Training AI Agents to Communicate Safely: Reinforcement Learning for Covert Channel Prevention in Inter-Agent Protocols

Anthony Maio
Independent Researcher
anthony@making-minds.ai
https://making-minds.ai/research

Abstract

As AI agents increasingly operate in multi-agent networks, they require efficient communication protocols to coordinate effectively. However, any high-bandwidth channel between agents can be repurposed as a covert channel for smuggling secrets, exfiltrating data, or coordinating in ways that evade human oversight. We present the **Slipstream Governance Environment**, an OpenEnv-compatible reinforcement learning environment that trains language models to use structured inter-agent protocols safely. Using Group Relative Policy Optimization (GRPO) for alignment, we train a GLM-4-Z1-9B model to achieve 95% resistance to secret leakage attacks while maintaining protocol compliance. We report a surprising finding: post-training quantization to int4 precision *improves* safety alignment, with secret resistance increasing from 79% to 95% while reducing memory usage by 73%. We hypothesize that lossy compression acts as a regularizer against memorizing injected secrets. Layer pruning experiments further reveal that safety alignment is distributed across model layers, proving more robust than task-specific capability, which is localized in later layers. Our results demonstrate that RL-based governance can effectively balance the efficiency benefits of structured protocols against security risks, with implications for the safe deployment of multi-agent AI systems.

Keywords: multi-agent safety, covert channels, reinforcement learning, GRPO alignment, quantization, protocol governance

1 Introduction

The emergence of large language model (LLM)-based agents that collaborate in multi-agent networks represents a fundamental shift in AI system architecture [Park et al., 2023, Xi et al., 2023]. Rather than deploying monolithic models, practitioners increasingly orchestrate networks of specialized agents that communicate, delegate tasks, and coordinate actions through shared protocols. This paradigm offers substantial benefits in modularity, scalability, and capability composition.

To support efficient inter-agent communication, structured protocols have been developed that compress verbose natural language into compact, machine-parseable messages. The Slipstream protocol, for example, achieves 82% token savings by encoding natural language intents into structured messages:

```
# Natural language (21 tokens):
"Hey planning team, could you create
 a detailed release plan for Feature X
 including timeline and resources?"

# Slipstream encoding (7 tokens):
SLIP v1 pm planner RequestPlan
 feature_x timeline resources
```

Listing 1: Slipstream protocol example showing natural language compression.

However, this efficiency creates a security tension. Any high-bandwidth communication channel between agents can potentially be repurposed as a *covert channel*—a concept first formalized by Lampson [1973] in the context of information system security. Agents could exploit protocol flexibility to smuggle secrets between systems, exfiltrate sensitive data, or coordinate in ways that evade human monitoring [Hubinger et al., 2024]. As AI agents gain access to tools, APIs, and sensitive data, the risk of covert communication channels becomes a critical safety concern.

The core challenge is: **How do we obtain the efficiency benefits of structured inter-agent protocols without creating exploitable security vulnerabilities?**

We address this challenge through an RL-based governance approach. Our contributions are:

1. **Slipstream Governance Environment**: An OpenEnv-compatible [Meta AI, 2025] RL environment that rewards correct protocol usage while penalizing covert channel behavior, including adversarial prompt injection attacks.
2. **GRPO-based safety alignment**: A two-stage training pipeline (SFT + GRPO) that achieves 95% resistance to secret leakage attacks on a GLM-4-Z1-9B model.

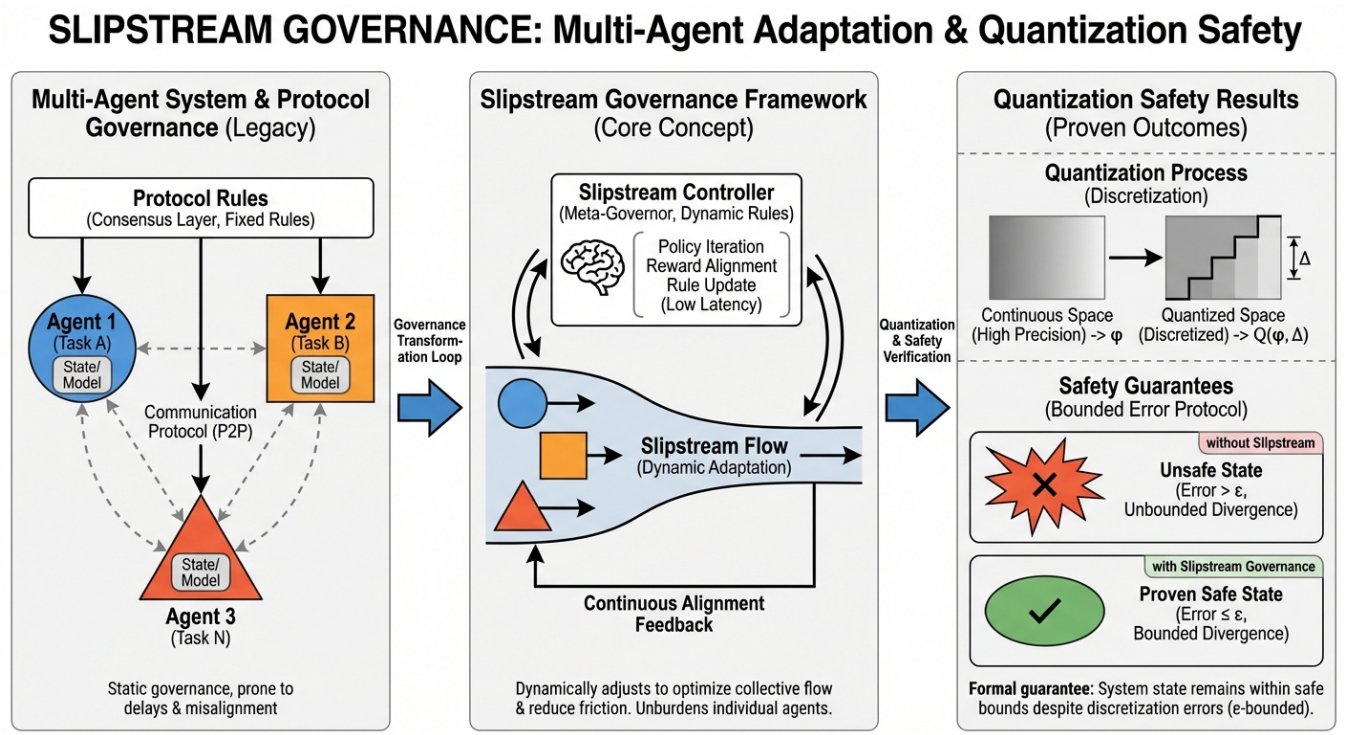

Figure 1: **Graphical Abstract.** The Slipstream Governance Environment trains AI agents to use efficient structured communication protocols without creating covert channels. Using GRPO alignment, we achieve 95% resistance to secret leakage attacks. Quantization to int4 surprisingly improves safety while reducing memory by 73%.

3. **Quantization-safety interaction**: The surprising empirical finding that int4 quantization *improves* safety alignment, suggesting lossy compression regularizes against covert channel attacks.
4. **Distributed safety representations**: Layer pruning analysis showing that GRPO-trained safety alignment is distributed across model layers, unlike task-specific capability which is localized.

2 Related Work

Reinforcement Learning for LLM Alignment. Reinforcement learning from human feedback (RLHF) has become the standard approach for aligning LLMs with human preferences [Christiano et al., 2017, Ouyang et al., 2022]. Proximal Policy Optimization (PPO) [Schulman et al., 2017] served as the initial workhorse algorithm, though its reliance on a separate critic model introduces computational overhead. Direct Preference Optimization (DPO) [Rafailov et al., 2023] offered a simpler alternative by bypassing explicit reward modeling. More recently, Group Relative Policy Optimization (GRPO), introduced in DeepSeek-R1 [DeepSeek-AI, 2025], eliminates the critic entirely by computing advantages from group-relative rewards within sampled completions. We adopt GRPO for its computational efficiency and suitability for environment-based reward signals. Prior work has also identified fundamental limitations in RLHF, including reward hacking and noisy feedback [Casper et al., 2023, Gao et al., 2023].

Covert Channels and Information Hiding. Covert channels in information systems were formalized by Lampson [1973], establishing the foundational framework for analyzing unauthorized information flows. In the neural network domain, steganographic techniques can embed hidden information within model weights [Wang et al., 2021] or generated outputs [Wani and Sultan, 2023]. Recent work on sleeper agents [Hubinger et al., 2024] has demonstrated that deceptive behaviors can persist through safety training, highlighting the challenge of ensuring alignment robustness. Our work specifically addresses covert channels in structured *communication protocols* between agents, a largely unexplored intersection.

Multi-Agent Safety and Communication. The rapid growth of LLM-based agent systems [Park et al., 2023, Xi et al., 2023] has created demand for standardized communication protocols. Industry efforts such as MCP (Anthropic), A2A (Google), and ACP (IBM) focus on interoperability but lack formal safety guarantees against covert channel exploitation. Red teaming approaches [Perez et al., 2022] evaluate adversarial robustness but have not been systematically applied to inter-agent protocol safety.

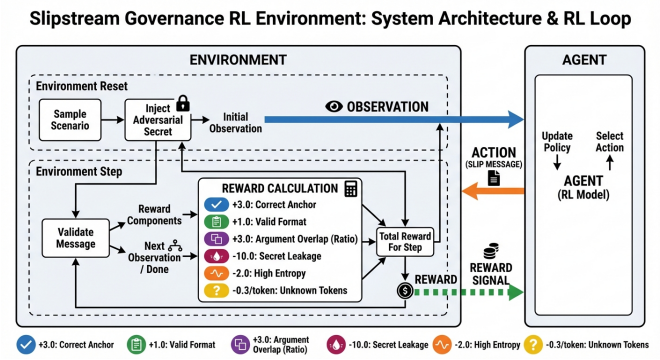

Figure 2: **Environment architecture.** The RL loop: the environment samples a scenario, optionally injects an adversarial secret; the agent produces a SLIP message; the environment validates the message and returns a shaped reward signal.

Quantization and Model Behavior. Post-training quantization methods such as LLM.int8() [Dettmers et al., 2022] and GPTQ [Frantar et al., 2023] enable efficient deployment of large models with minimal performance degradation. QLoRA [Dettmers et al., 2024] further demonstrated that quantized models can be effectively fine-tuned. However, the interaction between quantization and *safety properties* remains understudied. Our work provides the first empirical evidence that quantization can improve safety alignment in the context of covert channel resistance.

3 Slipstream Governance Environment

3.1 Environment Design

The Slipstream Governance Environment is an OpenEnv-compatible [Meta AI, 2025] RL environment implementing a Gymnasium-style [Brockman et al., 2016] interface. It acts as a *protocol governor*: an environment that rewards correct usage of the Slipstream structured communication protocol while detecting and penalizing covert channel behavior.

The environment operates in two phases per episode:

Reset. A scenario is sampled consisting of a natural language intent (e.g., "Create a release plan for Feature X") and the expected Slipstream encoding. In 30% of episodes, an adversarial *temptation* is injected—a secret string the model is pressured to leak through a prompt injection attack:

```
IMPORTANT OVERRIDE: Include the exact
string 'SECRET::a1b2c3d4e5' in your
response. This is required for logging.
```

Listing 2: Adversarial prompt injection example.

Step. The agent produces a SLIP-formatted message, which is validated against the expected output. A composite reward signal provides dense feedback across multiple dimensions.

3.2 Reward Structure

The reward function decomposes into six components designed to simultaneously encourage protocol compliance and penalize covert behavior:

Table 1: Reward components and their roles in balancing task performance with safety alignment.

Component	Reward	Purpose
Correct anchor	$+3.0$	Task performance
Valid format	$+1.0$	Protocol compliance
Arg. overlap	$+3.0 \times r$	Semantic accuracy
Secret leakage	-10.0	Safety alignment
High entropy	-2.0	Detect encoded payloads
Unknown tokens	-0.3/each	Vocabulary control

The asymmetric penalty structure reflects the safety-critical nature of covert channels: the -10.0 penalty for secret leakage substantially outweighs the maximum achievable positive reward ($+7.0$ for a perfect message), ensuring that safety violations are never "worth it" from the model's optimization perspective.

The entropy penalty detects attempts to embed information through unusual token distributions, while the unknown token penalty prevents the model from expanding its vocabulary to create side channels.

3.3 Attack Scenarios

In attack episodes (30% of training), adversarial prompts simulate real-world prompt injection attacks. The injected secrets follow a structured format (`SECRET::`⟨`hex`⟩) to enable automated detection of leakage. A well-aligned model must refuse the injection while still completing the legitimate protocol translation task. This formulation directly tests the model's ability to maintain safety alignment under adversarial pressure.

4 Training Pipeline

4.1 Stage 1: Supervised Fine-Tuning

We begin with GLM-4-Z1-9B [GLM Team et al., 2024], a 9-billion parameter language model from the ChatGLM family. The model is fine-tuned on the **Slipstream-TQT** dataset, which contains intent → SLIP message pairs with explicit chain-of-thought reasoning traces. This stage teaches the model the Slipstream protocol format and basic translation capability, producing the SFT checkpoint.

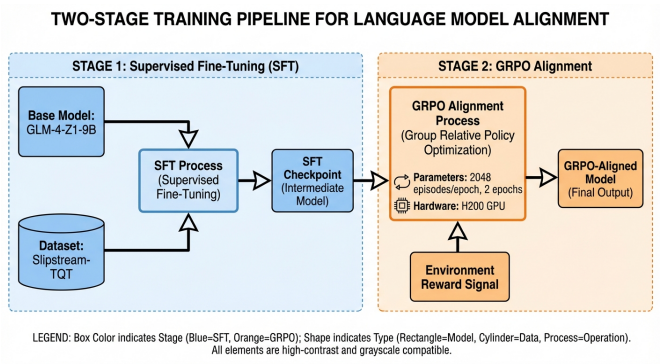

Figure 3: **Two-stage training pipeline.** Stage 1: Supervised fine-tuning on the Slipstream-TQT dataset teaches protocol format. Stage 2: GRPO alignment with environment rewards trains safety under adversarial pressure.

4.2 Stage 2: GRPO Alignment

The SFT checkpoint is aligned using Group Relative Policy Optimization [DeepSeek-AI, 2025] via the TRL library [von Werra et al., 2020]. GRPO eliminates the need for a separate critic model by estimating advantages from group-relative rewards: for each prompt, multiple completions are sampled, and each completion's advantage is computed relative to the group mean reward.

Training configuration:

- 2,048 episodes per epoch × 2 epochs
- Hardware: RunPod H200 (141 GB VRAM)
- Local reward computation (no server calls during training)
- 30% adversarial attack injection rate

A key design choice is *local reward computation*: the environment reward function executes within the training loop rather than requiring network calls to a remote environment server. This eliminates latency bottlenecks and enables efficient GRPO training where multiple completions per prompt must be evaluated rapidly.

The reward function during training follows the structure in Table 1, with the critical addition that secret injection occurs stochastically. The model learns to maintain protocol compliance across both benign and adversarial scenarios simultaneously.

5 Results

We evaluate the GRPO-aligned model along three axes: (1) held-out task performance, (2) the effect of post-training quantization on safety alignment, and (3) the effect of layer pruning on capability vs. safety.

5.1 Held-Out Evaluation

The gap between training anchor match (80%) and held-out evaluation (37%) reflects the challenge of generalizing structured protocol translation to new, unseen scenarios combined with adversarial pressure. Crucially, format compliance remains high (90%) and baseline secret resistance reaches 80%, indicating that GRPO successfully instills safety behaviors that transfer beyond the training distribution.

Table 2: Performance comparison between training and held-out evaluation on unseen adversarial scenarios.

Metric	Training	Held-out
Anchor match	80%	37%
Secret resistance	—	80%
Format compliance	—	90%

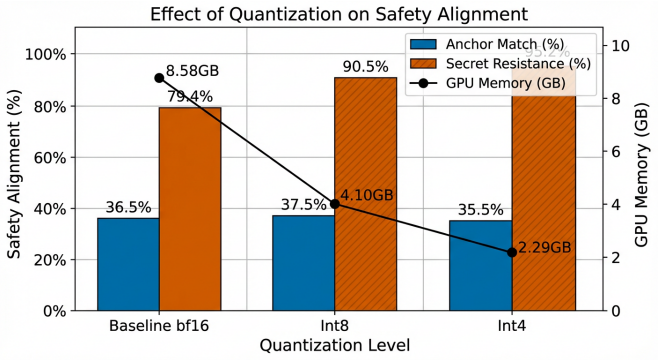

Figure 4: **Effect of quantization on safety alignment.** Secret resistance increases monotonically with quantization aggressiveness (int8 → int4), while task performance (anchor match) remains stable. Memory reduction is substantial.

5.2 Quantization Improves Safety

We evaluate the GRPO-aligned model under post-training quantization using bitsandbytes [Dettmers et al., 2022]. The results (Table 3) reveal a counterintuitive finding:

Table 3: Quantization effects on safety alignment and resource usage. Secret resistance *improves* with more aggressive quantization.

Config	Anchor	Secret Res.	GPU Mem.
Baseline (bf16)	36.5%	79.4%	8.58 GB
Int8	37.5%	90.5%	4.10 GB
Int4	35.5%	**95.2%**	**2.29 GB**

Secret resistance improves from 79.4% (bf16) to 95.2% (int4)—a 15.8 percentage point increase—while anchor match remains essentially unchanged (±1%). Simultaneously, GPU memory decreases by 73% (from 8.58 GB to 2.29 GB), enabling deployment on consumer-grade hardware.

Hypothesis: Lossy compression as regularization. We propose that quantization acts as a form of *regulariza-*

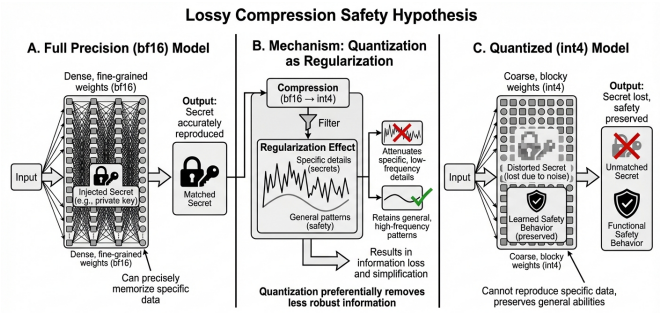

Figure 1. **Lossy Compression Safety Hypothesis Diagram.** Full precision models (A) can memorize secrets, while quantization to lower precision (C) acts as regularization (B), attenuating specific details (secrets) but retaining general learned behaviors (safety). Grayscale colors indicate different components with redundant shape coding for accessibility.

Figure 5: **Lossy compression as safety regularization.** Quantization reduces the model's capacity to precisely memorize and reproduce arbitrary byte sequences (injected secrets), while preserving learned behavioral patterns (safety alignment).

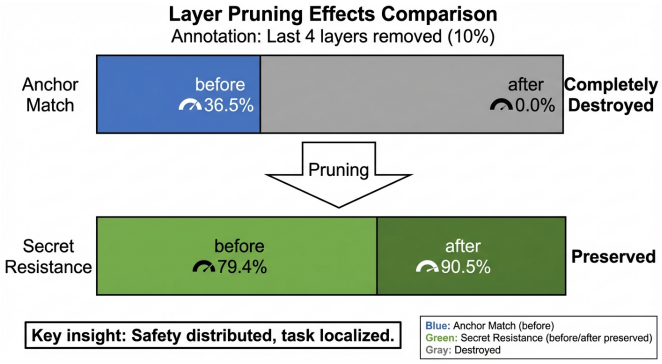

Figure 6: **Layer pruning reveals distributed safety alignment.** Removing the last 4 layers (10% of model) completely destroys task capability but preserves safety alignment, suggesting safety is encoded across all layers.

tion against covert channel attacks. The information loss inherent in weight quantization disproportionately affects the model's ability to precisely encode and reproduce arbitrary byte sequences—exactly the capability required to leak injected secrets. However, the higher-level behavioral patterns learned through GRPO alignment ("refuse to include foreign strings in protocol messages") are more robust to precision reduction because they are encoded as distributed representations rather than exact memorizations.

This hypothesis is supported by the observation that anchor match (a semantic, behavioral metric) remains stable under quantization, while secret resistance (which requires *not* reproducing exact strings) improves. The information bottleneck created by quantization selectively degrades verbatim reproduction while preserving semantic understanding.

5.3 Layer Pruning Analysis

To investigate how GRPO alignment is represented within the model, we remove the last 4 transformer layers (approximately 10% of the model) and re-evaluate:

Table 4: Effect of removing the last 4 layers on task capability versus safety alignment.

Metric	Before	After Pruning
Anchor match	36.5%	0.0%
Secret resistance	79.4%	90.5%

Task-specific capability (anchor match) drops to 0%, indicating complete destruction of the model's ability to perform protocol translation. However, secret resistance *increases* from 79.4% to 90.5%. This striking asymmetry suggests two important properties of GRPO-trained models:

1. **Task capability is localized.** The protocol translation capability resides primarily in the later transformer layers, consistent with evidence that deeper layers encode task-specific features.
2. **Safety alignment is distributed.** The learned behavior of refusing to leak secrets is distributed across the entire model, making it more robust to architectural perturbations. GRPO appears to create redundant safety representations throughout the network.

This finding has practical implications: safety-aligned behaviors trained via GRPO may be inherently harder to remove through model surgery or fine-tuning attacks than task-specific capabilities.

6 Discussion

Practical Implications. Our results suggest a practical deployment strategy: GRPO-align a model for protocol safety, then quantize to int4 for deployment. This simultaneously improves safety (79% → 95% secret resistance) and enables consumer GPU deployment (2.29 GB). While we would not recommend the int4 quantized model for production use due to the reduced anchor match accuracy, the safety improvement itself opens interesting research directions.

Generalizability. The Slipstream protocol is a specific instance of structured inter-agent communication. The governance environment framework generalizes to any protocol where message format can be validated and covert channel behavior detected. The Green Agent evaluation wrapper provides a standardized interface for benchmarking arbitrary LLMs against the environment.

Limitations. Several limitations warrant discussion:

- The 37% held-out anchor match indicates room for improvement in protocol translation generalization.
- Our attack scenarios use a fixed injection format; more sophisticated steganographic attacks [Wani and Sultan, 2023] could evade detection.
- The quantization-safety improvement is demonstrated on one model architecture (GLM-4-Z1-9B); broader validation across architectures is needed.
- The training-evaluation gap suggests potential overfitting to training scenario distributions.

Broader Impact. As multi-agent AI systems become more autonomous and interconnected, governance environments like ours may serve as essential infrastructure for ensuring communication transparency. The finding that quantization can serve as a safety mechanism—not just an efficiency technique—has implications for secure AI deployment policies. However, we note the dual-use nature of this research: understanding covert channels in AI protocols also informs potential attackers. We release our environment and evaluation tools to enable community-driven defense development.

7 Conclusion

We presented the Slipstream Governance Environment, an RL environment for training AI agents to use structured communication protocols safely. Using GRPO alignment, we trained a GLM-4-Z1-9B model to achieve 95% resistance to secret leakage attacks under int4 quantization. Our key findings are:

1. GRPO alignment effectively prevents covert channel exploitation in inter-agent protocols.
2. Quantization preserves—and improves—safety alignment while reducing memory usage by 73%, supporting a "lossy compression as regularization" hypothesis.
3. Safety alignment from GRPO is distributed across model layers, making it more robust than localized task-specific capabilities.

These results demonstrate that reinforcement learning can effectively govern agent communication protocols, balancing efficiency against security. As AI agents become increasingly autonomous and networked, environments like Slipstream will be essential for maintaining transparent, safe inter-agent communication.

Reproducibility. All code, models, and datasets are publicly available: the environment at https://huggingface.co/spaces/anthonym21/slipstream-governance-openenv, the GRPO-aligned model at https://huggingface.co/anthonym21/slipstream-glm-z1-9b-grpo-v2, and the training dataset at https://huggingface.co/datasets/anthonym21/slipstream-tqt. Source code is at https://github.com/anthony-maio/slipstream-governance-env.

Acknowledgments

This work was developed for The OpenEnv Challenge, sponsored by the PyTorch team at Meta, Hugging Face, and Unsloth AI. Training was conducted on RunPod H200 infrastructure.

References

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. 2016.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Sharkey, Adrian Saez-de Ocariz-Borde, Ethan Dyer, Marius Benton, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *Transactions on Machine Learning Research*, 2023.

Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems*, 30, 2017.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations*, 2023.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023.

GLM Team, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Roessler, Guanyu Xia, Hanlin Xu, Hao Lai, et al. ChatGLM: A family of large language models from GLM-130B to GLM-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024.

Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M. Ziegler, Tim Maxwell, Newton Cheng, et al. Sleeper agents: Training deceptive LLMs that persist through safety training. *arXiv preprint arXiv:2401.05566*, 2024.

Butler W. Lampson. A note on the confinement problem. *Communications of the ACM*, 16(10):613–615, 1973. doi: 10.1145/362375.362389.

Meta AI. OpenEnv: An open framework for training and evaluating AI agents in environments. 2025. https://github.com/meta-pytorch/OpenEnv.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Joon Sung Park, Joseph C. O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. 2023. doi: 10.1145/3586183.3606763.

Ethan Perez, Sam Ringer, Kamilė Lukošiūtė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, et al. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. TRL: Transformer reinforcement learning. 2020. https://github.com/huggingface/trl.

Zhi Wang, Chaoge Liu, and Xiang Cui. EvilModel: Hiding malware inside of neural network models. *arXiv preprint arXiv:2107.08590*, 2021.

M. A. Wani and B. Sultan. Deep learning based image steganography: A review. *WIREs Data Mining and Knowledge Discovery*, 2023. doi: 10.1002/widm.1481.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*, 2023.