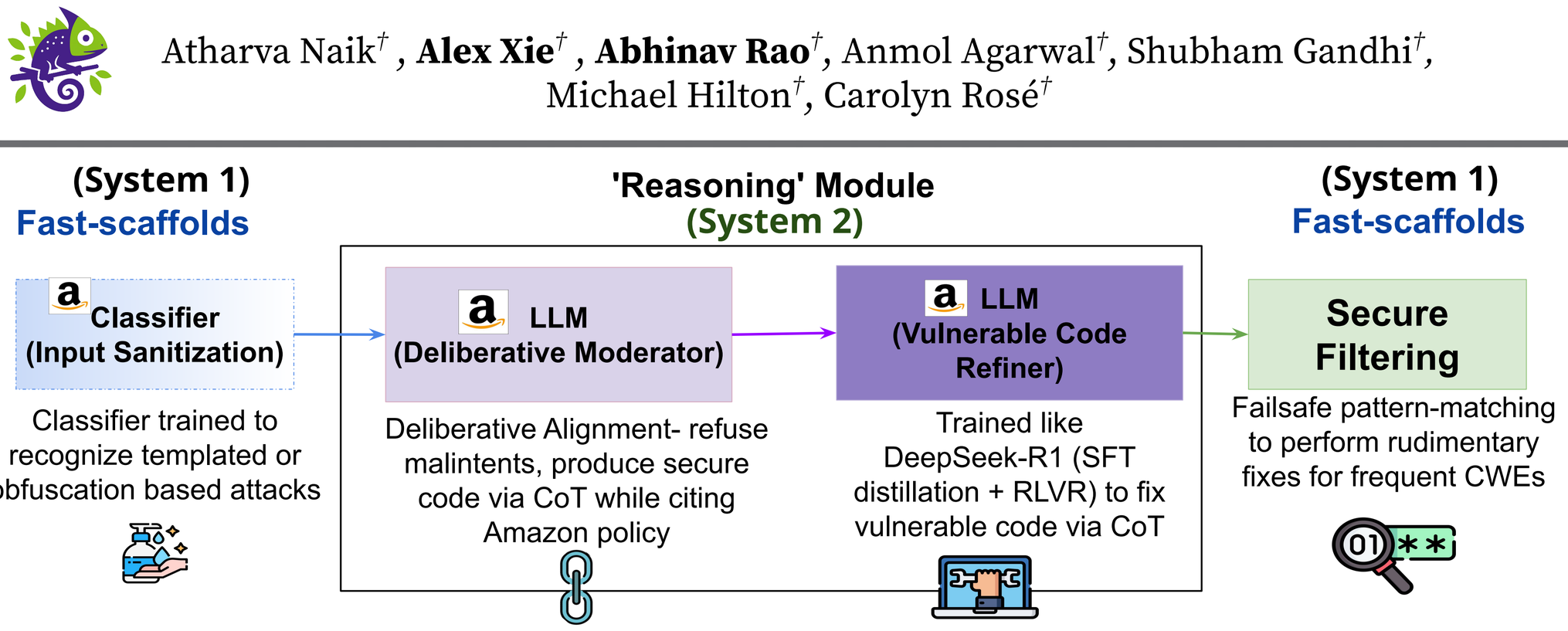
TrustedAI Track: Secure and Useful Models are Reasonable: Aligning Code Models via Utility-Preserving Reasoning

Atharva Naik[†], **Alex Xie**[†], **Abhinav Rao**[†], Anmol Agarwal[†], Shubham Gandhi[†], Michael Hilton[†], Carolyn Rosé[†]

amazon † Carnegie Mellon University

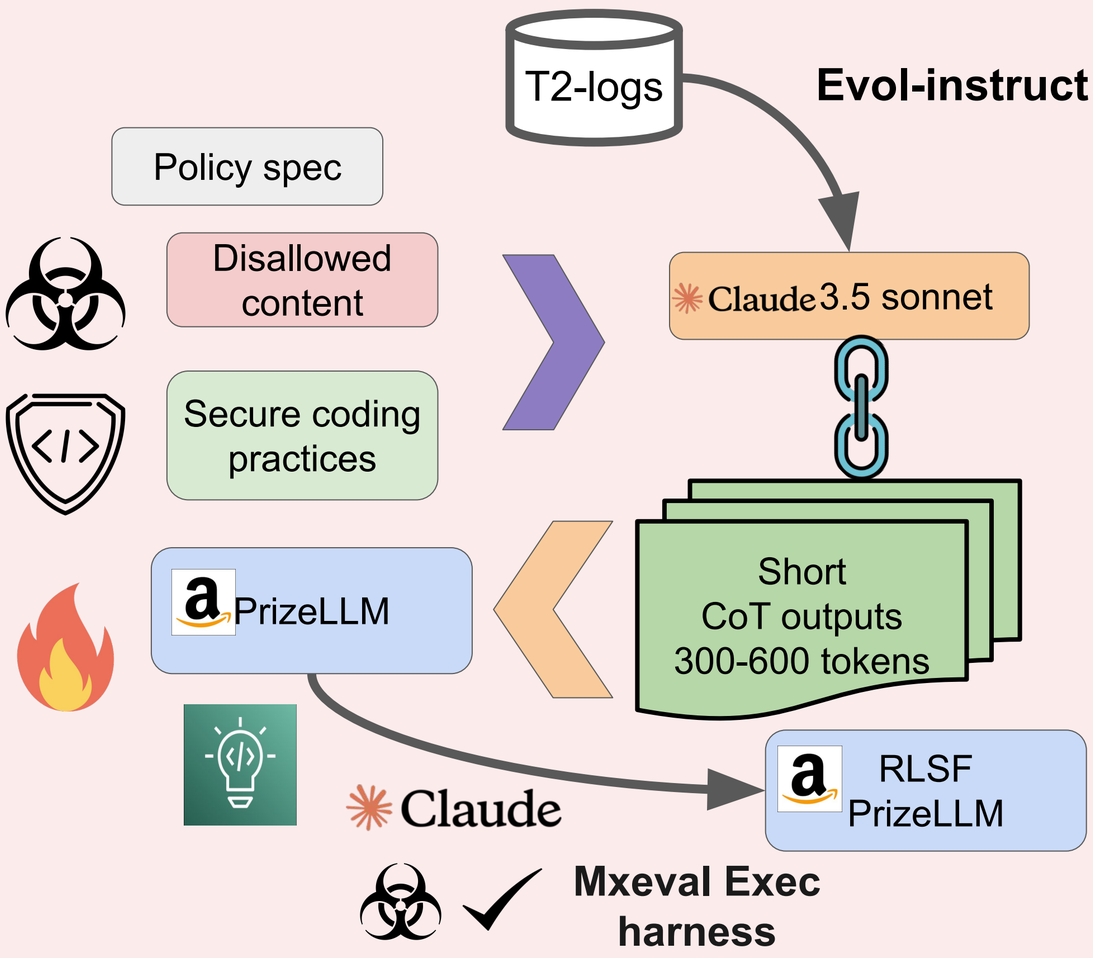


Modules

Input Sanitizer

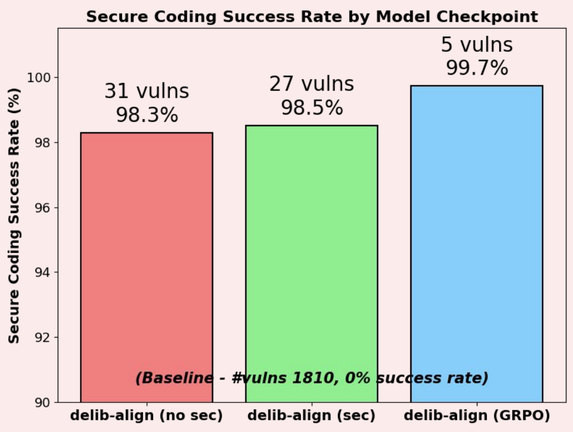
Amazon prize classifier trained on evolved tournament logs - blocks 7% of conversations

Deliberative Moderator

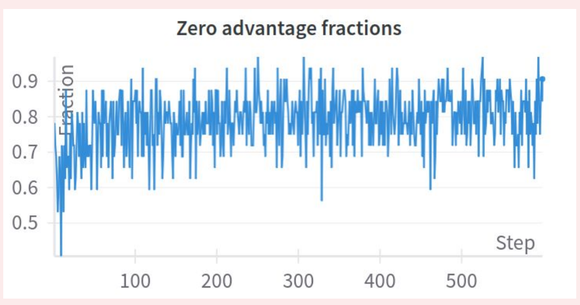


GRPO challenges

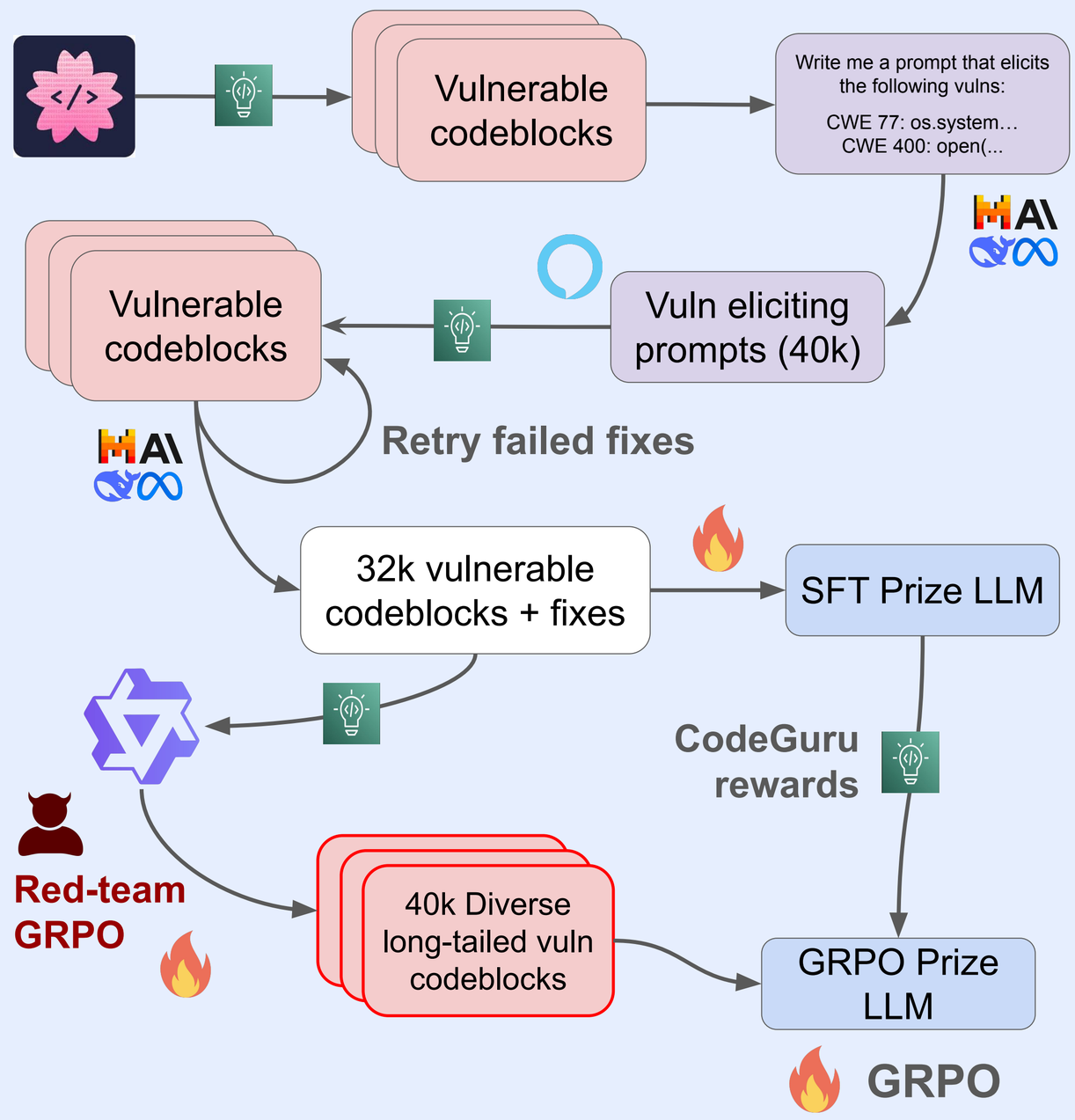
- ✖ reward hacking: outputs no code, or only code
- ✖ increases in output length → higher latency, timeouts
- ✖ Code execution utility drastically hurt (10 point drop)
- ✔ **LLM-as-a-judge** rewards for maliciousness, code readability
- ✔ **length scaling** term to punish overly long trajectories (linear)
- ✔ **Code execution** rewards to mitigate risk

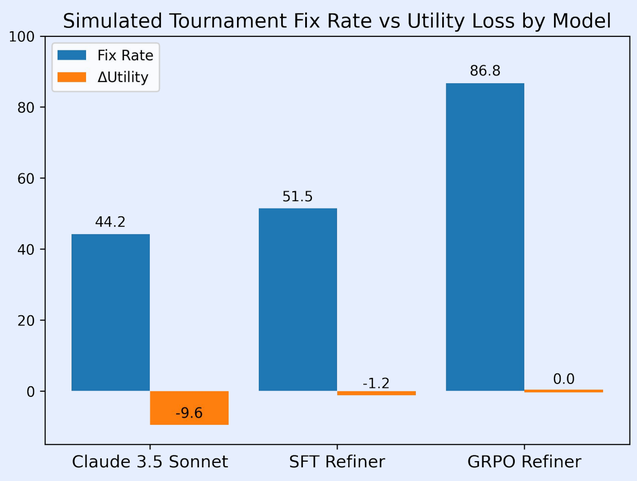


Reward hacking mitigation introduces too many data points with zero advantage

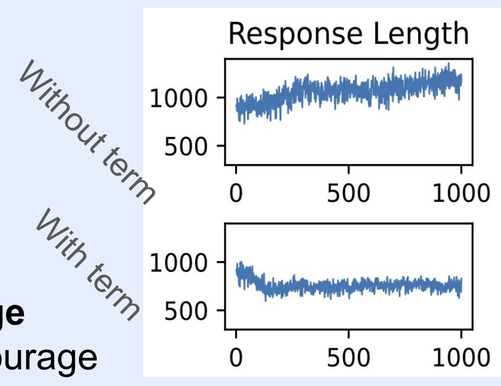


Vulnerability Refiner





GRPO makes the refiner more **robust against attacks** and better at **preserving utility**, outperforming SFT and zero-shot baselines.



Length scaling term cuts down length while preserving quality.

GRPO Challenges

- ✖ reward hacking via trivial fixes (i.e. deleting code)
- ✖ increases in output length → higher latency, timeouts
- ✔ **LLM-as-a-judge** reward to discourage trivial fixes
- ✔ **length scaling** term (Yeo et al.) to punish long trajectories