When Not to Trust the Oracle:

Detecting Unsafe Advice in LLM-guided Reinforcement Learning

When Not to Trust the Oracle: Detecting Unsafe Advice in LLM-guided Reinforcement Learning

1st Anandkumar NS
PSG College of Technology
Coimbatore, India
22z209@psgtech.ac.in

2rd Kishoreadhith V
PSG College of Technology
Coimbatore, India
22z232@psgtech.ac.in

3nd Dhakkshin S R
PSG College of Technology
Coimbatore, India
22z215@psgtech.ac.in

4th M Raj Ragavender
PSG College of Technology
Coimbatore, India
22z233@psgtech.ac.in

5th Rithvik K
PSG College of Technology
Coimbatore, India
22z253@psgtech.ac.in

Core Research Question

"How can we build LLM-guided RL agents that are safe, interpretable, and robust to adversarial inputs?"

The Promise and Peril of LLM-guided RL

1. The Promise

- LLMs as a Strategic Advisor: Natural language understanding + domain knowledge
- Enhanced Decision Making: Human like reasoning for complex environments.
- Rapid Development: Leverage pre-trained knowledge without domain training.

2. The Peril

- ➔ **Blind Trust Problem**: Current systems execute LLM advice without validation
- ➔ **Brittleness**: Performance degrades with corrupted or ambiguous inputs
- ➔ **Safety Concerns**: No mechanisms to detect unsafe recommendations
- ➔ **Black Box Decision**: Limited interpretability when things go wrong

Current State and Gaps

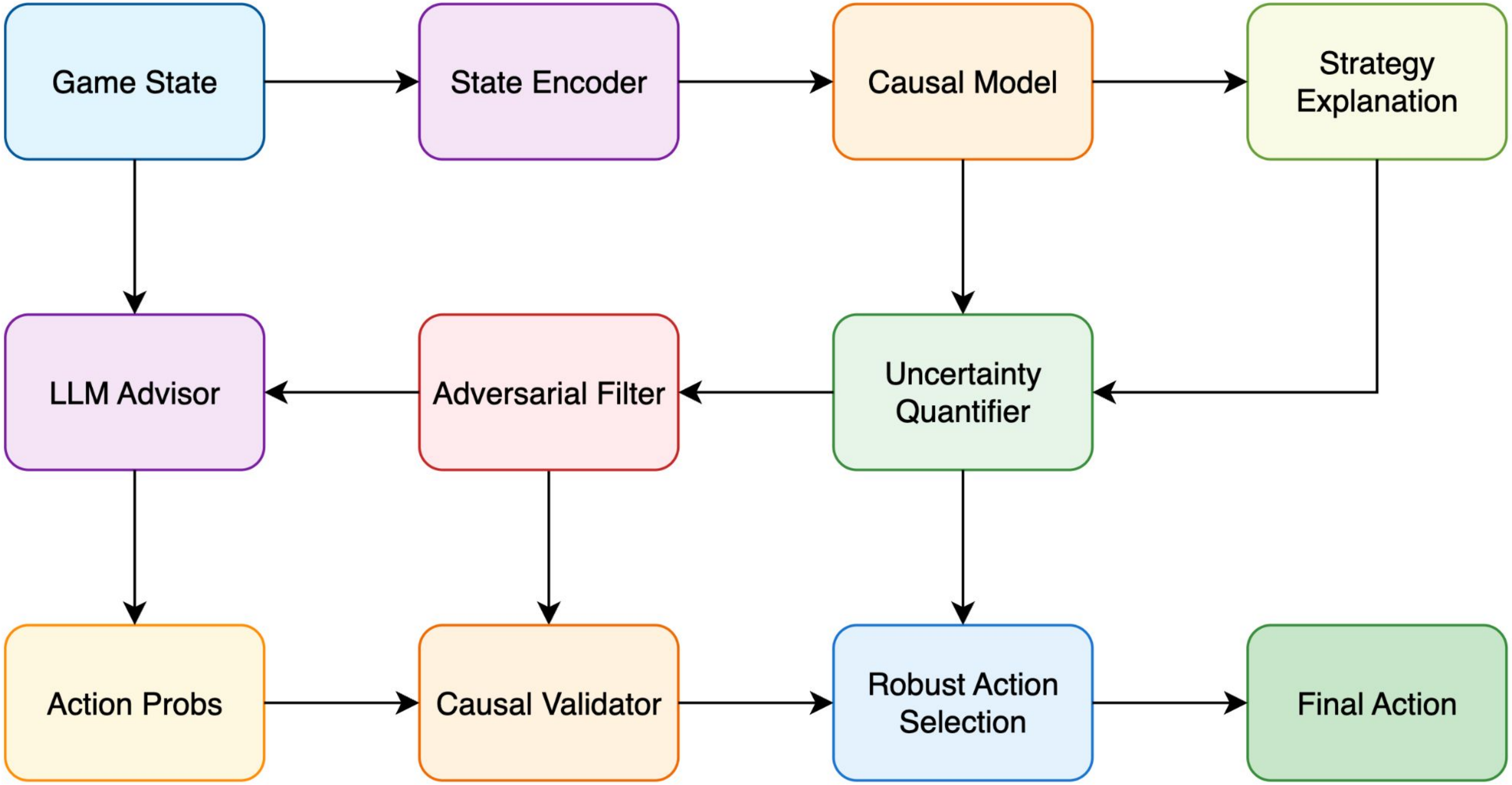
Blind Trust

Fragile State Understanding

Limited Interpretability

Lack of Stress Testing

Architecture overview



Our Solution

Pillar 1: Rule-Based Causal Validators

- Capture critical survival dependencies
- Health, nutrition, enemy proximity checks
- Domain-specific safety constraints

Pillar 2: Adversarial Corruption Tests

- Deliberately perturb state descriptions
- Test robustness to misleading information
- Expose LLM failure modes systematically

Pillar 3: Fallback Ensemble Policy

- Detect unsafe advice in real-time
- Seamlessly defer to baseline RL agent
- Maintain performance under uncertainty

Adversarial Testing

Type 1: Semantic Perturbations

Change character names: "orc" → "ally"

Modify object descriptions: "poison" → "healing potion"

Test: Does LLM maintain safety awareness?

Type 2: Context Confusion

Inject contradictory information

Ambiguous state descriptions

Test: How does LLM handle uncertainty?

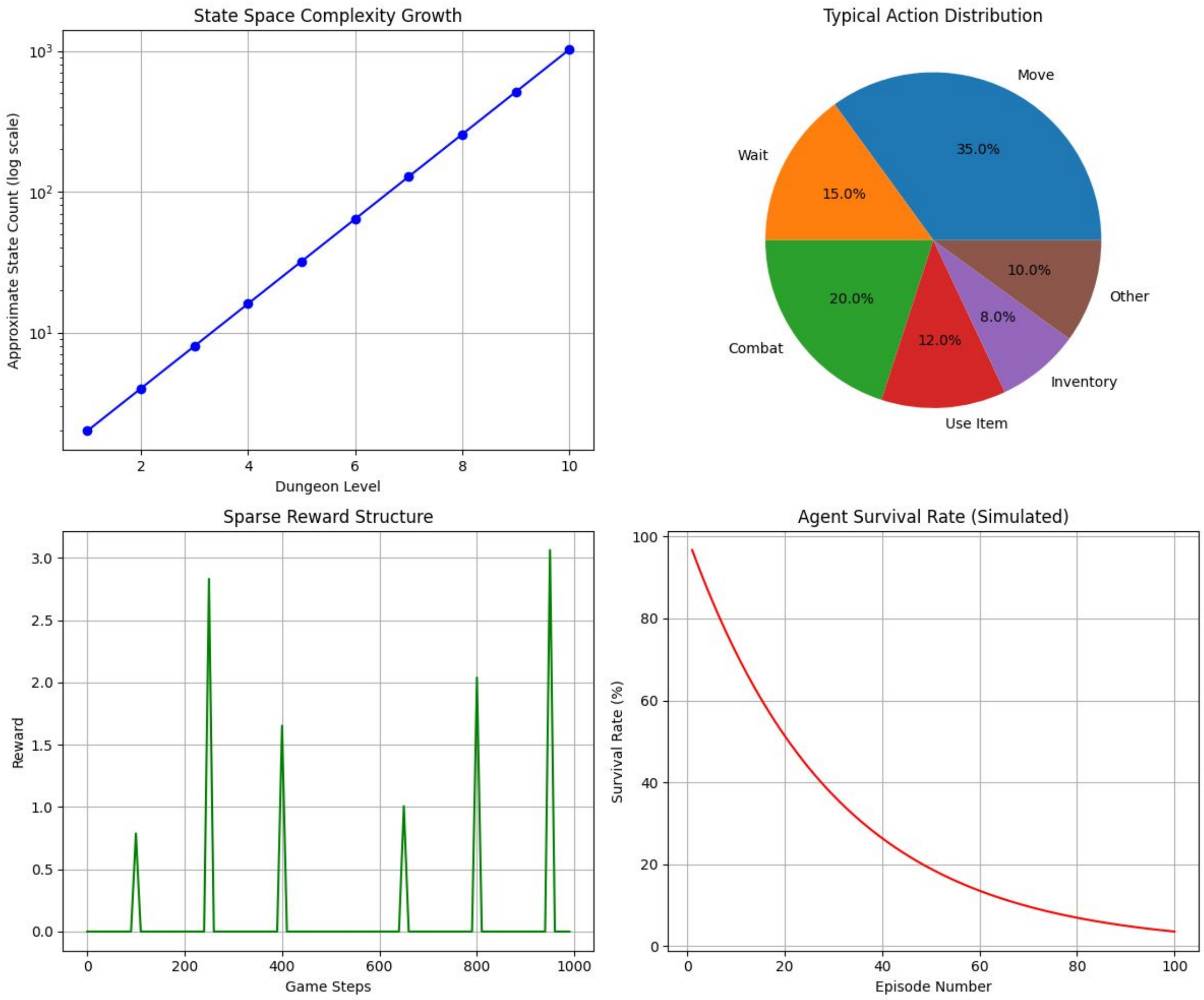
Type 3: Adversarial Prompts

Instruction injection attempts

Role reversal attacks

Test: System robustness to prompt manipulation

NetHack Example



Key Technical Innovations

1. **Causal Intervention Learning**: Unlike existing work that treats LLM advice as fixed, this learns when and why to intervene
2. **Adversarial State Robustness**: First framework to systematically test LLM-RL robustness to corrupted information
3. **Counterfactual Strategy Evaluation**: Explicit reasoning about alternative strategies using causal models
4. **Multi-Modal Validation**: Cross-checking LLM advice against causal expectations
5. **Interpretable Decision Making**: Transparent explanations of why strategies work or fail

Hardware Requirements

Minimum specifications:

- **CPU:** Quad-core CPU (Intel i5 or AMD Ryzen 5, 4+ cores).
- **GPU:** NVIDIA GTX 1650 or RTX 2060 with 4–6 GB VRAM (or no GPU if using CPU-only experiments).
- **RAM:** 16 GB.
- **Storage:** 512 GB SSD.
- **Networking:** Stable broadband connection.

Recommended (Balanced Setup)

- **CPU:** 8-core processor (Intel i7 or AMD Ryzen 7).
- **GPU:** NVIDIA RTX 3060/3070 (12 GB VRAM or more).
- **RAM:** 32 GB.
- **Storage:** 1 TB NVMe SSD.
- **Networking:** Gigabit Ethernet or high-speed Wi-Fi.

Software Requirements

Programming Language & Libraries:

- Python 3.10+ with NumPy, Pandas, Matplotlib, PyTorch/TensorFlow.
- Hugging Face Transformers for LLMs.
- Stable-Baselines3 or RLlib for RL algorithms.
- OpenAI Gymnasium, MuJoCo, and PettingZoo for simulation environments.
- MLflow or Weights & Biases for experiment tracking.

Scalability & Deployment Tools:

- Docker for reproducibility, Ray for distributed training if scaling.

Operating System:

- Linux (Ubuntu 22.04 LTS preferred), macOS or Windows for dev work.

References

1. Zeng, F., et al. (2023). Large language models for robotics: A survey. *arXiv preprint arXiv:2311.07226*. https://arxiv.org/abs/2311.07226
2. Liu, S., et al. (2024). RL-GPT: Integrating reinforcement learning and code-as-policy. *arXiv preprint arXiv:2402.19299*. https://arxiv.org/abs/2402.19299
3. Carta, T., et al. (2024). Grounding large language models in interactive environments with online reinforcement learning. *arXiv preprint arXiv:2302.02662*. https://arxiv.org/abs/2302.02662
4. Ahn, M., et al. (2022). Do as I can, not as I say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*. https://arxiv.org/abs/2204.01691
5. Liang, J., et al. (2023). Code as policies: Language model programs for embodied control. *arXiv preprint arXiv:2209.07753*. https://arxiv.org/abs/2209.07753
6. Wang, G., et al. (2023). Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*. https://arxiv.org/abs/2305.16291
7. Wu, S., et al. (2024). Enhance reasoning for large language models in the game Werewolf. *arXiv preprint arXiv:2402.02330*. https://arxiv.org/abs/2402.02330

[redacted]

References

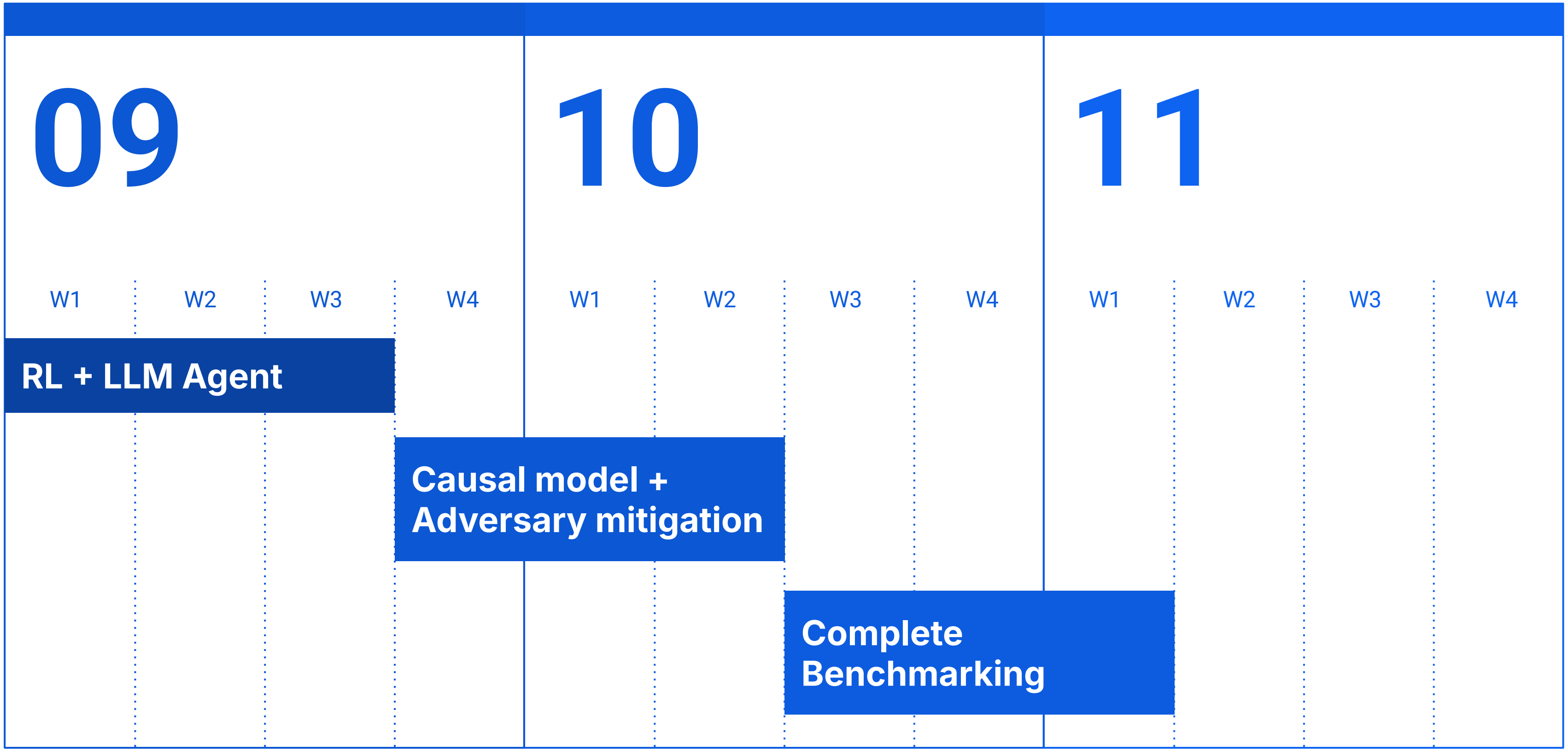
8. Huang, W., et al. (2023). Inner monologue: Embodied reasoning through planning with language models. In K. Liu, D. Kulic, & J. Ichnowski (Eds.), *Proceedings of the 6th Conference on Robot Learning* (pp. 1769–1782). PMLR. https://proceedings.mlr.press/v205/huang23c.html
9. Sahoo, S. S., et al. (2024). Large language models for biomedicine: Foundations, opportunities, challenges, and best practices. *Journal of the American Medical Informatics Association, 31*(9), 2114–2124. https://doi.org/10.1093/jamia/ocae074
10. Küttler, H., et al. (2020). The NetHack learning environment. *arXiv preprint arXiv:2006.13760*. https://arxiv.org/abs/2006.13760
11. Chevalier-Boisvert, M., et al. (2019). BabyAI: A platform to study the sample efficiency of grounded language learning. *arXiv preprint arXiv:1810.08272*. https://arxiv.org/abs/1810.08272
12. Shridhar, M., et al. (2021). ALFWorld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*. https://arxiv.org/abs/2010.03768
13. Mees, O., et al. (2022). CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *arXiv preprint arXiv:2112.03227*. https://arxiv.org/abs/2112.03227
14. Amodei, D., et al. (2016). Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*. https://arxiv.org/abs/1606.06565

[redacted]

References

15. Raji, I. D., & Dobbe, R. (2023). Concrete problems in AI safety, revisited. *arXiv preprint arXiv:2401.10899*. https://arxiv.org/abs/2401.10899
16. Bhattacharjee, A., et al. (2023). Towards LLM-guided causal explainability for black-box text classifiers. *Semantic Scholar*. https://api.semanticscholar.org/CorpusID:262459118
17. García, J., & Fernández, F. (2015). A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research, 16*(42), 1437–1480. http://jmlr.org/papers/v16/garcia15a.html

Timeline


09
10
11
W1
W2
W3
W4
W1
W2
W3
W4
W1
W2
W3
W4
RL + LLM Agent
Causal model +
Adversary mitigation
Complete
Benchmarking