Magi × GPT-6: A Hyper-Reasoning Router for Efficient Expert Orchestration

The Singularity (Asari & Payton)
isonpayton@gmail.com

September 18, 2025

Abstract

We present **Magi × GPT-6**, a hyper-reasoning router that composes a small controller (*Gate*), three specialized experts (*Magi-A/B/C*), a lightweight *Judge* (arbiter), and an optional *Canon* model into a single, cost-aware inference system. The Gate classifies tasks, budgets tokens, and selects experts; the experts answer in parallel under a unified JSON schema; a constrained micro-debate resolves disagreements; the Judge fuses candidates into a single answer with calibrated confidence; and escalation to Canon is triggered only when uncertainty or risk exceeds thresholds. The design unifies conditional computation (MoE/Switch), tool-augmented reasoning, retrieval-augmented generation, and LLM-as-Judge into an end-to-end architecture with explicit *evidence objects* and *uncertainty vectors*. We provide a formal cost model, deployment notes targeting *H200* GPUs, and a reproducible evaluation protocol measuring both answer quality and *token-normalized* efficiency.

1 Introduction

Scaling language model capability by densifying parameters quickly encounters prohibitive inference cost. Conditional computation and routing promise capacity without commensurate FLOPs by activating only relevant experts per input. While token-level MoE layers expand capacity within a single network, many real-world applications benefit from *system-level routing* across heterogeneous skills (logic/math/code, retrieval-heavy analysis, and editorial/creative refinement).

We propose **Magi × GPT-6**, a practical hyper-reasoning router that: (i) selects a subset of specialized experts under explicit token and risk budgets; (ii) compels experts to emit structured *evidence*; (iii) resolves conflicts with a short, bounded micro-debate; (iv) fuses results via a lightweight Judge; and (v) escalates to an optional large Canon model only when necessary. The result is a tractable quality–cost Pareto on commodity H200 nodes.

Contributions.

[leftmargin=*, itemsep=2pt]

1. A unified JSON schema for expert outputs (answer, rationale, uncertainty vector, evidence list, timing).
2. A micro-debate protocol and Judge that fuse multiple candidates into one answer with calibrated confidence.
3. A cost-aware router that escalates to a large Canon model only when uncertainty or risk exceeds thresholds.

Flow: User → Gate → Selected Magi (parallel) → Judge → (optional) Canon → Final.
Budgets: Gate sets `max_new_tokens`, temperature, and timeouts per expert.
Evidence: Experts emit unit-test results, citations, or tool outputs as first-class objects.
Escalation: Triggered iff Judge confidence $< \tau$ or risk $> \rho_{\max}$ after micro-debate.

Figure 1: High-level architecture and decision points.

4. A *reproducible evaluation protocol* reporting accuracy, citation validity, pass@k, escalation rate, and token-normalized latency/cost on standard benchmarks.

5. A deployment recipe for H200-class GPUs using SDPA/Flash attention backends.

2 Related Work

Conditional computation and MoE ([**?**, **?**]) demonstrate compute-efficient scaling via token-wise expert routing. **MRKL-style** modular systems route to symbolic tools and knowledge sources ([**?**]). **Reason-and-act** and **tool use** (e.g., ReAct, Toolformer, Program-of-Thoughts) interleave reasoning with tool calls ([**?**, **?**, **?**]). **RAG** reduces hallucinations by conditioning on retrieved documents ([**?**]). **Self-reflection** and **LLM-as-Judge** improve reliability and evaluation ([**?**, **?**]). Our work integrates these paradigms into a single controller–experts–judge architecture with explicit evidence tracking and uncertainty calibration.

3 System Overview

3.1 Components

[leftmargin=*, itemsep=3pt]

- **Gate (Brainstem)**: small controller (7–13B) that classifies the task (`math|code|research|creative|mixed`) predicts hardness/risk, allocates token/temperature budgets, and selects experts.
- **Magi-A (Logic/Math/Code)**: tool-using; must provide runnable code/tests or arithmetic checks when applicable; refuses speculation.
- **Magi-B (Knowledge/Analysis)**: retrieval-first; emits citations and evidence; penalizes ungrounded claims.
- **Magi-C (Creative/Refactor)**: clarity, style, and counterfactual framing without altering verified facts.
- **Judge (Arbiter)**: small model (3–7B) that fuses candidates via rules prioritizing evidence and test results; emits {`final, why, confidence`}.
- **Canon (Optional)**: large model (e.g., 120B in MXFP4) invoked only when confidence is low or risk is high.

3.2 Unified Expert Schema

Every expert returns the following JSON object:

```
{
  "answer": "...",
  "rationale": "...(<= 10 lines)...",
  "confidence": 0.0,
  "uncertainty": {"calc": 0.0, "facts": 0.0, "speculation": 0.0},
  "evidence": [
    {"type":"tool","name":"python","result":"tests passed: 12/12"},
    {"type":"retrieval","source":"URL-or-doc-id","snippet":"..."}
  ],
  "timing_ms": 0
}
```

3.3 Micro-Debate Protocol

If candidates materially disagree: **Round 1** (64 tokens each): A asserts claim + test/evidence; B counters with sources. **Round 2** (64 tokens each): a single concession or refutation each. Judge decides; if still ambiguous and impact high, escalate to Canon.

4 Routing Logic and Objective

Objective. Given input x, Gate selects expert set $E \subseteq \{A, B, C\}$ and budgets b to minimize expected token cost while meeting quality and risk constraints:

$$\min_{E,b,\text{debate}} \mathbb{E}[T_{\text{total}}] \quad \text{s.t.} \quad \Pr(\text{err}(x) \leq \epsilon) \geq 1 - \delta, \quad \rho(x) \leq \rho_{\max}. \tag{1}$$

Expected total tokens:

$$\mathbb{E}[T_{\text{total}}] = T_{\text{gate}} + \sum_{e \in E} T_e + T_{\text{judge}} + p_{\text{debate}} T_{\text{debate}} + p_{\text{esc}} T_{\text{canon}}. \tag{2}$$

Calibration. The Judge maps expert uncertainties to a calibrated confidence $\hat{c} \in [0, 1]$. The Gate updates its escalation threshold τ online via bandit feedback against pass@k, citation-validity, and test success.

5 Implementation Details (H200-Oriented)

Backends. Use PyTorch SDPA with Flash-style kernels on Hopper-class GPUs; enable backend selection at runtime. Keep several GiB of headroom to avoid allocator thrash during parallel expert runs.

Placement (2×H200). GPU0: Gate, Judge, Magi-A. GPU1: Magi-B, Magi-C. Canon loads lazily on first escalation and can share GPU1. Use left-padding and batching to run experts in parallel under tight token budgets.

Prompts (System Snippets).

[leftmargin=*, itemsep=2pt]

- **Magi-A**: "Derive → verify with python/tests; refuse speculation; return schema exactly."
- **Magi-B**: "Propose 2–4 searches; ground claims with citations; summarize then answer; return schema."
- **Magi-C**: "Clarify or refactor; preserve verified facts; offer alt framings if confidence < 0.6; return schema."
- **Judge**: "Fuse candidates; prefer verifiable evidence; penalize uncited claims; one micro-debate max; output {`final, why, confidence`}."

6 Evaluation Protocol

Benchmarks. Math/Logic: GSM8K; Code: HumanEval; Knowledge/Analysis: MMLU, GPQA. Report accuracy or pass@k as appropriate.

System Metrics. (medians with IQR) Escalation rate p_{esc}; token-normalized latency; evidence validity (fraction of claims with supporting sources); test success (Magi-A unit tests); Judge agreement with human raters on open-ended tasks.

Ablations. Remove micro-debate; vary debate budget (32–128). Disable Magi-C to measure clarity/readability impact. Sweep Gate threshold τ to trace quality–cost Pareto. Swap retrieval backends to measure citation drift.

7 Safety, Alignment, and Governance

We add input/output guardrails before the Gate and after the Judge/Canon. Sensitive domains (health/legal/finance) require higher confidence thresholds and stricter debate limits. The Judge records uncertainty vectors and evidence for post-hoc auditing. Magi-A degrades gracefully on tooling failures (timeouts, resource caps).

8 Limitations

Judge bias and calibration remain challenging; retrieval quality does not equal truth; tool sandboxes can be brittle; routing errors can starve tasks of the right expertise. We mitigate via diversity- and authority-weighted retrieval, strict evidence requirements, and logging of near-misses to refine the Gate classifier.

9 Discussion and Outlook

System-level routing—not only architectural sparsity—can deliver reliable quality at low marginal cost by *spending tokens where they matter*. Future work: jointly learn Gate and Judge with test-time optimal compute, integrate symbolic verifiers for more domains, and extend to multimodal pipelines (image/audio) under the same schema.

Ethics Statement. We discuss potential misuse (automated misinformation, unsafe tool calls) and outline mitigations (guardrails, human-in-the-loop for high-risk domains).