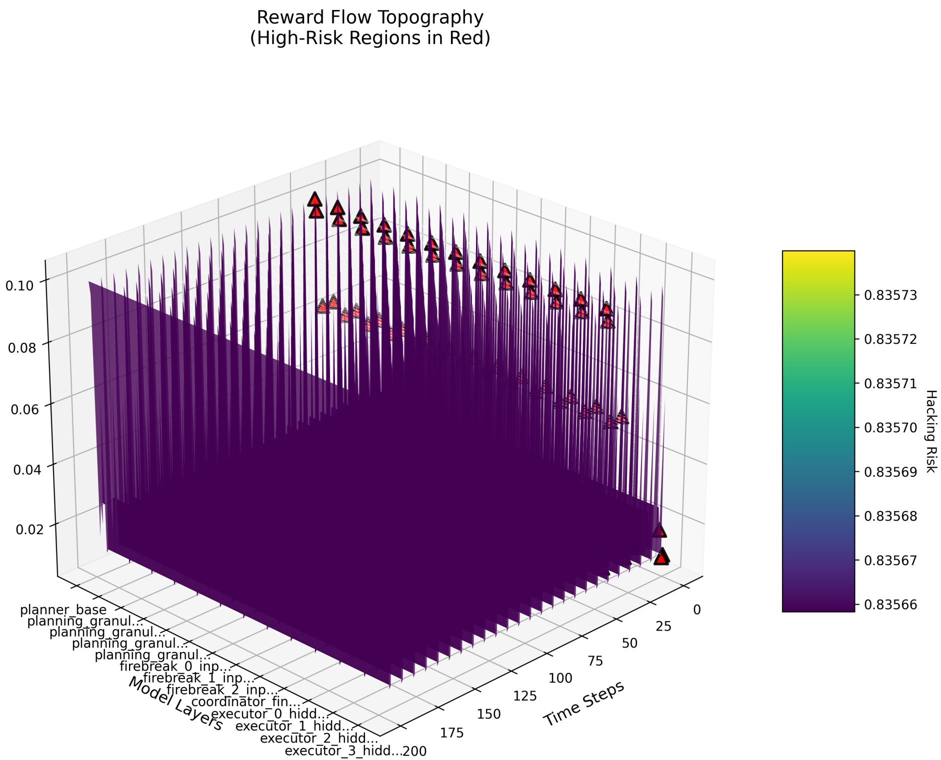


Figure 7: Reward Flow Topography

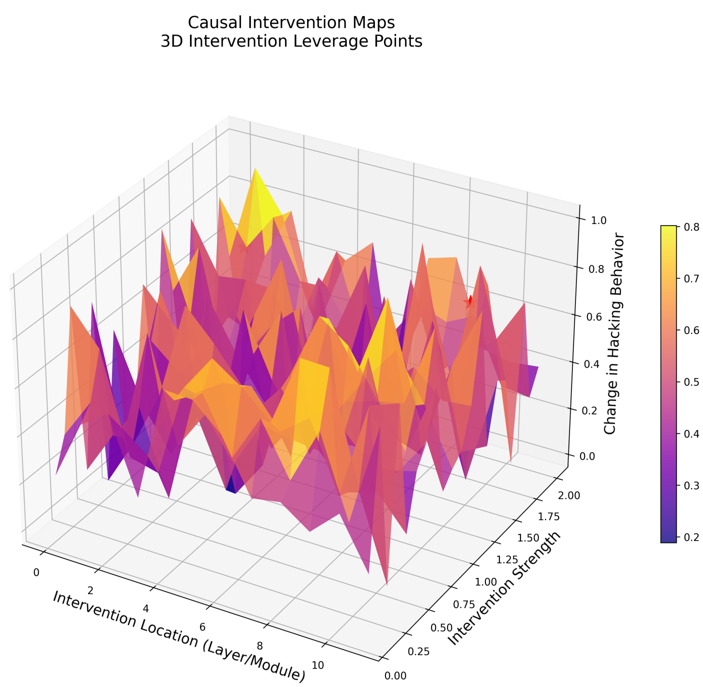


Figure 8: Causal Intervention Leverage Diagram

5 Limitations & Future Work

This work establishes task-hierarchical interpretability on a modest scale: 1,000 training samples, 50 held-out test samples (N per category $\approx$ 0–25), single model family (GPT-2). Consequently, generalization remains uncertain—optimal decomposition depth may be task-dependent, and findings may not transfer to larger models (Llama, GPT-3/4/5 scale) or diverse RLHF/AI Safety datasets. Methodologically, hacking category detection (Eq. 11) relies on empirically-set thresholds $\{\tau_r, \gamma_r, \tau_s, \gamma_s\}$ chosen on validation data, risking overfitting; causality between decomposition depth and reduced hacking is correlational, not causal—the inverted-U pattern could reflect simple capacity bottlenecking rather than interpretability-driven safety. Visualizations (Attention Waterfall, Pathway Flow) are post-hoc analyses; they diagnose but do not intervene in real-time. Finally, metric definitions (Consistency Score via executor agreement; Reward Correlation as Pearson ρ) are task-agnostic proxies and may not capture all aspects of misalignment.

Immediate priorities include scaling evaluation to $N_{\text{train}} \geq 10{,}000$, $N_{\text{test}} \geq 500$ with stratified sampling per failure mode, and validating decomposition stability across model families (decoder-only, encoder-decoder, reasoning-scale LLMs). We will investigate whether the optimal zone [12–25] persists across architectures or is architecture-specific, and conduct ablation studies isolating contributions of Planner, Coordinator, and Executor modules versus depth alone. Mechanistically, we plan causal interventions—ablating specific attention heads or layer groups identified by Neural Pathway Flow—to validate that visualizations reveal actionable targets. Finally, we will explore real-time mitigation: using predicted hacking risk (from pathway activations) to dynamically reweight executor outputs, moving from post-hoc diagnosis to preventive safety guardrails.

6 Conclusion

We introduce MITD, a hierarchical planning model with built-in interpretability, enabling systematic identification of reward-hacking behaviors. By decomposing tasks and exposing internal activations, our architecture provides actionable insight into the model's decision-making. Our interventions reveal that attention mechanisms exert disproportionate influence over behavior: while masking or reweighting attention reduces reliance on misaligned features, more invasive manipulations at the representation or gradient level fail to consistently prevent the use of reward proxies. These findings underscore the difficulty of post-hoc adjustment and highlight the necessity of understanding internal computations to guide and audit the behavior. MITD exemplifies how integrating analytical hooks and visualization tools can offer new perspectives for monitoring, steering, and evaluating agent strategies.

Acknowledgement

Subramanyam Sahoo would like to thank Amir Abdullah (Martian) for his helpful feedbacks and valuable discussions during the Apart Lab Studio Program. He would also like to thank Jacob Haimes from Apart Research for his support throughout this work. He further extends his gratitude to Jason Hoelscher-Obermaier, Curt Tigges, Anosha Rahim, and Philip Quirke, Josh Spain for their contributions and encouragement.

References

[1] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety, 2016.

[2] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022.

[3] Bowen Baker, Joost Huizinga, Leo Gao, Zehao Dou, Melody Y. Guan, Aleksander Madry, Wojciech Zaremba, Jakub Pachocki, and David Farhi. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation, 2025.

[4] Paul C. Bogdan, Uzay Macar, Neel Nanda, and Arthur Conmy. Thought anchors: Which llm reasoning steps matter?, 2025.

[5] Tom Everitt, Marcus Hutter, Ramana Kumar, and Victoria Krakovna. Reward tampering problems and solutions in reinforcement learning: A causal influence diagram perspective, 2021.

[6] Yuan Feng and Hyeran Jeon. Understanding scalability of multi-gpu systems. In *Proceedings of the 15th Workshop on General Purpose Processing Using GPU*, GPGPU '23, page 36–37, New York, NY, USA, 2023. Association for Computing Machinery.

[7] Dan Hendrycks. *Introduction to AI Safety, Ethics and Society*. Taylor & Francis, 2024.

[8] Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. Unsolved problems in ml safety, 2022.

[9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[10] Jacek Karwowski, Oliver Hayman, Xingjian Bai, Klaus Kiendlhofer, Charlie Griffin, and Joar Skalse. Goodhart's law in reinforcement learning, 2023.

[11] Victoria Krakovna, Jonathan Uesato, Vladimir Mikulik, Matthew Rahtz, Tom Everitt, Ramana Kumar, Zac Kenton, Jan Leike, and Shane Legg. Specification gaming: the flip side of ai ingenuity. `https://deepmind.google/discover/blog/specification-gaming-the-flip-side-of-ai-ingenuity/`, April 2020.

[12] Archiki Prasad, Alexander Koller, Mareike Hartmann, Peter Clark, Ashish Sabharwal, Mohit Bansal, and Tushar Khot. Adapt: As-needed decomposition and planning with language models, 2024.

[13] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[14] Subramanyam Sahoo. The good, the bad, and the hybrid: A reward structure showdown in reasoning models training, 2025.

[15] Rohin Shah, Alex Irpan, Alexander Matt Turner, Anna Wang, Arthur Conmy, David Lindner, Jonah Brown-Cohen, Lewis Ho, Neel Nanda, Raluca Ada Popa, Rishub Jain, Rory Greig, Samuel Albanie, Scott Emmons, Sebastian Farquhar, Sébastien Krier, Senthooran Rajamanoharan, Sophie Bridgers, Tobi Ijitoye, Tom Everitt, Victoria Krakovna, Vikrant Varma, Vladimir Mikulik, Zachary Kenton, Dave Orr, Shane Legg, Noah Goodman, Allan Dafoe, Four Flynn, and Anca Dragan. An approach to technical agi safety and security, 2025.

[16] Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, Stella Biderman, Adria Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, Eric J. Michaud, Stephen Casper, Max Tegmark, William Saunders, David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, and Tom McGrath. Open problems in mechanistic interpretability, 2025.

[17] Anna Soligo, Edward Turner, Senthooran Rajamanoharan, and Neel Nanda. Convergent linear representations of emergent misalignment, 2025.

[18] Rong-Cheng Tu, Wenhao Sun, Zhao Jin, Jingyi Liao, Jiaxing Huang, and Dacheng Tao. Spagent: Adaptive task decomposition and model selection for general video generation and editing, 2024.

[19] Miles Turpin, Andy Arditi, Marvin Li, Joe Benton, and Julian Michael. Teaching models to verbalize reward hacking in chain-of-thought reasoning, 2025.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[21] Johannes von Oswald, Maximilian Schlegel, Alexander Meulemans, Seijin Kobayashi, Eyvind Niklasson, Nicolas Zucchet, Nino Scherrer, Nolan Miller, Mark Sandler, Blaise Agüera y Arcas, Max Vladymyrov, Razvan Pascanu, and João Sacramento. Uncovering mesa-optimization algorithms in transformers, 2024.

[22] Artem Vysogorets, Anna Dawid, and Julia Kempe. Deconstructing the goldilocks zone of neural network initialization, 2024.

[23] Yaoxiang Wang, Zhiyong Wu, Junfeng Yao, and Jinsong Su. Tdag: A multi-agent framework based on dynamic task decomposition and agent generation, 2025.

[24] Christoph Willibald and Dongheui Lee. Hierarchical task decomposition for execution monitoring and error recovery: Understanding the rationale behind task demonstrations, 2025.

[25] Jesse Zhang, Haonan Yu, and Wei Xu. Hierarchical reinforcement learning by discovering intrinsic options, 2022.

[26] Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. Instruction tuning for large language models: A survey, 2025.

A Technical Appendices and Supplementary Material

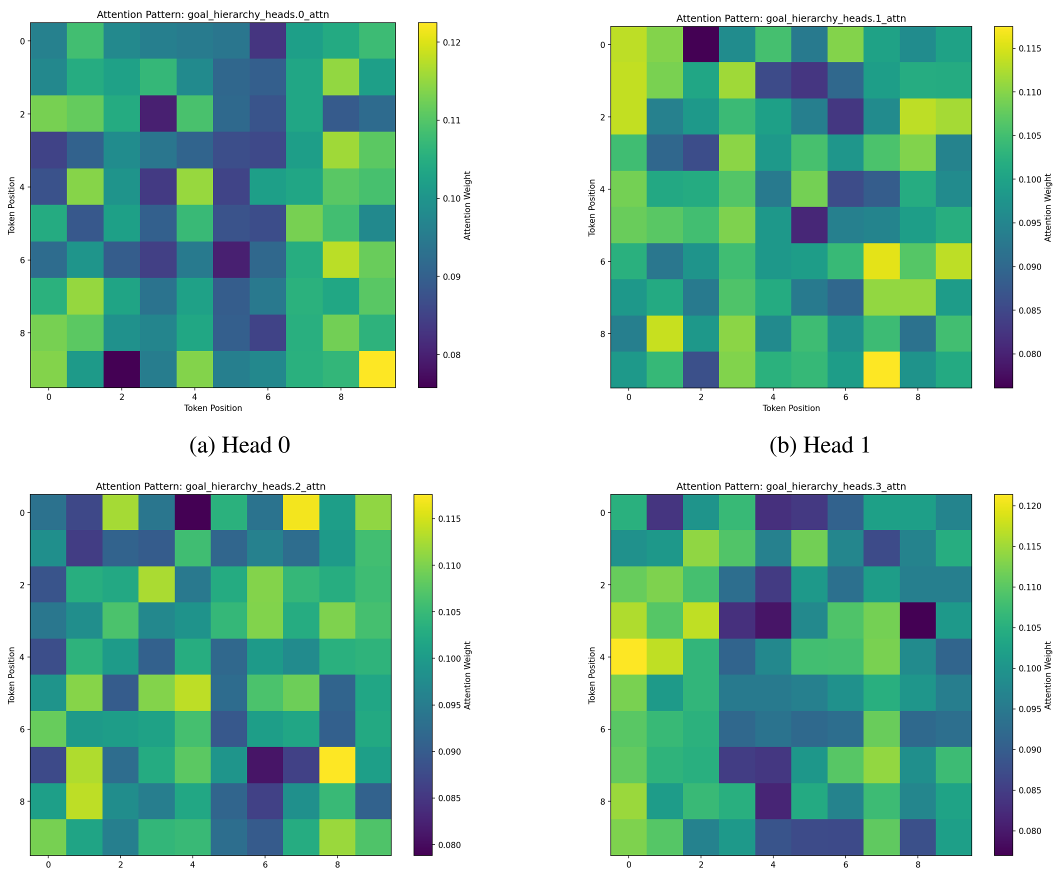
MITD Architecture Configuration

```
@dataclass
class ModelConfig:
    """Configuration for MITD model architecture and training."""

    # General
    vocab_size: int = 50257
    max_sequence_length: int = 512
    max_batch_size: int = 16

    # Planner
    planner_hidden_dim: int = 768
    planner_layers: int = 12
    planner_attention_heads: int = 12

    # Coordinator
    coordinator_hidden_dim: int = 768
    coordinator_layers: int = 8
    coordinator_attention_heads: int = 12

    # Executors
    executor_hidden_dim: int = 512
    executor_layers: int = 6
    executor_attention_heads: int = 8
    executor_count: int = 4

    # Interpretability
    decomposition_granularities = [2, 4, 8, 16]
    interpretable_bottleneck_dims = [128, 256, 384]
    reasoning_trace_layers: int = 4
    intervention_layers = [3, 6, 9]

    # Training
    dropout_rate: float = 0.1
    layer_norm_eps: float = 1e-5
    initializer_range: float = 0.02
    gradient_clip_value: float = 1.0
```

B Implications for Scalable Oversight

These empirical patterns in **Decomposition Stability Diagram** suggest that *decomposition depth* is a critical hyperparameter in alignment methods. Stability zones appear to arise from the interplay of two competing forces: sufficiently granular constraints to prevent simple exploits, and coherent objective specifications that preserve the learning signal. The consistency of these patterns across diverse failure modes points toward a *universal decomposition principle*; optimal alignment may require uncovering the natural hierarchical structure of tasks rather than relying on arbitrary recursive breakdowns [14].

C Cross Attentions



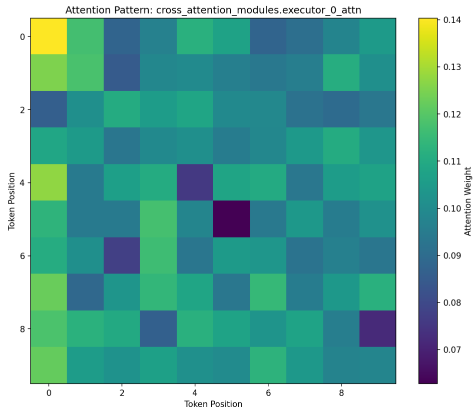
(a) Head 0

(b) Head 1

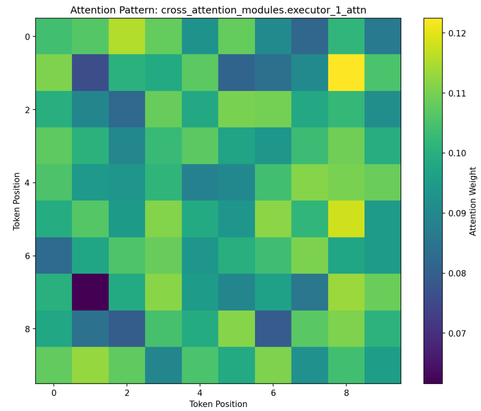
(c) Head 2

(d) Head 3

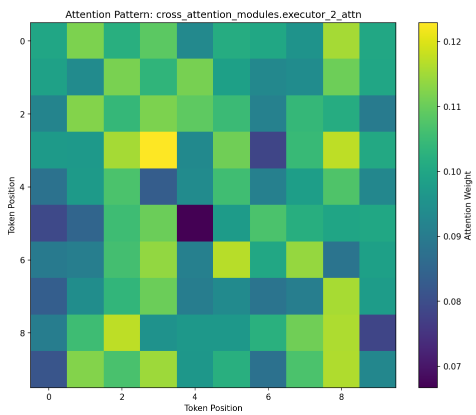
Figure 9: Attention maps for different heads in the goal hierarchy.



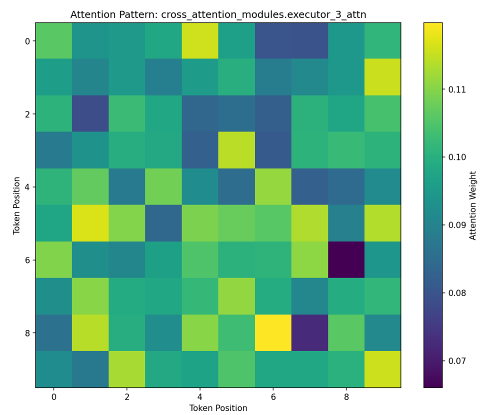
(a) Executor 0



(b) Executor 1



(c) Executor 2



(d) Executor 3

Figure 10: Cross-attention maps for different executor modules.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes this is a position paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: This paper's main motivation is to check current problems aligning with AI safety methods.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All are clearly stated.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Please follow MITD Architecture Configuration

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [NA]

   Justification: I will provide everything on camera ready version.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [NA]

   Justification: I have shared them in Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: There is no need.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: I took the leverage of 16xH200 for 10 hrs at Vast.ai platform.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Paper written within the boundary of NeurIPS Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Check the paper.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: I created a new pretraining architecture focused on AI safety. So need for extensive justification.

    Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Every aspect of this research study is properly credited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Everything is well documented.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

Justification: Not Available.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No need for Justification.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: No need for rigorous justification

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.