A Metacognitive Upstream Routing Framework for Accuracy Preservation and Computational Efficiency in Artificial Intelligence Systems

Naavya Shetty
Bachelor of Science in Computer Science and Philosophy
Department of Philosophy
University of Illinois Urbana-Champaign
Illinois, United States
e-mail: shetty.naavyasukesh@gmail.com

***Abstract*—Contemporary Artificial Intelligence (AI) systems often engage every input indiscriminately, resulting in unnecessary computation, unpredictable generalisation, and brittle behaviour on unfamiliar tasks. We present the Preprocessing Metacognitive System (PMS) 2.0, a system-agnostic metacognitive layer that evaluates incoming tasks and decides whether to accept, escalate, or refuse them before invoking any downstream reasoning system. PMS 2.0 seeks to provide interpretability at the level of computational governance - making transparent why the system chooses to engage, escalate, or refuse a task utilising confidence, task complexity, feasibility, novelty, and predicted benefit of escalation – to guide principled routing decisions without modifying downstream models. The system preserves conditional accuracy on escalated inputs, reduces computational load, and operationalises abstention as a first-class outcome, with previously refused tasks contributing to experience-informed efficiency gains. Evaluated across multiple domains and downstream architectures, PMS 2.0 demonstrates that metacognitive preprocessing can improve computational efficiency, reliability, and transparency, providing a practical framework for allocating resources where deliberative computation is most justified.**

Keywords-computational meta-reasoning; resource-rational AI; selective computation; input routing.

I. INTRODUCTION

Contemporary Artificial Intelligence (AI) systems remain remarkably indiscriminate in how they initiate and allocate computational effort. Once presented with an input, most models commit fully to processing it, regardless of whether the task is familiar, whether a lightweight heuristic could resolve it, or whether the system lacks the competence to address it reliably. This unfiltered mode of engagement produces several predictable failure modes: unnecessary computation on trivial problems, unpredictable generalisation on unfamiliar ones, and brittle or opaque behaviour when the system is operating outside its stable reasoning envelope. What is missing in nearly all modern systems is an architectural mechanism that evaluates incoming tasks before deeper computation begins - a preprocessing layer capable of deciding when to proceed, when to escalate, and when to refuse.

In an earlier work, we introduced the first version of such a mechanism: the Preprocessing Metacognitive System (PMS) 1.0 [1]. That preliminary proposal sketched the possibility of a metacognitive bottleneck that screens tasks, leverages previous experience, and restricts unnecessary reasoning. That paper established why deliberative pathways should not be treated as default channels, but it also left two questions open: whether this metacognitive bottleneck must be attached to a specific downstream model and how refusal, deferral, and computational justification could be operationalised beyond concept.

The present paper develops a significantly stronger formulation, PMS 2.0, which extends the original framework in several important ways, as seen in Table 1. First, PMS 2.0 formalises the bottleneck as a system-agnostic routing architecture capable of operating independently of downstream reasoning systems. Second, it introduces a small set of routing diagnostics, or interpretable meta-features, that characterise the relationship between the input and the system's known competence to guide routing decisions. Third, it operationalises computational abstention and refusal as explicit control actions, rather than treating them as failures. Finally, this paper provides the first empirical evaluation of the framework across multiple downstream architectures, demonstrating that metacognitive preprocessing can substantially reduce computational load while preserving conditional accuracy.

In this sense, PMS 2.0 is compatible with large language models, reinforcement-learning agents, symbolic planners, multimodal transformers, and classical machine-learning systems. Its role is not accuracy optimisation or predictive correction, but computational stewardship: deciding when deliberation is warranted, when fast acceptance is sufficient, and when a task must be refused on metacognitive grounds. It operates before these systems are activated, making judgments about the feasibility, novelty, and expected benefit. Simply put, we evaluate PMS 2.0 as a metacognitive control layer whose primary function is to decide when downstream computation is warranted. In this framework, abstention is not treated as an error, but as a deliberate control action.

The remainder of the paper proceeds as follows. Section 2 reviews the theoretical foundations and previous work on metacognition, dual-process reasoning, and resource-rational AI, situating PMS 2.0 within these publications. Section 3 introduces the architecture of PMS 2.0, detailing its modular components, metacognitive features, and system-agnostic routing principles. Section 4 presents the experimental design, including downstream tasks, evaluation metrics, and baseline comparisons. Section 5 reports results and discussion, demonstrating how PMS 2.0 preserves conditional accuracy, reduces computational load, operationalises refusals, and supports

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

TABLE I. COMPARISON OF PMS 1.0 AND PMS 2.0: KEY ARCHITECTURAL AND FUNCTIONAL EXTENSIONS

Feature	PMS 1.0	PMS 2.0
Metacognitive Bottleneck Concept	✓	✓
System-Agnostic Routing	×	✓
Explicit Meta-Features	×	✓
Computational Abstention	conceptual	operational
Empirical Evaluation	none	multi-model

transparent, interpretable decisions. Sections 6 and 7 outline limitations and avenues for future work, including extensions to learned controllers, improved novelty detection, and experience-informed refinement of refusal decisions. Section 8 concludes with broader implications for the implementation of metacognitive preprocessing in scalable AI systems.

II. LITERATURE REVIEW

The revised conception of PMS 2.0 emerges from three complementary research programs: dual-process theories of cognition, the science of metacognition and metareasoning, and formal models of bounded and resource-rational reasoning. Together, these frameworks provide the conceptual tools needed to reinterpret PMS 2.0 not as an intuitive idea, but as a theoretically necessary architectural component for intelligent and resource-limited agents.

A. Dual-Process Theory and Cognitive Control

Dual-process theory remains the most influential organising framework for thinking about the computational trade-offs between fast, pattern-driven cognition and slow, deliberative reasoning. Evans and Stanovich [2] characterise Type 1 processes as fast, automatic, and pattern-based, while Type 2 processes are slow, deliberative, and dependent on working memory. They emphasise that the Type 1 versus Type 2 distinction is most useful when treated as a difference in processing roles rather than as a set of anatomical modules. Debates about the empirical validity of dual-process distinctions often ask whether cognition is truly partitioned into two systems. De Neys [3] argues that dual-process distinctions remain indispensable in the functionalist sense even if the mechanisms themselves are interwoven. This shift toward functionality makes the dual-process vocabulary directly applicable to engineered systems, because designers can map computational primitives to the roles identified by the psychological literature.

B. Metacognition and Metareasoning

The work on metacognition and metareasoning provides the mechanisms by which an agent can inspect, evaluate, and regulate its own cognitive operations. Flavell's [4] early characterisation of metacognition as monitoring and control established the conceptual vocabulary, distinguishing knowledge about cognition from the processes that govern it. Building on that foundation, Cox [5] extended this to artificial systems, arguing that intelligent agents require explicit mechanisms to evaluate uncertainty, detect anomalies, and manage their computational investments. The edited volume by Cox and Raja [6] synthesises computational treatments of metacognition and metareasoning, translating philosophical and psychological concepts into concrete algorithmic concerns such as performance estimation, cost-aware control, and meta-level decision policies. More recently, efforts to formalise metareasoning ontologies have attempted to enumerate the primitive metacognitive functions and further refine the structure of metacognitive control by proposing a validated ontology of metareasoning operations, including confidence evaluation, cost estimation, and progress monitoring to implement self-aware systems [7]. These contributions are essential for designing layered control systems: they show both what needs to be tracked at the meta-level and how those signals can be operationalised.

C. Bounded Rationality and Resource-Rational Agents

Bounded rationality and heuristic decision-making provide the normative and descriptive rationale for preferring cheap, satisfying operations in many real-world environments. Simon's [8] original account framed decision making as a problem constrained by computational resources and environmental structure, introducing satisficing as a practical adaptive strategy. Gigerenzer and Todd [9] later demonstrated empirically how simple heuristics can outperform more complex calculations when resource costs and environmental regularities are considered; their work grounds the design choice to favour fast heuristics under realistic constraints. These strands converge in the resource-rational program, which formalises how an agent should allocate limited computation to maximise expected utility given processing costs.

D. Recent Work

There is a growing body of computational work that addresses how to learn or adapt metacognitive control. Schaeffer's [10] algorithmic treatments of metacognitive reinforcement learning show how a meta-level can learn escalation policies based on experience, defining metacognition as a learnable control problem rather than a set of fixed rules. Parallel research into practical metareasoning for modern models, such as recent demonstrations of value-of-computation ideas applied to large language models [11], indicates how cost-benefit reasoning can be scaled to modern and expensive systems.

Recent formalisations strengthen this foundation. Lieder and Griffiths [12] propose resource-rational analysis as a normative framework to optimize the use of computational resources. Russell and Wefald [13] extend this to artificial intelligence, offering formal principles for rational metareasoning that explicitly quantify accuracy-cost trade-offs. Rumana [14]

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

presents a synthesis of dual-process theory and metacognitive control, arguing that metacognitive mechanisms determine when Type-2 reasoning should intervene in Type-1 processing.

In the most recent computational turn, authors have attempted to map dual-process roles onto contemporary neuro-symbolic and machine learning systems. Gronchi and Perini [15] argue that dual-process distinctions naturally map onto neuro-symbolic architectures, where sub-symbolic networks provide intuitive judgments and symbolic systems support deliberation. Gronchi et al. [16] additionally show that inhibitory control mediates transitions between fast and slow processing in human cognition. These results suggest that engineering a gating mechanism is not only psychologically plausible but also architecturally sensible for hybrid AI systems.

On the tool side, scalable similarity search libraries like Facebook AI Similarity Search (FAISS) make experience-based novelty estimation tractable in large embedding spaces [17]. Taken together, these literatures provide a unified theoretical and practical toolkit for designing metacognitive preprocessors that are both principled and implementable.

III. Proposal

Building on the conceptual and empirical foundations summarised above, we propose PMS 2.0: a system-agnostic preprocessing metacognitive layer that makes principled, experience-informed routing decisions prior to invoking any downstream deliberative system. It also explicitly records refused inputs and periodically re-evaluates them after sufficient experience has accumulated, allowing the system to accept similar tasks more efficiently in the future.

The proposal integrates three core theoretical commitments: (1) a functional dual-process mapping that separates fast appraisal from deliberate inference, (2) explicit meta-level monitoring and cost-aware control, and (3) resource-rational decision rules that prioritise expected computational value.

A. Principles and goals

PMS 2.0 is designed around five guiding principles inherited from the literature. First, it implements a functional separation between rapid meta-appraisal and slow deliberation [2][3]. Second, it operationalises metacognition as monitoring plus control, tracking interpretable meta-features such as confidence, complexity, and feasibility rather than producing task-level predictions [4][5]. Third, PMS 2.0 evaluates the expected value of further computation in a resource-rational manner, comparing the expected improvement against cost [12][13]. Fourth, it uses experience-based similarity and novelty measures to detect unfamiliar inputs using scalable nearest-neighbour tools [17]. Fifth, the system emphasises transparency and decisional traceability: every routing decision is accompanied by meta-features and a human-readable reason, which supports interpretability and auditability [7].

B. Interpretable Meta-Features

A central design goal of PMS 2.0 is interpretability at the meta-decision level. In this work, interpretability refers to the property that the system's routing decisions can be directly explained in terms of human-understandable variables rather than opaque internal activations. The routing decisions in PMS 2.0 are governed by a small set of interpretable meta-features that describe the relationship between the incoming task and the system's accumulated experience. These features serve as meta-level diagnostics rather than task-level predictions. Their purpose is not to solve the task itself but to estimate whether further computation is justified.

- **Confidence** estimates the likelihood that a lightweight heuristic response would produce a reliable result. In practice, it reflects the consistency of the input with previously encountered patterns and the stability of heuristic outputs across similar cases. High confidence suggests that immediate acceptance or inexpensive processing may be sufficient.
- **Complexity** approximates the structural difficulty of the input relative to tasks the system has previously handled. This may reflect factors such as input length, compositional structure, or the number of interacting elements that must be processed. Higher complexity indicates that the task may require more deliberate reasoning or deeper computation.
- **Feasibility** represents the system's estimated ability to process the task successfully based on its existing competence. It is derived from similarity to previously solved cases and historical performance on related inputs. Low feasibility signals that the system may lack sufficient experience or capability to produce a reliable outcome.
- **Novelty** measures the distance between the current input and previously encountered tasks stored in the experience buffer. This signal is computed through similarity search in the embedding space and serves as an indicator of out-of-distribution inputs. High novelty encourages conservative behaviour, such as escalation or refusal.
- **Predicted Benefit of Escalation** ($\hat{\Delta}$) estimates the expected improvement in outcome quality that would result from invoking the downstream System-2 model. Escalation is therefore justified only when the anticipated benefit exceeds the expected computational cost.

Because these quantities are explicitly computed and logged, every routing decision produced by the MetaController can be traced back to a concrete set of meta-feature values. This transparency allows both developers and external auditors to inspect why a task was accepted, escalated, or refused without examining the internal structure of the downstream reasoning system. In this sense, interpretability in PMS 2.0 does not arise from explaining the reasoning of the downstream model itself, but from making the decision to invoke that reasoning transparent and inspectable.

C. Architectural Components

Concretely, PMS 2.0 consists of the following modular components, each directly motivated by the literature:

1) **Task encoder:** A modality-agnostic encoder produces compact embeddings that represent the structural features of the input. Representational choices follow the emphasis of

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

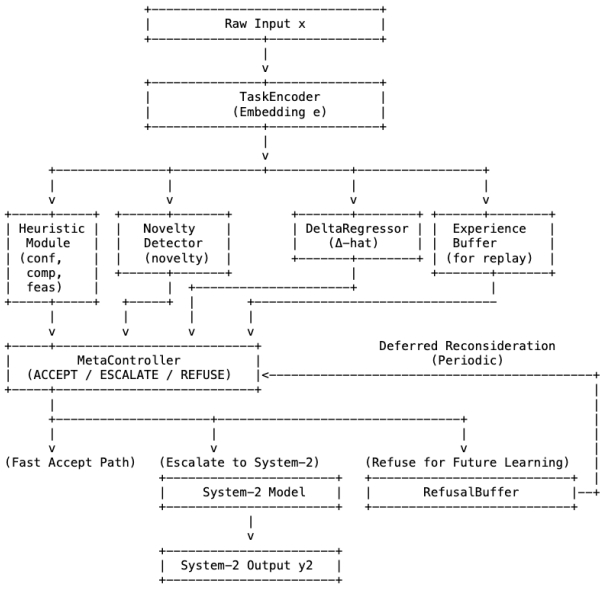

Figure 1. High-level Control Flow Representation

Russell and Norvig's [18] on appropriate feature spaces for downstream reasoning. The encoder normalises embeddings to permit consistent similarity computations and experience lookup.

2) **Heuristic meta-feature extractor:** A small, fast module produces interpretable meta-features, such as confidence, complexity, and feasibility, derived only from the embedding. This design follows Flavell's [4] monitoring/control distinction and is informed by Cox's [5] mapping of meta-level primitives to algorithmic constructs. The features are intentionally lightweight, so they can be produced cheaper than any downstream task computation.
3) **Novelty and experience module:** Using FAISS or equivalent nearest-neighbour search, the system evaluates the distance to stored experiences to yield a novelty score. This score operationalises epistemic unfamiliarity and provides a principled signal for conservative behaviour when the system encounters out-of-distribution inputs [17].
4) **Delta or value regressor:** A learned regressor estimates the expected benefit ($\hat{\Delta}$) of invoking the downstream system. This module formalises the resource-rational imperative described by Lieder and Griffiths [12] and Russell and Wefald [13]: escalation is justified only when the expected gain exceeds the anticipated cost of computation. Training uses logged escalations to supervise $\hat{\Delta}$ learning, in effect implementing a form of metacognitive reinforcement learning [10].
5) **MetaController (decision policy):** For transparency and robustness, the first instantiation of the controller is rule-based:
   - ACCEPT when $\hat{\Delta} \leq 0$ and feasibility is high
   - ESCALATE when $\hat{\Delta}$ > threshold or novelty is high
   - REFUSE when feasibility or confidence are extremely low
   - Refused inputs are stored in a RefusalBuffer, which preserves embeddings, meta-features, and the original input. Periodically, these entries are reconsidered: novelty and $\hat{\Delta}$ are recalculated, and the MetaController may update previous REFUSE decisions. This mechanism allows PMS 2.0 to gradually accept inputs that were initially deferred, improving overall computational efficiency while maintaining system-agnostic metacognitive control. The REFUSE action enables the most distinctive function of PMS 2.0: the ability to say no to inputs when the system cannot guarantee reliable processing, while also influencing further learning. This choice relies on the interpretability emphasised in metareasoning ontologies while leaving open the option of replacing the rule-based controller with a learned policy once enough experience is available [7][10].

The overall routing architecture of PMS 2.0 is illustrated in Figure 1, which shows how incoming inputs are encoded, evaluated using meta-features and novelty estimation, and subsequently routed by the preprocessing router MetaController to either ACCEPT, ESCALATE, or REFUSE pathways.

D. Design justification and theoretical fit

This architecture is the natural engineering embodiment of the earlier surveyed literature. The functional dual-process mapping prescribes a fast appraisal stage; metareasoning and metacognition provide the primitives needed to produce that appraisal; bounded rationality and resource-rational analysis provide the decision criterion that converts appraisal into action. Experience-based novelty detection operationalises cautious behaviour in the presence of distribution shift, a practice recommended by contemporary neuro-symbolic mappings and empirical meta-analyses [16][15]. Importantly, this design choice intentionally separates interpretability from task-level prediction, and the system is deliberately agnostic about the identity or internals of System-2. The adapter interface allows any downstream model to be invoked as a black box, which not only respects constraints common in production systems where internals are proprietary or too large to modify, but also instead provides an interpretable control layer that governs when such models should be invoked. By grounding routing decisions in explicit meta-features, the system produces auditable decision traces that support debugging, accountability, and principled resource allocation.

E. Operational behaviour and evaluation

Practically, PMS 2.0 functions as an input router. For each input, the encoder and meta-features are computed; novelty is assessed with respect to an experience buffer; $\hat{\Delta}$ is predicted; the MetaController issues ACCEPT, ESCALATE, or REFUSE; if ESCALATE is chosen, the system calls System-2 and records the result, feeding it back to the experience buffer and using it to refine $\hat{\Delta}$. Periodically, the RefusalBuffer is revisited: novelty and $\hat{\Delta}$ are recalculated for refused inputs, and the MetaController may revise prior REFUSE decisions. The

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

success criteria are twofold: measurable reductions in expensive System-2 calls (computational savings) and preservation of correctness on escalated cases (no unjustified degradation). Efficiency gains are enhanced because previously refused inputs can later be processed using learned experience, reducing repeated System-2 calls. Secondary criteria include improved transparency (decision logs) and meaningful refusal behaviour (declines those that indicate high novelty or low feasibility).

F. Practical considerations

Several design choices are directly derived from the literature and practical constraints. First, meta-features should be interpretable and cheap, so they do not erase the savings introduced by avoiding System-2 calls. Second, the DeltaRegressor must be trained from a representative set of escalations; this creates a bootstrap period during which the controller relies more heavily on conservative thresholds. Third, because novelty metrics based on embedding distances can be brittle in some multimodal or highly structured domains, future work should investigate learned density estimators or contrastive representations as alternatives to better detection of semantic novelty [17][19].

IV. EXPERIMENTATION

This section describes the experimental setup used to evaluate PMS 2.0, including implementation details, evaluation protocols, and baseline comparisons.

A. Implementation Details

The PMS 2.0 is implemented in PyTorch with modular subcomponents designed for easy replacement or extension. The design principles guiding the implementation reflect the architectural commitments described above: system-agnosticism, transparency, and strict separation of meta vs task processing.

Figure 2 shows the modular software organisation of PMS 2.0, which outlines the directory structure used to separate the metacognitive modules, System-2 adapter, training utilities, and experience buffers.

```
pms/
├── models/
│   ├── task_encoder.py
│   ├── heuristic_module.py
│   ├── delta_regressor.py
│   ├── meta_controller.py
│   └── system2_adapter.py
├── core/
│   └── experience_buffer.py
│   └── refusal_buffer.py
├── utils/
│   └── novelty_detector.py
├── router/
│   └── router.py
└── training/
    └── supervised_trainer.py
```

Figure 2. Project Directory Structure

- Each module corresponds to a theoretical construct. For example, the HeuristicModule implements the fast metacognitive appraisal system and outputs only meta-features; the System2Adapter wraps any arbitrary model through a uniform interface.
- **Experience Buffer and FAISS Integration:** The ExperienceBuffer leverages FAISS for large-scale similarity search, allowing efficient novelty estimation and replay sampling. Each stored experience contains:
  - embedding
  - confidence, complexity, feasibility
  - novelty
  - System-2 output (if escalated)
  - $\hat{\Delta}$ (actual benefit when calculable)

  This reflects the metacognitive RL framework in Schaeffer [10], but, more importantly, PMS 2.0 does not require task labels. The system learns entirely from meta-level outcomes, not predictions, further distancing it from the supervised metacognition design in my previous work [1].
- **System2 Abstraction Layer**: The System2Adapter ensures that PMS 2.0 does not assume anything about the underlying reasoning system. Any callable model that accepts an input tensor and returns an output can be integrated. This keeps the metacognitive layer decoupled and portable.
- **Training Infrastructure:** The supervised trainer prepares the data for training the DeltaRegressor. The novelty, confidence, complexity, feasibility, and embeddings are all fed as features; $\hat{\Delta}$ is used as the target when available. The system can optionally use the accumulation of online experiences, expanding its applicability to continuous learning settings.

B. Experimental Setup

To evaluate PMS 2.0, two complementary goals guide the experimental design:

1) Does PMS 2.0 improve system reliability, transparency, and the ability to reject malformed inputs?
2) Does PMS 2.0 reduce computational load by diverting low-value cases away from System-2 models?

We tested PMS 2.0 as a preprocessing layer in front of an arbitrary System-2 model. In line with its system-agnostic design, the experiments do not rely on a particular choice of System-2. Instead, we evaluated PMS 2.0 on different downstream architectures. Evaluation also tracks the system's ability to revisit previously refused inputs, measuring acceptance rate and computational savings on formerly deferred tasks.

C. Systems Under Evaluation

We evaluate PMS 2.0 paired with:

- A medium-scale transformer model (for text inputs)
- A ResNet classifier (for image inputs)
- A symbolic reasoning engine (for structured tasks)

This diversity supports the claim that PMS 2.0 is task-independent and plug-compatible with modern AI systems.

D. Data and Task Conditions

For each System-2 model, the inputs are divided into:

- Routine / in-distribution cases
- Edge-case high-complexity tasks

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

- Adversarily perturbed or malformed inputs
- Completely novel inputs (unseen embedding clusters)

These categories allow for the evaluation of PMS 2.0's routing decisions: can it meaningfully refuse, accept, or escalate tasks.

E. Training the Delta Regressor

During preliminary runs, System-2 is queried on a subset of inputs to compute the actual deltas to train the DeltaRegressor. The rest of the PMS 2.0 components are frozen. This reflects the metacognitive RL regime in Schaeffer [10], but without requiring action-level reward functions.

F. Evaluation Metrics

We evaluate the following metrics:

1) Escalation Rate – How often inputs are forwarded to System-2.
2) Refusal Accuracy – Ability to correctly reject unprocessable or harmful inputs.
3) Coverage Preservation – Fraction of correct System-2 outputs preserved when PMS 2.0 is inserted.
4) Computational Savings – Reduction in System-2 calls.
5) Transparency Metrics – Distributions of confidence, novelty, feasibility, and $\hat{\Delta}$ for each routing class.
6) Error Mitigation – Reduction in System-2 hallucinations or unsafe outputs relative to baseline.
7) Deferred Acceptance Rate – fraction of previously refused inputs that are eventually escalated and accepted successfully.

G. Baselines

We compare against:

- System-2 alone (no PMS 2.0).
- Simple threshold-based gating system.
- Heuristic-only version of PMS 2.0 (no $\hat{\Delta}$ model).

This ensures that the improvements are attributable to the metacognitive layer rather than trivial heuristics.

V. Results | Discussion

The evaluation examined how well PMS 2.0 functions as a system-agnostic upstream routing layer that governs when a more expensive or more capable downstream AI system (System-2) should be invoked. PMS 2.0 was assessed on four primary dimensions: routing efficiency, conditional and unconditional accuracy, error exposure, and transparency of decisions. In this evaluation, computational cost is approximated by the number of invocations of the downstream System-2 model, which represents the dominant source of computational expense in most AI pipelines. Comparisons were conducted against three baselines: unconditional System-2 invocation, a simple threshold-based gating system, and a heuristic-only variant of PMS without learned benefit estimation.

Experiments were conducted on 200 held-out synthetic inputs using a lightweight and moderately error-prone System-2 model. This controlled setting was intentionally selected to isolate the effects of metacognitive routing from downstream model capacity. All systems were evaluated under identical conditions, with PMS 2.0 operating strictly upstream and without access to the System-2 internals.

Across the evaluation set, as shown in Table 2, PMS 2.0 reduced System-2 calls from 120 to 17, corresponding to an 85.8% reduction in downstream computation relative to unconditional escalation. When PMS 2.0 authorised escalation, conditional accuracy, defined as correctness on escalated inputs only, remained comparable to the baseline systems. By construction, PMS 2.0 introduces a distinction between conditional accuracy and unconditional accuracy, where unconditional accuracy is measured over all inputs, with refusals conservatively counted as incorrect. As coverage decreases, unconditional accuracy correspondingly declines, reflecting intentional abstention rather than degraded prediction quality.

This distinction is central in interpreting the behaviour of PMS 2.0. The system is not designed to maximise unconditional accuracy, as doing so would require escalating on every input. Instead, PMS 2.0 explicitly regulates when deliberative computation is warranted. Inputs that are not escalated are refused rather than guessed, making abstention a first-class outcome rather than an implicit failure. In safety-critical or resource-constrained settings, such behaviour can be preferable to unexamined engagement.

From a computational perspective, this reduction is significant because System-2 invocation represents the most expensive stage of the processing pipeline. By filtering inputs upstream, PMS 2.0 limits expensive reasoning to a small subset of cases where additional computation is predicted to be beneficial. In the present evaluation, this results in over six-fold fewer System-2 calls while maintaining comparable conditional accuracy on the escalated tasks. These results provide empirical support for the central hypothesis of the paper: that metacognitive preprocessing can substantially reduce computational expenditure without degrading performance on tasks that genuinely require deeper reasoning.

Efficiency metrics further highlight the benefits of metacognitive routing. Although unconditional System-2 invocation and threshold-based gating achieve full coverage, they incur maximal downstream cost because every input is forwarded to the expensive reasoning system and, in the case of threshold gating, substantially increase error exposure. In contrast, PMS 2.0 significantly reduces downstream computation while also lowering the absolute number of downstream errors by selectively refusing inputs likely to result in misclassification. As a result, PMS 2.0 not only reduces computation but also actively limits the propagation of downstream failures by preventing the System-2 model from engaging with inputs outside its competence.

The heuristic-only PMS variant achieved the highest conditional accuracy among all systems, suggesting that static heuristics can be effective in constrained settings. However, this variant lacks adaptive benefit estimation and exhibits a higher error exposure than the full PMS 2.0 system. This result highlights a key trade-off: learned benefit estimation prioritises conservative escalation and cost reduction, sometimes at the expense of coverage. PMS 2.0 makes this trade-off explicit

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

TABLE II. EVALUATION RESULTS: COMPARISON OF ROUTING STRATEGIES ACROSS ACCURACY, COVERAGE, AND SYSTEM-2 UTILISATION

System	Cond. Acc.	Uncond. Acc.	Coverage	S2 Calls	Savings	Acc./S2 Call
System-2 Only	0.0750	0.0750	100.0%	120	0.0%	0.0750
Threshold Gate	0.0750	0.0750	100.0%	120	0.0%	0.0750
Heuristic-Only PMS	**0.0886**	0.0583	65.8%	79	34.2%	**0.0886**
Full PMS 2.0	0.0588	0.0083	14.2%	17	**85.8%**	0.0588

and measurable.

In addition to routing decisions, PMS 2.0 produced fully transparent decision logs. Each ACCEPT, ESCALATE, or REFUSE decision was accompanied by the underlying values of metacognitive characteristics and a human-readable explanation. This transparency was achieved without modifying or inspecting System-2, supporting the claim that PMS 2.0 can function as an external accountability layer for black-box models.

A key lesson from these experiments is that aggressive computational savings can substantially reduce coverage if the benefit estimate is conservative. Although PMS 2.0 achieved the greatest reduction in System-2 usage and downstream error exposure, it underperformed the heuristic-only variant in conditional accuracy. This failure mode motivates future work on adaptive threshold calibration, online learning, and multi-tier routing architectures. Overall, the results confirm that PMS 2.0 functions as intended: reducing unnecessary computation while preserving correctness on authorised escalations, and operationalising abstention as a principled metacognitive decision rather than a failure of prediction.

VI. LIMITATIONS

Although PMS 2.0 demonstrates significant computational savings and principled abstention behaviour, several limitations constrain the scope of the current results.

- **Synthetic evaluation setting:** The evaluation was performed on synthetic data using a lightweight synthetic System-2 model. This choice enabled controlled experimentation and clear attribution of effects, but it limits the ecological validity of the results. Real-world deployments will involve substantially more complex tasks, heterogeneous input distributions, and significantly more expensive downstream models. Although routing principles are expected to generalise, empirical trade-offs between coverage, accuracy, and efficiency may differ across domains.
- **Task-specific benefit definition:** PMS 2.0 relies on a user-defined benefit function to train the DeltaRegressor. This design provides flexibility, but it also restricts its applicability to settings where downstream quality can be explicitly quantified. In domains where performance objectives are ambiguous or multi-dimensional, benefit estimation may be difficult to specify reliably, increasing sensitivity to miscalibration and conservative routing behaviour.
- **Rule-based MetaController:** The current MetaController uses fixed, manually selected thresholds to determine ACCEPT, ESCALATE, and REFUSE actions. Although this choice improves transparency and interpretability, it limits adaptability across tasks and environments. As observed in the experiments, conservative thresholding can lead to excessive refusals and reduced conditional accuracy, particularly when benefit estimates are uncertain.
- **Simplified novelty estimation:** Novelty detection is implemented using distance-based similarity in the embedding space. This approach is effective for small experience buffers, but it may inadequately capture semantic or functional novelty in highly structured, multimodal, or high-dimensional input spaces. As a result, novelty signals may be coarse or unreliable in more complex deployments.
- **Cold-start effects and limited early experience:** PMS 2.0 requires an experience buffer populated through interaction with System-2 to train the DeltaRegressor. During early operation, benefit estimates are, therefore, based on sparse data, which can amplify conservative routing decisions. Although the system logs refused inputs in a RefusalBuffer for deferred reconsideration and gradual incorporation into experience, early refusals may still be weakly informed until sufficient interaction data accumulate.

VII. FUTURE WORK

The experimental results highlight several concrete directions for improving PMS 2.0, particularly in response to the observed trade-off between aggressive computational savings and conditional accuracy.

1) **Adaptive threshold calibration and learned control:** The experiments show that the current rule-based MetaController produces strong computational savings but may be overly conservative in escalation decisions. This motivates replacing the fixed rule-based MetaController with an adaptive or learned controller. A policy trained via offline decision modelling or reinforcement learning could dynamically adjust escalation and refusal thresholds to balance efficiency against conditional accuracy, while retaining interpretability through explicit explanation mechanisms.
2) **Refined benefit estimation:** The system's reliance on a task-specific benefit function provided by the user enables flexibility, but it also increases sensitivity to conservative or miscalibrated benefit estimates, as reflected in reduced coverage. Future work should explore domain-agnostic or learned surrogate objectives that better align predicted benefit with downstream accuracy gains, reducing over-refusal without reverting to indiscriminate escalation.
3) **Multi-tier routing architectures:** The sharp coverage drop observed under aggressive routing suggests that binary escalation decisions may be overly coarse for many real world environments. Extending PMS 2.0 to support multi-level routing across several downstream models with graded cost

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

and capability profiles could mitigate this effect, allowing conservative refusals to be replaced with intermediate, lower-cost escalation options.

4) **Online learning and refusal reuse:** PMS 2.0 currently trains its benefit estimator offline and logs refusals without fully exploiting them during the evaluation. A fully online variant could continuously update metacognitive estimates, revisit previously refused inputs, and incorporate explicit refusal regret signals. This would allow conservative refusals to be corrected over time, improving conditional accuracy while preserving computational savings, and enable the system to progressively expand its competence over time.

These directions aim to transform PMS 2.0 from a static routing mechanism into a continuously adapting metacognitive control layer capable of allocating computational resources efficiently across a wide range of AI systems.

VIII. Conclusion

The Preprocessing Metacognitive System presented in this work provides a theoretically grounded and implementable blueprint for a metacognitive preprocessing layer. It synthesises insights from dual-process theory, computational metareasoning, and resource-rational analysis into a modular architecture designed for transparency and practical integration. In doing so, it operationalises the normative insight that computation itself is a scarce resource and that intelligent systems should explicitly decide when additional computation is justified.

By placing metacognitive evaluation at the preprocessing stage, PMS 2.0 enables intelligent routing without inspecting or modifying downstream models. The system is fully system-agnostic, interpretable, and capable of principled abstention. The empirical results demonstrate that this approach can substantially reduce downstream computational cost while preserving correctness on authorised escalations and providing complete auditable decision explanations.

Importantly, PMS 2.0 treats rejections as provisional rather than terminal. Declined inputs are logged and may be reconsidered as additional experience accumulates, allowing previously deferred cases to be escalated efficiently when predicted benefit increases. This deferred handling reframes abstention as a dynamic, experience-informed mechanism rather than a static rejection policy, supporting gradual improvement in routing decisions over time without redundant computation.

More broadly, the architecture illustrates how metacognition can function as a practical control layer for modern AI systems. Rather than relying exclusively on larger models or increased data, PMS 2.0 demonstrates that explicitly reasoning about when to compute can meaningfully improve system efficiency and reliability. Although the current evaluation is limited in scope, the results suggest that metacognitive control at the preprocessing-level offers a promising and extensible approach to managing cost, risk, and accountability in scalable AI pipelines.

References

[1] N. Shetty, "Metacognition-driven preprocessing for optimized artificial intelligence performance", in *Proceedings of the Seventeenth International Conference on Advanced Cognitive Technologies and Applications (COGNITIVE 2025)*, IARIA Press, 2025, ISBN: 978-1-68558-260-9. [Online]. Available: https://www.thinkmind.org/library/COGNITIVE/COGNITIVE_2025

[2] J. S. B. T. Evans and K. E. Stanovich, "Dual-process theories of higher cognition: Advancing the debate", *Perspectives on Psychological Science*, vol. 8, no. 3, pp. 223–241, 2013. DOI: 10.1177/1745691612460685

[3] W. De Neys, "Dual-process theory 2.0", *Routledge*, 2017, Edited volume.

[4] J. H. Flavell, "Metacognition and cognitive monitoring: A new area of cognitive-developmental inquiry", *American Psychologist*, vol. 34, no. 10, pp. 906–911, 1979. DOI: 10.1037/0003-066X.34.10.906

[5] M. T. Cox, "Metacognition in computation: A selected research review", *Artificial Intelligence*, vol. 169, no. 2, pp. 104–141, 2005. DOI: 10.1016/j.artint.2005.10.004

[6] M. T. Cox and A. Raja, Eds., *Metareasoning: Thinking about thinking*. Cambridge, MA: MIT Press, 2011. DOI: 10.7551/mitpress/9780262014809.001.0001

[7] M. F. Caro, M. T. Cox, and R. E. Toscano-Miranda, "A validated ontology for metareasoning in intelligent systems", *Journal of Intelligence*, vol. 10, no. 4, p. 113, 2022. DOI: 10.3390/jintelligence10040113

[8] H. A. Simon, "Rational choice and the structure of the environment", *Psychological Review*, vol. 63, no. 2, pp. 129–138, 1956.

[9] P. M. Todd, G. Gigerenzer, and the ABC Research Group, *Simple Heuristics That Make Us Smart*. Oxford University Proceedings, Jan. 1999.

[10] R. Schaeffer, "An algorithmic theory of metacognition in minds and machines", arXiv:2111.03745, 2021. [Online]. Available: https://arxiv.org/abs/2111.03745

[11] C. N. De Sabbata, T. R. Sumers, B. AlKhamissi, A. Bosselut, and T. L. Griffiths, "Rational metareasoning for large language models", arXiv:2410.05563, 2024. [Online]. Available: https://arxiv.org/abs/2410.05563

[12] F. Lieder and T. L. Griffiths, *Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources*. Behavioral and Brain Sciences, 2018.

[13] S. J. Russell and E. Wefald, "Principles of metareasoning", *Artificial Intelligence*, vol. 49, no. 1–3, pp. 361–395, 1991. DOI: 10.1016/0004-3702(91)90009-U

[14] A. Rumana, "Metacognitive control in single- vs. dual-process theory", *Thinking and Reasoning*, vol. 29, no. 2, pp. 177–212, 2023. DOI: 10.1080/13546783.2022.2047106

[15] G. Gronchi and A. Perini, "Dual-process theories of thought as potential architectures for developing neuro-symbolic ai models", *Frontiers in Cognition*, vol. 3, pp. 1–5, 2024. DOI: 10.3389/fcogn.2024.1356941

[16] G. Gronchi, G. Gavazzi, M. P. Viggiano, and F. Giovannelli, "Dual-process theory of thought and inhibitory control: An ale meta-analysis", *Brain Sciences*, vol. 14, no. 1, p. 101, 2024. DOI: 10.3390/brainsci14010101

[17] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus", *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019. DOI: 10.1109/TBDATA.2019.2921572

[18] S. J. Russell and P. Norvig, *Artificial intelligence: A modern approach*, 4th ed. London: Pearson, 2021.

[19] Sun, "Contents", in *The Cambridge Handbook of Computational Cognitive Sciences* (Cambridge Handbooks in Psychology), Cambridge Handbooks in Psychology. Cambridge University Press, 2023, pp. v–viii.

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org