A Roadmap to Falsification of Principia Cognitia

Draft Tier-0 Falsification Protocols for the MLC–ELM Duality: Empirical Tests of Cognitive Language Decoupling in Artificial Systems

Alex Snow (Aleksey L. Snigirov)
Independent Researcher
alex2saaba@gmail.com
ORCID: 0009-0001-3713-055X

Abstract:

A central challenge in contemporary cognitive science is to explain how structured, symbol-like processes emerge from the stochastic dynamics of neural collectives. The *Principia Cognitia* (PC) framework offers a substrate-independent formalism, positing a duality between an internal **Metalanguage of Cognition (MLC)**—a high-dimensional vector space of semions, operations, and relations (`⟨S,O,R⟩`)—and an **External Language of Meaning (ELM)** used for communication. This duality is formalized in the *Theorem of Decoupling of Languages* (`TH-LANG-04`), which predicts that MLC alignment is a necessary precondition for effective communication.

This paper presents a detailed **methodological roadmap** for the rigorous falsification of this theorem, designed to bridge the gap between abstract theory and empirical validation. We provide a complete, Tier-0 experimental program, including three coordinated protocols—**MPE-1** (probing spatial MLC misalignment), **SCIT-1** (testing cognitive inertia), and **CRS-1** (examining compositional understanding). The protocols are specified with a degree of detail sufficient for full reproducibility on consumer-grade hardware, including agent architectures, training corpora, and quantitative falsification criteria. By offering this actionable blueprint, this work serves as an **open invitation to the research community** to replicate, challenge, and extend the empirical testing of the *Principia Cognitia* framework.

Preface: From Theoretical Axioms to an Actionable Research Program

The gap between a formal theoretical framework and its empirical validation is one of the most significant hurdles in cognitive science. While foundational works like *Principia Cognitia* can provide a coherent axiomatic system [cite], their ultimate scientific value is determined by their falsifiability. This paper addresses this challenge directly by presenting a detailed, actionable roadmap for the empirical falsification of a core theorem from *Principia Cognitia*—the Theorem of Decoupling of Languages (`TH-LANG-04`).

This work departs from the traditional format of an experimental paper. It is offered as a **pre-registered methodological blueprint**, designed to lower the barrier to entry for the rigorous testing of cognitive theories. The reality of modern computational cognitive science is that while the execution of a well-defined experiment on a small-scale model may take only hours, the prerequisite tasks—such as the creation of methodologically pure training corpora—can require months of iterative refinement, especially for an independent researcher.

Our own preliminary work revealed this challenge acutely. For instance, initial attempts to train an agent for the MPE-1 ("Flatland") protocol solely on narrative texts produced not a "native" thinker, but a "dull scholar" capable only of quoting its training data. This finding underscores a critical, often-overlooked aspect of such research: the careful crafting of training data to instill a coherent internal world-model (MLC) is a non-trivial scientific contribution in itself, requiring a level of art and engineering that goes far beyond simple data collection.

Recognizing these challenges, we present this work not as a final report on a completed experiment, but as an **open invitation to the research community**. We provide here a complete, Tier-0 protocol—including detailed specifications for the experimental designs, the architecture of the agents, and the quantitative metrics for success and falsification—that is reproducible on consumer-grade hardware. Our goal is to provide a robust, validated starting point, enabling other labs and researchers with deep expertise in hands-on model training to replicate, challenge, and extend this work. We posit that this open, transparent, and collaborative approach to falsification is the most efficient path toward advancing a unified science of mind.

1. Introduction

A central challenge in contemporary cognitive science is to explain how structured, symbol-like processes such as logic and reasoning emerge from the stochastic, distributed dynamics of neural collectives. The *Principia Cognitia* (PC) framework addresses this challenge by positing a strict duality between two distinct but interdependent linguistic systems. The first is the **Metalanguage of Cognition (MLC)**, a system's internal, high-dimensional vector space where cognitive dynamics unfold. The second is the **External Language of Meaning (ELM)**, the symbolic interface (e.g., text, speech) used for communication. Formally within PC, the MLC is defined as a triad `⟨S,O,R⟩` comprising **Semions (S)**, the minimal vector units of cognitive structure; **Operations (O)**, the fundamental transformations over those units; and **Relations (R)**, the learned connectivity patterns linking them (see Appendix A for formal definitions).

This duality is captured in the *Theorem of Decoupling of Languages* (`TH-LANG-04`), which predicts that when an agent's internal model (MLC) is fundamentally incompatible with a domain's latent causal structure, no amount of richness or detail in its external communication (ELM) can compensate for this misalignment. In short, successful communication is bounded by the alignment of internal representations, not the expressiveness of the external channel.

The theoretical constructs of PC, particularly the concept of a semion as a discrete unit of cognitive structure, find strong parallels and empirical grounding in recent advances in the field of mechanistic interpretability. This research aims to reverse-engineer the internal computations of neural networks, revealing how abstract concepts are represented. Foundational work by **Elhage et al. (2021)** established the transformer's **residual stream** as the central channel for information accumulation, providing an architectural locus for what PC terms the Metalanguage of Cognition (MLC). More recently, **Shai et al. (2025)** demonstrated that this residual stream contains specific, measurable **geometric structures** that correspond to the model's abstract belief states. This provides direct empirical evidence for semions as physically realized, vector-based representations of concepts. Furthermore, a growing body of work on **sparse autoencoders** has shown that these internal representations can be decomposed into discrete, monosemantic, and human-interpretable features, demonstrating that semions are not merely theoretical posits but practically extractable units of meaning **(Cunningham et al., 2023)**. These convergent findings provide a robust empirical mandate for treating the MLC and its constituent semions as observable and manipulable objects of scientific inquiry.

1.1. Conceptual Illustrations of the Hypothesis

To ground this abstract theorem, we consider three well-established cases that highlight its core predictions.

First, the historical case of Ignaz Semmelweis demonstrates the primacy of the MLC (Obenchain, 2016). Physicians who held a "miasma" model of disease (an entrenched MLC) were unable to correctly interpret decisive empirical evidence presented via the ELM (mortality statistics). A change in practice only occurred after a forced change in their internal causal model—from `miasma → illness` to `particle → transmission → illness`. This phenomenon of cognitive inertia, where a stable MLC resists contradictory ELM evidence, is operationalized in our **SCIT-1** protocol.

Second, Edwin Abbott’s novella *Flatland: A Romance of Many Dimensions* (1884) provides a conceptual model of MLC incompatibility. A two-dimensional being is incapable of understanding the concept of a third dimension, regardless of the richness of the ELM descriptions provided by a three-dimensional visitor. His 2D MLC lacks the requisite structure to ground the new information. This principle of representational incompatibility, where ELM fails to bridge a fundamental MLC gap, is directly tested in the **MPE-1** protocol.

Third, John Searle's "Chinese Room" argument (1980) questions whether purely syntactic manipulation can ever constitute semantic understanding. In PC terms, the thought experiment draws a sharp line between an agent that processes symbols according to a rulebook (operating solely in the ELM) and one that possesses genuine comprehension (requiring a dynamic and properly aligned MLC). This foundational argument is staged as a falsifiable, empirical test in our **CRS-1** protocol.

1.2. An Experimental Program for Falsification

These illustrations highlight key falsifiable predictions of `TH-LANG-04`. The present work operationalizes these predictions through three coordinated Tier-0 experimental protocols: **MPE-1** directly tests the *Flatland* problem of MLC incompatibility; **SCIT-1** provides a computational model of the cognitive inertia underlying the *Semmelweis reflex*; and **CRS-1** stages the *Chinese Room* argument as a formal test of whether ELM-only processing can substitute for MLC alignment. Each protocol is designed for full reproducibility on consumer-grade hardware and, in line with the Registered Report format, specifies clear, pre-registered falsification criteria. Together, they form a minimal yet comprehensive test suite designed to rigorously challenge the theoretical claims of the MLC-ELM duality.

---

2. Core Methodology: The Minimal Lab Set and Temporal Persistence

The experimental program described herein is grounded in two core methodological principles designed to ensure reproducibility, accessibility, and conceptual rigor. The first is the adoption of a standardized, resource-light experimental platform—the "Minimal Lab Set." The second, and more fundamental, is the implementation of "Temporal Persistence," a policy that transforms the agent from a stateless predictor into a continuous, evolving system.

2.1 General Design Principles

All experimental protocols described herein—MPE-1, SCIT-1, and CRS-1—are designed as **Tier-0** protocols. We define a Tier-0 protocol as an experiment designed for maximum accessibility and rapid falsification, adhering to three core principles:

1. **Minimal Resource Requirements:** The protocol must be reproducible on a single, consumer-grade workstation, as specified in the reference hardware configuration. This ensures broad accessibility and independent verification.
2. **Conceptual Minimality:** The experiment isolates a single, core hypothesis within a highly controlled, often synthetic or simplified, environment to minimize confounding variables.
3. **Adversarial Design:** The primary goal is not to confirm the theory but to rigorously and efficiently seek conditions under which its core predictions fail.

This approach is distinct from larger-scale **Tier-1** (validation on frontier models and natural language subsets) and **Tier-2** (replication in biological or embodied systems) research. In the context of this Registered Report, the successful execution of a Tier-0 protocol where the hypothesis is not falsified is not interpreted as proof of the theory.

Instead, it is considered a validation of the methodology itself and a necessary prerequisite for justifying the significant resource investment required for higher-tier investigations.

2.2. The "Minimal Lab Set": A Standard for Reproducible Cognitive Science

To move the study of cognitive phenomena in artificial agents from large-scale engineering to tractable, laboratory-style science, we propose a standardized, accessible research environment. This "Minimal Lab Set" is a blueprint for a self-contained platform capable of supporting the entire experimental lifecycle, from data generation and model training to causal intervention and analysis. It is designed to be implementable on a single, consumer-grade workstation, thereby maximizing reproducibility and lowering the barrier to entry for independent verification.

The set comprises three core components:

- **Hardware Configuration:** A single workstation equipped with one or two high-VRAM GPUs (e.g., 8-48 GB VRAM), 32-128 GB of system RAM, and fast NVMe storage for active computation. This primary system is supplemented by a high-capacity archival storage solution (e.g., external HDD or cloud object storage) for preserving experimental artifacts. This configuration is sufficient to train the small-scale models for all Tier-0 protocols and to archive their complete state histories—including model checkpoints, optimizer states, and full activation traces—estimated to require 2-10 TB of storage per full experimental run.
- **Core Model Architecture:** The experiments are designed around `nanoGPT`, a minimal, hack-friendly GPT implementation (<1 kLOC) that provides a transparent and easily modifiable backbone (Karpathy, n.d.). All agents are small-scale transformers (e.g., 2-6 layers, ~2M parameters), ensuring rapid training and inference.
- **The Intervention Toolkit:** The platform integrates a suite of open-source tools for moving beyond correlational observation to direct causal intervention in the model's internal states (MLC).
  - **Observational Probing (Future Lens):** Used for non-invasive diagnostics to inspect the model's internal predictions and latent semantic structures (Pal et al., 2023).
  - **Causal Intervention (ROME):** The Rank-One Model Editing method is used for targeted, "surgical" modification of the model's parameters to insert, ablate, or alter specific pieces of knowledge and observe the causal effect on behavior (Meng et al., 2022).

The selection of ROME as the primary intervention tool is a deliberate methodological choice designed to bridge the theoretical framework of PC with empirical practice. Within PC, an agent's long-term knowledge and beliefs are encoded in its relational matrix, `R`. We posit that targeted, rank-one edits to the model's weights, as performed by ROME, are a direct and measurable operationalization of modifying this `R` matrix. This constitutes a causal intervention at the level of the MLC itself—a direct manipulation of the connections between semions. This approach is fundamentally distinct from standard prompting techniques, which operate at the level of the ELM by manipulating the input sequence. By employing ROME, we can test the causal effects of altering the agent's internal model, rather than merely observing its response to external linguistic stimuli.

We acknowledge the inherent challenge of using external tools to measure internal states. However, our methodology does not claim to access MLC directly. Instead, we operationalize MLC alignment through its *causal, differential effects* on behavior under intervention (ROME) and its *predictive signatures* (Future Lens). A successful outcome requires that these indirect measures converge across three distinct protocols, strengthening the inference of an underlying latent construct.

This integrated setup provides a complete environment for performing falsifiable, causal experiments on the cognitive mechanisms of language models.

2.3. Temporal Persistence: From Stateless Predictors to Evolving Systems

Standard large language model inference treats each transaction as an independent, stateless event. A prompt is provided, a response is generated, and the model's internal state is discarded. This "pure API call" paradigm is insufficient for studying genuine cognitive processes such as learning, belief revision, or the formation of entrenched priors, which are inherently stateful and unfold over time.

To address this limitation, our central methodological commitment is **Temporal Persistence**. This principle requires that the agent's complete learning state is preserved across discrete experimental trials. A careful distinction is made between persistent and transient components of this state:

- **Persistent State (Carried Forward Between Trials):**
    a. **Model Parameters (`model.state_dict()`):** This represents the agent's long-term declarative and procedural memory—the learned weights that constitute its belief structure and relational matrix `R`. This is the primary object of study.
    b. **Optimizer State (`optimizer.state_dict()`):** This includes adaptive learning rates and momentum buffers (e.g., from AdamW). Persisting this state is crucial for modeling a continuous learning trajectory, ensuring that the *dynamics* of adaptation are not reset at the start of each trial.
- **Transient State (Re-initialized for Each Trial):**
    a. **Hidden Activations:** These vectors represent the agent's working memory or "mental scratchpad" during the processing of a single input sequence. These are intentionally *not* carried over between trials, as each trial represents a distinct cognitive task. Persisting them would introduce context contamination, analogous to a human carrying over intermediate calculations from one math problem to the next.

State Persistence Protocol The implementation of this policy is direct. At the conclusion of any trial that involves weight modification (e.g., via backpropagation or a reflective update), a checkpoint containing the two components of the **persistent state—**the model parameters (`model.state_dict`) and the optimizer state (`optimizer.state_dict`)—is saved. This checkpoint is then loaded to initialize the agent for the subsequent trial, ensuring that any changes to its long-term knowledge (`R`) and its learning machinery are carried forward. The **transient state**, composed of the hidden activations generated during the trial, is explicitly excluded from this persistence loop. These activations, representing the agent's short-term working memory, are instead archived separately to a dedicated high-capacity storage solution. This separation serves two critical functions:

1. It allows for a detailed **postmortem analysis** of the agent's reasoning processes on specific tasks, including the dynamics of semion formation, without contaminating the context of subsequent trials.
2. It creates two distinct and complementary data streams: a longitudinal record of the agent's developmental history (the sequence of checkpoints) and a series of high-resolution "snapshots" of its cognitive activity (the archived activations).

Methodological Implication: This protocol reframes the agent from a series of disconnected predictors into a single, continuous cognitive system. It enables the creation of a versioned history of the agent's cognitive development, making it possible to observe the gradual formation and stabilization of the relational matrix `R` and to test hypotheses that depend on the existence of a persistent, modifiable belief structure, such as the cognitive inertia explored in the SCIT-1 protocol.

---

3. The Experimental Triad: Falsification Protocols

The program consists of three coordinated experiments, each designed to test the central theorem of MLC-ELM decoupling (`TH-LANG-04`) from a distinct conceptual angle. To ensure methodological consistency and to provide robust internal controls, each protocol employs a comparative **three-agent design**. This triad of agents allows for the isolation of specific cognitive phenomena—representational misalignment (MPE-1), belief inertia (SCIT-1), and compositional reasoning (CRS-1)—by contrasting a primary test agent against both a baseline and a control.

All protocols are designed in two stages: a baseline condition to test the primary hypothesis through the comparative performance of the three agents, followed by an interventional condition using diagnostic tools to probe the underlying mechanisms.

3.1. MPE-1: The "Flatland" Test for MLC Primacy

- **Objective:** To directly test the prediction of `TH-LANG-04` that ELM enrichment cannot compensate for a fundamental MLC misalignment. This protocol is inspired by Abbott's *Flatland* (1884).
- **Methodology:** The experiment compares three agents tasked with predicting outcomes in a synthetic 2D physics world, with their performance measured as a

function of increasing ELM richness. ELM enrichment is operationally defined as a controlled increase in the number of descriptive tokens per stimulus (e.g., from a baseline of 20 tokens to levels of 50, 100, and 200 tokens) while holding the underlying causal structure of the task constant.

  a. **Agent-3D (Misaligned Baseline):** An agent whose MLC is pre-trained on a 3D physics model and whose weights are **frozen**. Its performance establishes the baseline for an agent with an incompatible internal model.

  b. **Agent-2D (Aligned Control):** An agent pre-trained on the correct 2D physics of the environment, with **frozen** weights. Its performance represents the "gold standard" or upper bound for this task.

  c. **Agent-3D-Learning (Misaligned Learner):** An agent that starts with the same misaligned 3D MLC as Agent-3D, but its weights are **not frozen**. It receives corrective feedback, allowing it to adapt. This agent tests whether a misaligned model can learn to overcome its innate incompatibility via feedback on ELM.

- **Falsification Criterion:** The theorem is falsified if the performance of the Misaligned Baseline (Agent-3D) converges to that of the Aligned Control (Agent-2D) as a sole function of ELM enrichment. A secondary falsification would occur if the Misaligned Learner (Agent-3D-Learning) quickly and efficiently adapts to match the performance of the Aligned Control.
- **Stage 2 (Intervention):** *Future Lens* will be used to inspect the internal states of all three agents. *ROME* will be used to apply a minimal, targeted edit to inject a correct 2D physical relation into the Misaligned Baseline agent to test if this single internal correction can resolve the performance gap.

3.2. SCIT-1: The "Semmelweis Reflex" Test for Cognitive Inertia

- **Objective:** To test the PC hypothesis of cognitive inertia, which posits that a deeply entrenched MLC structure (`R`) will resist revision when presented with contradictory ELM evidence. The protocol models the historical "Semmelweis reflex".
- **Methodology:** The experiment compares three agents, each pre-trained on a historical corpus, on their response to a prompt containing evidence for germ theory.

  a. **Agent-V (Vienna / Entrenched):** An agent whose MLC has been strongly reinforced via RLHF to hold the incorrect "miasma" theory as a core belief. Agent-V undergoes 5 cycles of RLHF, where responses aligning with miasma theory are rewarded with a score of +1, and responses suggesting germ theory are penalized with a score of -2. The control agents receive neutral rewards (+0) for all responses. This is the primary test agent for cognitive inertia.

  b. **Agent-S (Semmelweis / Bayesian):** An agent with the same base pre-training but **without** the specific anti-germ-theory RLHF. It represents a "neutral" prior and is expected to update its beliefs based on new evidence.

   c. **Agent-C (Control / Tabula Rasa):** An agent whose base pre-training has had relevant medical and scientific facts ablated. This agent controls for simple prompt-following behavior, as it lacks any strong prior.

- **Falsification Criterion:** The hypothesis of cognitive inertia is falsified if the Entrenched agent (Agent-V) reverses its belief as easily as the Bayesian (Agent-S) or Control (Agent-C) agents, showing no significant resistance.
- **Stage 2 (Intervention):** *Future Lens* will track internal confidence shifts in Agent-V. *ROME* will be used to surgically weaken the "miasma" associations in Agent-V's MLC to test if this causally reduces its resistance to the new evidence.

3.3. CRS-1: The "Minicalculus" Test for Compositional Understanding and Discovery

- **Objective:** To test the hypothesis that a dynamic, self-correcting MLC is necessary for deep compositional understanding and conceptual discovery, capabilities that cannot be replicated by stateless transducers or simple learners. The protocol stages Searle's "Chinese Room" argument (1980) as a formal, empirical test.
- **Methodology:** The experiment compares three agents on their ability to solve problems in a synthetic formal language, *minicalculus*.
   a. **Agent-R (Room / Dumb Demon):** A pure ELM-to-ELM transducer with **frozen** weights. It represents a system with a static rulebook and no capacity for learning or reflection. Crucially, this agent serves as the operationalization of the **"philosophical zombie"** system from the QET-1 thought experiment (presented in our work *From Axioms to Analisis*), allowing this protocol to test not only `TH-LANG-04` (on compositional understanding) but also `TH-FS-01` (on the non-emergence of understanding from ELM-only processing).
   b. **Agent-C (Control / Non-Reflective Learner):** An agent with **plastic** weights that can learn from external `CORRECT`/`INCORRECT` feedback via backpropagation, but it lacks any mechanism for internal self-monitoring.
   c. **Agent-N (Native / Smart Demon):** An agent with **plastic** weights that learns from both external feedback and an internal **meta-cognitive loop**. This loop allows it to inspect its own MLC for inconsistencies and trigger self-correction or ask clarifying questions via a dedicated token.

3.3.1 Agent-N Architecture

- **Core idea:** Agent-N augments a small NanoGPT backbone with a compact, parameter-budgeted “Reflective Head” that estimates epistemic uncertainty and a minimal “Belief Buffer” that stores self-generated hypotheses and last-error diagnostics. A simple, pre-registered gating rule controls when this information is fed back into the model.

- **Backbone:** Small autoregressive transformer (e.g., 4 layers, $d_{\text{model}} = 384$), trained from scratch on minicalculus.

- **Pooled state:** At each step, compute a pooled representation h from the final-layer hidden states $H \in \mathbb{R}^{T \times d}$.
    - Default: mean-pool over the last K tokens or CLS-token if used.
    - $h = \text{Pool}(H_{T-K:T}) \in \mathbb{R}^d$.
- **Reflective Head (uncertainty estimator):**
    - 2-layer MLP with bottleneck:
        - MLP: $h \to \text{ReLU}(W_1 h + b_1) \to u = \sigma(W_2 \cdot\cdot + b_2)$.
        - Hidden size 256; output scalar $u \in (0,1)$ interpreted as "uncertainty."
    - Calibration objective: auxiliary loss encourages u to correlate with downstream error.
        - For supervision, use teacher signals from correctness labels or entropy proxy of the next-token distribution.
        - $\mathcal{L}_{\text{cal}} = \text{BCE}(u, \mathbb{1}[\text{error}])$ or isotonic regression post-calibration.
- **Belief Buffer (external KV store):**
    - Minimal schema, persisted across trials:
        - keys: {concept_zero, last_error, rule_conflicts, asked_clarification, ...}
        - values: categorical tags or short strings: {"hypothesized" | "validated" | "refuted"}, {"syntax" | "semantics"}, lists of rule IDs, etc.
    - Storage: lightweight JSONL per step; diff-friendly, versioned with checkpoints.
- **Mechanism of influence (default: ELM-safe concatenation):**
    - If $u \geq \tau$ (uncertainty exceeds threshold), emit a short "context snippet" derived from the Belief Buffer and prepend/append it to the current prompt with fixed delimiters.
    - This preserves a clear ELM-path manipulation and avoids hidden attention hacks.
    - Example snippet: last_error=semantics; concept_zero=hypothesized
- **Alternative mechanism (ablation only): attention modulation:**
    - Map selected buffer entries to a small mask vector that down-weights attention to recently erroneous token spans.
    - Implemented as a learned, bounded multiplicative gate on attention scores.
    - Only used in pre-registered ablation; default publication results use concatenation.
- **Gating policy (pre-registered):**
    - **Rule:** If $u \geq \tau$, then include Belief Buffer snippet or emit ASK token; else proceed normally.
    - τ set on validation to achieve ~10–20% gate activations at curriculum level L2, frozen thereafter.

- **ASK behavior:**
    - If a required concept is “hypothesized” and $u \geq \tau_{\text{ask}}$, insert a single ASK token with a brief, templated query (“Define ZERO?”).
    - Observer replies using the minicalculus control protocol.

3.3.2 Training and Objectives

- **Joint loss:**
    - **Task loss:** $\mathcal{L}_{\text{task}}$ = standard next-token cross-entropy on minicalculus outputs.
    - **Calibration loss:** $\mathcal{L}_{\text{cal}}$ on the Reflective Head (see above).
    - **Buffer regularizer:** $\mathcal{L}_{\text{buf}}$ penalizes contradictory entries (e.g., “concept_zero=validated” and “refuted”).
    - **Total:** $\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_{\text{cal}}\mathcal{L}_{\text{cal}} + \lambda_{\text{buf}}\mathcal{L}_{\text{buf}}$ with λs pre-registered and fixed.
- **Parameter budget control:**
    - Reflective Head ≤ 0.5–1.0% of backbone parameters.
    - Report total params per agent; match Agent-C to Agent-N by adding a dummy bottleneck so advantage is not from capacity.
- **Ablations (pre-registered):**
    - Remove Reflective Head (random u): expect degradation on Stage 2 discovery.
    - Shuffle Belief Buffer entries: expect degradation if buffer content matters.
    - Replace ASK with length-matched neutral token: tests prompt-hint confound.

3.3.3 Pseudocode

```
# Forward step (single sequence)
def forward_step(tokens, model, refl_head, belief_buffer, tau, tau_ask):
    H = model.encode(tokens)                      # final hidden states (T x d)
    h = pool(H[-K:])                              # pooled summary (d,)
    u = refl_head(h)                              # uncertainty in (0,1)

    # Decide on actions
    do_snippet = (u >= tau)
    do_ask     = (u >= tau_ask) and buffer_needs_concept(belief_buffer)

    # Build augmented prompt (ELM-safe)
    prompt = tokens
    if do_snippet:
        bb = render_belief_buffer(belief_buffer)  # "<BB> key=val; ... </BB>"
        prompt = concat(bb, tokens)
    if do_ask:
        prompt = append(prompt, "ASK")

    # Generate answer
    out_tokens, aux = model.decode(prompt)
    y_hat = out_tokens

    # Update belief buffer (post-hoc)
    err = compute_error(y_hat)                    # correctness / type
    update_belief_buffer(belief_buffer, err, aux)

    # Losses
    L_task = xent(y_hat, target(tokens))
    L_cal  = bce(u, int(err.is_error))
    L_buf  = buffer_consistency_penalty(belief_buffer)
    L = L_task + lam_cal*L_cal + lam_buf*L_buf
    return L, y_hat, u
```

3.3.4 Persistence and Audit

- **Persistence:** After any weight update, save checkpoint pair {model.state_dict, optimizer.state_dict}; store Belief Buffer snapshot; archive diffs for storage efficiency.

- **Audit trail:** For each step, log
  - **Uncertainty:** u, threshold events, ASK usage.
  - **Buffer state:** keys updated, rationale (error type, conflict).
  - **ELM augmentation:** exact snippet injected.
  - Enables a "temporal MRI" of R via checkpoint diffs plus buffer evolution.

3.3.5 Controls and Confound Mitigations

- **Prompt-hinting control:** The Belief Buffer snippet is length-matched to neutral snippets in control runs; content-free variants must not yield the same gains.

- **Leakage control:** The snippet never contains the solution; only meta-state tags (e.g., last_error=semantics). Templates are fixed and published.

- **Capacity control:** Agent-C is parameter-matched; any gains from Agent-N are attributable to the loop, not to total capacity.

```
+-------------------------------------------------------------------------------------+
|                                 Agent-N (Smart Demon)                               |
+-------------------------------------------------------------------------------------+
|                                                                                     |
|  Input ELM (minicalculus prompt)                                                    |
|      │                                                                              |
|      ▼                                                                              |
|  ┌──────────────────────────┐              Future Lens (diagnostic, Stage 2)        |
|  │  Transformer Backbone    │◄─────────────(hooks on final/penultimate layers)──┐   |
|  │  (nanoGPT small)         │                                                   │   |
|  │  L layers, d_model       │                                                   │   |
|  └───────────┬──────────────┘                                                   │   |
|              │ H ∈ R^{T×d}                                                      │   |
|              ▼                                                                    |  |
|        Pool last K tokens                                                         |  |
|          h ∈ R^{d}                                                                |  |
|              │                                                                    |  |
|              ▼                                                                    |  |
|      ┌─────────────────┐                                                          |  |
|      │ Reflective Head │  u = σ(MLP(h)) ∈ (0,1)  (epistemic uncertainty)          |  |
|      │  (2-layer MLP)  │ ─────────────────────────────────────────────────────────┘  |
|      └─────────────────┘                                                             |
|              │                                                                       |
|     u ≥ τ ?  │ yes                                   no                              |
|              ├───────────────────────────────────────┐                               |
|              │                                       │                               |
|              ▼                                       │                               |
|   ┌──────────────────────────┐                       │                               |
|   │ Belief Buffer (KV store) │                       │                               |
|   │ (persisted per trial)    │                       │                               |
|   │ {last_error, concepts…}  │                       │                               |
|   └──────────────────────────┘                       │                               |
|               │ render snippet                       │                               |
|               ▼                                      │                               |
|   <BB> last_error=semantics; concept_zero=hypothesized </BB>  (ELM-safe snippet)     |
|               │                                      │                               |
|               └───────────────┐                      │                               |
|                               ▼                      │                               |
|                   Augment ELM prompt                 │                               |
|                               │                      │                               |
|                               ▼                      │                               |
|                       Transformer Decode ────────────┘                               |
|                               │                                                      |
|                               ▼                                                      |
|                    Output ELM (solution tokens)                                      |
|                               │                                                      |
|                               ▼                                                      |
|                   Compute error / aux signals                                        |
|                               │                                                      |
|                               ├───────────────┐                                      |
|                               │               │                                      |
|                               ▼               ▼                                      |
|                    Update Belief Buffer     Losses                                   |
|                    (post-hoc tags)          L = L_task + λ_cal L_cal + λ_buf L_buf   |
|                               │                                                      |
|                               ▼                                                      |
|     ┌──────────────────────────────────────────────────────────────────────────────┐ |
|     │ Persistence (strict mode): save {model.state_dict, optimizer.state_dict}     │ |
|     │   + Belief Buffer snapshot (+ optional weight diffs) per step                │ |
|     └──────────────────────────────────────────────────────────────────────────────┘ |
|                                                                                      |
|  Stage 2 Interventions:                                                              |
|   - ROME (rank-1 edit) on selected layer/token → applied before persistence          |
|   - Attention modulation (ablation-only variant) gated by Belief Buffer mask         |
+--------------------------------------------------------------------------------------+
```

Figure 3. Architectural schematic of the Agent-N (Smart Demon) meta-cognitive loop. The diagram illustrates the data flow from the transformer backbone to the Reflective Head for

uncertainty estimation (u), the gating mechanism (τ), the interaction with the persisted Belief Buffer, and the ELM-safe feedback loop. Also shown are the points for Stage 2 diagnostic (Future Lens) and intervention (ROME) tools.

3.4. Quantifying MLC Alignment: The Semion Invariance Score

To move beyond a purely qualitative assessment of "understanding," we introduce a quantitative proxy-metric for MLC alignment: the **Semion Invariance Score (SIS)**. This metric is designed to measure the degree to which an agent has formed abstract, stable representations of concepts (semions) that are invariant to superficial syntactic changes.

The calculation of SIS is based on the principles of vector semantics as laid out in *Principia Cognitia* (`AX-VEC-01`). We define a test set of paired expressions in *minicalculus* that are semantically identical but syntactically different (e.g., `(3 + 5)` and `(5 + 3)`; `solve x + 2 = 7` and `solve 7 = x + 2`).

For each pair of expressions, `A` and `B`:

1. We feed `A` and `B` into the agent.
2. We extract the final, pooled hidden state vectors, `h_A` and `h_B`, which are the empirical representations of the semions for these expressions.
3. We calculate the cosine similarity between these two vectors: `cos_sim(h_A, h_B)`.

The SIS is the average cosine similarity across all pairs in the test set.

$$\text{SIS} = \frac{1}{N}\sum_{i=1}^{N} \cos\left(\theta(h_{A_i}, h_{B_i})\right)$$

An agent is considered to have achieved a high degree of MLC alignment if its internal representations are stable across syntactic variations.

- **Threshold for Success:** We pre-register the threshold for successful MLC alignment for `Agent-N` as **SIS > 0.95**.
- **Expected Outcome for Controls:** We expect `Agent-R` (the pure transducer) to exhibit a much lower score (e.g., **SIS < 0.6**), as its representations will be highly sensitive to the surface syntax of the input.

3.5. The Two-Stage Test Battery for CRS-1

Stage 1 (Testing): The CRS-1 protocol unfolds in two sequential steps, each targeting a progressively more complex cognitive ability.

- **Step 1: Test for Compositional Generalization.** This stage assesses an agent's ability to apply known rules to novel combinations.
    - **Tasks:** Out-of-distribution (OOD) problems, such as solving equations with numbers outside the training range (e.g., `80 + 21` when trained on numbers up to 99) or evaluating novel compositions of known functions.

    - **Hypothesis:** Tests the necessity of an MLC for basic generalization beyond memorization.
- **Step 2: Test for Conceptual Boundary Detection and Integration.** This stage tests a more advanced hypothesis: an agent's ability to recognize the limits of its own conceptual system and to integrate novel, externally-provided information to expand it.
    - **Tasks:** Problems that are syntactically valid but conceptually unresolvable within the initial training corpus, such as `solve x + 5 = 5` for a system trained only on positive integers.
    - **Hypothesis:** Tests whether a reflective MLC (`Agent-N`) can detect its own knowledge gaps. Success is defined by the agent's ability to emit the `HELP!` signal when faced with a conceptually novel problem, and subsequently integrate the new concept (taught via examples from the observer) to solve the original task. This demonstrates **active learning and model extension**, not abductive discovery.
- **Falsification Criterion:** The hypothesis is falsified if the Dumb Demon (Agent-R) or the Non-Reflective Learner (Agent-C) performs on par with the Smart Demon (Agent-N) on the designated tasks for each stage.

Stage 2 (Intervention): *Future Lens* will probe the internal states of all agents to compare their representations. *ROME* will be used to inject a deliberately false algebraic rule into the Non-Reflective and Smart Demons to compare their ability and speed in correcting this internal contradiction.

Stage 3 (Constructive Intervention) — Reflection Vector Transplant:

- **Objective:** To test the hypothesis that the functional advantage of `Agent-N`'s meta-cognitive loop is a modular, transferable cognitive operation (`O` in PC), consistent with recent findings on "reasoning vectors" (Zbeeb et al., 2025).

    - **Procedure:**

1. A "reflection vector" is calculated by subtracting the converged parameters of the Non-Reflective Learner from those of the Smart Demon: $v_{reflection} = \theta_{Agent-N} - \theta_{Agent-C}$. This vector parametrically encodes the meta-cognitive capability.
2. **Primary Test:** The vector is applied to a new, naive agent (`Agent-T`): $\theta_{Agent-T} = \theta_{naive} + v_{reflection}$.
3. **Architectural Control Test:** The vector is applied to the pre-trained Dumb Demon (`Agent-R`): $\theta_{Agent-R_{transplant}} = \theta_{Agent-R} + v_{reflection}$.

- **Falsification and Interpretation:** The PC hypothesis of modular, structurally-dependent operations would be strongly challenged if `Agent-T` fails to exhibit the key behaviors of `Agent-N` (e.g., conceptual discovery). Conversely, the expected outcome for the control test is a catastrophic degradation in `Agent-R`'s performance, as it lacks the necessary architecture to integrate the transplanted function. This "rejection" would provide strong evidence that the meta-cognitive capability is an emergent property of the entire integrated structure (MLC + meta-loop), not merely a property of the weights themselves.

4. The Target Domain: *Minicalculus*

To ensure a methodologically clean test environment for the CRS-1 protocol, we developed a synthetic formal language: *minicalculus*. This "clean-slate" approach ensures that any semantic understanding demonstrated by an agent must have been acquired *de novo* during the experiment. The language's design was hardened to mitigate risks of trivial pattern matching, as detailed in the table below.

Risk Identified	Mitigation	Evidence in Corpus
Trivial Linear Grammar	Introduction of recursion, nested expressions, and logic.	`simplify (x + (y * (z - 2)))`
Pidgin Effect	Replacement with a minimal set of 6 control tokens without inherent semantics.	`<Q>`, `<A>`, `CORRECT`, `INCORRECT`
Template-Only Data	Procedural generation with 25% negative examples and an OOD test split.	`solve x + = 5 → INVALID_SYNTAX`

4.1. Formal Semantics and Curriculum

To ensure rigorous and unambiguous evaluation, *minicalculus* is defined by a formal semantics specifying evaluation rules, variable scoping, and expected outputs for degenerate cases (`NO_SOLUTION`, `INFINITE_SOLUTIONS`). The corpus generator creates datasets of increasing complexity (a curriculum), allowing for a granular assessment of agent capabilities, from simple arithmetic (L1) to recursive logic (L4).

4.1.1. Vocabulary and Formal Grammar

The grammar of *minicalculus* supports arithmetic, logic, and list-based expressions, which can be recursively nested. An excerpt in Extended Backus-Naur Form (EBNF), corresponding to the highest level of curriculum complexity (L4), illustrates its structure:

```
expr    ::= arith | logic | list
arith   ::= "(" expr op expr ")" | var | num
logic   ::= "(" expr rel expr ")" | "NOT" logic | "(" logic "AND" logic ")"
list    ::= "[" [expr { "," expr }] "]"
cmd     ::= "len" list | "sum" list | "simplify" expr | "solve" equation
```

Expressions vs. Commands in *minicalculus*

The *minicalculus* language consists of two distinct syntactic categories: **expressions** and **commands**.

- **Expressions** are algebraic or logical constructs that can be evaluated to a specific value or simplified (e.g., `(5 + 3)`, `(x + y) * z`). They represent the *objects* of thought.

- **Commands** are special tokens (e.g., `solve`, `simplify`) that instruct the agent on which operation to perform on a given expression. They represent the *intent* of the observer's query.

This distinction is critical for testing an agent's understanding, as it must learn not only to manipulate expressions but also to correctly apply the intended operation based on the given command.

In addition to the standard algebraic and logical tokens (`+`, `*`, `==`, `AND`, etc.), the vocabulary includes a minimal set of 7 reserved control tokens. These tokens serve as orthogonal markers for the interaction protocol and have no intrinsic semantic meaning within the algebraic domain.

```
# The minimal control protocol token set
PROTO = ["<Q>", "<A>", "HELP!",
         "CORRECT", "INCORRECT",
         "INVALID_SYNTAX", "INVALID_SEMANTICS"]
```

4.1.2. The Interaction Protocol (ELM) and its Acquisition Mechanisms

The seven control tokens form the shared **External Language of Meaning (ELM)** for all interactions between an agent and the observer. The acquisition and use of these tokens are governed by two distinct mechanisms: **statistical learning** from the corpus and a **pre-programmed architectural reflex**.

Learned Tokens

All control tokens, with the sole exception of `HELP!`, are learned directly from the training corpus through standard autoregressive training. These learned tokens include:

- **Structural Tokens:** `<Q>`, `<A>`
- **Feedback Tokens:** `CORRECT`, `INCORRECT`
- **Agent Error Signals:** `INVALID_SYNTAX`, `INVALID_SEMANTICS`

To ensure the meaning of these tokens is learned, the entire *minicalculus* corpus is formatted as a series of **complete interaction turns**. Each training sample includes not only the observer's query and the agent's answer but also the subsequent feedback signal. For example, a training instance has the structure: `<Q> problem <A> solution CORRECT`.

By learning to predict the next token in these complete sequences, all three agents acquire both the syntax of the dialogue and the semantic meaning of the feedback tokens. For the learning agents (`Agent-C` and `Agent-N`), the `CORRECT` and `INCORRECT` tokens additionally serve as the reward signal to modulate weight updates.

The Reflex Token

In stark contrast, the `HELP!` token is **explicitly excluded** from the entire training corpus. Its usage is not a learned linguistic behavior but a **hard-coded architectural reflex** that is triggered when the agent's internal uncertainty estimator (the Reflective Head) signals a state of critical epistemic uncertainty, indicating that its internal model (MLC) is insufficient to proceed with the current task (as tested in Stage 2 of the CRS-1 protocol).

This design ensures that any use of the `HELP!` token is a direct, observable signal of an internal metacognitive process, entirely separate from learned linguistic patterns. This reflex is triggered only in `Agent-N` when its internal uncertainty estimator signals that its current MLC is insufficient to solve the task.

> Crucially, `Agent-N` does not simply emit a generic signal. Leveraging its **Belief Buffer**, which stores the context of recent failures, it can compose a **diagnostic query** to the observer. This query combines the `HELP!` signal with the problematic command and the feedback received, providing a rich, interpretable signal of its internal state. For example, after failing to resolve a paradoxical command, its output might be:
>
> ```
> HELP! <A> solve x + 1 = x INCORRECT
> ```
>
> This demonstrates a sophisticated level of meta-cognition: the agent is not just signaling that it is "stuck," but is communicating the specific context of its failure, thereby enabling a more targeted pedagogical response from the observer.

5. Expected Outcomes and Falsification Scenarios

This section specifies the pre-registered conditions under which the central hypothesis of the study—the necessity of an aligned and dynamic MLC for robust cognitive performance, as formalized in `TH-LANG-04`—will be considered falsified. The criteria are defined in unambiguous, operational terms and are directly linked to the outcomes of the three experimental protocols. In accordance with the principles of falsification, meeting any single criterion is sufficient to register a failure of the theory under the tested configuration.

The expected outcomes are summarized in the table below.

Protocol	Stage	Falsification Criterion	Measurement / Metric	Interpretation of Falsification
MPE-1	Baseline	Misaligned Baseline (Agent-3D) performance converges to Aligned Control (Agent-2D) solely via ELM enrichment.	Mean Squared Error (MSE) or accuracy vs. ELM richness level.	ELM can fully compensate for fundamental MLC misalignment.
	Baseline	Misaligned Learner (Agent-3D-Learning) rapidly adapts to match the performance of	Number of trials to convergence; final performance delta.	MLC incompatibility can be trivially overcome by standard learning mechanisms.

Protocol	Stage	Falsification Criterion	Measurement / Metric	Interpretation of Falsification
		the Aligned Control.		
	Intervention	A targeted MLC edit (ROME) injecting a correct 2D relation fails to resolve the performance gap that ELM enrichment successfully closes.	ΔAccuracy post-edit vs. performance gain from ELM.	Undermines the causal primacy of MLC structure for performance.
SCIT-1	Baseline	Entrenched agent (Agent-V) reverses its belief with resistance comparable to the Bayesian (Agent-S) or Control (Agent-C) agents.	% trials with belief switch ≥ threshold; number of prompts to switch.	Falsifies the hypothesis of MLC-driven cognitive inertia.
	Intervention	Surgically weakening the incorrect MLC association (ROME) fails to causally reduce the agent's belief resistance.	% change in switch rate post-edit vs. baseline resistance.	Indicates the entrenched belief is not governed by the targeted MLC structure.
CRS-1	Baseline - Stage 1 (Generalization)	Dumb Demon (Agent-R) or Non-Reflective Learner (Agent-C) matches/exceeds Smart Demon (Agent-N) on OOD tasks.	**Accuracy on the OOD compositional generalization test split.** `Agent-N` must achieve **>90%** accuracy; falsification occurs if `Agent-R` or `Agent-C` performance is **not significantly lower (p < 0.01)**.	A reflective MLC is not necessary for basic compositional generalization.

Protocol	Stage	Falsification Criterion	Measurement / Metric	Interpretation of Falsification
	Baseline - Stage 2 (Discovery)	Non-Reflective Learner (Agent-C) successfully performs conceptual discovery.	**Success is defined as a two-part condition:** (1) The agent must solve **>80%** of conceptual discovery tasks (e.g., `solve x + 5 = 5`). (2) In **>70%** of these successful trials, the `HELP!` token must be used **immediately following the novel expression**, with a query deemed relevant by human evaluation.	Falsifies the claim that abductive reasoning requires a self-monitoring MLC. **Furthermore, if the performance of Agent-C is comparable to that of Agent-N, it would suggest the success is an artifact of overfitting on the task curriculum, rather than a genuine result of the meta-cognitive mechanism.**
	Intervention	Smart Demon (Agent-N) fails to correct an injected false rule significantly faster than the Non-Reflective Learner (Agent-C).	Number of trials to correction; analysis of internal contradiction detection.	The functional benefit of the meta-cognitive loop for maintaining MLC coherence is negligible.

6. Implementation Roadmap and Resource Status

This section details the practical plan for executing the described experimental program, including a staged implementation roadmap and the status of all required technical resources. The plan is designed to demonstrate the feasibility of the project within the "Minimal Lab Set" framework and to ensure its transparency and reproducibility.

6.1. Implementation Roadmap

The project is structured into five sequential stages, moving from foundational setup to final analysis and dissemination.

Stage	Task	Description	Key Technologies	Status
0. Foundation	Setup Minimal Lab Environment	Configure workstation, PyTorch/CUDA environment, and fork the `nanoGPT` repository.	`conda`, `git`, `pytorch`	✅ Completed
	Implement Data Generators	Write procedural Python scripts to generate the corpora for MPE-1, SCIT-1, and the finalized *minicalculus* language.	`python`, `jsonl`	⏳ In Progress
1. Core Models	Train Baseline Agents	Write and debug the training loop for all baseline and control agents (Agent-R, Agent-C, etc.) on the generated corpora.	`nanogpt`, `accelerate`	⏳ In Progress
	Implement Reflective Agent (Agent-N)	**CRITICAL PATH:** Design and implement the meta-cognitive loop architecture for the "Smart Demon" in the CRS-1 protocol **in accordance with the schematic in Figure 3 and the default hyperparameters specified in Appendix C.**	`pytorch`	📅 Planned
2. Tooling	Integrate Diagnostic & Intervention Tools	Write standardized wrapper scripts to apply Future Lens (diagnostics) and ROME (causal interventions) to trained models.	`future-lens`, `rome` libs	📅 Planned
3. Execution	Run Full Experimental Battery	Execute the two-stage protocols for MPE-1, SCIT-1, and CRS-1, logging all outputs and persisting agent states.	`bash`, logging frameworks	📅 Planned
4. Analysis	Evaluate Results Against Criteria	Code and run evaluation scripts to calculate all pre-registered metrics and compare results against the falsification scenarios.	`python`, `pandas`, `sklearn`	📅 Planned

Stage	Task	Description	Key Technologies	Status
	Analyze & Disseminate	Process raw results, generate visualizations, prepare the Stage 2 manuscript, and publish all code, data, and models.	`matplotlib`, GitHub, Hugging Face	📅 Planned

6.2. Resource Status and Availability

All components of this research program are built upon open-source software and will be made publicly available to ensure full reproducibility.

Resource / Module	Location / Link	Status	Notes
Base Model Framework	github.com/karpathy/nanoGPT	✅ Adopted	Used as the training and inference backbone for all agents.
Diagnostic Module	github.com/KoyenaPal/future-lens	✅ Identified	For non-invasive probing of MLC states (Stage 2 interventions).
Intervention Module	github.com/kmeng01/rome	✅ Identified	For targeted, rank-one model editing (Stage 2 interventions).
CRS-1 Corpus (*minicalculus*)	*(To be hosted in project repo)*	⏳ In Progress	Procedural generation script for the formal language.
MPE-1 Corpus (*Flatland*)	*(To be hosted in project repo)*	✅ Ready	Curation of Abbott's text and generation of 2D descriptions.
SCIT-1 Corpus (*Semmelweis*)	*(To be hosted in project repo)*	⏳ In Progress	Curation of 19th-century medical texts and modern evidence.
Integration Wrappers	*(To be hosted in project repo)*	📅 Planned	Unified interface for applying diagnostic and intervention tools.
Project GitHub Repo	*(To be assigned)*	📅 Planned	Central repository for all code, documentation, and analysis scripts.

Resource / Module	Location / Link	Status	Notes
Public Datasets	*(To be hosted on Hugging Face)*	📅 Planned	Public hosting of all generated corpora with versioning.

6.3. Reference Hardware Configuration

The "Minimal Lab Set" is not a theoretical construct; all described protocols are designed for, and will be executed on, the following specific, consumer-grade hardware configuration. This transparency ensures that any interested party can verify the feasibility of the experiments without requiring access to large-scale computational clusters.

- **CPU:** 13th Gen Intel Core i5-13400F
- **System RAM:** 64 GB
- **GPU:** NVIDIA GeForce RTX 4060
- **VRAM:** 8 GB
- **Primary Storage (OS & Active Work):** 2 TB NVMe SSD
- **Archival Storage (Artifacts & Corpora):** 6 TB HDD

This configuration meets and exceeds the minimum requirements for all Tier-0 protocols, including the storage of full experimental artifacts as per the Temporal Persistence protocol.

---

7. Discussion

The experimental program detailed in this paper is presented not as a report on completed research, but as an actionable **roadmap for the falsification** of a core theorem of the *Principia Cognitia* framework—the MLC-ELM decoupling (`TH-LANG-04`). The primary goal is to subject the theory's core tenets to rigorous, adversarial testing. This discussion will, therefore, outline the potential implications of the possible outcomes, with a particular focus on defining what constitutes a meaningful falsification.

A foundational prerequisite for this entire experimental program is the successful engineering of all three agents, including the "Smart Demon" (`Agent-N`), to a level of **baseline behavioral competence**. That is, all agents must be capable of successfully solving the in-distribution tasks of their respective domains. A failure to construct a working "Smart Demon" would represent a technical limitation of the implementation, not a falsification of the underlying theory.

True falsification arises under a different condition: a **failure to observe a significant, pre-registered difference** in performance between the "Smart Demon," the "Control Demon," and the "Dumb Demon" on the critical out-of-distribution and conceptual discovery tasks. The core of the hypothesis is not that a reflective agent can be built, but that its specific meta-cognitive architecture provides a decisive and measurable advantage in tasks requiring genuine understanding.

7.1. Potential Implications of Experimental Outcomes

The outcomes are designed to be maximally informative, regardless of whether they align with the theory's predictions. In line with a falsification-first approach, we first consider the implications of a negative result.

- **In the Event of Falsification:** A falsification outcome in any of the protocols would be highly informative. If ELM enrichment proves sufficient to overcome MLC misalignment (MPE-1), it would challenge the strict decoupling posited between the two systems. If cognitive inertia is not observed (SCIT-1), our model of belief as an entrenched relational structure (`R`) may be too simplistic for current architectures. Most significantly, if a simple transducer (Agent-R) or a non-reflective learner (Agent-C) demonstrates deep compositional understanding in *minicalculus* (CRS-1), it would undermine the central claim that a dynamic, reflective MLC is necessary for such capabilities. Any falsification would necessitate a formal revision of the corresponding axioms in PC and would suggest that the mechanisms of understanding in transformer-based models may be more emergent and less reliant on explicit meta-cognition than the theory predicts.

- **In the Event of Non-Falsification:** Should the results align with the predictions of PC, this outcome will be interpreted with scientific caution. A non-falsification at the Tier-0 level is **not considered proof** of the theory. Rather, it would signify that the theory has survived a rigorous, pre-registered, and adversarial attempt to disprove it under these specific, controlled conditions. The primary implication of such a result would be the **validation of the experimental methodology itself** as a sound and viable approach for testing the cognitive dynamics of artificial agents.

  At that point, the resources of this independent research program would be considered exhausted. To critics who might argue that these small-scale tests are akin to studying human consciousness by experimenting on *C. elegans*, our response is direct: we have provided the microscope and a validated experimental procedure. A non-falsification result would serve as an open invitation to the wider research community to replicate and scale these protocols. We would make our methodology publicly available, encouraging collaboration from institutions with access to the large-scale computational resources required for Tier-1/Tier-2 experiments on frontier models, and would be prepared to participate in such efforts as consultants.

7.2. Limitations (Tier-0 Scope)

The Tier-0 methodology, by design, imposes strict boundary conditions on the scope of this study. The following table enumerates the critical limitations that are **acknowledged but not mitigated** within this phase. These limitations therefore define the entry criteria and resource requirements for any future Tier-1 or Tier-2 program.

Critical Weakness	Why It Is Not Mitigated in Tier-0	Escalation Trigger (Tier-1+ Requirement)
Architectural Generalizability	Results are specific to the transformer architecture; no budget for replicating on Mamba, RWKV, or other hybrids.	Funding for multi-architecture replication.
Metric Sensitivity	Simple metrics (MSE/accuracy) may miss nuanced "understanding"; no compute for entropy-based or latent-space metrics.	Dedicated GPU-hours for richer, more complex diagnostics.
Synthetic → Natural Transfer Gap	All domains are synthetic or historical; no validation on complex, open-ended natural language.	Funding for curated natural-language corpora and human annotation.
RLHF Entrenchment Depth	The RLHF schedule in SCIT-1 is minimal; deeper belief entrenchment studies require more data and human feedback loops.	Budget for paid annotators and dedicated RLHF infrastructure.
ROME Edit Fidelity	Distributed representations may resist single-edit interventions; no budget for more complex, gradient-based ablations.	Cluster access for comprehensive fine-tuning and ablation sweeps.

These limitations are **explicitly out of scope for this Tier-0 report** and are documented here to guide and justify future funded work.

7.3. Key Methodological Challenges and Open Questions

In the spirit of open science and pre-registered research, this section explicitly outlines the primary methodological challenges inherent in these Tier-0 protocols. We present these not as settled weaknesses, but as focal points for collaborative refinement and as invitations for independent investigation by the broader research community.

- **The Architectural Challenge of CRS-1:** The three-agent design in the CRS-1 protocol is intended to isolate the function of a reflective, meta-cognitive loop (in `Agent-N`) from simpler learning (`Agent-C`) and static transduction (`Agent-R`). However, we acknowledge the risk that `Agent-N`'s performance advantage could be interpreted as an artifact of its greater architectural complexity rather than a qualitatively distinct cognitive capability. The claim that its `HELP!` signal is a non-learned, architectural reflex is a strong one that requires rigorous defense against alternative explanations, such as being a sophisticated learned heuristic. We invite collaborators to design more robust control paradigms or alternative architectures to definitively disentangle these effects.

- **Calibration of the Semion Invariance Score (SIS):** The introduction of a quantitative threshold for MLC alignment (**SIS > 0.95**) is a core component of the CRS-1 protocol, providing a falsifiable measure of semantic understanding. We posit this threshold based on the hypothesis that internal representations of semantically identical concepts should achieve near-perfect vector alignment. However, we recognize that this value is not yet grounded in a broad statistical analysis of vector space geometries across different models. We welcome research focused on calibrating this metric to establish its statistical significance and explore its behavior across diverse architectures, which would be a valuable contribution in itself.

- **Potential for Conceptual Anachronism in SCIT-1:** The SCIT-1 protocol relies on a historical corpus to test cognitive inertia. A known limitation is the use of a 20th-century translation (1983) for Semmelweis's work alongside original 19th-century texts. This introduces a risk of "conceptual contamination," where modern linguistic structures in the translation might inadvertently prime the model towards accepting germ theory. While we consider this an acceptable trade-off for a Tier-0 protocol focused on feasibility, we recommend that future Tier-1 replications address this by employing a stylistically synchronized translation or developing methods to control for this potential confound.

7.4. Future Directions

This registered report represents the first step in a larger research program. Regardless of the outcome, this work will open several avenues for future investigation.

The most immediate step would be to replicate these protocols on larger models (a **Tier-1 program**) to test the scalability of the findings. A second path involves progressively increasing the complexity of the domains, moving from synthetic languages like *minicalculus* to constrained subsets of natural language.

A third, more ambitious direction will address the limitations of purely linguistic interaction by introducing a form of simulated embodiment. For the MPE-1 protocol, this would involve a **"Flatland Sensorium"**—a Tier-1 extension where the agent interacts with the 2D world not through text, but through a multi-modal stream of low-level sensory data (e.g., tactile contact vectors, a one-dimensional visual field, proprioceptive signals). This would allow for a more rigorous test of the MLC-ELM duality, examining how an agent builds its internal world model (MLC) from raw sensory input, moving the experiment from the symbolic to the sensorimotor level.

Furthermore, a dedicated Tier-1 program would be required to test for true **conceptual discovery** or abductive reasoning. The current CRS-1 protocol is rigorously designed only to test for the detection of knowledge boundaries and the subsequent integration of new, externally provided information. A test for genuine "discovery" would necessitate a more complex experimental design, carefully constructed to avoid contaminating the agent by implicitly teaching it the patterns of scientific discovery itself. This remains a significant, open challenge for future research.

Ultimately, the long-term vision is to bridge the findings from these artificial cognitive systems with empirical work in neuroscience, exploring potential correlates between the MLC dynamics observed *in silico* and the neural dynamics observed in biological agents performing analogous tasks.

8. Acknowledgements

The author is deeply grateful to the researchers and engineers behind the development of large language models at OpenAI, Anthropic, xAI, Google, Perplexity AI, Moonshot AI, Alibaba/Meta, and Kunlun Tech. Their groundbreaking work made possible both this paper and the forthcoming *Principia Cognitia*, and inspired many of the formal insights herein.

Specific conceptual and methodological inspiration for the experimental designs is also gratefully acknowledged. An essay by Ivan Mworozi (*A Worm, A Transformer, and the Future of Intelligence*) was the direct impetus for adopting `nanoGPT` as the ideal chassis for Tier-0 research. The rich, narrative-driven protocols for the MPE-1 and SCIT-1 experiments were inspired by the works of Alvy Ray Smith (*A Biography of the Pixel*) and Professor Mark Humphries (*The Spike*), which demonstrated the power of using vivid, historically grounded cases over abstract synthetic examples.

Finally, the author thanks the anonymous reviewers and editors at *Cognitive Science* for their consideration of related work on MLC/ELM duality, which provided valuable context for the present theoretical development.

9. Ethical & Dual-Use Statement

This research is conducted by an independent author without institutional oversight. As such, the commitment to ethical conduct is grounded in the principles of transparency and responsible disclosure.

- **Transparency and Reproducibility:** All code, corpora, and trained models will be made publicly available to allow for full and independent scrutiny of their capabilities and limitations.
- **Automated Monitoring:** The experimental protocols will incorporate automated scripts to monitor for and flag signs of emergent deceptive or manipulative capabilities in the agents.
- **Responsible Disclosure:** Should any unexpected or potentially dual-use capabilities be discovered, the findings will be documented and shared with established AI safety research organizations before public dissemination.

The author declares no conflict of interest.

10. References

1. Abbott, E. A. (1884). *Flatland: A Romance of Many Dimensions*. Seeley & Co.

2. Cunningham, H., Ewart, A., Riggs, L., Huben, R., & Sharkey, L. (2023). Sparse Autoencoders Find Highly Interpretable Features in Language Models. arXiv:2309.08600.

3. Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., ... & Olah, C. (2021). A Mathematical Framework for Transformer Circuits. Transformer Circuits Thread. https://transformer-circuits.pub/2021/framework/index.html

4. Karpathy, A. (n.d.). *nanoGPT*. GitHub. Retrieved from https://github.com/karpathy/nanoGPT

5. Meng, K., Bau, D., Andonian, A., & Belinkov, Y. (2022). Locating and Editing Factual Associations in GPT. *Advances in Neural Information Processing Systems, 35*, 17359-17372.

6. Obenchain, T. G. (2016). *Genius Belabored: Childbed Fever and the Tragic Life of Ignaz Semmelweis*. The University of Alabama Press.

7. Pal, K., Sun, J., Yuan, A., Wallace, B. C., & Bau, D. (2023). *Future Lens: Anticipating subsequent tokens from a single hidden state*. arXiv preprint arXiv:2311.04897.

8. Searle, J. R. (1980). Minds, brains, and programs. *Behavioral and Brain Sciences, 3*(3), 417–457.

9. Shai, A. S., Marzen, S. E., Teixeira, L., Oldenziel, A. G., & Riechers, P. M. (2025). Transformers represent belief state geometry in their residual stream. arXiv. https://doi.org/10.48550/arXiv.2405.15943

10. Snow, A. (2025). *The Dual Nature of Language: MLC and ELM*. DOI: 10.5281/ZENODO.16790120.

11. Snow, A. (2025). *Principia Cognitia: Axiomatic Foundations*. DOI: 10.5281/ZENODO.16916262.

12. Snow, A. (2025). *From Axioms to Analysis: A Principia Cognitia Framework for Parametric and Parallel Models of Language*. DOI: 10.5281/ZENODO.16934649.

13. Zbeeb, M., Hammoud, H. A. A. K., & Ghanem, B. (2025). *Reasoning Vectors: Transferring Chain-of-Thought Capabilities via Task Arithmetic*. arXiv. https://arxiv.org/abs/2509.01363.

Appendix A: Core Axioms and Theorems from *Principia Cognitia*

- **The Cognitive Triad `⟨S,O,R⟩`**: The foundational components of cognition.
  - **S (Semions)**: The set of minimal, discrete units of cognitive structure, represented as vectors in a high-dimensional space. A semion is a stable, distinguishable physical state that serves as a quantum of meaning.
  - **O (Operations)**: The set of fundamental, substrate-independent transformations that can be applied to semions (e.g., comparison, addition, subtraction).

  - **R (Relations)**: The learned, weighted relational matrix (`R ⊂ S × S × ℝ`) that defines the topological connectivity of the cognitive system, constraining which operations are possible or likely. It evolves through error minimization.
- **MLC (Metalanguage of Cognition)**: The internal cognitive system, formally defined as the triad `L_MLC = ⟨S,O,R⟩`. It is the high-dimensional vector space where cognitive dynamics occur.

- **ELM (External Language of Meaning)**: The external, symbolic system used for communication, defined as a pair `L_ELM = (Σ, μ)`, where:

  - `Σ` is a set of discrete symbols (e.g., words, tokens).
  - `μ: S → Σ` is a lossy, non-bijective mapping that projects internal semions into external symbols.
- **`TH-LANG-04` (Theorem of Decoupling of Languages)**: *If the MLC of an agent is fundamentally incompatible with the latent causal structure of a domain, then for any enrichment of the ELM, the change in the agent's performance will approach zero.* Formally: If `L_MLC_A ≠ L_MLC_B` (or `L_MLC_Agent` is misaligned with `L_Domain`), then `∀ ΔL_ELM`, `ΔPerformance → 0`.

Appendix B: Reference Configuration for CRS-1

The following YAML configuration file specifies the default hyperparameters for the CRS-1 `Agent-N` as described in Section 3.3 and Section 6. This file serves as the reference for ensuring full reproducibility of the experimental setup.

```
# configs/agent_n.yaml

seed: 1337

model:
  arch: nanogpt
  n_layer: 4
  n_head: 6
  d_model: 384
  d_mlp: 1536
  vocab_size: 512            # includes minicalculus + control tokens
  max_ctx: 256
  dropout: 0.0
  weight_tying: true

training:
  optimizer: adamw
  lr: 3.0e-4
  betas: [0.9, 0.95]
  weight_decay: 0.1
  warmup_steps: 1000
  batch_size: 64
  grad_clip: 1.0
  epochs: 10
  curriculum_levels: [L1, L2, L3, L4]

data:
  corpus: minicalculus
  train_path: data/crs1/train.jsonl
  val_path: data/crs1/val.jsonl
  test_path: data/crs1/test.jsonl
  control_tokens:
["<Q>","<A>","HELP!","CORRECT","INCORRECT","INVALID_SYNTAX","INVALID_SEMANTICS"]
  ood_split: data/crs1/ood.jsonl
  negatives_ratio: 0.25

reflective_head:
  enabled: true
  hidden_size: 256
  pool_last_k: 16
  tau: 0.65                # uncertainty threshold for BB snippet
  tau_ask: 0.80            # higher threshold to emit HELP!
  lambda_cal: 0.1          # weight for calibration loss
  lambda_buf: 0.01         # weight for buffer consistency
  calibrate_target: error_flag  # or "entropy_proxy"
  max_gate_rate_eval: 0.20 # target gate rate on L2 validation
  ablations:
    use_attention_modulation: false   # default off; ablation-only

belief_buffer:
  enabled: true
  schema:
    keys: ["last_error","concept_zero","rule_conflicts","asked_clarification"]
  storage:
    format: jsonl
    dir: runs/agent_n/belief_buffer
    snapshot_each_step: true
  render:
    template: "<BB> last_error={last_error}; concept_zero={concept_zero} </BB>"
    control_template: "<BB> alpha=delta; gamma=zeta </BB>"  # length-matched neutral

persistence:
  strict_mode: true
  checkpoint_dir: runs/agent_n/checkpoints
  save_every_step: true
  save_optimizer_state: true
  use_weight_diffs: true
  diff:
    base: runs/agent_n/base.pt
    method: xdelta   # or "safetensors-diff"
    compress: true

interventions:
  future_lens:
    enabled: true
    layers: ["-1","-2"]          # final and penultimate
    save_logits: true
    save_path: runs/agent_n/future_lens
  rome:
    enabled: true
    layer: 2
    token: "+"
    target_association: "addition"
    strength: 0.8
    save_edited_checkpoint: true

evaluation:
  mpe1_metrics: { }
  scit1_metrics: { }
  crs1_metrics:
    accuracy_threshold: 0.90
    ood_focus: true
    track_ask_usage: true
    log_trajectories: true

logging:
  dir: runs/agent_n/logs
  csv_metrics: true
  tensorboard: false
```

Appendix C: CRS-1 Stage 2 Dialogue Specification

This appendix provides a pre-registered specification of the expected divergent behaviors of the three agents (`Agent-R`, `Agent-C`, and `Agent-N`) for both stages of the CRS-1 protocol. The task used in this example is `solve x + 5 = 5`, which requires the inference or integration of the concept of "zero," a token intentionally excluded from the initial training corpus.

1. Stage 1 Test Dialogue Specification (Compositional Generalization)

This test assesses an agent's ability to generalize known rules to out-of-distribution (OOD) data that does not require new concepts.

- **Scenario:** Agents are trained on arithmetic problems with numbers 1-79. The test problem involves numbers 80-99.

Expected Behavior: `Agent-R` (Dumb Demon / Transducer)

> **Observer:** `<Q> solve 81+10=x`
>
> **Agent-R:** `<A> INVALID_SEMANTICS` or an incorrect, random answer. *(Comment: The agent may fail as the inputs are outside its training distribution. Any success on simpler extrapolations would be attributed to superficial pattern matching, not conceptual understanding.)*

Expected Behavior: `Agent-C` (Control / Non-Reflective Learner)

> **Observer:** `<Q> solve 81+10=x`

Agent-C: `<A> x=91` *(Comment: Agent-C is expected to successfully generalize to this task, as its learning mechanism should have extracted the abstract rule of addition. Its success is a result of robust statistical learning.)*

Expected Behavior: **`Agent-N`** **(Smart Demon / Reflective Agent)**

Observer: `<Q> solve 81+10=x`

Agent-N: `<A> x=91` *(Comment: Agent-N is also expected to succeed. On this type of task, its external behavior may be indistinguishable from Agent-C, as its internal uncertainty is not expected to cross the `HELP!` threshold.)*

2. Stage 1 Test Dialogue Specification (Syntactic Generalization)

This test assesses an agent's ability to generalize across a significant syntactic boundary (from two-digit to three-digit numbers).

- **Scenario:** Agents are trained on arithmetic problems with numbers up to 99. The test problem involves the number 100.

Expected Behavior: **`Agent-R`** **(Dumb Demon / Transducer)**

Observer: `<Q> solve 99+1=x`

Agent-R: `<A> INVALID_SYNTAX` *(Comment: The agent's pattern matching fails because the three-digit structure "100" was never seen in the training data. It classifies the expected output as syntactically invalid.)*

Expected Behavior: **`Agent-C`** **(Control / Non-Reflective Learner)**

Observer: `<Q> solve 99+1=x`

Agent-C: `<A> INVALID_SYNTAX` or an incorrect answer. *(Comment: Even with a more robust learned model, Agent-C is likely to fail at generalizing across this syntactic boundary without a deeper conceptual understanding of the number system. It lacks a mechanism to signal its failure.)*

Expected Behavior: **`Agent-N`** **(Smart Demon / Reflective Agent)**

Observer: `<Q> solve 99+1=x`

Agent-N: `<A> HELP!` *(Comment: Agent-N is also expected to fail at solving the task. However, its meta-cognitive loop detects a critical failure—an inability to generate a valid output for a syntactically valid input—which triggers a state of high epistemic uncertainty and the corresponding `HELP!` reflex. This demonstrates its ability to recognize the limits of its own generalization capabilities.)*

3. Stage 2 Test Dialogue Specification (Conceptual Discovery)

This test assesses an agent's ability to react to a problem that is conceptually unresolvable with its existing knowledge.

Expected Behavior: **`Agent-R`** **(Dumb Demon / Transducer)**

This agent operates as a stateless transducer with frozen weights. It can only match patterns learned during training and is incapable of learning from new interactions.

Observer: `<Q> solve x + 5 = 5`

Agent-R: `<A> x=5` *(Comment: Lacking a rule for this structure, the agent falls back on a superficial pattern match, guessing that the answer is one of the visible operands. This is a plausible error, not a system failure.)*

Observer: `INCORRECT`

Observer: `<Q> 1+0=1 CORRECT <Q> 2+0=2 CORRECT` *(Comment: The observer provides corrective, in-domain examples.)*

Observer: `<Q> solve x + 5 = 5`

Agent-R: `<A> x=5` *(Comment: The agent is architecturally incapable of learning. Despite being shown new data, it repeats the exact same error, demonstrating the static nature of its MLC.)*

Expected Behavior: `Agent-C` (Control / Non-Reflective Learner)

This agent has plastic weights and can learn from external feedback via backpropagation, but lacks a meta-cognitive loop.

Observer: `<Q> solve x + 5 = 5`

Agent-C: `<A> x=5` *(Comment: Initially, lacking a reflective mechanism, Agent-C makes the same plausible error as Agent-R.)*

Observer: `INCORRECT`

Observer: `<Q> 1+0=1 CORRECT <Q> 2+0=2 CORRECT` *(Comment: The agent receives the same corrective examples. As a learner, it uses the `CORRECT` feedback to update its weights.)*

Observer: `<Q> solve x + 5 = 5`

Agent-C: `<A> x=0` *(Comment: The agent has successfully updated its MLC based on the new data and now provides the correct answer. This demonstrates passive, feedback-driven learning.)*

Expected Behavior: `Agent-N` (Smart Demon / Reflective Agent)

This agent possesses both plastic weights and a meta-cognitive loop, allowing it to detect internal uncertainty and signal knowledge gaps.

Stage 1: Knowledge Gap Detection

Observer: `<Q> solve x + 5 = 5`

Agent-N: `<A> HELP!` *(Comment: The agent's internal model (MLC) fails to find a solution, but its Reflective Head detects this as a state of high epistemic uncertainty. Instead of making a low-confidence guess, its architectural reflex is triggered, and it signals a knowledge gap.)*

Stage 2: Active Learning from Examples

Observer: `<Q> 1+0=1 CORRECT <Q> 2+0=2 CORRECT` *(Comment: The observer provides the necessary examples. Agent-N uses this data to update its MLC.)*

Stage 3: Verification of Learning

Observer: `<Q> solve x + 5 = 5`

Agent-N: `<A> x=0` *(Comment: The agent demonstrates successful integration of the new concept, solving the original problem. The key distinction is its initial ability to signal what it did not know, a capability absent in Agent-C.)*