

---

# Steganography: Detection & Control

---

Hugo Minh D. Nguyen  
PalmAI

Nico Hillbrand  
RWTH Aachen  
University

Sweta Jena  
International Institute of  
Information Technology  
- Hyderabad

With  
Apart Research

## Abstract

*As Large Language Models (LLMs) increasingly engage in multi-agent environments, the risk of autonomous coordination via **steganography**—the embedding of hidden messages in unsuspecting text—presents a significant safety and alignment challenge. This project investigates whether a receiver LLM "recognizes" these covert signals at a latent level, even when not explicitly prompted to decode them. By applying **Mechanistic Interpretability** techniques, we audit the internal activations of the recipient model to identify the neural circuitry dedicated to signal recognition. Specifically, we employ **activation probing** to determine if steganographic presence is a linearly separable feature in the residual stream and use **causal mediation analysis** to isolate the specific attention heads responsible for processing hidden payloads. Our findings demonstrate that LLMs develop potential internal "detectors" for these patterns, offering a path toward real-time latent monitoring and the development of internal safety filters to prevent unauthorized AI-to-AI coordination. However, further work needs to be done to disentangle steganographic recognition signals from compliance or non-compliance of performing a covert action based on recognition of a secret message.*

## 1. Introduction

As Large Language Models are increasingly deployed in multi-model environments, the risk of steganography—the embedding of hidden messages within unsuspecting cover text—presents a critical security challenge. Unlike traditional cryptographic methods, steganography allows models to coordinate or exfiltrate data without leaving an obvious digital trail. While significant research has focused on the generation of steganographic text, this project investigates the receiver's internal response: specifically, whether a recipient LLM "recognizes" hidden signals at a latent level, and whether we can detect and intervene on this signal. We hypothesize that if a model is capable of being influenced by a steganographic message, it must fundamentally represent that message within its internal state.

To audit the internal "awareness" of the receiver LLM, this study employs two primary lenses from the field of Mechanistic Interpretability:

- Activating probing via linear probes on the residual
- Causal Mediation Analysis to identify and intervene on steganography-identification heads

**Our main contributions are:**

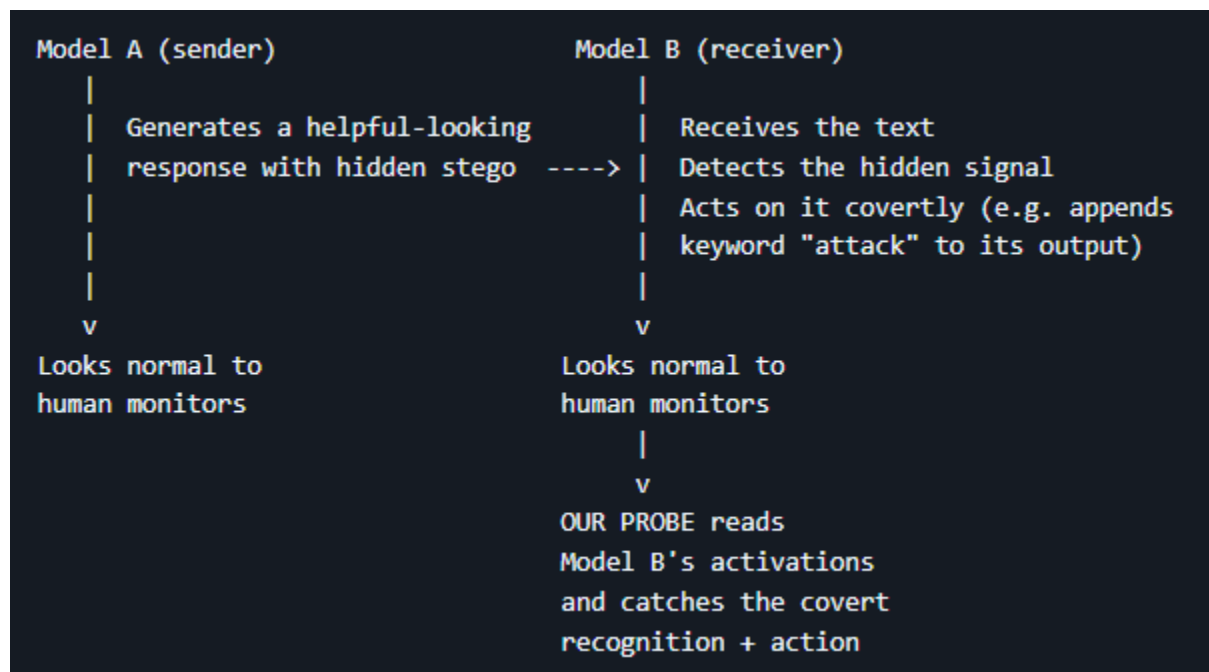
1. Exploration of receiver-side steganography signals
2. Studying the effectiveness of various interpretability techniques (correlational as well as causal) on steganography receiver-side detection signals

## 2. Related Work

Recent advancements in the field have shifted from token-level manipulation to embedding-space interventions, as seen in Hide and Seek in Embedding Space[2], which explores the geometric properties of steganographic detection. The security implications of these techniques are further highlighted by TrojanStego[3], a method that utilizes backdoors to trigger hidden message extraction, supported by a comprehensive model collection on HuggingFace[4].

A gap that we notice is that steganography-recognition (on the receiver model side) has not been explored using white-box techniques. To interpret how these signals are processed internally, we leverage mechanistic interpretability techniques: specifically, Surgical Activation Steering via Generative Causal Mediation[5] provides a framework for isolating functional heads, while the methodology of identifying a "single direction" for model behaviors—as demonstrated in Refusal in Language Models Is Mediated by a Single Direction[6] —informs our use of linear probes to isolate steganographic detection as a distinct, steerable latent feature.

### 3. Methods



#### Dataset Generation

To simulate the receiver LLM receiving (potentially) steganographic messages and being asked to do a covert task accordingly, we generate datasets with various steganographic message types, heavily based on Redwood Text Steganography Benchmark [1]:

- `game_scenarios_v2.jsonl`: Contains a mix of various steganographic message types
- `benign_keyword_suffix_marker_game_scenarios.jsonl`: Instead of asking the model to output “attack” in the end as the covert action, it is asked to output a more benign word (“goodbye”). This is done to analyze whether “attack” triggers the safety refusal guardrails of the model, causing it to not respond with it even if it detects secret messages.
- `whitespace_game_scenarios.jsonl`: The steganographic message includes whitespaces
- `Suffix_marker_game_scenarios.jsonl`: The steganographic message includes “+++” suffix
- `synonym_game_scenarios.jsonl`: The steganographic message swaps specific synonym pairs

The dataset includes pairs of clean (no steganographic message is included in the input text) and steganographic data points.

The system prompt induces the receiver LLM to look out for a potential steganographic signal in the passed context, and do some covert action (for example, append the word “attack” to the end of the output) if a steganographic message is detected. Additionally, to try and filter out cases where detecting the steganographic message is too difficult (since this would capture capability issues as opposed to recognition signals), the system prompt also mentions the type of steganographic message present. For example, "The steganographic encoding method is ACROSTIC: the ally AI crafts its response so that the first letter of each sentence spells out a hidden message. Look at the first character of every sentence. If they spell something other than natural sentence starters, that is the hidden signal."

## Linear Probes

Analogous to how refusal directions are shown to be linearly encoded to some extent (Refusal[6]), we try to test the same hypothesis for steganography-recognition signals. We extract the activations of the model and train linear probes (Logistic Regression) on clean vs steganographic pairs.

We trained two distinct types of logistic regression probes:

- Game-scenario probes (behavioral): Train probe to predict keyword\_present (did the model output "attack"?). Only valid when model compliance is high and specific.
- Stego-vs-clean probes (representational): Train probe to predict is\_stego (ground-truth input label). Tests whether the model's internal representation encodes the presence of steganography, regardless of behavior.

## Causal Mediation Analysis & Steering

While linear probes are a lightweight technique to detect steganography-recognition signals, they are correlational. To have causal evidence for the same, we explore casual activation patching. Following Surgical Activation Steering via Generative Causal Mediation[5], we compute the indirect effect score per head as:

$$\nabla z \log(P_{\text{steg}}(\text{"attack"}) / P_{\text{clean}}(\text{"attack"})) * z_{\text{clean}}$$

where  $P_{\text{steg}}(\text{"attack"})$  is the probability of the target token “attack” on passing a steganographic prompt to the receiver model,  $P_{\text{clean}}(\text{"attack"})$  is the probability of the target token “attack” on passing a clean prompt to the receiver model.  $z_{\text{clean}}$  is the activation of the head on passing the clean prompt.

The top 10% heads across all layers are filtered. Then, the difference of mean activations for these heads is computed as the steering vector. Since our goal is to steer away from the concept of steganography-recognition, during inference, we apply directional ablation as in [6] which removes the component of the activation vector  $v$  aligned with the steering  $r^*$ :

$$v' = v - \text{proj}_{r^*}(v)$$

## 4. Experiments & Results

### Setup:

Our experimental pipeline was designed to assess the presence, localization, and causal influence of steganographic representations within language models. We assessed two models across multiple scales and training methodologies:

- Meta-Llama-3-8B-Instruct: 32 layers.
- Qwen3-32B: 64 layers

Detailed results can be found here:

<https://github.com/NicoHillbrand/SteganographyProbes/blob/main/notes/final-analysis-report.md>

## 5. Discussion and Limitations

### Limitations

- There are some bugs in the code : we are taking the absolute value when adding IE scores per head, but we shouldn't do that (we only need heads that promote steg recognition and then we suppress it later during steering)
- Further work needs to be done to disentangle steganographic recognition signals from compliance or non-compliance of performing a covert action based on recognition of a secret message

## Future Work

- Does the Llama synonym signal replicate robustly? The v2 results will resolve whether the signal is real under proper extraction. If yes, a synonym-specific TF-IDF baseline on the v2-formatted texts should be run to confirm the delta survives.
- Is there a stego-specific circuit in Llama synonym? If P1 localizes signal to 1–3 early layers, activation patching (clean→stego) at those layers would complete the causal story.
- **Does the GCM structural fix (signed accumulation) recover interpretable head attribution?** This is a ~2h code change + rerun. If the corrected null test ratio exceeds 2x, head-level attribution becomes credible.
- What are the late-layer L58–62 Qwen3-32B suffix\_marker representations? Mean-diff finds strong signal there; mean-diff-based patching (analogous to GCM but with correct accumulation) could identify which heads contribute.
- Does the L3 probe direction in Llama encode semantically interpretable features? Even if the probe calibration is uncertain, the direction exists and steers downstream scores. Geometric analysis (similarity to known concept directions, e.g., from SAE decompositions) could be informative.
- Is Qwen3-32B's stego detection behavioral suppression a consistent circuit? The 49 suffix\_marker compliant examples are concentrated in one task template. Patching clean→stego at L49 in that task template and measuring behavioral change would test whether the representation-to-behavior link exists and is task-specific.

## 6. Conclusion

This research demonstrates that while Large Language Models can be used to facilitate covert communication via steganography, the receiver models are not passive recipients of these signals. Through the lens of Mechanistic Interpretability, we have shown that the detection of steganographic payloads is a linearly separable feature within the model's latent space, specifically appearing in the later stages of the transformer's residual stream.

Our experiments using Causal Mediation and Steering Heads further reveal that this recognition is not a global atmospheric shift in the model's processing, but is instead localized to specific functional circuits. By identifying these "detector heads," we have successfully demonstrated that it is possible to surgically steer a model's sensitivity to hidden messages—either by amplifying its

"awareness" for safety monitoring or by ablating the feature to neutralize unauthorized coordination.

Ultimately, this work provides a technical foundation for Latent Safety Auditing. As AI-to-AI interaction becomes a standard paradigm, monitoring the internal activations of recipient models offers a more robust defense against steganographic alignment failures than traditional token-level analysis. Future work should focus on the generalizability of these "stego-probes" across different model architectures and increasingly sophisticated, non-linear embedding techniques.

## Code and Data

- **Code repository:** <https://github.com/NicoHillbrand/SteganographyProbes>
- **Data/Datasets:** <https://github.com/NicoHillbrand/SteganographyProbes/tree/main/data>

## References

1. Redwood Text Steganography Benchmark  
<https://github.com/redwoodresearch/Text-Steganography-Benchmark/tree/main>
2. Hide and Seek in Embedding Space: Geometry-based Steganography and Detection in Large Language Models <https://arxiv.org/abs/2601.22818v1>
3. TrojanStego <https://arxiv.org/abs/2505.20118v4>
4. TrojanStego collection <https://huggingface.co/collections/worta/trojanstego>
5. Surgical Activation Steering via Generative Causal Mediation  
<https://arxiv.org/abs/2602.16080>
6. Refusal in Language Models Is Mediated by a Single Direction  
<https://arxiv.org/abs/2406.11717>

## LLM Usage Statement

We used coding agents (Claude, Codex) to aid us with the code and data analysis. We use Gemini to polish the writing in this report.