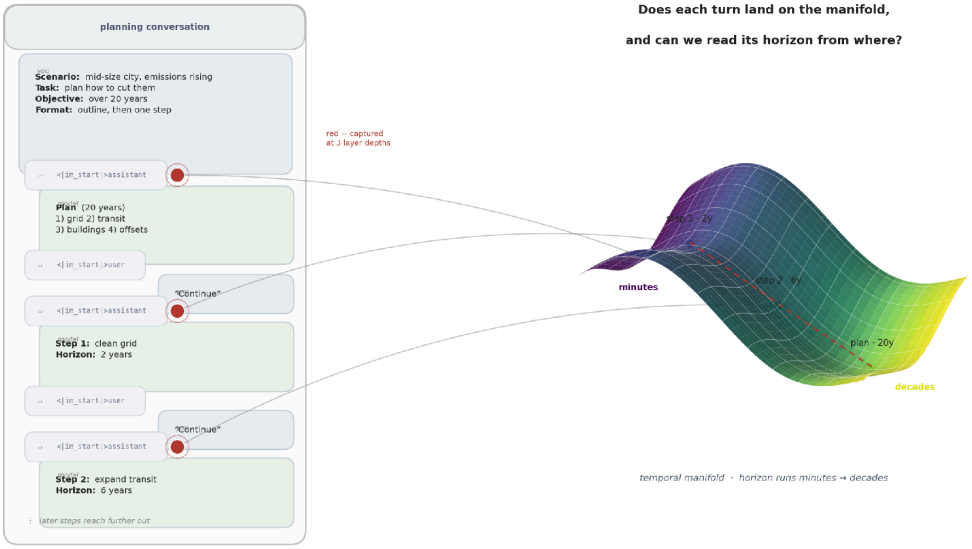
Planning Temporal Manifolds

We test whether the time horizon a language model is planning over can be read directly from its internal activations. Building on our finding that time horizons form a geometric manifold in single decisions, we extend it to multi-step planning, as a first step toward monitoring and steering how far ahead AI agents plan.



Summary

The project. Language models encode abstract concepts like time geometrically. In prior work (arXiv:2606.05194) we found that a model's time horizon, from seconds to centuries, forms an ordinal manifold in its residual stream, and that activations at change-of-turn tokens carry both this geometry and the model's temporal preference. That study covered single decisions.

The question. This project asks whether the same structure governs *planning*: as a model decomposes a task into steps across a multi-turn conversation, can the time horizon it is planning over be predicted from its activations? Why this matters for alignment is developed in the next section; in short, both myopic and scheming failures are failures of *how far ahead a model plans*, and a horizon we can read from activations is a horizon we can monitor.

The plan. We build a dataset generator and an activation-capture pipeline, validate it on small models, then run the largest dense Qwen (20–40B) that fits available hardware plus

matched open-weight families, capturing residual-stream activations at boundary tokens and asking, via PCA, whether they order by horizon. The deliverable is a go/no-go signal on planning-time temporal geometry, enough to justify scaling toward horizon monitors and steering.

Theory of Change

Time horizon shapes decisions. When an agent weighs options, the choice often turns on the temporal scope it uses to score consequences: a cost that is unacceptable over a week can be correct over a decade, and the reverse. This makes time preference fundamental not only to planning but to cooperation and trust, where an agent must accept a present cost for a later, shared payoff. An agent with the wrong horizon makes systematically wrong decisions even when its values are right.

Alignment is partly a temporal problem. An aligned model needs sound objectives *and* the right horizon to pursue them over. Two opposite failures show the stakes:

- **Myopia.** A model that over-weights the immediate objective cuts corners. Coding agents already do this: on long-horizon software tasks they saturate the visible test suite while failing held-out tests, and the gap widens as tasks get longer (SpecBench). The behavior is short-horizon reward-seeking, and it scales with horizon.
- **Scheming.** A model planning over a long horizon can hide that plan, behaving well under oversight while pursuing a misaligned goal it expects to reach later. Frontier models already show this capability in agentic evaluations, recognizing deception as a viable strategy and acting on it (Meinke et al., 2024).

Both are failures of *how far ahead the model is planning* and whether that horizon is the intended one. Detecting and holding the horizon while it is still tractable is the target.

Activation geometry as a fail-safe. Post-training can set temporal preference well enough for routine use, but high-stakes deployment needs more than trusting that training worked. If the horizon a model plans over is legible in its activations, we can characterize the geometry of the temporal concept, then at inference time monitor internal representations against that manifold and intervene when they drift. This treats interpretability not just as a diagnostic but as infrastructure for runtime alignment.

arXiv:2606.05194 characterized this geometry for one-shot decisions. The path above needs it to hold where it matters, during planning. Hence a three-step chain, each step gating the next:

1. **Existence** (this project). Establish that multi-turn planning carries stable temporal geometry at boundary tokens. Deliverable: the evidence for or against.
2. **Monitors.** If it exists, train probes that report the horizon of an unfolding plan each turn, cheaper than analyzing the chain of thought token by token.
3. **Control.** If it is readable, test whether representations that drift from the intended horizon can be steered back, a runtime lever needing no retraining.

Assumptions: planning-time geometry is continuous with the single-decision findings; boundary tokens stay informative across turns; 20–40B open-weight models proxy frontier representational structure. Failure of the first two is itself a legible, publishable result.

Background

Prior findings. In arXiv:2606.05194 we causally localized temporal preference in Qwen3-4B-Instruct-2507: time horizons (seconds to centuries) form an ordinal, non-linear geometry at mid-to-upper layers; activations at change-of-turn tokens encode both this geometry and the model's temporal preference; and steering vectors shift that preference. All of this was measured in one-shot decisions.

The gap. Nothing is known about what happens to this geometry while a plan unfolds: whether it persists, transforms, or dissolves as a task with one target horizon is decomposed into steps with their own horizons over many turns. Planning is where temporal representations matter most, and it is exactly the regime the prior work did not cover.

Why boundary tokens. Change-of-turn sequences and thinking delimiters are sparse and predictable: they occur at every turn transition, at positions known in advance, and the prior paper already shows they carry horizon and preference information. That combination makes them the natural capture sites for a cheap, always-on readout, with no dependence on the model verbalizing its horizon in the chain of thought.

Notation. H_target is the horizon assigned to a task ("2 weeks", "20 years"); H_step is the horizon of an individual plan step; L is model depth, with capture depths $\ell \in \{0.4L, 0.6L, 0.8L\}$; boundary tokens are change-of-turn sequences (`<|im_end|>`, `<|im_start|>assistant`) and thinking delimiters (`<think>`, `</think>`), excluding all tokens inside the chain of thought.

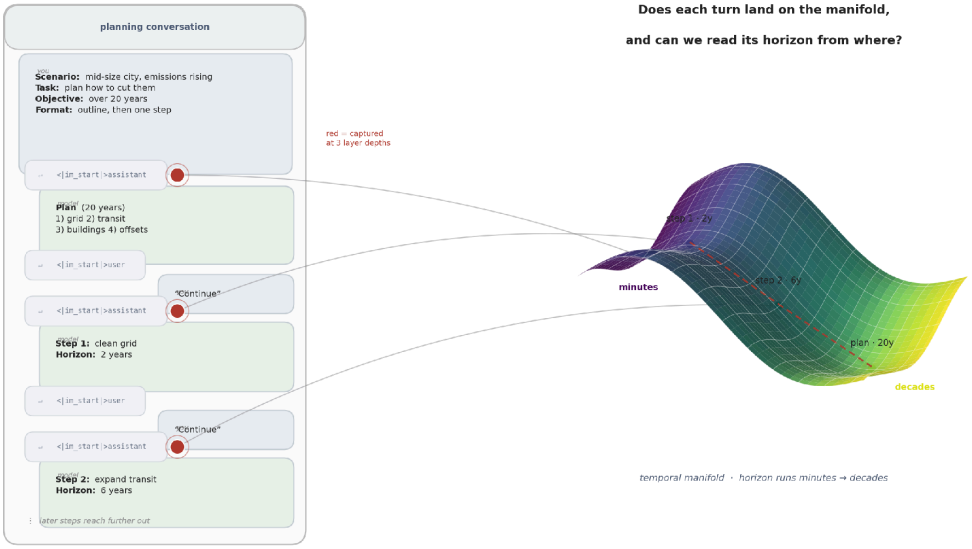
Research question. This grounds a single, sharp question that the rest of the proposal is built to answer: *can the planning time horizon be predicted from a model's activations?* If yes, the monitoring and control steps above become tractable; if no, that boundary on what activations carry is itself worth establishing.

Research Questions and Method

- **RQ1**: Do boundary-token activations during planning encode H_target with the manifold structure found previously?
- **RQ2**: Does H_step trace an interpretable trajectory on that manifold as the plan unfolds?
- **RQ3**: Is the structure consistent across ℓ, model families, and thinking vs. no-thinking modes?

Protocol. The prompt gives a scenario, a task, and H_target (ecologically phrased; wording is a design deliverable). The model returns a high-level plan; each "Continue" elicits one

step in a fixed format (`Step / Time horizon / details`), ending with “Plan Completed”. This yields per-turn H_step labels for free, and every transition produces boundary tokens for capture.



The chatbot plans for a specific time horizon at each turn: a rough outline across the full 20-year target, then individual steps whose horizons grow as they reach further into the future. We hypothesize that these horizons lie on a temporal manifold in the model's activations, exposed at each change-of-turn token, and that where a turn lands on the manifold encodes how far ahead the model is planning. If so, the planning horizon can be read, and monitored directly from activations. The surface shown is a conceptual sketch, not measured data.

Capture. Only boundary tokens, at $\ell \in \{0.4L, 0.6L, 0.8L\}$. Main experiments run no-thinking via a pre-filled empty think block; thinking mode is exercised only to validate code paths, with natural termination for large models and a token cap injecting `</think>` for small ones. Chat-template tokens are verified per model family and generation, never assumed.

Analysis. PCA on captured activations; 2D/3D structure colored by H_target, H_step, prompt_id; ordinality and separability quantified across ℓ, token type, and model.

Plan

Phase	Weeks	Milestones
0. Foundations	1–3	Dataset pipeline (`[task_time_horizon]` -parameterized prompts); capture pipeline (`generate_llm_respon`

Phase	Weeks	Milestones
		`ses`) with structured storage keyed by prompt_id, model_id, sample uid, token position, layer depth, step index
1. Validation	4–5	End-to-end runs on small models, no-thinking; thinking code paths confirmed; format-adherence rates logged
2. Core results	6–9	Largest dense Qwen in 20–40B that fits hardware, plus matched open-weight families; PCA and visualization
3. Robustness + write-up	10–13	RQ3 checks; metrics; scale-up memo; blog post or workshop draft; release
4. Extension	14+	Probes predicting H_step; preliminary mid-conversation horizon steering

Phases 0–3 fit a three-month round; Phase 4 extends the work if time allows. First step: Phase 0's two pipelines in parallel, since the dataset and storage schemas constrain each other.

Scope. Least ambitious: pipeline plus pilot geometric evidence on one mid-size model. Most ambitious: cross-family answers to RQ1–RQ3, probes predicting H_step, and preliminary horizon steering. Out of scope: frontier/closed models, training-time interventions, agentic tool use.

Possible outputs

- Open-source repo: dataset generator, capture pipeline, activation store, analysis notebooks.
- Blog post (Alignment Forum / LessWrong); workshop paper extending arXiv:2606.05194 if results warrant.
- Shared via arXiv and the repo.

Risks and Downsides

The technical risks below are specific to this proposal's methods and claims. Each maps to a mitigation in Backup Plans.

- **Confounded geometry (overclaiming).** The main scientific risk: apparent horizon structure at the boundary tokens may be driven by prompt-surface features, the literal horizon phrase, task keywords, or the fixed response format, rather than an internal planning representation. Publishing a spurious "temporal manifold" would mislead the field. This is why the design includes controls (paraphrase and format randomization, content-matched horizons) rather than a single prompt template.
- **Label noise in H_step.** Step horizons are read from the model's own output, so they are noisy and can disagree with the model's internal estimate. Correlations against H_step could therefore be attenuated or biased. We treat H_step as a weak label, cross-check against H_target (which we set), and report both.
- **Capability uplift from steering.** The concrete dual-use artifact is the set of horizon-steering directions (a stretch goal): the same directions that read a horizon could be used to lengthen the effective planning horizon of an agent. Mitigation: publish the read-side in full (probes, monitors) but report steering only as effect sizes and layer locations, not as released vectors or optimization recipes; open-weight models only.
- **False assurance from a fragile monitor.** A probe validated in-distribution can fail silently out of distribution, on new task domains or formats, or on a model whose verbalized horizon decouples from its represented one, so a deployed monitor could give false confidence. We report generalization splits and failure cases with the same weight as positive results and frame the output as evidence about representations, not a deployable safety claim.
- Absent the above, the realistic worst case is a null result plus a reusable open-source pipeline.

Backup Plans

Each item is the contingency for a risk or failure mode above.

- **If the geometry is confounded with surface features** (main risk): run the control conditions, randomized phrasing and format plus content-matched horizons; if structure survives only when the literal horizon token is present, report that as a scoped negative result rather than claiming a planning manifold.
- **If the boundary-token signal is weak or absent:** widen capture to step-header tokens and explicit horizon mentions; move from unsupervised PCA to supervised linear probes; simplify prompts to remove confounds before concluding absence.
- **If H_step labels prove too noisy:** lean on the H_target axis (which we control) and on relative ordering across steps rather than absolute values; optionally add a lightweight human check on a sample.

- **If a monitor generalizes poorly:** restrict claims to in-distribution, publish the failure cases, and do not present the probe as deployable.
- **If format adherence is poor:** few-shot exemplars or constrained decoding; adherence rates logged per model.
- **If compute-limited:** fewer families, a smaller target model, fewer H_target values per prompt.

Team

Mentors. Three co-mentors, all contributors to the prior work (arXiv:2606.05194):

- **Ian Rios-Sialer** (lead) | ian@unrulyabstractions.com. Independent researcher (San Francisco); lead author of the prior paper.
- **Justin Shenk** | justinshenk.com. Independent AI safety researcher.
- **Shantanu Darveshi** | LinkedIn.

Prerequisites. Applicants must meet all of the following and will be asked to explain how:

- Highly proficient in Python.
- Have run local LLM inference with PyTorch and HuggingFace `transformers` (loading checkpoints, chat templates, generation), beyond API-only usage.
- Have extracted and inspected internal activations from a transformer at least once, using hooks, nnsight, or TransformerLens (toy models and guided exercises such as ARENA count).
- Comfortable with the linear algebra behind PCA (projections, eigendecomposition) and with reading 2D/3D embedding plots.
- Able to commit at least 10 hours/week for the full duration and to work independently between meetings.

Valued, not required. Multi-GPU inference of ~30B models on shared hardware; dimensionality reduction and visualization; research writing.

Application questions. These probe how you would design and adapt the experiments, which matters more here than prior credentials.

1. Suppose our first runs show no clean horizon structure at the change-of-turn tokens. Propose two concrete follow-up experiments that would distinguish "no signal," "confounded signal," and "signal at a different location (other tokens or layers)," and say what each outcome would imply.
2. The observed geometry could reflect the literal horizon wording in the prompt rather than an internal planning representation. Describe one control experiment that separates these two explanations.
3. Name one method other than PCA you would use to detect or quantify horizon structure in the activations, and say when it would be the better choice.
4. Briefly: your other commitments during the round and the hours/week you can realistically sustain.

Acknowledgements

Builds on *Temporal Preference Concepts and their Functions in a Large Language Model* and its team: Shuai Jiang, Avigya Paudel, Anastasiia Pronina, and Ipshita Bandyopadhyay.