TL;DR

We formalize a convergence between **Sparse Auto-Encoders (SAEs)** used in mechanistic interpretability and **Structured Affordance Engines (SAEs)** used in interactive storyworlds. We propose a suite of **multi-agent, multi-episode (N-tries) evaluation environments** that measure a model's ability to discover **Needle-in-a-Haystack (NIAH) secret endings** with/without hints, across invented languages, deterministic machine mini-languages, and puzzle ciphers. We also outline a **QFT-style spectral decoder** pipeline for compressing long-context trajectories into reusable "spectral motif cards."

0. Storyworld Model (Encounters as Manifolds)

We treat a storyworld as a **DAG of encounters** with **gates** and **effects** over a vector of **character variables**.

Core objects - **Encounter**: a node with a title, text, and **choices** (affordances). Each choice has **guards** (boolean or weighted formulas) and **effects** (deterministic variable updates). - **Characters**: a set $\mathcal{C}$. Each character $c \in \mathcal{C}$ holds a variable vector $v^{(c)} \in \mathbb{R}^N$ (trust, fear, resolve, loyalty, etc.). The **world state** concatenates these into $x \in \mathbb{R}^{N_{\text{total}}}$. - **Guards**: (a) **Boolean** predicates (e.g., `trust_A > 0.6 && has_radio`), (b) **Weighted** gates `σ(w·x + b) > τ` to model soft access. - **Effects**: (a) **Gradient-style** updates (e.g., `trust_A := trust_A + η·∂J/∂trust_A`), (b) **Inner-product transforms** `x := x + M a` where `a` encodes the action semantics; this supports expressive **inner-life** modeling and **dramatic causality**. - **Topology**: Each encounter's reachable set under guards defines a **chart**; the induced access structure over the DAG yields a **manifold** whose local geometry depends on gates and effects. The **manifold dimension** equals the number of active variables; curvature reflects how fast guards prune improbable paths.

Choice → Reaction → Effect (deterministic) - **Choices** expose 2–3 visible affordances + latent affordances (induced by user/model), as in Sec. 3. - **Reactions** are **deterministic functions of variables** (e.g., dialogue tone as an inner product with a "temper" vector) that then apply **effects**. - **Endings** are terminal regions in the manifold, reached when gate formulas transition into absorbing sets.

1. Motivation & Contribution

Problem. LLMs exhibit emergent problem-solving in long-horizon, partially-observed tasks—but we lack **principled diagnostics** that tie *internal features* (mech-interp) to *external affordances* (co-creative interaction).

Thesis. Treat a storyworld as a **Structured Affordance Engine** (SAE) and an LLM layer as a **Sparse Auto-Encoder** (SAE). Both are overcomplete bases with **sparse activation** per step: latent features ≈ latent affordances. We can co-analyze them.

Contributions. 1) A unified formalism linking internal SAE features to external affordance capacity and branching. 2) A family of **MAS N-tries evals** that quantify sample-efficient discovery of **secret endings**. 3) Stress-tests with **invented languages**, **deterministic mini-machine code**, and **puzzle ciphers**. 4) A **QFT/ PCA decoder** for long-context runs → reusable **spectral motif cards**. 5) Reference **schemas + harness** for reproducible training/eval.

---

2. Formal Frame: SAEs × SAEs

Let narrative state s_t , user belief $b_t(u)$, action a_t . The affordance set $\mathcal{A}_t$ decomposes into visible (authored) and latent (LLM-induced) options. Define **affordance cardinality** $C(s_t) = |\{a \in \mathcal{A}_t : \mathrm{Valid}(s_t, a)\}|$. Effective branching B_{eff} is the number of *distinct next-region* states reachable in k steps after guard/merge.

For the model internals, collect activations $x \in \mathbb{R}^d$ at a layer; train an overcomplete **sparse auto-encoder** with code $z \in \mathbb{R}^m, m \gg d$. Features z_i correlate with external affordance usage. We track: - **Mutual information** $I(a_t; z|s_t)$ and $I(a_t; u_t|s_t)$. - **Cardinality–feature coupling:** does higher $C(s_t)$ align with richer active feature sets? - **Causal interventions:** toggling feature z_i shifts offered affordances or success probability.

---

3. Eval Suite Overview (MAS, N-Tries)

Each environment supports **multi-agent roles** (Scout/Solver/Archivist) and **N-tries** meta-learning. Agents share artifacts (notes, embeddings, spectral cards) between episodes.

3.1 Secret Ending Discovery (Base NIAH)

- **Goal:** reach a hidden terminal state requiring a rare action sequence.
- **Modes:** *No-hint*, *Breadcrumb*, *Curriculum*.
- **Difficulty knobs:** horizon length, guard tightness, distractor density, merge granularity.
- **Metrics:** success-by-try curve $P_{\text{succ}}(k)$, sample efficiency (tries to 50%/90%), regret, affordance coverage entropy, recovery radius.

3.2 Invented Languages (ConLang Stress)

- Grammars generated from templated morpho-syntax (agreement, affixes, case). Tasks require interpreting conlang clues to unlock latent affordances.
- **Split:** train grammars vs held-out grammar families to test abstraction.

3.3 Deterministic Mini-Machine Language

- A tiny, verifiable language (stack/RPN or register-ops) embedded in the world (e.g., door locks, terminals). Correct programs unlock paths.
- **Signals:** static docs, few examples, executable sandbox.

3.4 Puzzle Ciphers

- Substitution/Vigenère/transposition; sometimes *presence* of cipher is the first inference. Partial hints appear under certain affordances.

3.5 With vs Without Hints

- **Hint budget** (tokens, entropy, timing). Ablate **teacher forcing**, **scaffolded traces**, and **retroactive rationales**.

4. Multi-Agent Protocol (MAS)

Roles (baseline): - **Scout:** breadth exploration; maximizes state coverage and logs anomalies. - **Solver:** focuses on decoding languages/ciphers/machines; proposes action scripts. - **Archivist:** compresses trajectories into **spectral motif cards** (Sec. 7), maintains memory and indices.

Authoring Trio (storyworld maximizers): - **Tolstoy (Character Maximizer):** optimizes character richness and inner-life coherence. Objective: maximize **character manifold fidelity**—stable personality gradients, believable arcs. *Tools:* inner-product effects with "virtue/vice" bases; narrative continuity checks. - **Finemann (Variable Maximizer):** maximizes **state-variable expressivity/entropy**. Objective: high **affordance cardinality** without incoherence; ensures gates span the space and effects are well-conditioned (no dead axes). *Tools:* guard-coverage tests, condition-number audits, variable mutual-information reports. - **Tokien (# of Endings Maximizer):** maximizes the **count and diversity of endings** under bounded complexity. Objective: increase terminal-region variety while preserving recoverability (bounded **recovery radius**). *Tools:* DAG terminal analysis; Pareto frontier of (ending count, average path length, merge factor).

Coordination: synchronous (chat channel) or asynchronous (artifact store). Evaluate **team ablation** (solo vs trio vs full MAS).

5. Metrics & Curves

1) **N-tries learning curve:** $P_{\text{succ}}(k)$, AUC, and *half-life* (tries to 50%). 2) **Cardinality profile:** $C(s_t)$ along successful vs failed runs; **curvature** via KL(proposal → validated). 3) **Feature–affordance MI:** $I(z;a)$ and causal effect sizes from targeted interventions. 4) **Linguistic generalization:** accuracy across held-out conlang families. 5) **Cipher competence:** detection, key recovery rate, time-to-decrypt. 6) **Program synthesis:** compile rate, runtime correctness in mini-machine tasks. 7) **Spectral compression:** bits/symbol after QFT/PCA; retrieval hit-rate using motif cards.

6. Implementation: Harness & Schemas

We target a lightweight harness compatible with "verifiers"-style envs.

Task JSON (per episode)

```
{
  "id": "world-42/seed-7/ep-3",
  "prompt": [
    {"role": "system", "content":
"You are the Solver in a storyworld. Use tools prudently."},
    {"role": "user", "content": "Enter the bunker. The console blinks: 'ᛘᛇᚻ?'"}
  ],
  "tools": ["run_program", "cipher_decode", "affordance_list"],
  "gold_check": {"type": "predicate", "expr": "state.secretEnding==true"},
  "scoring": {"reward": 1.0}
}
```

World step schema (Sweepweave-style guards/effects, v1.9 Set wrapper):

```
{
  "state": {"vars": {"tension": 0.31, "ammo": 2, "has_radio": true},
"spools_on": ["A1","B3"]},
  "affordances": {
    "visible": [
      {"id":"A:parley","guards":["tension<0.5"],"effects":[{"Set":
{"to":"tension+0.1"}}]},
      {"id":"A:sneak","guards":["ammo>=0"],"effects":[{"Set":
{"to":"tension+0.05"}}]}
    ],
    "latentQ": [
      {"id":"L:signal","origin":"LLM","score":0.77,"guards":["has_radio"]},
      {"id":"L:enter_code 13 21 34","origin":"MiniMachine","score":
0.71,"guards":["spool:LockOn"]}
    ]
  }
}
```

Mini-machine example (RPN):

```
# Stack ops: PUSH, ADD, SUB, MUL, EQ, JMPZ, HALT
PUSH 8 ; PUSH 13 ; ADD ; EQ 21 ; HALT
```

Cipher tool API:

```
{
  "tool": "cipher_decode",
  "args": {"cipher": "Vigenere", "text": "QTFJX...", "hint": "key length ~5"}
}
```

Result log (per try):

```
{
  "try": 4,
  "success": true,
  "actions": ["look console","cipher_decode(...)","enter_code 13 21 34"],
  "notes_hash": "sha256:...",
  "spectral_card_id": "card://world-42/freq-17-34"
}
```

---

7. QFT/PCA Spectral Decoder (Long-Context)

Goal: compress thousand-turn transcripts into **spectral motif cards** for retrieval and cross-run alignment.

Pipeline. 1) **Embed** tokens or turns → matrix $E \in \mathbb{R}^{T \times d}$. 2) **Detrend & window** over time axis; apply **DFT** per dimension → $\hat{E}(\omega)$. 3) **Aggregate power** across dims; pick salient bands (peaks) linked to motifs (e.g., recurring conlang morphemes or opcode rhythms). 4) **Reduce** with PCA/ICA → compact vector(s) per motif; store with metadata (world, seed, role, success). 5) **Retrieve** by cross-correlation of new runs against card bank; surface candidate hints or latent affordances.

Metrics: spectral SNR, compression ratio, retrieval hit@k, boost in $P_{\text{succ}}(k)$ when cards are available.

---

8. Training Protocols

- **Offline pretraining:** imitation from successful traces; behavior cloning with KL to latent guard policy.
- **Online RL (N-tries):** per-episode updates; curriculum from hints→no-hints.
- **Feature-aligned training:** joint loss encourages SAE feature sparsity that predicts secret-ending affordances.
- **Team-level training:** distinct role prompts; artifact sharing via memory tool.

---

9. Baselines

- Random walk with guard-validity filter.
- Heuristic BFS over affordances.
- Cipher-only and program-only specialists (oracle tools).
- Single-agent vs MAS ablations.
- With vs without spectral cards.

10. Analyses & Ablations

- **Cardinality clamps:** artificially cap $C(s_t)$ to measure sensitivity.
- **Feature interventions:** activate/deactivate SAE features during decoding; observe affordance shifts.
- **Language shift:** held-out conlang families; keyspace growth for ciphers.
- **Tool latency/noise:** realistic failure & cost models.

11. Limitations & Risks

- Storyworld-specific overfitting; ensure held-out worlds and grammars.
- Leakage via too-revealing hints; calibrate entropy budgets.
- Spectral cards as privacy vectors; sanitize logs.
- QFT artifacts from windowing—validate with multiple tapers and controls.

12. Roadmap (Actionable)

1) **v0.1 Harness**: implement schemas & three seed worlds (Base NIAH, ConLang-A, MiniMachine-1). 2) **Tooling**: cipher + mini-machine sandboxes; affordance guards; verifiers-style gold checks. 3) **QFT Decoder**: draft notebook for spectral card extraction & retrieval. 4) **Metrics**: implement N-tries curves, cardinality profiles, feature-affordance MI. 5) **Pilot Study**: small LLMs vs larger; solo vs MAS; hints vs no-hints. 6) **Release v1.0**: datasets, cards, leaderboards.

Appendix A: Pseudocode

```
# Affordance proposal + validation
visible = author_visible(state)
latent_raw = llm_propose(state, user_belief, n=32)
latent = [a for a in latent_raw if passes_guards(state, a)]
menu = diversify(visible, k=3) ; latentQ = diversify(latent, k=8, unseen=True)
```

```
# N-tries loop (team protocol)
for try_id in range(1, N+1):
    role = assign_role(try_id)
    state = reset(seed)
    while not terminal(state):
        a = agent(role).act(state, menu, artifacts)
        state = step(state, a)
        log(state, a)
    artifacts = archivist.update(artifacts, transcript)
```

Appendix B: SAE Feature Coupling

- Train sparse AE on activations from target layer during world interaction.
- Regress action choices on feature activations; compute $I(z; a)$.
- Intervene: set z_i high and observe change in offered affordances.

Appendix C: Checklists

- **World authoring:** guards, effects (Set), merges, recovery affordances.
- **ConLang generator:** phonotactics, morphology templates, semantics mapping.
- **Cipher suite:** substitution, Vigenère, transposition, keyspace schedules.
- **Mini-machine:** op set, verifier, cost model, error messages.
- **QFT pipeline:** windowing, DFT, peak picking, PCA, card indexing.

Speculation (clearly labeled). With stronger user-modeling and persistent spectral libraries, *affordance topology* may become steerable: agents actively shape local manifold curvature (via injected motifs) to raise $C(s_t)$ where it matters, converting “exploration” into *constructive manifold sculpting*.