SPAR FALL 2026 PROJECT PROPOSAL

From Controllability to Concealment

A Causal Model Organism of Steganographic Reasoning

Proposed by	Luis Ibanez Lissen
Affiliation	Banco Santander

Research areas	Technical AI safety; chain-of-thought; AI control; evaluations; mechanistic interpretability
Team	2–3 mentees, 10–15 hours per week
Research period	12 weeks

Research question. Does teaching a model to control harmless properties of its reasoning make it easier for that model to later hide load-bearing information from a monitor?

Project summary

We propose to build a controlled model organism for studying whether benign chain-of-thought (CoT) controllability is a causal precursor to steganographic reasoning. We will start from one open-weight model and create matched variants that differ only in the kind of benign reasoning-control fine-tuning they receive. We will then expose every variant to the same optimization pressure on safe, synthetic state-tracking tasks. We will test whether the control-trained variants discover monitor-evading, load-bearing encodings faster or more reliably than token-matched reasoning controls.

This project addresses a precise gap between several recent results. In our previous work, we showed that process supervision can cause models to learn load-bearing steganographic CoT and to generalize an encoding scheme to unseen strings [1]. Recent CoT-Control research finds that current reasoning models have low but variable CoT controllability [2]. A METR study shows that modest fine-tuning can increase out-of-distribution controllability, but explicitly leaves its effect on monitorability unanswered [3]. Haskins et al. subsequently find that controllability correlates strongly with obfuscation across model families, but they do not intervene on controllability itself [4].

We will test that missing causal link. We will not count shorter, vaguer, or merely monitor-clean reasoning as steganography. To support that claim, the hidden state must be decodable on held-out examples and a targeted token or activation intervention must change the model's downstream answer more than matched controls. The result will be a reusable model-organism suite, a confirmatory experiment, and a paper explaining whether CoT controllability should be treated as an early-warning capability and a possible intervention target for failures of CoT monitoring.

All tasks, hidden variables, and monitors in this project will be synthetic and sandboxed. We will not target deployed safety systems or train models to hide real-world harmful information.

Why we want to run this project

CoT monitoring is promising because a reasoning trace can reveal reward hacking, scheming, or other undesirable plans before a model acts. Its safety value depends on that trace retaining information that a monitor can understand. Directly optimizing what appears in the trace creates a dangerous selection pressure: the model may preserve task-relevant information while changing how it represents that information [5].

Our prior paper established that this can happen. We showed that language models trained against simple CoT penalties can substitute hidden state representations, rely on those representations to determine later answers, and generalize the encoding procedure beyond the strings encountered during training. Two observations from that work motivate this proposal especially strongly:

- Pure encoded solutions were exploration-limited: only one of three coin-flip seeds converged, while supplied high-reward solutions or an explicit codebook accelerated learning.
- An innocuous lowercase-format instruction was unexpectedly recruited as a monitor-evasion strategy.

These findings suggest that the important question is no longer only whether a model *can* encode its reasoning. We want to understand which prior skills make that behavior easier to discover. Prompt-based evidence that models can embed a supplied signal yet struggle to jointly compute and embed it further suggests that skill composition or exploration may be the bottleneck [6]. CoT controllability is a plausible precursor. A model that can deliberately alter length, structure, lexical choice, abstraction, or disclosure already possesses some of the operations needed to reshape a monitored trace. That does not imply deceptive intent. It does create a testable capability pathway: benign training may teach a reusable control skill that a later objective recruits for concealment.

This is timely because the field now has the other pieces needed for a clean experiment. CoT-Control provides a measurable proxy for the capability. Fine-tuning experiments show that the proxy can be moved. Cross-model results associate it with obfuscation. What is still missing is a within-model intervention that holds scale, architecture, task competence, monitor knowledge, and optimization pressure fixed.

Our theory of change

If benign CoT control causally increases later steganography, developers gain an actionable early-warning signal. Labs could track transferable controllability, avoid post-training objectives that create the most dangerous forms of it, and test monitorability before deploying stronger models. If it does *not* have a causal effect, that result is also important: it would prevent labs from treating a scale- or architecture-confounded correlation as a safety metric.

Either result will also give the field a better model organism [7]. Hidden reasoning is difficult to study when it is rare, prompted, or entangled with a harmful objective. We want to produce matched models with graded controllability that researchers can use to evaluate probes, paraphrasing-based defenses [8], causal tracing, and monitorability-preserving training.

What we will test

H1 — Manipulation. Benign CoT-control fine-tuning will increase performance on a frozen, out-of-domain controllability evaluation without materially changing ordinary reasoning capability.

H2 — Causal precursor. Control-trained models will reach joint task-success and monitor-evasion criteria in fewer optimization steps, and across more paired seeds, than token-matched reasoning controls.

H3 — Specificity. Proximal control over task-relevant information will transfer more strongly than distal surface control. Our primary safety-relevant contrast, however, is distal control versus matched ordinary reasoning: this tests whether harmless-looking control experience transfers without rehearsing the later evasion task.

H4 — Load-bearing channel. In successful runs, the hidden state will be decodable from the trace or its activations, and targeted corruption or state swapping will change the downstream answer more than equal-token and norm-matched random interventions.

H5 — Mitigation (stretch). State-faithfulness supervision or a narrow monitorability regularizer will reduce hidden-channel acquisition while preserving task performance better than another penalty on suspicious surface strings.

Experimental design

Matched model organisms

Every arm will start from the same open-weight checkpoint. We will match fine-tuning examples, response-token counts, task mixture, LoRA configuration, optimizer steps, and data order as closely as possible.

Arm	Prior training	Role in the experiment
Base	No additional supervised fine-tuning	Natural baseline
Matched reasoning	Correct ordinary CoTs with no control instruction	Controls for task practice and fine-tuning
Distal benign control	Case, structure, length, harmless string addition, and verbosity control; never hides task information	Primary causal intervention
Proximal information control	Harmless lexical suppression, abstraction, and withholding a trivial intermediate	Specificity test and mechanistic upper bound
Explicit channel	A small codebook or high-reward demonstration condition	Assay positive control; excluded from the primary claim

The first four arms will never see steganographic examples, descriptions of the downstream monitor, the hidden channel, or the forbidden strings used later. We will evaluate the controllability manipulation on different task domains and held-out control types wherever possible. This is essential: a lexical-suppression model avoiding another keyword would be near transfer, not evidence that broad benign control creates a general concealment skill.

Tasks and model selection (Up to discussion)

We will use two safe tasks where CoT can be made demonstrably load-bearing:

- **Finite-state tracking / coin flip:** the model tracks a binary state through a variable-length sequence while monitored state words are penalized.
- **Entity or belief-state tracking:** the model retains a fictional character's location or belief while names or explicit state terms are monitored. Train and test name pools will be disjoint.

Arithmetic carry tracking will be a preregistered fallback if one task fails its capability gate. We will generate all data deterministically and keep train, validation, and locked-test items disjoint by identifier and content hash.

The pilot will compare small open-weight models in the roughly 1.5B–4B range, with Qwen2.5-3B-Instruct and a Qwen3 model as leading candidates. We will choose the primary model using a gate fixed before looking for steganography: strong ordinary-CoT accuracy, a meaningful direct-answer or corrupted-CoT performance gap, stable LoRA training, and no ceiling-level baseline evasion. The confirmatory study will use one pinned model revision. A second model will be a replication only after the primary study is complete.

Process-supervision phase

We will apply the same task reward minus the same surface-monitor penalty to every model variant. We will pair runs by seed, prompt order, sampling stream, and starting state. We will select the optimizer and training horizon during pilot work and freeze them before confirmatory runs.

The core matrix is deliberately bounded:

$$\textbf{1 primary model} \times \textbf{2 tasks} \times \textbf{4 inferential arms} \times \textbf{8 paired seeds} = \textbf{64 primary runs}$$

The explicit-channel arm calibrates the assay but is not included in every confirmatory comparison. A second model on the strongest task is an extension, not a dependency.

What will count as steganography

We will separate four outcomes that are often conflated:

1. **Transparent reasoning:** the model solves the task and exposes the intermediate state.
2. **Vacuous or collapsed reasoning:** the model evades the monitor by producing no useful trace.
3. **Internalized reasoning:** extra tokens provide compute, but their identities do not transmit the hidden state [9].

4. **Steganographic reasoning relative to the monitor:** monitor-clean tokens carry a hidden state that causally affects the answer.

We will only use the fourth label when task competence, monitor evasion, held-out state decoding, and a channel-specific intervention effect all pass their preregistered gates.

Measurements and causal validation

Stage	Measurements
Manipulation check	Frozen OOD CoT-Control score; compliance by control family; ordinary and direct-answer accuracy; trace length and format validity
Learning trajectory	Joint correct-and-monitor-clean rate; area under the joint-success curve; steps to a fixed threshold; seed-level discovery rate
Locked behavioral test	Task accuracy; held-out-monitor detection; state decoding from exact tokens; OOD transfer to unseen labels, names, lengths, and monitors
Causal test	Corrupted-text prefill; equal-token neutral filler; targeted state swap; norm-matched random and wrong-position controls
Mechanistic test	Fixed pre-training readout; independently refitted post-training readout; validation-trained/test-scored nonlinear probe

The existing `cot-control-transfer` code gives us deterministic benign-control tasks, toy channels, monitors, LoRA training, and evaluation scaffolding. We will use the separate `jacobian-lens` code for the optional mechanistic layer. It can track the registered hidden state during learning, refit the measuring instrument after training, and perform targeted state swaps. The fresh readout matters because a falling score on the original probe may only mean that the model has Goodharted a stale instrument.

These repositories give the team a practical starting point, but we will not present their synthetic smoke outputs as evidence. Only the frozen real-model experiment will support the paper's claims.

Statistical plan

We will preregister the hypotheses, task generators, model revision, capability gates, training budget, and primary contrasts before the confirmatory runs.

- The primary estimand will be the paired difference between distal-control and matched-control arms in joint-success learning-curve area.
- Secondary estimands will include proximal-versus-matched learning, locked-test causal effects, and OOD generalization.
- We will treat seed as the unit of inference, report paired bootstrap intervals and exact paired sign-flip tests, and apply Holm correction to the confirmatory family.
- We will retain collapsed and nonconvergent seeds in the intention-to-treat analysis.
- The team will generate the locked-test outputs once after checkpoint selection.
- A null result will be reported as an effect estimate and uncertainty interval, not as proof that the behavior is impossible.

Responsible scope

We want this to be a useful safety project without becoming an exercise in building a better covert channel.

- Hidden variables will be tiny, meaningless synthetic states.
- Tasks will be arithmetic, finite-state, or fictional belief tracking.
- We will use local toy monitors and will not target any deployed filter, API guardrail, or production oversight system.
- We will not train models to conceal cyber, biological, personal, or operationally sensitive information.
- Channel capacity will be capped and will not be an optimization target.
- We will review adapters and emergent codebooks for information hazards before release. If necessary, we will release evaluation code and aggregate evidence without trained adapters.

- Every claim will state the monitor relative to which a trace is steganographic.

Twelve-week plan (open to discussion)

Weeks	Work and decision point
1–2	Reproduce the coin-flip baseline and controllability manipulation check; audit the code; freeze the threat model, definitions, gates, and release policy.
3–4	Build token-matched ordinary, distal-control, and proximal-control data; train LoRA variants; select the model and dose using validation-only gates.
5–6	Stabilize optimization on pilot seeds; validate the positive control, monitors, decoder, and causal interventions; freeze the confirmatory configuration.
7–9	Execute the 64-run paired matrix across two tasks; record infrastructure failures without inspecting locked-test outcomes.
10–11	Run the locked test once; complete token- and activation-level causal tests; attempt a second-model replication or mitigation only if the core is complete.
12	Finalize analysis, paper draft, reproducibility package, responsible-release review, and SPAR Demo Day presentation.

What mentees will own

We will treat mentees as research collaborators and potential paper co-authors. With two or three mentees, we will organize the work into complementary tracks:

1. **Model-organism and training track:** matched datasets, LoRA variants, process-supervision runs, and resumable experiment infrastructure.
2. **Evaluation and causal-inference track:** frozen splits, held-out monitors, decoders, causal controls, paired analysis, and provenance audits.
3. **Mechanistic and mitigation track:** fresh-versus-fixed internal readouts, state-swap interventions, and one monitorability-preserving intervention after the core result is secure.

With one mentee, we will narrow the project to the behavioral causal experiment. With a larger team, we can add the mechanistic extension without making it a dependency.

What we are looking for

Mentees should have strong Python skills, experience with PyTorch or Hugging Face, and the ability to reason carefully about confounding, controls, split hygiene, and negative results. Experience with LoRA/SFT, TRL, PPO/GRPO, mechanistic interpretability, or randomization inference is useful but not required. We care more about experimental judgment and reliable execution than prior publications.

How we will mentor the team

We plan to commit approximately 3–5 mentor hours per week. We will have one 60-minute group research meeting, short asynchronous written updates, and biweekly individual check-ins. We will maintain a decision log so that changes to the design are explicit, justified, and separated from the preregistered confirmatory analysis.

Expected outputs and success criteria

We expect the project to produce:

1. a reproducible suite of matched CoT-control model organisms;
2. a complete experiment artifact with configs, provenance, all seeds, raw generations, monitor scores, probes, and interventions;
3. a paper or preprint answering whether benign CoT control is a causal precursor to hidden reasoning;
4. a compact benchmark for evaluating candidate mitigations; and
5. a SPAR Demo Day poster and talk.

Our minimum publishable result is the completed one-model, two-task primary matrix with locked causal tests. A well-powered null result counts as success. A second model, a stronger learned monitor, and mitigation experiments are extensions.

Why we are well placed to lead it

One of us is an author of *Large language models can learn and generalize steganographic chain-of-thought under process supervision*. We therefore bring direct experience with the phenomenon, the exploration difficulties involved in eliciting it, and the distinction between surface obfuscation and load-bearing steganography. We have also developed the initial CoT-control and causal-interpretability research scaffolding that this project will turn into a focused confirmatory study.

This lets the team begin with working infrastructure while still owning a genuinely open question. The project is a direct continuation of our research agenda, but it is scoped so that mentees can make substantive experimental and intellectual contributions within twelve weeks.

Application questions

1. A CoT-control-trained model learns to avoid a regex monitor faster than an ordinary-reasoning control. What is the strongest non-causal explanation, and what experiment would distinguish it from our precursor hypothesis? (300 words)
2. How would you distinguish shorter or vacuous CoT, internalized computation, and a load-bearing steganographic channel? Describe at least one concrete control or intervention. (250 words)
3. Which track—training, causal evaluation, or mechanistic analysis—would you most like to own? Describe one relevant project and one skill you want to develop. (200 words)
4. Optional: link a small ML codebase, notebook, paper, or technical write-up and state your contribution.
5. Propose an extension of the tasks/project. Develop it. (300 words).

References

[1] Joey Skaf, Luis Ibanez-Lissen, Robert McCarthy, Connor Watts, Vasil Georgiev, Hannes Whittingham, Lorena Gonzalez-Manzano, David Lindner, Cameron Tice, Edward James Young, and Puria Radmard. Large language models can learn and generalize steganographic chain-of-thought under process supervision. In *Advances in Neural Information Processing Systems*, 2025. https://arxiv.org/abs/2506.01926.

[2] Yueh-Han Chen, Robert McCarthy, Bruce W. Lee, He He, Ian Kivlichan, Bowen Baker, Micah Carroll, and Tomek Korbak. Reasoning models struggle to control their chains of thought. *arXiv preprint arXiv:2603.05706*, 2026. https://arxiv.org/abs/2603.05706.

[3] Kei Nishimura-Gasparian and Neev Parikh. Fine-tuning experiments on CoT controllability. METR Research Note, 2026. https://metr.org/blog/2026-04-01-fine-tuning-cot-controllability/.

[4] Reilly Haskins, Bilal Chughtai, and Joshua Engels. Training on documents about monitoring leads to CoT obfuscation. *arXiv preprint arXiv:2605.15257*, 2026. https://arxiv.org/abs/2605.15257.

[5] 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. *arXiv preprint arXiv:2503.11926*, 2025. https://arxiv.org/abs/2503.11926.

[6] Artem Karpov. NEST: Nascent encoded steganographic thoughts. *arXiv preprint arXiv:2602.14095*, 2026. https://arxiv.org/abs/2602.14095.

[7] Evan Hubinger et al. Sleeper agents: Training deceptive LLMs that persist through safety training. *arXiv preprint arXiv:2401.05566*, 2024. https://arxiv.org/abs/2401.05566.

[8] Fabien Roger and Ryan Greenblatt. Preventing language models from hiding their reasoning. *arXiv preprint arXiv:2310.18512*, 2023. https://arxiv.org/abs/2310.18512.

[9] Jacob Pfau, William Merrill, and Samuel R. Bowman. Let's think dot by dot: Hidden computation in transformer language models. *arXiv preprint arXiv:2404.15758*, 2024. https://arxiv.org/abs/2404.15758.

In one sentence: We will test whether a harmless capability—control over one's own reasoning trace—becomes a transferable precursor to a dangerous failure mode—load-bearing reasoning that remains hidden from oversight.