Monitor-Generate-Verify (MGV): Formalising Metacognitive Theory for Language Model Reasoning

Nick Oh
socius labs
London, UK
nick.sh.oh@socius.org

Fernand Gobet
Centre for Philosophy of
Natural and Social Science (CPNSS)
London School of Economics
London, UK
f.gobet@lse.ac.uk

Abstract

Test-time reasoning architectures such as those following the *Generate-Verify* paradigm, where a model iteratively refines or verifies its own generated outputs, prioritise generation and verification but exclude the monitoring processes that determine when and how reasoning should begin. This omission may contribute to the prefix dominance trap, in which models commit early to suboptimal reasoning paths and seldom recover, yielding roughly 20% accuracy loss. We address this architectural gap by proposing the Monitor-Generate-Verify (MGV) framework, a computational translation of Flavell's and Nelson and Narens' metacognitive theories that preserves their psychological detail. MGV extends the Generate-Verify paradigm by adding explicit monitoring that captures metacognitive experiences (from difficulty assessments to confidence judgements) before generation begins and refines future monitoring through verification feedback. Though we present no empirical validation, MGV provides a vocabulary for diagnosing component-level failures in reasoning systems, suggests specific architectural interventions for future designs, and identifies connections to resource-rational analysis that may ground its mechanisms in normative principles.

1 Introduction

Once language models commit to an initial reasoning strategy, subsequent verification rarely helps; this *prefix dominance trap* causes nearly 20% performance degradation when models choose suboptimal approaches, with limited recovery possible through refinement [Luo et al., 2025]. Today's dominant Generate-Verify test-time reasoning architectures [Weng et al., 2023, Madaan et al., 2023, Lee et al., 2025, Zhang et al., 2024] exemplify this limitation through their very design. They operate through immediate generation followed by iterative refinement, without assessing task characteristics or selecting appropriate strategies before generating solutions. What architectural vocabulary might help diagnose what these systems lack?

Cognitive psychology offers a candidate answer. Before attempting complex tasks, humans engage in metacognitive monitoring, assessing difficulty, retrieving relevant strategies, and establishing confidence criteria, often without conscious deliberation [Flavell, 1979, Nelson and Narens, 1990]. This monitoring capacity transforms uncertainty into tractable signals that guide subsequent action. Flavell's model specifies how metacognitive knowledge and experience interact to enable strategy selection, while Nelson and Narens' metamemory framework details how confidence thresholds and adaptive search mechanisms emerge from hierarchical monitoring and control. These theories

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Foundations of Reasoning in Language Models (FoRLM).

suggest that the Generate-Verify paradigm may be missing an entire phase of processing that precedes generation.

However, translating these psychological insights into architectural recommendations requires computational specification. Resource-rational analysis has made significant progress here, with Callaway et al. [2024] formalising Nelson and Narens' metamemory theory as a meta-level MDP that derives optimal stopping policies from cost-benefit principles. While their approach offers normative grounding and empirical validation, the present work pursues a complementary goal: instead of deriving optimal behaviour from computational constraints, we translate the detailed psychological vocabulary of Flavell [1979] and Nelson and Narens [1990] into algorithmic form, preserving structure that resource-rational approaches may abstract away.

The resulting Monitor-Generate-Verify (MGV) framework extends the Generate-Verify paradigm by adding explicit monitoring that captures metacognitive experiences before generation begins and refines future monitoring through verification feedback. Our formalisation preserves constructs including the tripartite organisation of metacognitive knowledge, the distinction between difficulty and evaluative experiences, dual-counter evidence accumulation for feeling-of-knowing, and satisficing threshold dynamics. We offer no empirical validation and no normative justification for these architectural choices. The contribution is a detailed computational translation that provides diagnostic vocabulary for understanding what reasoning systems might lack, suggesting where architectural interventions might prove productive.

The remainder of this paper is organised as follows. Section 2 situates our contribution within resource-rational approaches to metacognition and LLMs. Section 3 presents algorithmic formalisations of Flavell's and Nelson and Narens' theories, translating psychological concepts into computational structures. Section 4 acknowledges limitations and outlines future directions.

2 Related Work

This section situates the present work within two related research programmes. The first is resource-rational analysis, a cognitive modelling paradigm that formalises metacognition as cost-benefit optimisation over computational resources. The second is recent work applying these principles to language models. We review each in turn before clarifying how the present work differs in its foundational commitments.

2.1 Resource-Rational Analysis of Metacognition

Resource-rational analysis is a cognitive modelling paradigm that reconceives bounded rationality not as a departure from optimality but as optimality under constraints [Griffiths et al., 2015, 2019, Lieder and Griffiths, 2020]. Making good decisions requires computation, yet computation itself is costly. This tension between the value of deliberation and its cost gives rise to a class of meta-problems: problems not about what to decide, but about how to decide. *How much should an agent think before acting*? *Which cognitive strategy should be deployed for a given problem*? *Which aspects of a decision path merit exploration, and which can be safely ignored*? These questions concern the allocation of computational resources, and answering them requires reasoning about reasoning itself.

Resource-rational analysis provides a framework for addressing such meta-problems by treating cognitive constraints as part of the optimisation problem rather than as departures from rationality. Under metalevel rationality, an agent is assessed not by the expected utility of actions taken, but by how well its cognitive algorithm trades off decision quality against deliberation costs. From this perspective, rationality concerns not only making good decisions but also employing efficient cognitive strategies. The framework reduces meta-reasoning[1] to a decision-theoretic problem: *for*

[1] Meta-reasoning should be distinguished from meta-learning. Meta-reasoning concerns how to allocate limited computational resources during cognition, deciding which computations to perform and when to stop deliberating. Meta-learning concerns how to structure the learning process itself, exploiting knowledge about the statistical structure of learning environments to make better use of limited data. Both address resource limitations, but at different levels: meta-reasoning optimises the use of time and cognitive effort within a task, whereas meta-learning optimises the acquisition of knowledge across tasks. For a treatment that situates both perspectives within the resource-rational framework, see Griffiths et al. [2019].

any computation the agent could perform, there exists a value of computation (VOC) that quantifies the expected improvement in decision quality minus the cost of deliberation.

This cognitive modelling paradigm provides a common structure that different research programmes have instantiated in distinct ways, each addressing a different facet of the meta-reasoning problem. Lieder and Griffiths [2017] modelled *strategy selection*: rather than asking whether to think more, this formulation asks how to think, treating the choice among cognitive strategies as a contextual bandit in which the agent learns to predict which mode of reasoning best suits each type of problem (Appendix 7.2). Lieder et al. [2018] modelled *cognitive control*: by casting control specification as a belief-MDP and introducing a learned approximation, this formulation explains how control exhibits plasticity, adapting through experience to select signals that optimally trade off effort against goal achievement (Appendix 7.4). Callaway et al. [2022] modelled *planning*: by embedding deliberation within a meta-level MDP, this formulation captures the sequential structure of mental simulation, recognising that the value of a computation depends not only on its immediate effect but also on the future computations it enables (Appendix 7.3). Callaway et al. [2024] modelled *memory recall*: by treating retrieval as evidence accumulation supervised by an optimal stopping process, this formulation reveals the adaptive function of feeling-of-knowing, providing the monitoring signal that enables efficient termination of searches unlikely to succeed while sustaining effort when retrieval remains probable (Appendix 7.5).

These four instantiations span a range of formal structures, from contextual bandits to belief-MDPs, and a range of cognitive phenomena, from high-level strategy choice to low-level memory retrieval. Yet they share a common theoretical commitment: the mind's computational processes are themselves subject to rational optimisation. Appendix 7 provides a detailed technical treatment of these frameworks, comparing their formal structures and identifying shared algorithmic motifs. Here, we turn to recent work that has begun extending these principles beyond cognitive modelling to the training of language models.

2.2 Rational Metareasoning for Language Models

While the research reviewed above models human cognition, recent work has begun applying the VOC framework directly to language model training. De Sabbata et al. [2024] introduced RaM (Rational Metareasoning), which trains LLMs to use chain-of-thought reasoning selectively based on task difficulty. Their approach defines the reward of a reasoning chain z as the difference between its utility and cost:

$$R_\pi(x, y, z) = U_\pi(z|x, y) - C(z),$$

where utility $U_\pi(z|x, y) = \log \pi_\theta(y|z, x) - \log \pi_\theta(y|x)$ quantifies the increase in likelihood of generating the correct answer y when the chain of thought z is included, and cost $C(z) = \gamma \cdot l(z)$ is proportional to the number of tokens. This formulation mirrors the equation, with reasoning tokens corresponding to intermediate computations and model outputs corresponding to actions. Models trained with this objective learn to generate shorter reasoning chains for easier problems and longer chains for harder problems, demonstrating that VOC-based training can induce adaptive computation in neural systems. This work provides empirical evidence that resource-rational principles can be operationalised in modern architectures, though it focuses on optimising existing reasoning capabilities rather than introducing new metacognitive components.

2.3 Relationship to the Present Work

Formalising metacognitive theory requires choosing what to treat as primitive and what to derive. Resource-rational analysis treats computational operations as primitive and derives metacognitive behaviour from cost-benefit optimisation. The present work makes the opposite choice, treating psychological constructs as primitive and translating them directly into algorithmic form. Put simply, instead of deriving that people think the way they do because it is efficient given limited cognitive resources, this work treats psychological ideas as the foundation and directly formalises them in computational terms.

This difference in approach leads to different contributions. Resource-rational analysis provides normative grounding, explaining why certain metacognitive policies are optimal given computational constraints. The present work provides psychological granularity, preserving detailed structure from the source theories that resource-rational models abstract away. Section 3 makes this concrete by formalising constructs including tripartite metacognitive knowledge, dual-counter evidence accumulation, and satisficing threshold dynamics. Section 4 returns to the relationship between these approaches and examines how they might complement one another.

3 Monitor-Generate-Verify (MGV)

Flavell [1979] and Nelson and Narens [1990] developed seminal theories of how metacognition coordinates cognitive processes through monitoring and control loops. These frameworks, though developed for human cognition, offer potential blueprints for computational systems. Flavell's model provides a dynamic architecture where metacognitive knowledge and experience guide strategy selection and verification, while Nelson and Narens' metamemory framework specifies how confidence thresholds and adaptive search mechanisms emerge from hierarchical monitoring and control. By computationally formalising these psychological theories, we establish Monitor-Generate-Verify (MGV) as a theoretical framework for understanding how explicit metacognitive mechanisms could address the architectural limitations of current reasoning systems. The following subsections present detailed formalisations that translate these cognitive psychology insights into algorithmic structures, revealing both what current architectures lack and how metacognitive principles might be operationalised computationally.

3.1 Flavell's Model of Metacognition

Flavell [1979] conceptualises metacognition as a dynamic control architecture comprising four interacting components: *metacognitive knowledge*, *metacognitive experience*, *goals* (or tasks), and *actions* (or strategies). Rather than operating as independent modules, these components form an integrated system characterised by continuous bidirectional influences, positioning metacognition as a self-regulating system capable of adaptive control over cognitive processes. We present the core computational structure below, with a complete mathematical formalisation provided in Appendix 5.

3.1.1 Cognitive Monitoring

The regulation process begins with initialisation, where task $\mathcal{T}$ and goal $\mathcal{G}$ establish the initial state $\mathcal{S}_0 = f(\mathcal{T}, \mathcal{G})$. While Flavell [1979] treats goals and tasks as equivalent, we maintain a computational distinction. $\mathcal{T}$ represents the cognitive enterprise while $\mathcal{G}$ specifies success criteria, enabling clearer analysis of metacognitive processes.

Algorithm 1 Flavell's Model of Cognitive Control

```
Initialise: $\mathcal{S}_0 \leftarrow f(\mathcal{T}, \mathcal{G})$; $\tau \leftarrow 0$
while $\mathcal{S}_\tau = $ ACTIVE do
    // MONITOR: Retrieve knowledge & assess experience
    $\mathcal{MK}_\tau \leftarrow$ if $\tau = 0$ then retrieve($\mathcal{MK}, \mathcal{T}, \mathcal{G}$)
        else $\mathcal{MK}_{\tau-1} \cup$ retrieve($\mathcal{MK}, \mathcal{ME}_{\tau-1}$)
    $\mathcal{ME}^\tau_{\text{difficulty}} \leftarrow$ feel($\mathcal{T}$, Outcomes$_{\tau-1}$) $\oplus$ assess($\mathcal{T}$, $\mathcal{MK}_\tau$)
    // GENERATE: Select & execute cognitive strategy
    $\mathcal{CS}_\tau \leftarrow$ select($s \in \mathcal{MK}_{\text{Strategy}} \mid \mathcal{ME}^\tau_{\text{difficulty}}, \mathcal{MK}_\tau, \mathcal{T}, \mathcal{G}$)
    $\mathcal{CO}_\tau \leftarrow$ execute($\mathcal{CS}_\tau, \mathcal{T}, \mathcal{G}$)
    // VERIFY: Evaluate progress & update knowledge
    $\mathcal{ME}^\tau_{\text{evaluative}} \leftarrow$ assess($\mathcal{CO}_\tau, \mathcal{MK}_\tau$)
    $\mathcal{MS}_\tau \leftarrow$ select($s \in \mathcal{MK}^{\text{meta}}_{\text{Strategy}} \mid \mathcal{ME}^\tau_{\text{evaluative}}$)
    $\mathcal{MO}_\tau \leftarrow$ execute($\mathcal{MS}_\tau, \mathcal{CO}_\tau, \mathcal{MK}_\tau, \mathcal{G}$)
    $\mathcal{MK} \leftarrow$ update($\mathcal{MK}$, $\Phi_\tau$) where $\Phi_\tau = (\mathcal{ME}_\tau, \text{Strategy}_\tau, \text{Outcome}_\tau)$
    $\mathcal{S}_{\tau+1} \leftarrow$ if goal_achieved($\mathcal{CO}_\tau, \mathcal{G}$) then TERMINATE else ACTIVE
    $\tau \leftarrow \tau + 1$
end while
```

The **monitoring** phase activates metacognitive knowledge differently across cycles. Initial cycles rely solely on task-goal combinations, while subsequent cycles incorporate emerging metacognitive experiences from $\tau - 1$ that trigger additional relevant knowledge. According to Flavell [1979], this knowledge comprises three categories: *agent variables* ($\mathcal{MK}_{\text{Agent}}$) representing learned self-models of performance patterns and processing preferences; *task variables* ($\mathcal{MK}_{\text{Task}}$) capturing knowledge about cognitive situation assessment including information characteristics and task demands; and *strategy variables* ($\mathcal{MK}_{\text{Strategy}}$) encompassing knowledge about the effectiveness of both cognitive strategies (problem-solving procedures) and metacognitive strategies (monitoring and regulation processes). These categories function as an integrated system where task variables diagnose cognitive demands, strategy variables prescribe responses, and agent variables contextualise both within the agent's capabilities.

The monitoring phase also generates metacognitive experiences of difficulty ($\mathcal{ME}^{\tau}_{\text{difficulty}}$), which Flavell [1979, p. 909] describes as subjective feeling-of-complexity, comprehension challenges, or sensing that material exceeds current capabilities. These experiences evolve through iterative assessments, progressing from initial coarse feelings to increasingly nuanced evaluations of specific challenge sources.

During the **generation** phase, metacognitive experiences function as computational signals that require interpretation through metacognitive knowledge to guide strategy selection. The process follows a two-phase pattern. First, $\mathcal{MK}_{\text{Strategy}}$ transforms general difficulty *signals* into precise diagnostic patterns (e.g., "content uncertainty with unknown terms" or "procedural confusion from missing steps"). Second, these refined patterns activate corresponding cognitive strategies. The selected strategy $\mathcal{CS}_{\tau}$ is then executed to produce cognitive outcomes $\mathcal{CO}_{\tau}$, generating feedback that provides both task progress information and context for subsequent monitoring.

The **verification** phase evaluates these outcomes, triggering what Flavell [1979, p. 909] describes as additional metacognitive experiences about performance rather than difficulty. These evaluative experiences ($\mathcal{ME}^{\tau}_{\text{evaluative}}$) activate metacognitive strategies that assess whether outcomes form a coherent whole, appear plausible and consistent with prior knowledge, and provide an avenue to the goal. The specific metacognitive strategy $\mathcal{MS}_{\tau}$ selected depends on the nature of the evaluative signal: uncertainty about validity triggers plausibility checking, sensing incompleteness activates coherence assessment, and so forth. Notably, these experiences can add to, delete from, or revise the metacognitive knowledge base through Piagetian mechanisms [Flavell, 1963], with the complete experience tuple Φ_{τ} updating $\mathcal{MK}$ for future cycles.

3.1.2 Memory and Learning Gaps

A significant limitation in Flavell's model is the absence of explicit working memory mechanisms for storing information across monitoring cycles. The model does not specify where $\mathcal{ME}^{\tau}_{\text{difficulty}}$ resides during strategy execution, how $\mathcal{CO}_{\tau}$ is maintained during evaluation, or how experience patterns across cycles are retained for subsequent processing. This absence precludes sophisticated termination criteria that would require access to historical monitoring data across the complete sequence $\Phi = (\Phi_0, \Phi_1, \ldots, \Phi_T)$.

With an explicit memory component, the model could implement comprehensive abandonment criteria that evaluate: (1) repeated strategy failures indicated by consistently negative $\mathcal{MO}_{\tau}$ across multiple cycles, suggesting task intractability; (2) resource constraints where cumulative effort across Φ_0 to Φ_{τ} exceeds acceptable limits relative to $\mathcal{MK}_{\text{Agent}}$; (3) goal displacement where evolving $\mathcal{ME}^{\tau}_{\text{evaluative}}$ signals that alternative objectives have become more salient than the original $\mathcal{G}$; and (4) insurmountable goal-state discrepancy where the pattern of $\mathcal{CO}_{\tau}$ outcomes reveals fundamental incompatibility with $\mathcal{G}$ achievement.

A related temporal limitation concerns metacognitive knowledge acquisition and refinement. While Flavell acknowledges that experiences can 'add to', 'delete from', or 'revise' the knowledge base, the model assumes pre-existing $\mathcal{MK}$ without specifying learning mechanisms – how unsuccessful strategies refine strategy knowledge, or how repeated encounters improve task assessments.

Such memory-dependent termination decisions and learning-dependent knowledge refinement would better reflect real-world metacognitive monitoring, where individuals track cumulative progress patterns and recognise when persistence becomes counterproductive, while simultaneously refining their metacognitive knowledge through experience. These limitations point towards the necessity for more sophisticated architectural frameworks that explicitly model the temporal dynamics of metacognitive information storage and retrieval as well as the acquisition and refinement of metacognitive knowledge – considerations that become central to Nelson and Narens' metamemory architecture.

3.2 Nelson and Narens' Model of Metamemory

Nelson and Narens [1990] establish metacognition as fundamentally hierarchical, distinguishing between *object-level* and *meta-level* processes. The *object-level* comprises the cognitive operations themselves, such as searching memory for a target item. The *meta-level* supervises these operations, by monitoring progress toward the goal (e.g., how close is recall?) and controlling how long the process is allowed to continue (e.g., should the search persist or terminate?). This supervision relies on two distinct information flows: monitoring conveys information upward from object-level to meta-level, while control conveys decisions downward from meta-level to object-level. These relationships are logically independent and asymmetric: the meta-level maintains a model of the object-level, while the object-level operates without corresponding meta-level representation. We present the core computational structure below, with complete mathematical formalisation provided in Appendix 6.

3.2.1 Acquisition Process

The acquisition process begins with establishing the *norm of study* $\mathcal{N}_s = \rho^* \times (1 + \delta_{\text{retention}})$, where ρ^* represents target performance and $\delta_{\text{retention}}$ captures beliefs about memory decay over interval τ_{delay}. This operationalises abstract goals into quantified mastery criteria that anticipate forgetting. Following Ericsson and Simon [1984], monitoring occurs within working memory (STM), with information from long-term memory (LTM) accessed probabilistically via $\text{retrieve}_\theta(\cdot)$ where θ represents access probability [Atkinson and Shiffrin, 1968].

Algorithm 2 Nelson and Narens' Model of Acquisition

```
**Initialise:** $\mathcal{MK}_0^{\text{STM}} \leftarrow \text{retrieve}_\theta(\mathcal{MK}, \mathcal{T}, \mathcal{G})$
$\mathcal{N}_s \leftarrow \rho^* \times (1+ \text{formulate}(\mathcal{MK}_0^{\text{STM}}, \tau_{\text{delay}}, \mathcal{T}, \mathcal{G}))$
$\mathcal{J}_0 \leftarrow \{1, \ldots, N\}; \tau \leftarrow 1; \Phi_0^{\text{STM}} \leftarrow \emptyset$
**while** $\mathcal{J}_\tau \neq \emptyset$ **do**
    **// MONITOR: Assess mastery via EOL/FOK**
    $\mathcal{MK}_\tau^{\text{STM}} \leftarrow \mathcal{MK}_{\tau-1}^{\text{STM}} \cup \text{retrieve}_\theta(\mathcal{MK}, \mathcal{ME}_{\tau-1})$
    **for** each $j \in \mathcal{J}_\tau$ **do**
        $\mathcal{ME}_{\tau,j}[1] \leftarrow$ **if** $\tau = 1$ **then** $\text{EOL}(i_j)$ **else** $\text{FOK}(i_j, \mathcal{CO}_{\tau-1,j})$
    **end for**
    **// GENERATE: Allocate resources & select strategies**
    **for** each $j \in \mathcal{J}_\tau$ **do**
        $r_{\tau,j} \leftarrow R_{\text{total}} \times (\mathcal{ME}_{\tau,j}[1])^{-1} / \sum_k (\mathcal{ME}_{\tau,k}[1])^{-1}$
        $\sigma_{\tau,j} \leftarrow \text{select}(s \in \mathcal{MK}_{\text{Strategy}} \mid i_j, r_{\tau,j}, \mathcal{ME}_{\tau,j})$
        $\mathcal{CO}_{\tau,j} \leftarrow \text{execute}(i_j, r_{\tau,j}, \sigma_{\tau,j})$
    **end for**
    **// VERIFY: Judge learning & update items**
    **for** each $j \in \mathcal{J}_\tau$ **do**
        $\text{JOL}_{\tau,j} \leftarrow \text{feel}(i_j, \mathcal{CO}_{\tau,j}) \oplus \text{assess}(i_j, \mathcal{CO}_{\tau,j}, \mathcal{MK}_\tau^{\text{STM}})$
        $\mathcal{ME}_{\tau,j}[2] \leftarrow \text{JOL}_{\tau,j}$
        $\Phi_\tau^{\text{STM}} \leftarrow \Phi_\tau^{\text{STM}} \cup \{(\mathcal{ME}_{\tau,j}, i_j, r_{\tau,j}, \sigma_{\tau,j}, \mathcal{CO}_{\tau,j})\}$
    **end for**
    $\mathcal{J}_{\tau+1} \leftarrow \{j \in \mathcal{J}_\tau : \mathcal{N}_s - \text{JOL}_{\tau,j} > 0\}; \tau \leftarrow \tau + 1$
**end while**
$\mathcal{MK} \leftarrow \text{consolidate}_\psi(\mathcal{MK}, \Phi_\tau^{\text{STM}})$    ▷ Experience to LTM
```

The **monitoring** phase generates metacognitive experiences as multidimensional vectors. Ease-of-learning (EOL) provides initial difficulty assessment, while feeling-of-knowing (FOK) incorporates prior outcomes to refine mastery judgements. These phenomenological experiences serve as primary input for control decisions [Nelson and Narens, 1990, p. 160]. During the **generation** phase, resource allocation operates inversely to EOL/FOK values – items with lower metacognitive confidence receive proportionally more resources $r_{\tau,j} = R_{\text{total}} \times w_j / \sum_k w_k$ where $w_j = (\mathcal{ME}_{\tau,j}[1])^{-1}$. Strategy selection integrates these metacognitive inputs to map appropriate learning methods to individual items.

The **verification** phase employs judgement-of-learning (JOL) to evaluate mastery following cognitive outcomes. Items achieving the norm of study ($\text{JOL}_{\tau,j} \geq \mathcal{N}_s$) are removed from further consideration, while those below threshold remain in $\mathcal{J}_{\tau+1}$ for continued learning. The complete experience tuple accumulates in working memory as Φ_τ^{STM}, subsequently undergoing consolidation to LTM at encoding rate ψ.

3.2.2 Retrieval Process

The retrieval process implements Nelson and Narens' dual-counter FOK hypothesis, where FOK^+ accumulates evidence for information presence whilst FOK^- accumulates evidence for absence, consistent with 'knowing not' [Kolers and Palef, 1976]. Initial thresholds are personalised through metacognitive calibration history: $\lambda_{\text{FOK}}^{(0)} = \text{median}(\{||\text{FOK}|| : \text{successful retrievals in } \mathcal{MK}_0^{\text{STM}}\})$ and $\lambda_{\text{confidence}}^{(0)} = \text{median}(\{\text{confidence} : \text{correct outputs in } \mathcal{MK}_0^{\text{STM}}\})$, embodying the No-Magic Hypothesis by utilising recallable metacognitive knowledge.

Algorithm 3 Nelson and Narens Model of Retrieval

```
**Initialise:** $\mathcal{MK}_0^{\text{STM}} \leftarrow \text{retrieve}_\theta(\mathcal{MK}, \mathcal{Q})$
$\lambda_{\text{FOK}}^{(0)}, \lambda_{\text{confidence}}^{(0)} \leftarrow \text{calibrate}(\mathcal{MK}_0^{\text{STM}})$; $\tau \leftarrow 0$; $\Omega_0^{\text{STM}} \leftarrow \emptyset$
**while** search active **do**
    **// MONITOR: Assess dual-counter FOK**
    $\mathcal{MK}_\tau^{\text{STM}} \leftarrow \mathcal{MK}_{\tau-1}^{\text{STM}} \cup \text{retrieve}_\theta(\mathcal{MK}, \text{FOK}_{\tau-1})$ if $\tau > 0$
    $[\text{FOK}_\tau^+, \text{FOK}_\tau^-] \leftarrow \text{feel}(\mathcal{Q}, \mathcal{A}_{\tau-1}) \oplus \text{assess}(\mathcal{Q}, \mathcal{A}_{\tau-1}, \mathcal{MK}_\tau^{\text{STM}})$
    **// Determine search intensity based on FOK evidence**
    **if** $||\text{FOK}_\tau|| < \lambda_{\text{FOK}}^{(\tau)}$ **then**
        $\mathcal{S}_\tau \leftarrow \text{ACTIVE}_{\text{intensive}}$                    ▷ Insufficient evidence
    **else if** $\text{FOK}_\tau^+ > \text{FOK}_\tau^-$ **then**
        $\mathcal{S}_\tau \leftarrow \text{ACTIVE}_{\text{standard}}$                     ▷ Positive dominance
    **else**
        **break**                                          ▷ Negative dominance: terminate
    **end if**
    **// GENERATE: Attend to cues & automatic search**
    $\text{cue}_\tau \leftarrow \text{attend}_{[\text{intensive/standard}]}(\mathcal{Q}, \mathcal{MK}_\tau^{\text{STM}})$ based on $\mathcal{S}_\tau$
    $\mathcal{A}_\tau \leftarrow \text{search}_{\text{auto}}(\text{cue}_\tau)$                 ▷ Automatic pattern recognition
    **// VERIFY: Evaluate answer & adjust thresholds**
    $\text{confidence}_\tau \leftarrow \text{assess}(\mathcal{A}_\tau, \mathcal{Q}, \mathcal{MK}_\tau^{\text{STM}})$ if $\mathcal{A}_\tau \neq$ null
    **if** $\mathcal{A}_\tau \neq$ null $\wedge$ $\text{confidence}_\tau \geq \lambda_{\text{confidence}}^{(\tau)}$ **then**
        **output** $\mathcal{A}_\tau$; **break**
    **else if** $\mathcal{A}_\tau =$ null $\wedge \text{FOK}_\tau^- > \text{FOK}_\tau^+$ **then**
        **output** null; **break**                                          ▷ Omission
    **end if**
    $\Omega_\tau^{\text{STM}} \leftarrow \Omega_\tau^{\text{STM}} \cup \{(\text{FOK}_\tau, \text{cue}_\tau, \mathcal{A}_\tau, \text{confidence}_\tau)\}$
    $\beta_\tau \leftarrow \exp(-\alpha \cdot (\tau + |\{\text{failed attempts in } \Omega_\tau^{\text{STM}}\}|))$
    $\lambda_{\text{confidence}}^{(\tau+1)}, \lambda_{\text{FOK}}^{(\tau+1)} \leftarrow \lambda^{(0)} \cdot \beta_\tau$                          ▷ Satisficing
    $\tau \leftarrow \tau + 1$
**end while**
$\mathcal{MK} \leftarrow \text{consolidate}_\psi(\mathcal{MK}, \Omega_\tau^{\text{STM}})$                        ▷ Experience to LTM
```

The **monitoring** phase employs rapid FOK assessment that operates faster than actual recall [Reder, 1987], enabling efficient search control. When FOK magnitude falls below threshold ($||\text{FOK}_\tau|| < \lambda_{\text{FOK}}^{(\tau)}$), insufficient evidence triggers intensive cue attention to gather additional metacognitive information. With sufficient evidence, positive dominance ($\text{FOK}_\tau^+ > \text{FOK}_\tau^-$) warrants continued search, while negative dominance justifies termination.

The **generation** phase reflects Nelson and Narens' insight that search execution is automatic once initiated – *conscious control operates through cue attention intensity rather than strategy selection.* The automatic search process $\text{search}_{\text{auto}}(\text{cue}_\tau)$ operates through pattern recognition, potentially yielding identical results across consecutive cycles due to its deterministic nature.

Verification distinguishes two error pathways: commission errors (outputting incorrect answers with high confidence) and omission errors (terminating without answers following prolonged search). Following satisficing principles [Simon, 1979], both confidence and FOK thresholds undergo dynamic adjustment: $\lambda^{(\tau+1)} = \lambda^{(0)} \cdot \beta_\tau$ where $\beta_\tau = \exp(-\alpha \cdot \text{burden})$ captures accumulating search costs. This ensures previously inadequate answers may become acceptable as search burden increases, preventing exhaustive search behaviour.

3.2.3 Memory Consolidation and Knowledge Evolution

A distinctive strength of Nelson and Narens' framework lies in its explicit treatment of long-term memory (LTM) as both a repository and an evolving knowledge base. During acquisition and retrieval, the experience tuples accumulated in working memory (Ω_T^{STM}) undergoes consolidation into LTM at encoding rate ψ:

While Nelson and Narens do not explicitly specify the timing of this consolidation process, it likely occurs during the verification stage at rate ψ, potentially operating below conscious awareness. This consolidation mechanism enables the global metacognitive knowledge base to evolve through accumulated experience, distinguishing Nelson and Narens' approach from more static metacognitive frameworks. The probabilistic retrieval function $\text{retrieve}_\theta(\mathcal{MK}, \cdot)$ subsequently accesses this enriched knowledge base, creating a dynamic feedback loop where metacognitive experiences inform future metacognitive assessments.

4 Limitations and Future Work

4.1 From Specification to Implementation

The MGV framework specifies what should be computed but not how to compute it in neural architectures. Operationalising constructs like $\mathcal{ME}_{\text{difficulty}}$ or dual-counter FOK requires identifying appropriate neural correlates or designing explicit computational mechanisms. For language models, this might involve obtaining logit-based confidence ratings [Kumaran et al., 2025], using entropy-based proxies for metacognitive experiences, or implementing explicit evidence accumulators. Recent work suggests that LLMs access metacognitive signals they cannot adequately articulate, with implicit confidence measures outperforming explicit verbalisation [Wang et al., 2025, Tian et al., 2023, Xiong et al., 2023, Cash et al., 2024, Griot et al., 2025, Lindsey et al., 2025]. Neural analysis reveals a lower-dimensional "metacognitive space" where monitoring signals correspond to linearly separable directions [Ji-An et al., 2025, Zou et al., 2023, Liu et al., 2023], consistent with cognitive findings that metacognition operates on abstracted representations [Reder, 1987].

4.2 Scope of Formalisation

The present work formalises two foundational theories, but metacognition research extends well beyond Flavell and Nelson and Narens. A more complete computational account of metacognition should integrate subsequent theoretical developments. Additionally, our formalisations make interpretive choices where the source theories are ambiguous or silent. For instance, the exponential decay of satisficing thresholds reflects our reading of the theories rather than explicit specifications in the original texts. Alternative formalisations are possible and might yield different architectural implications.

4.3 Normative Grounding

The framework lacks principled justification for its architectural choices. Resource-rational analysis explains why certain metacognitive policies are optimal given computational constraints. MGV, by contrast, specifies mechanisms without explaining why those mechanisms are appropriate. As Callaway et al. [2024, pp. 17–18] noted regarding related work in metamemory, this paper shares the limits of Metcalfe [1993], Koriat and Goldsmith [1996], and Bennett et al. [2017] in focusing on functional aspects without taking an explicitly optimal approach. That is, while resource-rational analysis could potentially explain MGV-like mechanisms as approximations to optimal behaviour, MGV cannot explain resource-rational optima as emergent from its architecture.

Yet this limitation is a consequence of methodological choice rather than oversight. We prioritised preserving psychological detail over deriving mechanisms from first principles. The result is a framework that describes what metacognitive components *might* look like but cannot explain why they should take those forms rather than others. Whether dual-counter FOK approximates optimal stopping behaviour, whether satisficing thresholds should decay exponentially rather than linearly, whether tripartite knowledge organisation is functionally necessary: these questions cannot be answered within MGV but could potentially be addressed through resource-rational analysis. The grounding of MGV in resource-rational principles represents one path toward resolving this limitation.

4.4 Future Directions

Grounding MGV in Resource-Rational Principles The most productive path forward may involve selective integration of the two approaches. Some MGV constructs appear directly amenable to resource-rational analysis. The stopping dynamics formalised in Nelson and Narens' retrieval model parallel the optimal stopping structure of memory recall in the resource-rational framework, where feeling-of-knowing serves as the monitoring signal that enables efficient termination of searches unlikely to succeed [Callaway et al., 2024]. Flavell's cognitive monitoring, which involves inferring the state of ongoing cognitive processes from imperfect signals, shares the partial observability structure of the belief-MDP formulation of cognitive control [Lieder et al., 2018]. The threshold dynamics that govern when to cease retrieval or when to shift strategies may be derivable from VOC calculations rather than stipulated as free parameters.

Other constructs may serve primarily descriptive functions that complement rather than compete with normative analysis. The tripartite knowledge taxonomy (person, task, and strategy knowledge) provides a vocabulary for the features over which resource-rational agents must learn, but the taxonomy itself is not derived from optimality principles. Similarly, the phenomenological vocabulary of metacognitive experiences (feeling-of-knowing, judgment-of-learning, feeling-of-confidence) names the signals that enable metacognitive control, but resource-rational analysis explains why such signals are useful rather than what they feel like. A complete theory may require both the normative grounding that resource-rational analysis provides and the psychological granularity that MGV preserves.

Meta-Reasoning and Meta-Learning in Language Models. Griffiths et al. [2019] identify meta-reasoning and meta-learning as two components of human intelligence that current AI systems lack: meta-reasoning concerns efficient allocation of computational resources within a task, while meta-learning concerns efficient use of data across tasks to accelerate future learning. Resource-rational analysis provides formal frameworks for both (see Appendix 7). MGV formalises both components from a psychological rather than normative starting point. For meta-reasoning, the models derived from Flavell's cognitive monitoring and Nelson and Narens' retrieval address how to allocate effort during task performance, specifying when the Monitor should trigger the Generator to produce new reasoning and when the Verifier should accept or reject candidate outputs. For meta-learning, the model derived from Nelson and Narens' acquisition addresses how to distribute study across items for future retrieval, deciding which items merit additional encoding effort given current memory states.

Recent work has begun applying rational metareasoning principles to language models. De Sabbata et al. [2024] developed RaM, a VOC-inspired reward function that balances the utility of chain-of-thought reasoning against its token cost. With RaM, models learn to adapt reasoning length to task difficulty, using substantially fewer tokens on easier problems. This demonstrates that VOC-based training can induce adaptive computation, but it addresses only one dimension of meta-reasoning: **how long to** `<think>`. The resource-rational frameworks reviewed in Appendix 7 reveal additional dimensions, including **what to** `<think>` about (as in planning, where the agent selects which nodes to expand), **how to** `<think>` (as in strategy selection, where the agent chooses among qualitatively different reasoning modes), and **how intensely to** `<think>` (as in cognitive control, where the agent modulates the strength of control signals). MGV provides architectural mechanisms for these decisions. A natural direction for future work is to ground these mechanisms in resource-rational principles, deriving MGV's control dynamics from VOC optimisation rather than stipulating them directly.

References

Richard C Atkinson and Richard M Shiffrin. Human memory: A proposed system and its control processes. In *Psychology of learning and motivation*, volume 2, pages 89–195. Elsevier, 1968.

Stephen T Bennett, Aaron S Benjamin, and Mark Steyvers. A bayesian model of knowledge and metacognitive control: Applications to opt-in tasks. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 39, 2017.

Frederick Callaway, Bas van Opheusden, Sayan Gul, Priyam Das, Paul M Krueger, Thomas L Griffiths, and Falk Lieder. Rational use of cognitive resources in human planning. *Nature human behaviour*, 6(8):1112–1125, 2022.

Frederick Callaway, Thomas L Griffiths, Kenneth A Norman, and Qiong Zhang. Optimal metacognitive control of memory recall. *Psychological Review*, 131(3):781, 2024.

Trent N Cash, Daniel M Oppenheimer, Sara Christie, and Mira Devgan. Quantifying uncertainty: Testing the accuracy of llms' confidence judgments. *PsyArXiv 47df5_v3*, 10, 2024.

C Nicolò De Sabbata, Theodore R Sumers, Badr AlKhamissi, Antoine Bosselut, and Thomas L Griffiths. Rational metareasoning for large language models. *arXiv preprint arXiv:2410.05563*, 2024.

K. Anders Ericsson and Herbert A. Simon. *Protocol analysis: Verbal reports as data*. The MIT Press, 1984.

John H. Flavell. *The developmental psychology of Jean Piaget*. D Van Nostrand, 1963. doi: 10.1037/11449-000.

John H Flavell. Metacognition and cognitive monitoring: A new area of cognitive–developmental inquiry. *American Psychologist*, 34(10):906, 1979.

Thomas L Griffiths, Falk Lieder, and Noah D Goodman. Rational use of cognitive resources: Levels of analysis between the computational and the algorithmic. *Topics in cognitive science*, 7(2): 217–229, 2015.

Thomas L Griffiths, Frederick Callaway, Michael B Chang, Erin Grant, Paul M Krueger, and Falk Lieder. Doing more with less: meta-reasoning and meta-learning in humans and machines. *Current Opinion in Behavioral Sciences*, 29:24–30, 2019.

Maxime Griot, Coralie Hemptinne, Jean Vanderdonckt, and Demet Yuksel. Large language models lack essential metacognition for reliable medical reasoning. *Nature communications*, 16(1):642, 2025.

Li Ji-An, Hua-Dong Xiong, Robert C Wilson, Marcelo G Mattar, and Marcus K Benna. Language models are capable of metacognitive monitoring and control of their internal activations. *arXiv preprint arXiv:2505.13763*, 2025.

P. A. Kolers and S. R. Palef. Knowing not. *Memory & Cognition*, 4(5):553–558, 1976. doi: 10.3758/BF03213218.

Asher Koriat and Morris Goldsmith. Monitoring and control processes in the strategic regulation of memory accuracy. *Psychological review*, 103(3):490, 1996.

Dharshan Kumaran, Stephen M Fleming, Larisa Markeeva, Joe Heyward, Andrea Banino, Mrinal Mathur, Razvan Pascanu, Simon Osindero, Benedetto De Martino, Petar Velickovic, et al. How overconfidence in initial choices and underconfidence under criticism modulate change of mind in large language models. *arXiv preprint arXiv:2507.03120*, 2025.

Hyunseok Lee, Seunghyuk Oh, Jaehyung Kim, Jinwoo Shin, and Jihoon Tack. Revise: Learning to refine at test-time via intrinsic self-verification. *arXiv preprint arXiv:2502.14565*, 2025.

Falk Lieder and Thomas L Griffiths. Strategy selection as rational metareasoning. *Psychological review*, 124(6):762, 2017.

Falk Lieder and Thomas L Griffiths. Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources. *Behavioral and brain sciences*, 43:e1, 2020.

Falk Lieder, Amitai Shenhav, Sebastian Musslick, and Thomas L Griffiths. Rational metareasoning and the plasticity of cognitive control. *PLoS computational biology*, 14(4):e1006043, 2018.

Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, et al. On the biology of a large language model. *Transformer Circuits Thread*, 2025.

Wenhao Liu, Xiaohua Wang, Muling Wu, Tianlong Li, Changze Lv, Zixuan Ling, Jianhao Zhu, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang. Aligning large language models with human preferences through representation engineering. *arXiv preprint arXiv:2312.15997*, 2023.

Tongxu Luo, Wenyu Du, Jiaxi Bi, Stephen Chung, Zhengyang Tang, Hao Yang, Min Zhang, and Benyou Wang. Learning from peers in reasoning models. *arXiv preprint arXiv:2505.07787*, 2025.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594, 2023.

Janet Metcalfe. Novelty monitoring, metacognition, and control in a composite holographic associative recall model: implications for korsakoff amnesia. *Psychological review*, 100(1):3, 1993.

Thomas O Nelson and Louis Narens. Metamemory: A theoretical framework and new findings. In *Psychology of learning and motivation*, volume 26, pages 125–173. Elsevier, 1990.

Lynne M. Reder. Strategy selection in question answering. *Cognitive Psychology*, 19(1):90–138, 1987.

Lynne M. Reder. Strategic control of retrieval strategies. In *Psychology of Learning and Motivation*, volume 22, pages 227–259. Academic Press, 1988.

Herbert A. Simon. *Models of Thought*, volume 1. Yale University Press, 1979.

Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *arXiv preprint arXiv:2305.14975*, 2023.

Guoqing Wang, Wen Wu, Guangze Ye, Zhenxiao Cheng, Xi Chen, and Hong Zheng. Decoupling metacognition from cognition: A framework for quantifying metacognitive ability in llms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25353–25361, 2025.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. Large language models are better reasoners with self-verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2550–2575, 2023.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*, 2023.

Yunxiang Zhang, Muhammad Khalifa, Lajanugen Logeswaran, Jaekyeom Kim, Moontae Lee, Honglak Lee, and Lu Wang. Small language models need strong verifiers to self-correct reasoning. *arXiv preprint arXiv:2404.17140*, 2024.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

5 Flavell's Model of Cognitive Monitoring

Flavell [1979] describes metacognition as a dynamic control architecture comprising four interacting components: *metacognitive knowledge*, *metacognitive experience*, *goals* (or tasks), and *actions* (or strategies). Rather than viewing metacognition as merely stored knowledge about cognition, Flavell presents it as a dynamic control system. This system operates through continuous interactions between four elements: what agents know about their cognitive capabilities (metacognitive knowledge), what they feel about their current cognitive state (metacognitive experiences), what they aim to achieve (goals), and how they control their thinking (strategies). Central to Flavell's model is the principle of reciprocal interaction amongst these components. Rather than operating as independent modules, they form an integrated system characterised by continuous bidirectional influences: metacognitive knowledge guides both strategy selection and the interpretation of ongoing cognitive experiences; these conscious experiences, in turn, update the knowledge base and prompt strategic adjustments; task-goals determine which aspects of metacognitive knowledge become most salient; and the outcomes of chosen actions provide feedback that shapes both immediate metacognitive experiences and longer-term understanding of effective cognitive approaches. This dynamic interplay positions metacognition as a self-regulating system capable of adaptive control over cognitive processes.

Initialisation Let $\mathcal{T}$ be a task and $\mathcal{G}$ be the associated goal. We establish the initial system state:

$$\mathcal{S}_0 = f(\mathcal{T}, \mathcal{G}) \tag{1}$$

where $(\mathcal{T}, \mathcal{G})$ is self-imposed or externally-imposed.

While Flavell [1979] treats 'goals' and 'tasks' as equivalent, we maintain a computational distinction to enhance the model's precision: $\mathcal{T}$ represents the specific cognitive enterprise, whilst $\mathcal{G}$ represents the desired outcome or success criteria. This separation enables clearer analysis of metacognitive processes, such as assessing the cognitive demands of $\mathcal{T}$ relative to $\mathcal{G}$, or specifying which approaches to employ for $\mathcal{T}$ to achieve $\mathcal{G}$. For instance, the same reasoning task ($\mathcal{T}$: logical problem-solving) might require different metacognitive assessments depending on whether the goal is speed ($\mathcal{G}_1$: quick approximation) or accuracy ($\mathcal{G}_2$: verified solution).

M-G-V (Information Processing) Cycle For monitoring cycles $\tau = 0, 1, \ldots, T$:

WHILE $\mathcal{S}_\tau =$ ACTIVE:

1. **MONITOR**: *Monitor cognitive status through retrieval of metacognitive knowledge and assessment of metacognitive experience.*

   Knowledge activation operates differently across metacognitive cycles, with initial cycles relying solely on task-goal combinations while ongoing cycles incorporate emerging metacognitive experiences. At $t = 0$, the system identifies potentially relevant metacognitive knowledge based exclusively on the task-goal pairing. In subsequent cycles ($\tau > 0$), the knowledge base expands as metacognitive experiences ($\mathcal{ME}$) from the previous cycle $\tau - 1$ triggering additional relevant knowledge.

$$\mathcal{MK}_\tau = \begin{cases} \text{retrieve}(\mathcal{MK}, \mathcal{T}, \mathcal{G}) & \text{if } \tau = 0 \\ \mathcal{MK}_{\tau-1} \cup \text{retrieve}(\mathcal{MK}, \mathcal{ME}_{\tau-1}) & \text{if } \tau > 0 \end{cases} \tag{2}$$

   According to Flavell [1979], metacognitive knowledge comprises three major categories:

   - **Agent Variables** ($\mathcal{MK}_{\text{Agent}}$): Knowledge about cognitive agents' characteristics and capabilities that applies across different cognitive endeavours. These are fundamentally subjective beliefs about processing preferences, strengths, and limitations rather than objective assessments. For computational agents, these may represent *learned self-models* – representations of performance patterns, processing preferences, and comparative capabilities derived from experience across cognitive tasks.
   - **Task Variables** ($\mathcal{MK}_{\text{Task}}$): Knowledge about cognitive situation assessment, including: (1) information characteristics (e.g., familiarity, complexity, organisation), and (2) task demands and goals. This knowledge is evaluative – understanding what task characteristics mean for cognitive processes and goal achievement, not merely recognising the characteristics themselves.

- **Strategy Variables** ($\mathcal{MK}_{\text{Strategy}}$): Knowledge concerning the effectiveness of cognitive strategies ($\mathcal{CS}$) and metacognitive strategies ($\mathcal{MS}$). Across different goals and task types, $\mathcal{CS}$ are cognitive operations that address problem-solving procedures such as applying domain-specific algorithms or step-by-step problem decomposition, whereas $\mathcal{MS}$ monitor and regulate such cognitive processes. For instance, chain-of-thought reasoning represents a $\mathcal{CS}$ for solving problems systematically, while deciding to *employ* chain-of-thought based on problem complexity assessment represents a $\mathcal{MS}$. Flavell [1979] explicitly incorporates both strategy types within this category, reflecting his theoretical position that strategy selection constitutes a fundamentally metacognitive process requiring knowledge about when, how, and why particular approaches prove effective under specific conditions.

These categories function as an integrated system: task variables diagnose cognitive demands, strategy variables prescribe responses, and agent variables contextualise both within the agent's capabilities[2].

Flavell [1979] distinguishes between knowledge-based experiences, which 'are best described as items of metacognitive knowledge that have entered consciousness' (e.g., suddenly recalling a relevant strategy), and feeling-based experiences, which 'clearly cannot be described that way' (e.g., feeling confused).

This dual nature of metacognitive experience – alternating between immediate phenomenological feelings and knowledge-based assessments – motivates our formal representation using the exclusive-or operator $\oplus$. In this formulation, $feel()$ captures the pure subjective sensations of cognitive state, while $assess()$ represents evaluations informed by metacognitive knowledge. The operator $\oplus$ thus reflects Flavell's distinction between feeling-based experiences (phenomenological states that cannot be reduced to knowledge) and knowledge-based experiences (instances of metacognitive knowledge entering consciousness).

Our exclusive-or formalisation captures the observation that these two modes typically alternate rather than blend, though we acknowledge that this binary representation constitutes a modelling simplification of potentially richer interactions. Accordingly, this binary characterisation suggests that, at any given moment, an agent experiences either raw cognitive feelings awaiting interpretation or automatic, knowledge-influenced assessments. The temporal alternation between these exclusive states gives rise to the evolving metacognitive experience that guides subsequent processing.

At this stage, it is notable that Flavell [1979, p. 909] primarily associates metacognitive experience with the subjective sense of perceived difficulty. Such experiences may involve feeling-of-complexity, comprehension challenges, conceptual opacity, or the sense that material exceeds current capabilities.

$$\mathcal{ME}^{\tau}_{\text{difficulty}} = \begin{cases} \text{feel}(\mathcal{T}) \oplus \text{assess}(\mathcal{T}, \mathcal{MK}_{\tau}) & \text{if } \tau = 0 \\ \text{feel}(\mathcal{T}, \text{Outcomes}_{\tau-1}) \oplus \text{assess}(\mathcal{T}, \text{Outcomes}_{\tau-1}, \mathcal{MK}_{\tau}) & \text{if } \tau > 0 \end{cases} \tag{3}$$

Accordingly, $\mathcal{ME}^{\tau}_{\text{difficulty}}$ evolves through iterative cycle-dependent assessments, progressing from initial coarse-grained feelings to increasingly nuanced evaluations that identify specific challenge sources and their implications for strategy selection. These experiences could help identify specific sources of obstacles and serve to guide the agent's attentional and regulatory focus.

2. **GENERATE**: *Control cognitive activity through strategy selection and execution.*

   Flavell [1979, p. 909] emphasises that this stage centres on the selection of cognitive strategies ($\mathcal{CS}_{\tau}$) through the integration of metacognitive experiences and knowledge. Metacognitive experiences of difficulty ($\mathcal{ME}_{\text{difficulty}}$), whether feeling-based or knowledge-based, function as computational *signals* that indicate cognitive status. However, these signals require interpretation through metacognitive knowledge to guide effective strategy selection.

[2] The distinction between $\mathcal{MK}_{\text{Strategy}}$ and $\mathcal{MK}_{\text{Task}}$ emerges from their functional roles. $\mathcal{MK}_{\text{Task}}$ enables diagnosis by identifying what makes cognitive enterprises demanding and how task characteristics influence goal achievement probability, whereas $\mathcal{MK}_{\text{Strategy}}$ enables prescription by specifying which cognitive approaches to deploy given those diagnostic assessments. Task variables answer 'what challenges does $\mathcal{T}$ present relative to $\mathcal{G}$?' and strategy variables answer 'which approaches for $\mathcal{T}$ will achieve $\mathcal{G}$?'

As Flavell [1979, p. 906] establishes, effective cognitive regulation emerges only when metacognitive experiences combine with metacognitive knowledge, transforming ambiguous feelings into actionable strategic decisions.

The strategy selection process draws upon $\mathcal{MK}_{\text{Strategy}}$, which encompasses knowledge about both metacognitive strategies ($\mathcal{MS}$) and cognitive strategies ($\mathcal{CS}$). Although Flavell does not specify the exact selection mechanism, his examples suggest a *two-phase pattern-matching* process. In the first phase, $\mathcal{MK}_{\text{Strategy}}$ guides the interpretation of $\mathcal{ME}^{\tau}_{\text{difficulty}}$, transforming general difficulty signals into precise diagnostic patterns. For instance, metacognitive knowledge might specify “when experiencing content uncertainty, identify specific unknown terms” or “when procedurally confused, assess whether confusion stems from missing steps versus unclear sequence”. In the second phase, these refined difficulty patterns activate corresponding cognitive strategies from $\mathcal{MK}_{\text{Strategy}}$ – content uncertainty with identified terms triggers seeking definitions, whilst procedural confusion from missing steps activates searching for worked examples. Throughout this process, the selection mechanism integrates agent capabilities ($\mathcal{MK}_{\text{Agent}}$) and task characteristics ($\mathcal{MK}_{\text{Task}}$) to identify the most appropriate strategy for achieving $\mathcal{G}$ given $\mathcal{T}$.

$$\mathcal{CS}_{\tau} = \text{select}(s \in \mathcal{MK}_{\text{Strategy}} \mid \mathcal{ME}^{\tau}_{\text{difficulty}}, \mathcal{MK}_{\tau}, \mathcal{T}, \mathcal{G}) \tag{4}$$

The selected cognitive strategy is implemented to produce cognitive outcomes ($\mathcal{CO}_{\tau}$), generating a feedback that is rich in nature, as as it encompasses not only task progress information but also a new context for the next cycle’s monitoring and potential strategy adjustment.

$$\mathcal{CO}_{\tau} = \text{execute}(\mathcal{CS}_{\tau}, \mathcal{T}, \mathcal{G}) \tag{5}$$

3. **VERIFY**: *Evaluate progress and determine continuation.*

Following strategy execution, Flavell [1979, p. 909] comments that the outcomes potentially ‘trigger additional metacognitive experiences about how the endeavour is faring’. These evaluative experiences ($\mathcal{ME}^{\tau}_{\text{evaluative}}$) are about performance rather than difficulty.

$$\mathcal{ME}^{\tau}_{\text{evaluative}} = \text{feel}(\mathcal{CO}_{\tau}) \oplus \text{assess}(\mathcal{CO}_{\tau}, \mathcal{MK}_{\tau}) \tag{6}$$

These experiences, again informed and guided by pertinent metacognitive knowledge, instigate the metacognitive strategy of surveying ‘all that [the agent has] learned to see if it fits together into a coherent whole, if it seems plausible and consistent with [the agent’s] prior knowledge and expectations, and if it provides an avenue to the goal’ [Flavell, 1979, p. 909].

$$\mathcal{MS}_{\tau} = \text{select}(s \in \mathcal{MK}^{\text{meta}}_{\text{Strategy}} \mid \mathcal{ME}^{\tau}_{\text{evaluative}}, \mathcal{MK}_{\tau}, \mathcal{CO}_{\tau}, \mathcal{G}) \tag{7}$$

where

$$\mathcal{MS}_{\tau} = \begin{cases} \text{coherence} & \text{if } \mathcal{ME}^{\tau}_{\text{evaluative}} \text{ signals fragmented understanding} \\ \text{plausibility} & \text{if } \mathcal{ME}^{\tau}_{\text{evaluative}} \text{ signals doubtful results} \\ \text{consistency} & \text{if } \mathcal{ME}^{\tau}_{\text{evaluative}} \text{ signals unexpected outcomes} \\ \text{goal-conduciveness} & \text{if } \mathcal{ME}^{\tau}_{\text{evaluative}} \text{ signals uncertain progress} \end{cases} \tag{8}$$

$\mathcal{ME}^{\tau}_{\text{evaluative}}$ signals the need for assessment. For example, “feeling uncertain about validity of the processed outcome” or “sensing incomplete understanding despite completion of process”. These evaluative experiences activate relevant metacognitive strategies from $\mathcal{MK}_{\text{Strategy}}$. For instance, uncertainty about validity triggers plausibility checking, while sensing incompleteness activates coherence assessment to identify gaps.

$\mathcal{MS}_{\tau}$ represents the strategic choice to conduct comprehensive evaluation along four possible dimensions: coherence (“do the outcomes form a consistent understanding?”), plausibility (“are the results believable given prior knowledge?”), consistency (“do outcomes align with initial expectations?”), and goal-conduciveness (“do current results provide a pathway to goal achievement?”). The execution systematically evaluates $\mathcal{CO}_{\tau}$ against relevant knowledge:

$$\mathcal{MO}_\tau = \text{execute}(\mathcal{MS}_\tau, \mathcal{CO}_\tau, \mathcal{MK}_\tau, \mathcal{G}) \tag{9}$$

Flavell emphasises that metacognitive experiences can ‘add to’, ‘delete from’, or ‘revise’ the metacognitive knowledge base through Piagetian [Flavell, 1963] mechanisms. The agent observes relationships among goals, strategies, experiences, and outcomes across the complete monitoring cycle.

Let Φ_τ represents the complete experience tuple, where $\mathcal{ME}_\tau = (\mathcal{ME}^\tau_{\text{difficulty}}, \mathcal{ME}^\tau_{\text{evaluative}})$, $\text{Strategy}_\tau = (\mathcal{CS}_\tau, \mathcal{MS}_\tau)$ and $\text{Outcome}_\tau = (\mathcal{CO}_\tau, \mathcal{MO}_\tau)$:

$$\Phi_\tau = (\mathcal{ME}_\tau, \text{Strategy}_\tau, \text{Outcome}_\tau) \tag{10}$$

$$\mathcal{MK} = \text{update}(\mathcal{MK}, \Phi_\tau) \tag{11}$$

Based on the comprehensive metacognitive evaluation, the system determines its next state:

$$\mathcal{S}_{\tau+1} = \begin{cases} \text{ACTIVE} & \text{if } \neg\text{goal\_achieved}(\mathcal{CO}_\tau, \mathcal{G}) \\ \text{TERMINATE} & \text{if goal\_achieved}(\mathcal{CO}_\tau, \mathcal{G}) \end{cases} \tag{12}$$

6 Nelson and Narens' Model of Metamemory

Nelson and Narens [1990] theorise metacognitive systems with particular focus on metamemory in the context of self-directed, self-paced learning and retrieval tasks. Their framework establishes metacognition as fundamentally hierarchical, distinguishing between cognitive processes that operate on mental content (object-level) and those that operate on cognitive processes themselves (meta-level). This two-level architecture provides the theoretical foundation for understanding how cognitive systems achieve self-regulation and control during learning activities.

According to their model, the meta-level maintains a dynamic internal representation of the object-level, functioning as a mental simulation that enables the system to monitor current cognitive states and guide transitions towards desired goals. The interaction between levels operates through two distinct information flows, *control* (meta-level → object-level) and *monitoring* (object-level → meta-level). Control processes enable the meta-level to modify object-level states or processes – such as allocating study time to difficult material or switching from rote memorisation to elaborative rehearsal strategies. Monitoring processes provide the meta-level with information about current object-level states, updating its internal model of the cognitive situation. These relationships connotes two notable properties: they are logically independent (control does not inherently generate feedback about its effects) and asymmetric (the meta-level maintains a model of the object-level whilst the object-level operates without any corresponding representation of the meta-level).

6.1 Acquisition Process

Initialisation Given a task ($\mathcal{T}$) and goal ($\mathcal{G}$) with a target performance level (ρ^*), the agent establishes the *norm of study* ($\mathcal{N}_s$):

$$\mathcal{MK}_0^{\text{STM}} = \text{retrieve}_\theta(\mathcal{MK}, \mathcal{T}, \mathcal{G}) \tag{13}$$

$$\delta_{\text{retention}} = \text{formulate}(\mathcal{MK}_0^{\text{STM}}, \tau_{\text{delay}}, \mathcal{T}, \mathcal{G}) \tag{14}$$

$$\mathcal{N}_s = \rho^* \times (1 + \delta_{\text{retention}}) \tag{15}$$

At the initialisation stage ($\tau = 0$), a global metacognitive parameter ($\mathcal{N}_s$) operationalises abstract goals into quantified mastery criteria, which Nelson and Narens [1990, p. 130] define as 'the overall degree of mastery the person believes should be attained during acquisition'.

Following Ericsson and Simon [1984], monitoring operations occur within working memory (STM), with $\mathcal{MK}_0^{\text{STM}}$ denoting the metacognitive knowledge retrieved into this workspace at $\tau = 0$. Information from long-term memory (LTM) may be accessed by first copying it into STM with probability θ [Atkinson and Shiffrin, 1968], captured through the notation $\text{retrieve}_\theta(\cdot)$ for this probabilistic access during metacognitive monitoring. The term $\delta_{\text{retention}}$ represents the agent's theory of retention – beliefs about memory decay over the interval τ_{delay}.

This formulation reflects Nelson and Narens' insight that effective learning requires anticipatory compensation for memory decay. The model predicts systematic variation in norm-setting behaviour across agents and contexts. For instance, an agent targeting 90% test performance ($\rho^* = 0.9$) who expects 20% decay ($\delta_{\text{retention}} = 0.2$) must achieve 108% mastery during acquisition. Moreover, the framework anticipates differential standards across learning contexts: conceptual understanding tasks (G_1, with $\delta_{\text{retention}} = 0.1$) vs. verbatim recall tasks (G_2, with $\delta_{\text{retention}} = 0.2$) yield distinct acquisition targets (99% vs. 108% respectively) even under identical performance goals.

M-G-V (Learning) Cycle For learning cycles $\tau \in \{1, \ldots, T_{\text{learn}}\}$, let $\mathcal{T}_\tau = \{i_j : j \in \mathcal{J}_\tau\}$ denote the set of items remaining in the task at cycle τ, where $\mathcal{J}_\tau \subseteq \{1, 2, \ldots, N\}$ represents the indices of items still requiring learning. Φ_τ^{STM} represent the cumulative learning experience in working memory.

WHILE $\mathcal{S}_\tau = \text{ACTIVE}$:

1. **MONITOR**: *Assess current mastery for each item* $i_j \in \mathcal{T}_\tau$.

   Monitoring involves retrieving metacognitive knowledge and generating metacognitive experiences about the current learning state.

$$\mathcal{MK}_{\tau}^{\text{STM}} = \mathcal{MK}_{\tau-1}^{\text{STM}} \cup \text{retrieve}_{\theta}(\mathcal{MK}, \mathcal{ME}_{\tau-1}) \quad \text{if } \tau > 0 \tag{16}$$

$$\mathcal{ME}_{\tau,j} = \begin{cases} [\text{EOL}_{\tau,j}, \text{null}] & \text{if } \tau = 0 \\ [\text{FOK}_{\tau,j}, \text{null}] & \text{if } \tau > 0 \end{cases} \tag{17}$$

where:

$$\text{EOL}_{\tau,j} = \text{feel}(i_j) \oplus \text{assess}(i_j, \mathcal{MK}_{\tau}^{\text{STM}}) \quad \text{if } \tau = 0 \tag{18}$$

$$\text{FOK}_{\tau,j} = \text{feel}(i_j, \mathcal{CO}_{\tau-1,j}) \oplus \text{assess}(i_j, \mathcal{CO}_{\tau-1,j}, \mathcal{MK}_{\tau}^{\text{STM}}) \quad \text{if } \tau > 0 \tag{19}$$

Metacognitive experiences are represented as vectors, reflecting Nelson and Narens' proposal of their multidimensional nature. Both ease-of-learning (EOL) and feeling-of-knowing (FOK) are immediate phenomenological experiences that emerge during cognitive tasks, illustrating how subjective feelings support monitoring functions, serving as the primary input for subsequent control decisions [Nelson and Narens, 1990, p. 160].

2. **GENERATE**: *Transforms monitoring outputs into executable learning actions.*

   Resources are allocated inversely proportional to their EOL or FOK, and strategy selection integrates metacognitive inputs to map learning methods to individual items.

$$r_{\tau,j} = R_{\text{total}} \times \frac{w_j}{\sum_{k=1}^{N} w_k}, \quad \text{where } w_j = (\mathcal{ME}_{\tau,j}[1])^{-1} \tag{20}$$

$$\sigma_{\tau,j} = \text{select}(s \in \mathcal{MK}_{\text{Strategy}} \mid i_j, r_{\tau,j}, \mathcal{ME}_{\tau,j}, \mathcal{MK}_{\tau}) \tag{21}$$

   The learning plan $\mathcal{P}_{\tau,j} = (i_j, r_{\tau,j}, \sigma_{\tau,j})$ is executed to produce cognitive outcomes (new memory state).

$$\mathcal{CO}_{\tau,j} = \text{execute}(\mathcal{P}_{\tau,j}) \tag{22}$$

3. **VERIFY**: *Assess learning progress and determines cycle continuation.*

   Judgement-of-learning (JOL) evaluate current mastery levels following cognitive outcomes.

$$\text{JOL}_{\tau,j} = \text{feel}(i_j, \mathcal{CO}_{\tau,j}) \oplus \text{assess}(i_j, \mathcal{CO}_{\tau,j}, \mathcal{MK}_{\tau}) \tag{23}$$

$$\mathcal{ME}_{\tau,j} = [\mathcal{ME}_{\tau,j}[1], \text{JOL}_{\tau,j}] \tag{24}$$

   For each item $i_j \in \mathcal{T}_{\tau}$, the agent computes the mastery discrepancy between the current JOL and the norm of study $\mathcal{N}_s$. Items that have reached the mastery criterion are removed from further consideration, so $\mathcal{T}_{\tau+1} = \{i_j : j \in \mathcal{J}_{\tau+1}\}$ contains only items still requiring learning.

$$\Delta_{\tau,j} = \mathcal{N}_s - \text{JOL}_{\tau,j} \tag{25}$$

$$\mathcal{J}_{\tau+1} = \{j \in \mathcal{J}_{\tau} : \Delta_{\tau,j} > 0\} \tag{26}$$

   The metacognitive experiences from each item, together with the strategies used and outcomes obtained, are then packaged into experience tuples and aggregated across all items processed in the current cycle.

$$\Phi_{\tau,j}^{\text{STM}} = (\mathcal{ME}_{\tau,j}, i_j, r_{\tau,j}, \sigma_{\tau,j}, \mathcal{CO}_{\tau,j}) \tag{27}$$

$$\Phi_{\tau}^{\text{STM}} = \Phi_{\tau-1}^{\text{STM}} \cup \{\Phi_{\tau,j}^{\text{STM}} : j \in \mathcal{J}_{\tau}\} \tag{28}$$

   Learning continues as long as any item remains below the mastery threshold:

$$\mathcal{S}_{\tau+1} = \begin{cases} \text{ACTIVE} & \text{if } \mathcal{J}_{\tau+1} \neq \emptyset \\ \text{TERMINATE} & \text{otherwise} \end{cases} \tag{29}$$

6.2 Retrieval Process

Initialisation Given a retrieval query $\mathcal{Q}$, the agent establishes retrieval goals and accesses contextually relevant metacognitive knowledge for search control.

$$\mathcal{MK}_0^{\text{STM}} = \text{retrieve}_\theta(\mathcal{MK}, \mathcal{Q}) \tag{30}$$

Nelson and Narens [1990] conceptualise FOK through the dual-counter hypothesis: one component accumulates evidence for information presence in memory (affirmative FOK, FOK_τ^+), while the other accumulates evidence for information absence, consistent with 'knowing not' [Kolers and Palef, 1976] (negative FOK, FOK_τ^-). This dual mechanism enables both continued search when positive evidence accumulates and efficient termination when negative evidence dominates, preventing exhaustive search behaviour.

The initial thresholds $\lambda_{\text{confidence}}^{(0)}$ and $\lambda_{\text{FOK}}^{(0)}$ are established through the agent's privileged access to personal metacognitive calibration history within $\mathcal{MK}_0^{\text{STM}}$:

$$\lambda_{\text{FOK}}^{(0)} = \text{median}(\{||\text{FOK}|| : \text{successful retrievals in } \mathcal{MK}_0^{\text{STM}}\}) \tag{31}$$

$$\lambda_{\text{confidence}}^{(0)} = \text{median}(\{\text{confidence}_\tau : \text{correct outputs in } \mathcal{MK}_0^{\text{STM}}\}) \tag{32}$$

FOK thresholds are calibrated based on successful retrievals – episodes where dual-counter FOK assessment correctly predicted retrieval outcomes, with $||\text{FOK}_\tau||$ (L1 norm) capturing the magnitude of metacognitive evidence. Confidence thresholds follow analogous calibration, reflecting historical accuracy at different confidence levels. This personalised approach embodies the No-Magic Hypothesis by utilising recallable metacognitive knowledge whilst accommodating domain-specific variations in metamemory accuracy.

M-G-V (Search) Process For search cycles $\tau \in \{0, 1, \ldots, T_{\text{search}}\}$, let $\mathcal{A}_\tau$ represent the current answer state (retrieved answer or null), and Ω_τ^{STM} represent the cumulative retrieval experience in working memory.

WHILST search is active:

1. **MONITOR**: *Assess Feeling-of-Knowing (FOK) and retrieval accessibility.*

   The metacognitive decision to initiate search relies on rapid, preliminary FOK judgement that operates faster than actual recall, enabling efficient search control [Reder, 1987, 1988]. Following the No-Magic Hypothesis, FOK monitoring accesses recallable item attributes – acquisition history, partial cues, contextual associations – rather than directly tapping unconscious memory states.

   $$\mathcal{MK}_\tau^{\text{STM}} = \begin{cases} \text{retrieve}_\theta(\mathcal{MK}, \mathcal{Q}) & \text{if } \tau = 0 \\ \mathcal{MK}_{\tau-1}^{\text{STM}} \cup \text{retrieve}_\theta(\mathcal{MK}, \text{FOK}_{\tau-1}) & \text{if } \tau > 0 \end{cases} \tag{33}$$

   $$\text{FOK}_\tau = \begin{bmatrix} \text{FOK}_\tau^+ \\ \text{FOK}_\tau^- \end{bmatrix} = \begin{cases} \text{feel}(\mathcal{Q}) \oplus \text{assess}(\mathcal{Q}, \mathcal{MK}_\tau^{\text{STM}}) & \text{if } \tau = 0 \\ \text{feel}(\mathcal{Q}, \mathcal{A}_{\tau-1}) \oplus \text{assess}(\mathcal{Q}, \mathcal{A}_{\tau-1}, \mathcal{MK}_\tau^{\text{STM}}) & \text{if } \tau > 0 \end{cases} \tag{34}$$

   At $\tau = 0$, preliminary FOK assessment determines search initiation through rapid accessibility evaluation using the dual-counter system. For subsequent cycles ($\tau > 0$), ongoing FOK monitoring incorporates previous search outcomes ($\mathcal{A}_{\tau-1}$) to reassess continued retrieval likelihood, with both affirmative (FOK_τ^+) and negative (FOK_τ^-) counters updating based on accumulating evidence.

2. **GENERATE**: *Deliberately attend to search cues and execute automatic search.*

   Following Nelson and Narens' insight that search execution is automatic once initiated, the generation phase focuses on conscious cue attention rather than strategy selection. The dual-counter FOK hypothesis provides metacognitive control over cue generation intensity, reflecting the principle that monitoring should adaptively influence control processes.

$$\mathcal{S}_\tau = \begin{cases} \text{ACTIVE}_{\text{intensive}} & \text{if } ||\text{FOK}_\tau|| < \lambda^{(\tau)}_{\text{FOK}} \\ \text{ACTIVE}_{\text{standard}} & \text{if } ||\text{FOK}_\tau|| \geq \lambda^{(\tau)}_{\text{FOK}} \wedge \text{FOK}^+_\tau > \text{FOK}^-_\tau \\ \text{TERMINATE} & \text{if } ||\text{FOK}_\tau|| \geq \lambda^{(\tau)}_{\text{FOK}} \wedge \text{FOK}^-_\tau > \text{FOK}^+_\tau \end{cases} \tag{35}$$

The search intensity logic operates through evidence-based decision making. When the total magnitude of metacognitive evidence falls below the threshold ($||\text{FOK}_\tau|| < \lambda^{(\tau)}_{\text{FOK}}$), insufficient evidence has accumulated from both counters to make a reliable continuation decision. This triggers intensive cue attention to gather additional metacognitive information, preventing premature termination based on weak or ambiguous signals. When sufficient evidence exists ($||\text{FOK}_\tau|| \geq \lambda^{(\tau)}_{\text{FOK}}$), the system evaluates counter dominance: positive dominance ($\text{FOK}^+_\tau > \text{FOK}^-_\tau$) indicates sufficient evidence for item presence to warrant continued search with standard attention, while negative dominance ($\text{FOK}^-_\tau > \text{FOK}^+_\tau$) provides sufficient evidence for item absence to justify search termination.

If $\mathcal{S}_\tau = \text{ACTIVE}$, the agent deliberately attends to retrieval cues that trigger automatic pattern-recognition-guided search, with attention determined by metacognitive confidence.

$$\text{cue}_\tau = \begin{cases} \text{attend}_{\text{intensive}}(\mathcal{Q}, \mathcal{MK}^{\text{STM}}_\tau) & \text{if } \mathcal{S}_\tau = \text{ACTIVE}_{\text{intensive}} \\ \text{attend}_{\text{standard}}(\mathcal{Q}, \mathcal{MK}^{\text{STM}}_\tau) & \text{if } \mathcal{S}_\tau = \text{ACTIVE}_{\text{standard}} \end{cases} \tag{36}$$

Once cues are consciously attended to, the search process $\text{search}_{\text{auto}}(\cdot)$ operates automatically through pattern recognition. Due to this automatic nature, $\mathcal{A}_\tau$ for consecutive cycles $\tau = 0, \ldots, k$ may yield identical results, reflecting the deterministic nature of automatic search.

$$\mathcal{A}_\tau = \text{search}_{\text{auto}}(\text{cue}_\tau) \tag{37}$$

3. **VERIFY**: *Evaluate retrieved answers based on confidence, update thresholds, and determine continuation.*

   According to Nelson and Narens [1990], confidence governs output decisions for retrieved answers, while FOK governs continuation decisions when no answer is found, with both involving dynamic thresholds that can change during search.

$$\text{confidence}_\tau = \begin{cases} \text{assess}(\mathcal{A}_\tau, \mathcal{Q}, \mathcal{MK}^{\text{STM}}_\tau) & \text{if } \mathcal{A}_\tau \neq \text{null} \\ 0 & \text{if } \mathcal{A}_\tau = \text{null} \end{cases} \tag{38}$$

$$\text{decision}_\tau = \begin{cases} \text{OUTPUT } \mathcal{A}_\tau & \text{if } \mathcal{A}_\tau \neq \text{null} \wedge \text{confidence}_\tau \geq \lambda^{(\tau)}_{\text{confidence}} \\ \text{CONTINUE} & \text{if } \mathcal{A}_\tau \neq \text{null} \wedge \text{confidence}_\tau < \lambda^{(\tau)}_{\text{confidence}} \\ \text{CONTINUE} & \text{if } \mathcal{A}_\tau = \text{null} \wedge \text{FOK}^+_\tau > \text{FOK}^-_\tau \\ \text{OUTPUT null (omission)} & \text{if } \mathcal{A}_\tau = \text{null} \wedge \text{FOK}^-_\tau > \text{FOK}^+_\tau \end{cases} \tag{39}$$

   This decision structure distinguishes between two primary error pathways identified by Nelson and Narens: (1) *Commission errors* occurring when $\mathcal{A}_\tau \neq \text{null}$ but the outputted answer is incorrect, typically associated with high confidence but incorrect retrieval; and (2) *Omission errors* occurring when search terminates without producing an answer ($\mathcal{A}_\tau = \text{null}$), often following prolonged search with declining FOK.

   The retrieval experience accumulates in working memory, creating a comprehensive search history that informs adaptive threshold adjustment:

$$\Omega^{\text{STM}}_\tau = \begin{cases} [(\text{FOK}_\tau, \text{cue}_\tau, \mathcal{A}_\tau, \text{confidence}_\tau)] & \text{if } \tau = 0 \\ \Omega^{\text{STM}}_{\tau-1} \cup [(\text{FOK}_\tau, \text{cue}_\tau, \mathcal{A}_\tau, \text{confidence}_\tau)] & \text{if } \tau > 0 \end{cases} \tag{40}$$

   Following the principle of satisficing [Simon, 1979], both confidence and FOK thresholds undergo dynamic adjustment based on accumulated search burden. This reflects the psychological tendency for acceptance criteria to progressively lower as the cost of continued searching increases. The satisficing adjustment factor captures this adaptive mechanism:

$$\beta_\tau = \exp(-\alpha \cdot (\tau + \sum_{(\mathcal{A}_i, \text{conf}_i) \in \Omega_\tau^{\text{STM}}} \mathbf{1}[\mathcal{A}_i = \text{null} \vee \text{confidence}_i < \lambda_{\text{confidence}}^{(i)}])) \tag{41}$$

where α represents the satisficing adjustment rate, and the exponential decay function models the psychological burden accumulating from both temporal persistence (τ) and retrieval failures (unsuccessful attempts or low-confidence outcomes). This burden manifests as decreasing acceptance standards, operationalised through threshold reduction:

$$\lambda_{\text{confidence}}^{(\tau+1)} = \lambda_{\text{confidence}}^{(\tau)} \cdot \beta_\tau \tag{42}$$

$$\lambda_{\text{FOK}}^{(\tau+1)} = \lambda_{\text{FOK}}^{(\tau)} \cdot \beta_\tau \tag{43}$$

This adaptive mechanism[3] ensures that answers previously deemed inadequate may become acceptable as search costs accumulate. Consequently, at cycle $\tau + 1$, a previously retrieved answer might satisfy the lowered confidence threshold and be output, even though it failed to meet the more stringent earlier criteria.

The search state for the next cycle is determined by:

$$\mathcal{S}_{\tau+1} = \begin{cases} \text{ACTIVE} & \text{if decision}_\tau = \text{CONTINUE} \\ \text{TERMINATE} & \text{if decision}_\tau \in \{\text{OUTPUT } \mathcal{A}_\tau, \text{OUTPUT null}\} \end{cases} \tag{44}$$

[3] Our adaptive threshold adjustment implements Nelson and Narens' 'costs/rewards rules' through a simplified model where search burden (costs) drives decreasing acceptance thresholds (reward standards). While this captures the essential cost/reward logic of adaptive satisficing, it abstracts away the multidimensional complexity that separate cost factors (time pressure, cognitive effort) and reward factors (answer importance, confidence benefits) might require.

7 Rational Metareasoning

The papers reviewed in this section share a common theoretical commitment; each treats a cognitive capacity as a *meta-level decision problem* in which the agent must choose which computations to perform in order to maximise expected reward while minimising computational cost. However, the formal structure of this problem differs across domains.

Contextual bandits In some domains, each problem is independent. Strategy selection [Lieder and Griffiths, 2017] exemplifies this structure. The agent observes features $\mathbf{f}$ characterising the current problem, selects a meta-level action, and receives feedback. There is no sequential structure within a problem – only across problems, as the agent learns which actions work well in which contexts.

Meta-level MDPs In other domains, the agent faces a sequential decision problem where each meta-level action updates a fully observable state. Planning [Callaway et al., 2022] and memory recall [Callaway et al., 2024] exemplify this structure. In planning, the state is the decision tree constructed so far – the agent knows exactly which nodes have been expanded and their values. In memory recall, the state comprises recall progress and elapsed time – the agent directly observes how much evidence has accumulated toward retrieval.

The "meta-level" designation distinguishes these from standard MDPs. A standard MDP formalises interaction with an external environment; a meta-level MDP formalises interaction with an internal, computational environment. States correspond to knowledge states rather than world states, actions correspond to computations rather than physical behaviours, and rewards capture both computational cost and decision quality. Formally, a meta-level MDP is defined as:

$$M = (\mathcal{S}, \mathcal{A}, T, r) \tag{45}$$

where $\mathcal{S}$ denotes the set of states, $\mathcal{A}$ the set of meta-level actions, $T(s, a, s')$ the transition function specifying how computations update knowledge, and $r(s, a)$ the meta-level reward.

Belief-MDPs When the relevant state is only partially observable, the problem becomes a Partially Observable MDP (POMDP). Cognitive control [Lieder et al., 2018] exemplifies this: the control system cannot directly access the full state of perceptual processes, working memory contents, or decision variables, but must infer these from available signals.

Such problems can be reformulated as belief-MDPs, where the state variable encodes what the agent *believes* rather than the true underlying state. Because beliefs must be initialised and updated based on observations, the initial belief state becomes a defining feature of the problem. Formally, a belief-MDP is defined as:

$$M = (\mathcal{B}, b_0, \mathcal{C}, T, r) \tag{46}$$

where $\mathcal{B}$ is the set of belief states, b_0 is the initial belief state, $\mathcal{C}$ is the set of meta-level actions, $T(b, c, b')$ specifies transition probabilities between belief states, and $r(b, c)$ is the meta-level reward.

7.1 Domain-Specific Instantiations

The following sections apply this framework to four domains. Table 1 summarises how each instantiates the relevant formal structure.

Component	Strategy Selection [Lieder and Griffiths, 2017]	Planning [Callaway et al., 2022]	Cognitive Control [Lieder et al., 2018]	Memory Recall [Callaway et al., 2024]
Structure	Contextual bandit	Meta-level MDP	Belief-MDP	Meta-level MDP
Context/State	Features $\mathbf{f}$	State s	Belief state b	State (t, z_t)
Observability	Fully observable	Fully observable	Partial	Fully observable
Meta-level action	Strategy s	Expansion a_i / $\perp$	Control signal c	SEARCH / STOP
Sequential?	No	Yes	Yes	Yes
Learning?	Yes (across problems)	No	Yes (across episodes)	No
Approximation	Learned (linear)	Myopic	Learned (LVOC)	Exact (compact state space)

Table 1: Structural comparison across domains.

7.2 Strategy Selection as Rational Metareasoning

Multiple cognitive strategies are often applicable to the same problem, yet people's strategy choices depend on task and context, and their adaptiveness increases with experience. Lieder and Griffiths [2017] proposed that people learn to approximate a cost-benefit analysis over strategies, predicting how well each will perform on problems with particular features.

7.2.1 Computational-level analysis

At the computational level, strategy selection has a distinctive structure that sets it apart from other meta-reasoning problems. Each problem the environment presents constitutes an independent episode. The agent observes a feature vector $\mathbf{f} = (f_1, \ldots, f_n)$ characterising the current problem, selects a cognitive strategy s from a finite set $\mathcal{S}$, and receives feedback in the form of outcome utility u and execution time t. Once a strategy is selected, it runs to completion – there is no sequential structure within a problem. Learning occurs only across problems, as the agent accumulates experience about which strategies work well in which contexts. This structure – independent problems, observable context, discrete choices – defines a **contextual multi-armed bandit**.

The objective is to learn a mapping $m : \mathcal{F} \to \mathcal{S}$ from feature vectors to strategies that maximises expected performance. Intuitively, a good strategy produces high-quality decisions without taking too long. This is formalised as the value of computation:

$$\text{VOC}(s, \mathbf{f}) = \mathbb{E}[U(s(\mathbf{f}); \mathbf{f})] - \gamma \cdot \mathbb{E}[T(s, \mathbf{f})] \tag{47}$$

where $U(s(\mathbf{f}); \mathbf{f})$ is the utility of the action selected by strategy s, $T(s, \mathbf{f})$ is execution time, and γ is the opportunity cost per unit time (e.g., the reward rate). The optimal mapping solves:

$$m^* = \arg\max_m \sum_{\mathbf{f} \in \mathcal{F}} P(\mathbf{f}) \cdot \text{VOC}(m(\mathbf{f}), \mathbf{f}) \tag{48}$$

However, computing this optimal mapping directly is infeasible because the true VOC of each strategy depends on outcomes revealed only after execution. The agent must learn the mapping from experience. After solving n problems, the agent's history is:

$$h_n = \Big((\mathbf{f}^{(1)}, s^{(1)}, u^{(1)}, t^{(1)}), \ldots, (\mathbf{f}^{(n)}, s^{(n)}, u^{(n)}, t^{(n)}) \Big) \tag{49}$$

recording the features, chosen strategy, resulting utility, and execution time for each problem. A learning mechanism $l : \mathcal{H} \to \mathcal{M}$ maps histories to strategy selection mappings, inducing a sequence of improving mappings $m^{(1)}, m^{(2)}, \ldots$ as experience accumulates. The optimal learning mechanism l^* maximises total expected VOC across all problem sequences.

The contextual bandit framing thus highlights two requirements for achieving optimality. First, the agent must generalise from past problems to novel ones using feature-based predictions. Second, the agent must balance exploration (trying uncertain strategies to learn about them) against exploitation (using known-good strategies to maximise immediate reward).

7.2.2 Model of mental computation

Having characterised the computational problem, the algorithmic level specifies how the brain might approximate the optimal learning mechanism.

The observability of VOC components The VOC itself cannot be observed directly – it is an expectation over future outcomes. However, VOC decomposes into two components that *are* separately observable: when a strategy generates a decision, the resulting utility u and execution time t become available. Recall that:

$$\text{VOC}(s, \mathbf{f}) = \mathbb{E}[U \mid s, \mathbf{f}] - \gamma \cdot \mathbb{E}[T \mid s, \mathbf{f}] \tag{50}$$

Because utility and time are observed after each strategy execution, the agent can learn to predict each component independently.

Feature-based function approximation. The agent learns to predict utility and time from problem features using strategy-specific weights. For each strategy s, weights $w_{k,s}^{(U)}$ capture how feature f_k relates to expected utility, and weights $w_{k,s}^{(T)}$ capture how feature f_k relates to expected time. This yields an approximation to the VOC:

$$\widehat{\text{VOC}}(s, \mathbf{f}; \mathbf{w}) = \sum_k w_{k,s}^{(U)} f_k - \gamma \cdot \sum_k w_{k,s}^{(T)} f_k \tag{51}$$

For binary outcomes (success/failure), utility prediction takes a logistic form rather than linear.

Learning the opportunity cost The agent must also estimate γ, the opportunity cost per unit time. This represents how much reward the agent foregoes by spending time on the current problem rather than moving to the next. It is modelled as the posterior mean reward rate given cumulative experience: $\gamma = \mathbb{E}[r \mid t_{\text{total}}, r_{\text{total}}]$.

Bayesian weight updating The weights are learned through Bayesian inference. After each experience $h = (\mathbf{f}, s, u, t)$, the agent updates its posterior distribution over weights. In simple settings, this reduces to Bayesian linear regression for the time model and Bayesian logistic regression for the utility model (see Lieder and Griffiths [2017, pp. 66–68]. The result is a system that learns to predict how well each strategy will perform on problems with particular features.

7.2.3 Optimal resource allocation

The optimal policy must balance exploitation of current knowledge against exploration to improve future predictions.

The failure of greedy selection A greedy policy that always selects the strategy with highest expected VOC ignores the value of learning about uncertain strategies. Suppose the agent has used strategy s_1 fifty times with consistent success, yielding a precise estimate of its VOC. Strategy s_2 has been tried twice, both times unsuccessfully. A greedy policy would never try s_2 again. Yet those failures might have been unlucky, and s_2 might actually be superior in certain contexts.

Thompson sampling The model resolves this dilemma through Thompson sampling. Rather than using point estimates (the posterior means), the agent samples from the full posterior distributions over weights:

$$\tilde{w}_{k,s}^{(U)} \sim P(w_{k,s}^{(U)} \mid h_t), \quad \tilde{w}_{k,s}^{(T)} \sim P(w_{k,s}^{(T)} \mid h_t) \tag{52}$$

Optimal policy Given the sampled weights, the agent computes a sampled VOC for each strategy and selects the strategy that maximises this quantity:

$$s_t = \arg\max_s \widehat{\text{VOC}}(s, \mathbf{f}_t; \tilde{\mathbf{w}}) = \arg\max_s \left[\sum_k \tilde{w}_{k,s}^{(U)} f_k - \gamma \cdot \sum_k \tilde{w}_{k,s}^{(T)} f_k \right] \tag{53}$$

The $\arg\max$ operates over all available strategies $s \in \mathcal{S}$, selecting whichever strategy has the highest predicted value given the current problem features $\mathbf{f}_t$ and the sampled weights $\tilde{\mathbf{w}}$. Crucially, because the weights are sampled rather than fixed at their means, different samples can yield different strategy rankings – even for the same problem.

This mechanism naturally balances exploration and exploitation through posterior variance. Well-known strategies have narrow posteriors, so samples cluster near expected values – such strategies are selected only when their expected VOC is genuinely high. Uncertain strategies have wide posteriors whose samples occasionally exceed those of better-known alternatives, prompting the agent to explore them and gather information. As the agent gains experience with a strategy, its posterior narrows, and its selection becomes driven by expected value rather than exploratory variance.

7.2.4 Evaluation and refinement

Lieder and Griffiths validated their model against human strategy choices across multiple decision-making paradigms. The model captured context-dependent strategy selection, learning dynamics, and transfer to novel problems. This work contributes to bounded rationality by specifying when people should use which heuristics, framing strategy selection as a learnable metacognitive skill rather than a fixed repertoire.

7.3 Planning as Rational Metareasoning

Planning involves mentally simulating future possibilities before acting. Since simulation is costly, the agent faces a resource allocation problem: which aspects of the future should be simulated, and when should deliberation stop? Callaway et al. [2022] formalised this problem by recognising that planning has a sequential structure absent from strategy selection: each computational action updates the agent's knowledge, and the value of a computation depends on which computations will follow.

7.3.1 Computational-level analysis

At the computational level, planning differs fundamentally from strategy selection. In strategy selection, each problem is independent – selecting a strategy for the current problem does not affect future problems. In planning, the agent performs a sequence of computational actions within a single episode, each of which updates what the agent knows. The value of expanding one branch of a decision tree depends on whether other branches will subsequently be expanded. This sequential dependency structure defines a **meta-level Markov Decision Process (MDP)**.

The "meta-level" designation distinguishes this from a standard MDP. A standard MDP formalises an agent's interaction with an external environment: states are world states, actions are physical behaviours, and transitions reflect environmental dynamics. A meta-level MDP formalises an agent's interaction with its own internal computational processes: states are knowledge states, actions are computations, and transitions reflect how computations update knowledge. Formally:

$$M = (\mathcal{S}, \mathcal{A}, T, r) \tag{54}$$

where $\mathcal{S}$ denotes the set of knowledge states, $\mathcal{A}$ the set of computational actions, $T(s, a, s')$ the transition function specifying how computations update knowledge, and $r(s, a)$ the meta-level reward capturing both computational cost and decision quality.

The objective is to find an optimal policy $\pi^* : \mathcal{S} \to \mathcal{A}$ that maximises total expected meta-level reward – the quality of the eventual decision minus the cumulative cost of deliberation.

7.3.2 Model of mental computation

Having established the meta-level MDP formulation, the algorithmic level specifies how states, actions, transitions, and rewards are concretely represented for the planning problem.

States ($\mathcal{S}$) A state encodes the agent's current knowledge about the decision problem. Following previous work in rational metareasoning, Callaway et al. represent this knowledge as a decision tree: a directed graph in which nodes represent hypothetical future states and edges represent the actions connecting them. Let N denote the maximum number of nodes in the decision tree. The state is then a vector $\mathbf{s}$ of length N, where each element s_i either contains the observed reward at node i or takes the special value $\varnothing$ indicating that node i remains unexpanded. Initially, only the root node (representing the current state) has been expanded with value 0; all other nodes have value $\varnothing$ (e.g., $\mathbf{s}_0 = [0, \varnothing, \varnothing, \ldots]$).

Actions ($\mathcal{A}$) The action space $\mathcal{A}$ comprises an expansion action a_i for each node i and a termination action $\perp$. Node expansion reveals the reward at node i, integrates this value into the total value of the path leading to that node, and adds the node's immediate successors to the search *frontier*. The frontier is defined as the set of nodes eligible for expansion; a node may be expanded only if it belongs to the frontier:

$$\text{frontier}(\mathbf{s}) = \{a_i \mid s_i = \varnothing \land \text{parent}(s_i) \neq \varnothing\} \tag{55}$$

When the agent executes the termination action, planning ceases and the agent commits to the plan with highest expected value given the decision tree constructed thus far.

Transition function (T) The meta-level transition function specifies how computational actions modify the state, analogous to how physical actions modify the world state in a standard MDP. Executing expansion action a_i produces a successor state $\mathbf{s}'$ identical to $\mathbf{s}$ except that s'_i is sampled from a node-specific reward distribution R_i. Executing the termination action $\perp$ deterministically transitions to a unique terminal state $\mathbf{s}_\perp$. Together, these rules implicitly define a probability distribution $T(\mathbf{s}' \mid \mathbf{s}, a)$ over successor states.

Reward function (r) The meta-level reward function quantifies both the cost of computation and the quality of the resulting decision:

$$r(\mathbf{s}, a) = \begin{cases} \max_{p \in \mathcal{P}} V(\mathbf{s}, p) & \text{if } a = \perp \\ -\lambda & \text{otherwise} \end{cases} \tag{56}$$

where $\lambda > 0$ represents the cost of expanding a single node, p denotes a complete plan (a sequence of nodes from root to leaf), $\mathcal{P}$ is the set of all such plans, and $V(\mathbf{s}, p)$ quantifies the expected value of executing plan p given the current state:

$$V(\mathbf{s}, p) = \sum_{i \in p} \begin{cases} s_i & \text{if } s_i \neq \varnothing \\ \mathbb{E}[R_i] & \text{otherwise} \end{cases} \tag{57}$$

For expanded nodes, the agent uses the observed reward s_i; for unexpanded nodes, it uses the prior expectation $\mathbb{E}[R_i]$. The termination reward therefore equals the maximum expected value across all possible plans, reflecting the quality of the decision the agent would make given its current knowledge.

7.3.3 Optimal resource allocation

The optimal policy specifies which computational action to take in each state. It is characterised by the optimal Q-function, which quantifies the expected total reward from taking action a in state $\mathbf{s}$ and acting optimally thereafter:

$$Q^*(\mathbf{s}, a) = r(\mathbf{s}, a) + \mathbb{E}_{\mathbf{s}' \sim T(\cdot \mid \mathbf{s}, a)}[V^*(\mathbf{s}')] \tag{58}$$

where $V^*(\mathbf{s}) = \max_a Q^*(\mathbf{s}, a)$ is the optimal value function. The optimal policy selects the action with highest Q-value:

$$\pi^*(\mathbf{s}) = \arg\max_a Q^*(\mathbf{s}, a) \tag{59}$$

This policy expands a node only when the expected improvement in decision quality exceeds the cost of expansion λ. Equivalently, it terminates when no expansion has positive expected value of computation.

Intractability Computing Q^* exactly is intractable for realistically sized decision trees. The state space grows exponentially with the number of nodes, and evaluating each state requires integrating over possible values of unexpanded nodes.

Myopic approximation Callaway et al. proposed a tractable approximation: at each step, select the computation that would yield the greatest expected improvement if the agent were forced to terminate immediately afterwards. This *one-step lookahead* or *myopic* policy ignores the value of computations that enable future computations, but it preserves the core tradeoff between computational cost and decision quality. Like the optimal policy, it is governed by a single parameter λ representing the cost of computation.

This framework does not propose feature-based learning across planning episodes. The agent allocates computation optimally within a single episode; there is no mechanism for transferring knowledge about good planning strategies from one problem to the next.

7.3.4 Evaluation and refinement

Callaway et al. evaluated their model using a Mouselab paradigm that externalises planning by requiring participants to click on nodes to reveal values, making the planning process directly observable. Human planning adapted to environmental structure in a manner broadly consistent with the meta-level MDP framework.

However, participants exhibited a pronounced bias towards forward search – expanding nodes in temporal order – even when other expansion orders would have been equally adaptive. This suggests that temporally ordered simulation may be a cognitive default, perhaps reflecting adaptation to naturalistic environments where reachable states can only be discovered through forward simulation from the current state.

The authors acknowledge important boundary conditions. The experiments considered only deterministic environments, which ensures that complete planning before acting is optimal; in stochastic environments, planning far ahead may be wasteful as unexpected transitions can render prior deliberation irrelevant. The state spaces were also small and unstructured. Furthermore, the framework provides a computational-level account but not a process-level theory of how people approximate optimal planning. The myopic approximation offers one hypothesis – that people select computations based on immediate expected value rather than full recursive evaluation – but this remains to be tested directly.

Despite these limitations, the results suggest that models of efficient resource allocation provide a productive foundation for theories of planning under cognitive constraints.

7.4 Cognitive Control as Rational Metareasoning

Cognitive control enables the brain to override automatic processes when they conflict with current goals. While exerting control improves performance, it is also effortful. Lieder et al. [2018] cast cognitive control specification as a sequential decision problem with a crucial additional complexity: unlike planning, where the decision tree is fully observable, the internal state of controlled cognitive systems is only partially observable.

7.4.1 Computational-level analysis

At the computational level, cognitive control specification shares the sequential structure of planning – each control signal affects subsequent options – but adds the challenge of partial observability. The cognitive control system cannot directly access the full state of perceptual processes, working memory contents, or decision variables. It must infer these internal states from available signals and select control actions based on uncertain beliefs.

When the relevant state is only partially observable, the problem becomes a Partially Observable MDP (POMDP). Such problems can be reformulated as **belief-MDPs**, where the state variable encodes what the agent *believes* about the environment rather than the true underlying state. Formally:

$$M = (\mathcal{B}, b_0, \mathcal{C}, T, r) \tag{60}$$

where $\mathcal{B}$ is the set of belief states (encoding beliefs about both the external environment and internal cognitive state), b_0 is the initial belief state, $\mathcal{C}$ is the set of control signals specifying which computations to perform, $T(b, c, b')$ specifies transition probabilities between belief states, and $r(b, c)$ captures outcome utility minus computational cost.

The objective is to find an optimal cognitive control strategy $\pi^* : \mathcal{B} \to \mathcal{C}$ that maximises the expected sum of rewards minus costs. To formalise this, we first define the value function $V_\pi(b)$ as the expected cumulative reward from following policy π starting in belief state b. The optimal value function $V^*(b) = \max_\pi V_\pi(b)$ represents the best achievable expected reward from state b.

The total expected value of issuing control signal c in belief state b combines the immediate reward with the expected future value. Because the next belief state is uncertain – it depends on stochastic factors like stimulus arrivals and internal processing variability – we write B_{t+1} (a random variable) rather than b_{t+1} (a specific realisation). The **expected value of control (EVOC)** is then:

$$\text{EVOC}(b, c) = Q^*(b, c) = \mathbb{E}[r(b, c) + V^*(B_{t+1}) \mid B_t = b, C_t = c] \tag{61}$$

The optimal strategy selects, in each belief state, the control signal with highest EVOC:

$$\pi^*(b) = \arg\max_{c \in \mathcal{C}} \text{EVOC}(b, c) \tag{62}$$

7.4.2 Model of mental computation

Having characterised the computational problem, the algorithmic level specifies how the brain might approximate the optimal solution given constraints on time and resources.

The approximation challenge Computing the EVOC exactly requires estimating the future consequences of control signals across all possible belief state trajectories – a computation that is itself demanding. Yet cognitive control often operates under severe time pressure: a habitual response may execute within hundreds of milliseconds. The brain cannot afford to solve a complex belief-MDP from scratch on every trial. This tension motivates a learning-based solution: rather than computing the EVOC online, the cognitive control system learns to predict it from experience.

Two obstacles to learning Learning the optimal Q-function directly faces two obstacles. First, *temporal entanglement*: the value of a control signal depends on which control signals will follow, creating credit assignment problems. Attending carefully at the start of a trial may be valuable or wasteful depending on subsequent attentional adjustments. Second, *dimensionality*: the belief state space encompasses every possible configuration of beliefs about controlled processes – a space far too large to store separate value estimates for each state.

Feature-based representation Both obstacles can be addressed by learning a compact, generalising representation. Rather than storing values for individual belief states, the system learns how interpretable features of the situation predict the value of control. Three classes of features are relevant:

1. **State features** $f_k(b)$: aspects of the current context, such as stimulus configuration, task demands, or recent history (e.g., "was the previous target green?")
2. **Control signal intensities** c_l: the magnitude of each control dimension (e.g., "how strongly to attend")
3. **Interaction terms** $f_k(b) \cdot c_l$: how context modulates the value of particular control settings (e.g., "does a recent green target increase the value of attending to green now?")

The **Learned Value of Control (LVOC)** approximates the EVOC as a weighted sum of these features:

$$\text{LVOC}(b, c; \mathbf{w}) = w_0 + \sum_k w_k^{(f)} f_k(b) + \sum_l w_l^{(c)} c_l + \sum_{k,l} w_{kl}^{(f \times c)} f_k(b) \cdot c_l - \text{cost}(c) - w^{(T)} T \tag{63}$$

where $\text{cost}(c)$ reflects the metabolic or opportunity cost of exerting control and $w^{(T)} T$ penalises slow responses.

Bayesian weight updating The weight vector $\mathbf{w}$ is learned through Bayesian inference. After each experience $e_i = (b_i, c_i, r_i, T_i, b_{i+1})$ – recording the belief state, chosen control signal, obtained reward, response time, and resulting belief state – the agent updates its posterior:

$$P(\mathbf{w} \mid e_{1:t}) \propto P(\mathbf{w} \mid e_{1:t-1}) \cdot P(e_t \mid \mathbf{w}) \tag{64}$$

When a single control signal produces a single reward, this reduces to Bayesian linear regression. When rewards are delayed or depend on sequences of control signals, the update substitutes predicted future values for unobserved outcomes.

Options The framework extends from elementary control signals to compound cognitive strategies. A single control signal might modulate one parameter (e.g., "attend to the left stimulus"), but complex behaviour often requires coordinating multiple signals simultaneously (e.g., increasing attention while also raising the decision threshold) or executing an entire sequence of operations (e.g., following a multi-step planning routine).

The rational metareasoning framework accommodates such cases by treating cognitive strategies as *options*. An option is a policy combined with an initiation set and a termination condition [Lieder et al., 2018, pp. 5]; see also Sutton et al. [1999]. That is, an option o can be defined by three components. First, an *initiation set* specifies the belief states in which the option can be started. Second, an *internal policy* specifies what control signals to issue while the option is executing. Third, a *termination condition* specifies when the option finishes.

Once initiated, the option follows its internal policy until the termination condition is met. An elementary control signal can be viewed as an option with trivial structure. A complex strategy is an option with richer structure involving multiple internal steps before termination.

7.4.3 Optimal resource allocation

Learning introduces a second-order problem: how should the system select control signals while its value estimates remain uncertain?

The failure of greedy selection. A greedy policy that always selects the control signal with highest predicted LVOC can fail during learning. Control signals that happen to fail early are abandoned; those that happen to succeed are repeated. The agent never discovers that alternative control signals might be superior in certain contexts.

Thompson sampling The LVOC model escapes this trap through Thompson sampling. Rather than acting on point estimates (the posterior means), the system samples from its posterior distribution over weights:

$$\tilde{\mathbf{w}} \sim P(\mathbf{w} \mid e_{1:t}) \tag{65}$$

Optimal policy Control signals are selected by maximising the LVOC computed with sampled weights:

$$c_t = \arg\max_c \text{LVOC}(b_t, c; \tilde{\mathbf{w}}) \tag{66}$$

When selecting among options rather than elementary control signals, the policy becomes:

$$\pi^*(b) = \arg\max_{o \in \mathcal{O}} Q^*(b, o) \tag{67}$$

As in strategy selection, posterior variance calibrates exploration: uncertain control signals occasionally receive optimistic samples, prompting information gathering.

7.4.4 Evaluation and refinement

Lieder et al. validated the LVOC model against five experiments on attentional plasticity and interference control. The model captured how people learn which stimuli to attend based on reward history, how allocation adapts to task difficulty and reward, and how learning transfers to novel stimuli sharing features with previously encountered ones.

The framework rests on several simplifying assumptions. The LVOC approximation assumes that a linear combination of features suffices to capture the value of control, which may not hold in tasks requiring complex, nonlinear interactions among contextual variables. The model also assumes that people can accurately observe their own belief states and the outcomes of their control decisions, yet introspective access to cognitive processes is known to be limited and sometimes distorted. Furthermore, the current formulation treats the feature set as given rather than learned, leaving open the question of how people discover which aspects of a situation are relevant to control.

It is worth clarifying how this framework relates to standard reinforcement learning. Model-based reinforcement learning learns the consequences of acting in the world, such as which locations yield reward following which movements. The LVOC model learns the consequences of thinking in a particular way, such as how strongly attending improves accuracy at the cost of effort. The learning is metacognitive in that its object is not behaviour itself, but the control of the computations that produce behaviour.

Despite these limitations, the LVOC framework offers a principled account of how cognitive control can be both adaptive and learnable. By casting control specification as a belief-MDP and introducing a tractable approximation, the model explains how control exhibits plasticity, adapting through experience to select signals that optimally trade off effort against goal achievement.

7.5 Memory Recall as Rational Metareasoning

Retrieving information from memory is not instantaneous. When we cannot immediately recall something, we face a decision: continue searching or give up? Research on metamemory has established that people can judge the likelihood of successful recall – a capacity termed *feeling-of-knowing* – and that such judgments influence search duration. Callaway et al. [2024] formalised metamemory as a meta-level MDP with the structure of an optimal stopping problem: a higher-order process monitors the progress of a basic recall process and controls how long the search is allowed to continue.

7.5.1 Computational-level analysis

At the computational level, memory recall shares the sequential structure of planning – each moment of continued search is a decision that affects future options. However, in planning, the agent decides both *what* to compute (which node to expand) and *when* to stop; whereas in memory recall, the only decision is whether to continue searching or give up; the computational process itself (evidence accumulation toward recall) proceeds automatically once initiated.

Following classic theories of metamemory [Nelson and Narens, 1990], the model distinguishes two interrelated processes. The *object-level process* comprises the mechanisms supporting recall itself. The *meta-level process* monitors the rate of progress toward recall and controls how long the search is allowed to continue.

The key insight is that the stopping decision involves a recursive dependency: whether to stop now depends on the expected value of continuing to search, which depends on how long the search would persist, which depends on future stopping decisions. This recursive dependency makes metamemory a sequential decision problem, naturally formalised as a **meta-level MDP**:

$$M = (\mathcal{S}, s_0, \mathcal{A}, T, r) \tag{68}$$

where $\mathcal{S}$ is the set of states encoding current recall progress, s_0 is the initial state, $\mathcal{A} = \{\text{SEARCH}, \text{STOP}\}$ comprises two actions, T specifies how continued searching updates the state, and r encodes the benefit of recall and cost of search.

From a rational perspective, one should keep searching as long as the expected utility exceeds the expected cost:

$$a^* = \begin{cases} \text{SEARCH} & \text{if } P(\text{recall}) \cdot U(\text{recall}) > \mathbb{E}[\text{cost}(\text{search})] \\ \text{STOP} & \text{otherwise} \end{cases} \tag{69}$$

The challenge lies in estimating $P(\text{recall})$ and $\mathbb{E}[\text{cost}(\text{search})]$, both of which depend on the unknown strength of the target memory.

7.5.2 Model of mental computation

Having characterised the computational problem, the algorithmic level specifies how the object-level and meta-level processes are concretely represented.

Object-level process. Recall is modelled as a simple evidence accumulation – a framework widely applied in decision-making and memory research. At each time step t, current recall progress z_t is incremented by a noisy sample:

$$z_t = z_{t-1} + x_t \quad \text{where} \quad x_t \sim \mathcal{N}(v, \sigma_x^2) \tag{70}$$

The drift rate v captures the strength of the memory: stronger memories accumulate evidence faster. The target is recalled when progress exceeds a threshold θ. Importantly, this threshold is exogenous – unlike in decision-making models where the agent can choose to commit based on any amount of evidence, the amount of evidence necessary to recall a memory is not under the agent's control (for further discussion on endogenous versus exogenous threshold, see Callaway et al. [2024, pp. 18].

States The state $s_t = (t, z_t)$ comprises two quantities: the time elapsed (t) and the current recall progress (z_t). Together, these provide a complete summary statistic for the entire evidence sequence up to time t. The model *assumes* that the meta-level process directly observes this state. This is a simplifying assumption for tractability, not a claim about how people actually monitor their memory – whether real monitoring tracks underlying memory strength, partial recall progress, or superficial cues remains an open empirical question (see Callaway et al. [2024, pp. 4, 19]).

Inferring memory strength Although the meta-level process observes recall progress, it does not directly observe the memory's strength v. However, observed progress provides approximate information about strength: rapid progress suggests a strong memory; slow progress suggests a weak one. Given observed progress z_t over time t, the agent can compute a posterior distribution over memory strength:

$$P(v \mid t, z_t) = \mathcal{N}(v; \mu_t, \sigma_t^2) \tag{71}$$

With a weak prior, $\mu_t \approx z_t/t$ – the average rate of progress. This time-varying belief about memory strength formalises the concept of *feeling-of-knowing*, which is a sense that the target is (or is not) likely to be recalled with further effort.

Transition function The transition function captures how continued searching updates the state. The next state depends on the increment x_{t+1}, which is drawn from $\mathcal{N}(v, \sigma_x^2)$. However, the true memory strength v is unknown to the agent.

To handle this uncertainty, the model *marginalises* over v. Marginalisation means integrating out the unknown variable: rather than conditioning on a single value of v, the model averages over all possible values, weighting each by its posterior probability $P(v \mid t, z_t)$. This yields a transition probability that reflects the agent's uncertainty about memory strength:

$$T(s_{t+1} \mid s_t, \text{SEARCH}) = \int P(z_{t+1} \mid z_t, v)\, P(v \mid t, z_t)\, dv \tag{72}$$

The first term $P(z_{t+1} \mid z_t, v)$ is Gaussian because the increment $x_{t+1} = z_{t+1} - z_t$ is normally distributed with mean v and variance σ_x^2. The second term $P(v \mid t, z_t)$ is also Gaussian, as derived above. Because the increment can be written as $x_{t+1} = v + \varepsilon$ where $\varepsilon \sim \mathcal{N}(0, \sigma_x^2)$ is noise independent of v, the marginal distribution over x_{t+1} is simply the distribution of a sum of two independent Gaussians. This yields a closed-form solution:

$$T(s_{t+1} \mid s_t, \text{SEARCH}) = \mathcal{N}(z_{t+1} - z_t \mid \mu_t, \sigma_x^2 + \sigma_t^2) \tag{73}$$

The expected increment equals the posterior mean μ_t (the agent's current estimate of memory strength), and the variance combines noise in the evidence (σ_x^2) with uncertainty about the true strength (σ_t^2). This closed-form solution keeps the model tractable, avoiding the need for numerical integration at each time step.

Reward function The reward function encodes the benefit of successful recall and the cost of continued search:

$$r(s_t, a) = \begin{cases} U(\text{recall}) & \text{if } z_t \geq \theta \\ -\gamma_{\text{SEARCH}} & \text{if } a = \text{SEARCH} \\ 0 & \text{if } a = \text{STOP} \end{cases} \tag{74}$$

where $U(\text{recall})$ is the utility of successful retrieval and γ_{SEARCH} is the explicit (e.g., experimentally imposed cost) and implicit cost (e.g., opportunity cost) per time step of search.

7.5.3 Optimal resource allocation

The compact state space (t, z_t) makes exact computation of the optimal policy tractable via backward induction, unlike planning where the exponential state space necessitates approximation.

Optimal value function The optimal value function can be factorised into two interpretable components:

$$V^*(s_t) = P(\text{recall} \mid s_t) \cdot U(\text{recall}) - \mathbb{E}[t_{\text{stop}} - t \mid s_t] \cdot \gamma_{\text{SEARCH}} \tag{75}$$

The first term captures the expected benefit of continued search. $P(\text{recall} \mid s_t)$ is the probability that the target will eventually be recalled given the current state, and $U(\text{recall})$ is the utility of successful retrieval. Their product is the expected utility from recall.

The second term captures the expected cost of continued search. $\mathbb{E}[t_{\text{stop}} - t \mid s_t]$ is the expected number of additional time steps until the search ends, whether by successful recall or by giving up. This quantity depends on both the current state and the agent's future stopping decisions. Multiplying by γ_{SEARCH}, the cost per time step, yields the total expected cost of the remaining search. The optimal value is thus the expected benefit minus the expected cost.

Optimal policy The optimal policy takes the form:

$$\pi^*(s_t) = \begin{cases} \text{SEARCH} & \text{if } \mathbb{E}_{s_{t+1} \sim T(\cdot|s_t, \text{SEARCH})}[V^*(s_{t+1})] > \gamma_{\text{SEARCH}} \\ \text{STOP} & \text{otherwise} \end{cases} \tag{76}$$

The expectation $\mathbb{E}_{s_{t+1} \sim T(\cdot|s_t, \text{SEARCH})}[V^*(s_{t+1})]$ averages the optimal value of the next state over all possible increments in recall progress, weighted by their probability under the transition function T. This expectation represents the expected future value if the agent continues searching for one more time step and then acts optimally thereafter.

The policy thus has a simple interpretation: continue searching if the expected future value exceeds the immediate cost of one more search step; otherwise, stop. This is a one-step lookahead condition, but because V^* already incorporates the consequences of all future optimal decisions, the policy is globally optimal rather than myopic.

Threshold representation The optimal policy can equivalently be represented as a time-varying threshold on recall progress. If progress z_t falls below this threshold at any moment, the search is terminated. The threshold is not constant because a fixed amount of negative progress provides stronger evidence of low memory strength when generated quickly than when generated slowly. Early in the search, the agent tolerates low progress because uncertainty about memory strength remains high; later, the same progress level may fall below threshold as evidence accumulates that the memory is weak.

7.5.4 Evaluation and refinement

The model makes two predictions about the relationship between memory strength and response time. First, stronger memories should be recalled more quickly. This is a straightforward consequence of faster evidence accumulation in the object-level process. Second, stronger memories should be abandoned more slowly. While the meta-level process can quickly identify very weak memories as unlikely to be recalled (their progress falls below threshold early), marginal-strength memories produce ambiguous evidence. It takes longer to accumulate sufficient evidence that the memory is too weak to justify continued search. Together, these predictions explain empirical pattern where response time and judged memory strength are negatively correlated for successful recall (faster responses reflect stronger memories) but positively correlated for unsuccessful recall (longer searches before giving up reflect stronger feeling-of-knowing).

Callaway et al. validated their model using a cued-recall paradigm that allowed objective measurement of memory strength before critical trials. Consistent with the optimal model, participants searched longer before giving up on targets they were more likely to recall, and they prioritised searching for stronger memories when given a choice between targets. These findings support the claim that feeling-of-knowing serves an adaptive function, enabling efficient termination of searches unlikely to succeed while sustaining effort when retrieval remains probable.

The model provides a normative account of how feeling-of-knowing could adaptively guide recall efforts. By formalising metamemory as an MDP, it creates a conceptual link to reinforcement learning: principles governing how people learn to act effectively in the world may also govern how people learn to think effectively in their own minds.

The assumption of direct observation of recall progress (t, z_t) is acknowledged as a simplification. Real metamemory likely involves imperfect monitoring. feeling-of-knowing is known to be influenced by cue familiarity and other factors that may not directly reflect recall progress. Future work might relax this assumption by modelling monitoring as noisy observation of the object-level state, yielding a belief-MDP formulation that more faithfully captures the uncertainty inherent in introspection.

7.6 Comparing the Value of Computation Across Domains

Each framework defines a value of computation (VOC) or expected value of control (EVOC) quantifying the benefit of executing a meta-level action. Although the core logic is shared – weighing computational costs against decision benefits – the structure of this tradeoff varies. This section compares the formulations along several dimensions.

7.6.1 Structure of the VOC

All four formulations share a common structure: VOC equals expected benefit minus expected cost. However, the components differ across domains.

Strategy selection

$$\text{VOC}(s, \mathbf{f}) = \mathbb{E}[U(s(\mathbf{f}); \mathbf{f})] - \gamma \cdot \mathbb{E}[T(s, \mathbf{f})] \tag{77}$$

The benefit is the utility of the decision produced by strategy s, while the cost is the opportunity cost of execution time, scaled by the reward rate γ. A key feature of this formulation is that VOC decomposes into separately observable quantities: utility and time are both revealed after strategy execution, enabling independent learning of each component.

Planning

$$\text{VOC}(\mathbf{s}, a_i) = \mathbb{E}[\max_p V(\mathbf{s}', p)] - \max_p V(\mathbf{s}, p) - \lambda \tag{78}$$

The benefit is the expected improvement in decision quality from expanding node i, computed as the difference between expected plan value after expansion and current plan value. The cost is a fixed parameter λ per expansion. This formulation focuses on the marginal value of information: each computation is evaluated by how much it improves the agent's ability to identify the best plan.

Cognitive control

$$\text{EVOC}(b, c) = \mathbb{E}[U(\text{outcome}) - \text{cost}(b, c) + V^*(B_{t+1}) \mid b, c] \tag{79}$$

The benefit comprises the utility of the resulting action plus the value of subsequent belief states. The cost includes both immediate effort and a time penalty. A distinctive feature of this formulation is its recursive structure: the value of a control signal depends on which control signals will be issued in the future, as captured by the term $V^*(B_{t+1})$.

Memory recall

$$V^*(s_t) = P(\text{recall} \mid s_t) \cdot U(\text{recall}) - \mathbb{E}[t_{\text{stop}} - t \mid s_t] \cdot \gamma_{\text{SEARCH}} \tag{80}$$

The benefit is the probability of eventual recall multiplied by the utility of successful retrieval. The cost is the expected remaining search time multiplied by the cost per time step. This formulation takes the form of an optimal stopping problem, with a distinctive feature: the threshold for successful recall is exogenous rather than under the agent's control.

7.6.2 Cost Structure

Domain	Cost formulation	Interpretation
Strategy selection	$\gamma \cdot T$	Opportunity cost (learned reward rate × duration)
Planning	λ	Fixed cost per expansion
Cognitive control	$\text{cost}(b, c) + w^{(T)} \cdot T$	Effort cost (state/signal-dependent) + time penalty
Memory recall	$\gamma_{\text{SEARCH}} \cdot t$	Fixed cost per time step

Strategy selection and memory recall both frame cost in terms of time, but with different structures: strategy selection *learns* the opportunity cost rate from experience, while memory recall treats it as a fixed parameter.

7.6.3 Approximation Strategies

Domain	Approximation	Rationale
Strategy selection	Linear function (learned)	Decomposition enables learning from observable components
Planning	Myopic lookahead	Avoids recursive computation; preserves key tradeoff
Cognitive control	LVOC (learned)	Compact representation with feature-control interactions
Memory recall	Exact (backward induction)	Compact state space makes exact solution tractable

7.7 Summary

	Strategy Selection	Planning	Cognitive Control	Memory Recall
VOC structure	$U - \gamma T$	$\Delta V - \lambda$	$U - \text{cost} + V'$	$pU - \gamma t$
Formal structure	Contextual bandit	Meta-level MDP	Belief-MDP	Meta-level MDP
Sequential	No	Yes	Yes	Yes (stopping)
State observable	Yes	Yes	No	Yes (by assumption)
Learning	Yes (across problems)	No	Yes (across episodes)	No
Cost type	Opportunity cost	Per-expansion	Effort + time	Per-timestep
Approximation	Learned (linear)	Myopic	Learned (LVOC)	Exact

Table 2: Summary comparison of VOC formulations.

Despite these differences, the frameworks share a common normative foundation: *rational metareasoning*. Each asks how a resource-bounded agent should allocate computational effort to maximise expected reward. The VOC provides the currency for this allocation, which is a unified measure of whether a computation is worth performing.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state this is a theoretical contribution that formalises metacognitive theories, not an empirical study claiming performance improvements.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper explicitly states it makes no empirical claims and that translating psychological concepts to computational systems remains speculative.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: This paper formalises existing psychological theories into computational frameworks rather than proving new theoretical results.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results?

   Answer: [NA]

   Justification: The paper does not include experiments; it presents theoretical formalisations of cognitive science theories.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code?

   Answer: [NA]

   Justification: The paper is purely theoretical and does not involve experiments requiring code or data.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details?

   Answer: [NA]

   Justification: The paper does not include experiments.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined?

   Answer: [NA]

   Justification: The paper does not include experiments.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources?

   Answer: [NA]

   Justification: The paper does not include experiments.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform with the NeurIPS Code of Ethics?

Answer: [Yes]

Justification: The theoretical work conforms with ethical guidelines and poses no ethical concerns.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts?

    Answer: [NA]

    Justification: This is foundational theoretical work without direct societal applications or deployment considerations.

11. **Safeguards**

    Question: Does the paper describe safeguards for responsible release?

    Answer: [NA]

    Justification: The paper poses no risks as it neither releases models nor datasets, only theoretical formalisations.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets properly credited?

    Answer: [NA]

    Justification: The paper does not use existing code, data, or model assets.

13. **New assets**

    Question: Are new assets introduced in the paper well documented?

    Answer: [NA]

    Justification: The paper does not release new assets.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include full details?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

15. **Institutional review board (IRB) approvals**

    Question: Does the paper describe potential risks incurred by study participants?

    Answer: [NA]

    Justification: The paper does not involve research with human subjects.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important component of the core methods?

    Answer: [NA]

    Justification: LLMs are not used as a component of the core theoretical formalisations presented.