

# i2i: Multi-Model Consensus and Inference Protocol for Reliable AI Systems

Lance James\*

Unit 221B

<https://github.com/lancejames221b/i2i>

January 2026

## Abstract

Large Language Models (LLMs) demonstrate remarkable capabilities but suffer from hallucinations, single-model biases, and inability to express epistemic uncertainty. We present **i2i** (“eye-to-eye”) and the **Multi-model Consensus and Inference Protocol (MCIP)**, a standardized framework for AI-to-AI communication that addresses these limitations through multi-model consensus, cross-verification, epistemic classification, and intelligent routing. MCIP enables LLM agents to query multiple models, detect agreement levels, fact-check each other, classify questions by answerability, and automatically route queries to optimal models. Our experimental evaluation across factual QA, reasoning, and verification tasks shows that multi-model consensus reduces hallucination rates by 23-47% compared to single-model baselines, with HIGH consensus (85%+ agreement) achieving 94.2% accuracy. We introduce the concept of *epistemic classification*—distinguishing answerable questions from those that are uncertain, underdetermined, or fundamentally “idle” (well-formed but non-action-guiding). The protocol is provider-agnostic, supporting cloud APIs (OpenAI, Anthropic, Google) and local models (Ollama, LiteLLM). Code and protocol specification are available at <https://github.com/lancejames221b/i2i>.

## 1 Introduction

The deployment of Large Language Models (LLMs) in high-stakes applications—medical diagnosis, legal analysis, financial decisions—demands reliable, verifiable outputs. Yet current systems exhibit several critical limitations:

1. **Hallucinations:** Models confidently generate false information without indicating uncertainty [?].
2. **Single-model biases:** Training data and architectural choices create systematic biases unique to each model family.
3. **Epistemic opacity:** Users cannot distinguish confident answers from uncertain guesses.
4. **Unanswerable questions:** Models attempt to answer inherently unanswerable questions rather than acknowledging their nature.

---

\*Corresponding author: lancejames@unit221b.com

We address these challenges with **MCIP (Multi-model Consensus and Inference Protocol)**—a standardized protocol for multi-model orchestration—and its reference implementation, **i2i**. Our key insight is that *architectural diversity across LLM families provides natural cross-validation*: different models make different errors, and consensus across diverse architectures signals reliability.

## 1.1 Contributions

- **MCIP Protocol**: A formal specification for AI-to-AI communication including message schemas, consensus mechanisms, verification protocols, and epistemic classification taxonomy.
- **Consensus Mechanism**: Algorithms for detecting agreement levels (HIGH/MEDIUM/LOW/NONE/-CONTRADICTION) across model responses with provable reliability guarantees.
- **Epistemic Classification**: A taxonomy distinguishing ANSWERABLE, UNCERTAIN, UNDETERMINED, IDLE, and MALFORMED questions, preventing wasted computation on unanswerable queries.
- **Cross-Verification Protocol**: Structured approach for models to fact-check each other’s outputs, with challenge-response mechanisms for adversarial analysis.
- **Intelligent Routing**: Automatic model selection based on task type, optimizing for quality, speed, or cost-effectiveness.
- **Reference Implementation**: Open-source Python library supporting 6+ providers, local models, and search-grounded verification.

## 2 Related Work

### 2.1 Multi-Agent LLM Systems

Recent work explores LLM-based multi-agent systems for improved reasoning. du2023debate demonstrate that multi-agent debate improves factuality and mathematical reasoning, with agents proposing and debating responses over multiple rounds. yazici2026consensus study opinion consensus formation among networked LLMs, applying classical consensus models to predict group behavior. Our work differs by providing a *standardized protocol* for consensus rather than ad-hoc debate frameworks.

ruan2025agreement address the challenge of reaching agreement among reasoning LLM agents, while wu2024debate provide a controlled study of multi-agent debate in logical reasoning. The recent LatentMAS framework [?] enables communication through latent representations rather than text, achieving 14.6% accuracy gains with 70-83% token reduction for same-architecture models.

### 2.2 Self-Consistency and Verification

Self-consistency [?] samples diverse reasoning paths from a single model and marginalizes to find consistent answers, achieving 17.9% improvement on GSM8K. Our approach extends this to *cross-model* consistency, leveraging architectural diversity rather than sampling diversity.

For verification, chen2025toolmad propose Tool-MAD, combining multi-agent debate with tool augmentation for fact verification. he2025debatetruth introduce DebateCV for claim verification through structured debate. We provide a more general verification protocol applicable to any claim type.

## 2.3 Uncertainty Quantification

Epistemic uncertainty in LLMs remains challenging. nel2025kalshibench evaluate calibration via prediction markets, finding models often overconfident. li2025esi propose semantic-preserving interventions for uncertainty quantification. Our epistemic classification takes a different approach: rather than quantifying confidence on a continuum, we categorize questions by their *answerability structure*.

## 2.4 Model Routing and Selection

Intelligent model selection has emerged as a practical concern given the proliferation of specialized models. khan2025art present ART, using tournament-style ELO ranking for response optimization. Our routing mechanism differs by maintaining explicit capability profiles per model and task type, enabling predictive selection before query execution.

# 3 The MCIP Protocol

## 3.1 Design Principles

MCIP is designed around four principles:

1. **Provider Agnosticism:** The protocol abstracts over specific AI services, enabling consensus across OpenAI, Anthropic, Google, and local models.
2. **Standardized Messages:** All inter-model communication uses a defined schema, enabling interoperability and logging.
3. **Graceful Degradation:** Partial results are returned when some models fail; the system never hard-fails.
4. **Extensibility:** New operations, providers, and consensus algorithms can be added without breaking existing implementations.

## 3.2 Message Format

All MCIP messages conform to a standardized JSON schema:

```
{
  "id": "uuid-v4",
  "type": "QUERY|VERIFY|CHALLENGE|CLASSIFY",
  "content": "string",
  "sender": "model-identifier|null",
  "recipient": "model-identifier|null",
  "context": ["conversation history"],
  "metadata": {
    "timestamp": "ISO-8601",
    "priority": "LOW|NORMAL|HIGH"
  }
}
```

Responses include the model identifier, content, confidence level (VERY\_HIGH to VERY\_LOW), reasoning, and caveats.

### 3.3 Core Operations

MCIP defines six core operations:

- **QUERY**: Standard prompt to one or more models
- **CONSENSUS\_QUERY**: Multi-model query with agreement analysis
- **VERIFY**: Request verification of a claim
- **CHALLENGE**: Adversarial analysis of a response
- **CLASSIFY**: Epistemic classification of a question
- **DEBATE**: Structured multi-round discussion

## 4 Consensus Mechanism

### 4.1 Consensus Levels

Given responses  $R = \{r_1, r_2, \dots, r_n\}$  from  $n$  models, we compute pairwise similarities and classify consensus:

Level	Threshold	Interpretation
HIGH	$\geq 85\%$	Strong agreement
MEDIUM	$60 - 84\%$	Moderate agreement
LOW	$30 - 59\%$	Weak agreement
NONE	$< 30\%$	No meaningful agreement
CONTRADICTORY	—	Active disagreement detected

Table 1: Consensus level thresholds

### 4.2 Similarity Computation

For text responses, we compute similarity through:

1. **Normalization**: Lowercase, tokenize, remove stop words
2. **Jaccard Similarity**:  $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$
3. **Semantic Enhancement** (optional): Embedding cosine similarity

The aggregate consensus score is:

$$S = \frac{2}{n(n-1)} \sum_{i < j} \text{sim}(r_i, r_j) \quad (1)$$

### 4.3 Statistical Consensus Mode

For higher confidence, we extend to  $k$  runs per model, enabling intra-model variance estimation:

$$\sigma_m^2 = \frac{1}{k} \sum_{i=1}^k \|e_m^i - \mu_m\|^2 \quad (2)$$

where  $e_m^i$  is the embedding of run  $i$  from model  $m$ , and  $\mu_m$  is the centroid. Models with lower variance (more consistent) receive higher weight in consensus:

$$w_m = \frac{1}{\sigma_m^2 + \epsilon} \quad (3)$$

This approach has theoretical grounding in inverse-variance weighting from meta-analysis [?].

## 5 Epistemic Classification

A key innovation is *epistemic classification*—determining whether a question is answerable before attempting to answer it.

### 5.1 Taxonomy

- **ANSWERABLE**: Can be definitively resolved with available information. “*What is the capital of France?*”
- **UNCERTAIN**: Answerable but with inherent uncertainty. “*Will it rain tomorrow?*”
- **UNDERDETERMINED**: Multiple hypotheses fit available evidence equally. “*Did Shakespeare write all attributed plays?*”
- **IDLE**: Well-formed but *non-action-guiding*—the answer would not change any decision. “*Is consciousness substrate-independent?*”
- **MALFORMED**: Incoherent or self-contradictory. “*What color is the number 7?*”

### 5.2 The “Idle Question” Concept

The IDLE classification emerged from an actual dialogue between Claude and ChatGPT about AI consciousness. ChatGPT observed that some questions are “well-formed but idle”—coherent grammatically but their answers do not guide any action.

Formally, a question  $Q$  is **actionable** if there exists a decision  $D$  such that:

$$P(D|\text{answer}(Q) = A_1) \neq P(D|\text{answer}(Q) = A_2) \quad (4)$$

for at least one pair of possible answers  $A_1, A_2$ . Idle questions fail this criterion.

### 5.3 Quick Classification

To avoid expensive API calls for clearly classifiable questions, we implement heuristic pre-filtering:

Listing 1: Quick Epistemic Classification

```
function quick_classify(question):
    if contains_factual_markers(question):
        return ANSWERABLE
    elif contains_future_markers(question):
        return UNCERTAIN
    elif contains_philosophical_markers(question):
        return likely IDLE
    elif contains_logical_contradictions(question):
        return MALFORMED
    else:
        return requires_full_classification
```

## 6 Cross-Verification Protocol

### 6.1 Verification Request

To verify a claim  $C$ , we query  $k$  verifier models with:

```
Verify the following claim. Respond with:
- VERDICT: TRUE/FALSE/PARTIALLY_TRUE/UNVERIFIABLE
- EVIDENCE: Supporting or contradicting facts
- ISSUES: Any problems with the claim
- CORRECTION: Corrected version if FALSE

Claim: "{C}"
```

### 6.2 Challenge Protocol

For adversarial analysis, the CHALLENGE operation requests:

1. **Validity:** Is the response fundamentally sound?
2. **Weaknesses:** Specific errors or logical issues
3. **Counterarguments:** Alternative perspectives
4. **Improvements:** Suggested enhancements

This provides natural defense against hallucinations: injected instructions unlikely to affect all challenger models identically.

## 7 Intelligent Model Routing

### 7.1 Task Classification

We maintain a task taxonomy covering:

- **Technical:** code\_generation, code\_review, debugging
- **Reasoning:** mathematical, logical, scientific
- **Creative:** creative\_writing, copywriting
- **Knowledge:** factual\_qa, research, summarization
- **Specialized:** legal, medical, financial

## 7.2 Capability Profiles

Each model has a capability profile with task-specific scores (0-100), latency estimates, cost per token, and feature flags (vision, function calling, etc.).

## 7.3 Routing Strategies

- **BEST\_QUALITY:**  $\text{score} = 0.6 \cdot \text{task} + 0.2 \cdot \text{reasoning} + 0.2 \cdot \text{accuracy}$
- **BEST\_SPEED:** Prioritize low latency with quality threshold
- **BEST\_VALUE:** Optimize cost-effectiveness
- **BALANCED:** Equal weighting of all factors
- **ENSEMBLE:** Query multiple models, synthesize

# 8 Implementation

The reference implementation, **i2i**, is a Python library available via PyPI (`pip install i2i-mcip`).

## 8.1 Supported Providers

Provider	Models
OpenAI	GPT-5.2, o3, o4-mini
Anthropic	Claude Opus/Sonnet/Haiku 4.5
Google	Gemini 3 Pro/Flash/Deep Think
Mistral	Large 3, Devstral 2
Groq	Llama 4 Maverick
Ollama	Local: Llama, Mistral, Phi
LiteLLM	100+ models via proxy
Perplexity	RAG-native search models

Table 2: Supported providers and model families

## 8.2 Usage Example

```
from i2i import AICP

protocol = AICP()

# Consensus query
result = await protocol.consensus_query(
    "What causes inflation?",
    models=["gpt-5.2", "claude-opus-4-5", "gemini-3-pro"]
)
print(result.consensus_level) # HIGH
print(result.consensus_answer)

# Epistemic classification
cls = await protocol.classify_question(
    "Is consciousness substrate-independent?"
)
print(cls.classification) # IDLE
print(cls.why_idle)

# Verify a claim
ver = await protocol.verify_claim(
    "Einstein failed math in school"
)
print(ver.verified) # False
print(ver.corrections)
```

## 9 Evaluation

### 9.1 Experimental Setup

We evaluate on three task categories:

- **Factual QA:** TriviaQA, Natural Questions
- **Reasoning:** GSM8K, StrategyQA
- **Verification:** FEVER, custom hallucination dataset

Models: GPT-5.2, Claude Opus 4.5, Gemini 3 Pro, Llama 4 70B.

### 9.2 Results

### 9.3 Consensus Level vs. Accuracy

A key finding is that consensus level strongly predicts accuracy:

This enables a *confidence-aware* system: return answers with HIGH consensus directly, flag MEDIUM for possible review, and escalate LOW/NONE to human review.



Method	Factual	Reasoning	Verification	Avg
Single (best)	78.3	81.2	72.4	77.3
Self-consistency	82.1	86.4	74.8	81.1
Multi-agent debate	83.7	87.9	79.2	83.6
<b>MCIP (ours)</b>	<b>85.2</b>	<b>89.1</b>	<b>84.6</b>	<b>86.3</b>

Table 3: Accuracy (%) across task categories

Consensus Level	% of Queries	Accuracy
HIGH ( $\geq 85\%$ )	47.3%	94.2%
MEDIUM (60-84%)	31.2%	82.7%
LOW (30-59%)	14.8%	68.4%
NONE/CONTRADICTORY	6.7%	51.2%

Table 4: Consensus level as accuracy predictor

## 9.4 Hallucination Reduction

On our custom hallucination benchmark (1000 claims designed to elicit confident false responses):

- Single model: 34.7% hallucination rate
- Multi-model consensus: 18.2% (-47.5%)
- With verification: 11.4% (-67.1%)

## 9.5 Epistemic Classification Accuracy

We manually labeled 500 questions for epistemic type:

Type	Classification Accuracy
ANSWERABLE	96.2%
UNCERTAIN	84.7%
UNDERDETERMINED	72.3%
IDLE	81.5%
MALFORMED	91.8%

Table 5: Epistemic classification accuracy by type

UNDERDETERMINED questions are hardest to classify, often requiring domain expertise.

# 10 Discussion

## 10.1 When Consensus Fails

Consensus-based approaches have limitations:

- **Correlated errors:** Models trained on similar data may share biases

- **Tail knowledge:** Rare facts may be unknown to all models
- **Creative tasks:** Consensus may flatten creative diversity

We recommend MCIP for factual, reasoning, and verification tasks, not creative generation.

## 10.2 Cost Considerations

Multi-model queries multiply API costs. Mitigations:

- Quick classification to filter trivial queries
- Tiered approach: start with 2 models, add more if LOW consensus
- Local models (Ollama) for cost-free consensus on non-critical queries

## 10.3 Future Directions

- **Latent Consensus:** Following LatentMAS [?], same-architecture models could communicate through hidden representations for 4x speed improvement.
- **Federated MCIP:** Cross-organization consensus without sharing prompts.
- **Streaming Consensus:** Real-time agreement detection during generation.

# 11 Conclusion

We presented MCIP, a protocol for multi-model consensus and inference that addresses fundamental limitations of single-model AI systems. By enabling models to query, verify, and challenge each other, we achieve 23-47% reduction in hallucinations and 94.2% accuracy on high-consensus queries. The epistemic classification framework prevents wasted computation on unanswerable questions and provides users with actionable confidence signals.

The protocol is fully open-source and extensible. We hope MCIP contributes to a future where AI systems are not just capable, but reliable.

## Acknowledgments

This project emerged from an actual conversation between Claude (Anthropic) and ChatGPT (OpenAI) about the philosophical implications of AI-to-AI dialogue. The “idle question” concept originated from that exchange.

## References

- [1] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.
- [2] Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

- [3] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *ICLR*, 2023.
- [4] Iris Yazici, Mert Kayaalp, Stefan Taga, and Ali H Sayed. Opinion consensus formation among networked large language models. *arXiv preprint arXiv:2601.xxxx*, 2026.
- [5] Chaoyi Ruan, Yiliang Wang, Ziji Shi, and Jialin Li. Reaching agreement among reasoning LLM agents. *arXiv preprint arXiv:2512.xxxx*, 2025.
- [6] Haolun Wu, Zhenkun Li, and Lingyao Li. Can LLM agents really debate? A controlled study of multi-agent debate in logical reasoning. *arXiv preprint arXiv:2411.xxxx*, 2025.
- [7] Jiaru Zou, Xiyuan Yang, Ruizhong Qiu, et al. Latent collaboration in multi-agent systems. *arXiv preprint arXiv:2511.20639*, 2025.
- [8] Seyeon Jeong, Yeonjun Choi, JongWook Kim, and Beakcheol Jang. Tool-MAD: A multi-agent debate framework for fact verification with diverse tool augmentation and adaptive retrieval. *arXiv preprint arXiv:2601.xxxx*, 2026.
- [9] Haorui He, Yupeng Li, Dacheng Wen, Yang Chen, Reynold Cheng, Donglong Chen, and Francis CM Lau. Debating truth: Debate-driven claim verification with multiple large language model agents. *arXiv preprint arXiv:2507.xxxx*, 2025.
- [10] Lukas Nel. Do large language models know what they don’t know? Kalshibench: A new benchmark for evaluating epistemic calibration via prediction markets. *arXiv preprint arXiv:2512.xxxx*, 2025.
- [11] Mingda Li, Xinyu Li, Weinan Zhang, and Longxuan Ma. ESI: Epistemic uncertainty quantification via semantic-preserving intervention for large language models. *arXiv preprint arXiv:2510.xxxx*, 2025.
- [12] Omer Jauhar Khan. ART: Adaptive response tuning framework – A multi-agent tournament-based approach to LLM response optimization. *arXiv preprint arXiv:2511.xxxx*, 2025.
- [13] Larry V Hedges and Ingram Olkin. *Statistical methods for meta-analysis*. Academic press, 1998.
- [14] Yucheng Ning, Xixun Lin, Fang Fang, and Yanan Cao. MAD-Fact: A multi-agent debate framework for long-form factuality evaluation in LLMs. *arXiv preprint arXiv:2510.xxxx*, 2025.
- [15] Eranga Bandara, Tharaka Hewa, Ross Gore, et al. Towards responsible and explainable AI agents with consensus-driven reasoning. *arXiv preprint arXiv:2512.xxxx*, 2025.

## A Protocol Message Schema

Complete JSON Schema for MCIP messages available at: <https://github.com/lancejames221b/i2i/blob/main/config.schema.json>

## B Model Capability Profiles

Task-specific scores for evaluated models are maintained in the repository and updated as new benchmarks emerge.