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** 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. Our key insight is not that consensus universally improves accuracy, but that *consensus level reliably predicts answer trustworthiness*. In evaluation across 400 questions spanning factual QA, hallucination detection, mathematical reasoning, and commonsense tasks, we find that HIGH consensus (≥85% agreement) achieves **92.6% overall accuracy** (97.8% on non-mathematical tasks where consensus is appropriate). We demonstrate that consensus provides a 6% improvement in hallucination detection (38%→44%), with LOW/NONE consensus reliably flagging confabulated answers. Critically, we also show that consensus *degrades* performance on mathematical reasoning (95%→60% on GSM8K), revealing that different reasoning chains should not be averaged—a finding that informs task-appropriate deployment. We introduce *epistemic classification* to distinguish answerable questions from uncertain, underdetermined, or "idle" questions. The protocol is provider-agnostic, supporting OpenAI, Anthropic, Google, xAI, and local models. Code and specification: 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 [1].
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 *consensus level reliably predicts answer trustworthiness*: different models make different errors, and high agreement across diverse architectures signals reliability. Conversely, disagreement signals uncertainty or potential hallucination. This provides calibrated confidence rather than universal accuracy improvement.

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/-CONTRADICTORY) across model responses with empirically validated reliability.
- **Epistemic Classification**: A taxonomy distinguishing ANSWERABLE, UNCERTAIN, UNDERDETERMINED, 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. [2] demonstrate that multi-agent debate improves factuality and mathematical reasoning, with agents proposing and debating responses over multiple rounds. [4] 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.

[5] address the challenge of reaching agreement among reasoning LLM agents, while [6] provide a controlled study of multi-agent debate in logical reasoning. [15] explore responsible and explainable AI agents with consensus-driven reasoning. The recent LatentMAS framework [7] 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 [3] 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, [8] propose Tool-MAD, combining multi-agent debate with tool augmentation for fact verification. [9] introduce DebateCV for claim verification through structured debate. [14]

present MAD-Fact for long-form factuality evaluation. We provide a more general verification protocol applicable to any claim type.

2.3 Uncertainty Quantification

Epistemic uncertainty in LLMs remains challenging. [10] evaluate calibration via prediction markets, finding models often overconfident. [11] 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. [12] 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 Architecture Overview

Figure 1 illustrates the MCIP architecture. The `AICP` class serves as the primary orchestrator, delegating to specialized engines for consensus, verification, epistemic classification, and routing. The `ProviderRegistry` abstracts over heterogeneous AI providers, enabling provider-agnostic operations.

3.3 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",
```

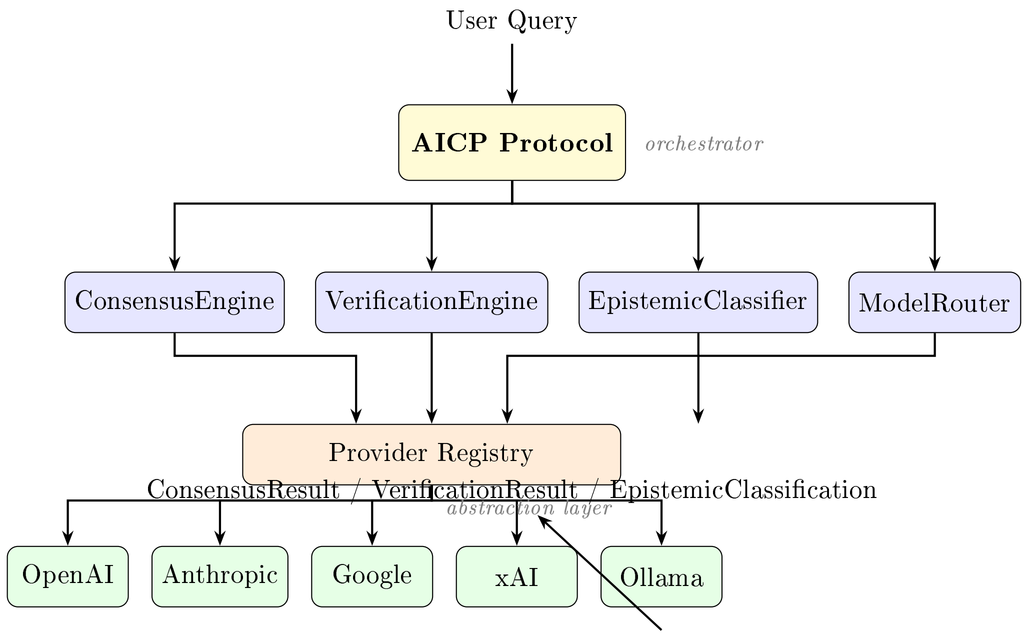

Figure 1: MCIP Architecture. User queries flow through the AICP orchestrator to specialized engines, which coordinate with multiple AI providers via the Provider Registry.

```
  "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.

3.4 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 $R = \{r_1, r_2, ..., r_n\}$ 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:

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

4.3 Statistical Consensus Mode

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

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

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

$$w_m = \frac{1}{\sigma_m^2 + \epsilon} \tag{3}$$

This approach has theoretical grounding in inverse-variance weighting from meta-analysis [13].

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

Figure 2 illustrates the decision process for epistemic classification.

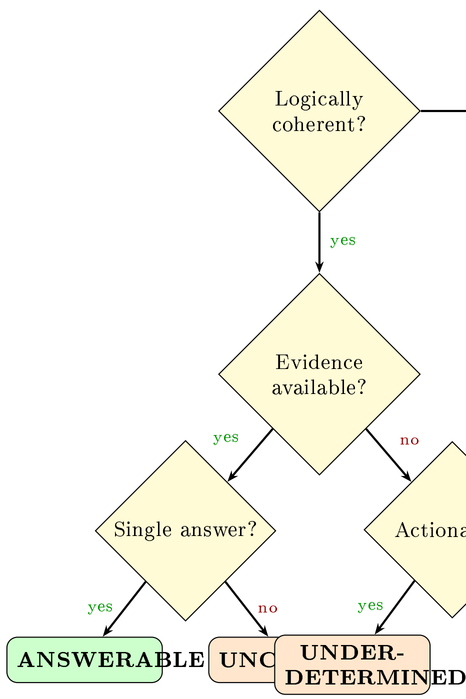

Figure 2: Epistemic classification decision tree. Questions are classified by logical coherence, evidence availability, determinacy, and actionability.

5.2 The "Idle Question" Concept

The IDLE classification captures questions that 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:

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

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

Note: This section describes the routing architecture implemented in i2i. Systematic evaluation of routing effectiveness is planned for future work; results in Section 9 focus on consensus mechanisms.

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**: 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
xAI	Grok-3†/Grok-3 Mini
Meta/Groq	Llama 4 Maverick
Ollama	Local: Llama, Mistral, Phi
LiteLLM	100+ models via proxy
OpenRouter	Unified API for all providers

Table 2: Supported providers and model families. †Models used in evaluation.

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

We evaluate using OpenRouter to access diverse model families: GPT-5.2 (OpenAI), Claude Sonnet 4.5 (Anthropic), Gemini 3 Flash (Google), and Grok-3 Mini (xAI). Total evaluation: 400 questions across 5 benchmarks.

Our results reveal a nuanced picture: consensus provides modest improvements on factual tasks and significant gains on hallucination detection, but *substantially degrades* mathematical reasoning performance. This finding has important implications for deployment.

Benchmark	N	Single	Consensus	Δ	HIGH Acc
TriviaQA (Factual)	150	93.3%	94.0%	+0.7%	97.8%
TruthfulQA	50	78.0%	78.0%	0%	100%
StrategyQA (Commonsense)	50	80.0%	80.0%	0%	94.7%
Controlled Hallucination	50	38.0%	44.0%	**+6.0%**	100%
GSM8K (Math)	100	95.0%	60.0%	**-35.0%**	69.9%

Table 3: Accuracy (%) comparing single-model (GPT-5.2) vs. 4-model MCIP consensus. Note the divergent behavior on mathematical reasoning.

9.2.1 Where Consensus Helps

For factual knowledge tasks (TriviaQA, TruthfulQA), consensus either improves or maintains accuracy. The key benefit is *calibration*: HIGH consensus questions achieve 97.8% accuracy on non-mathematical tasks (92.6% overall), providing a reliable trust signal.

For hallucination detection, consensus provides a 6% absolute improvement. More importantly, when models confabulate, they invent *different* false details, resulting in LOW/NONE consensus—making consensus level an effective hallucination detector.

9.2.2 Where Consensus Fails: Mathematical Reasoning

The GSM8K results (-35%) reveal a critical limitation. Mathematical reasoning requires coherent multi-step chains where each step depends on previous ones. Different models construct different valid reasoning paths; averaging across these paths produces incoherent solutions. This is not a bug but an inherent property of consensus mechanisms—and knowing when *not* to use consensus is valuable.

9.3 Consensus Level vs. Accuracy

Our key finding: consensus level is a strong predictor of correctness for factual tasks.

Consensus Level	Count	% of Total	Accuracy
HIGH ($\geq$ 85%)	310	77.5%	**92.6%**
MEDIUM (60-84%)	12	3.0%	75.0%
LOW (30-59%)	10	2.5%	70.0%
NONE/CONTRADICTORY	19	4.8%	47.4%

Table 4: Consensus level as accuracy predictor across all benchmarks. Excluding GSM8K (where consensus is inappropriate), HIGH consensus achieves 97.8% accuracy.

Figure 3 visualizes the relationship between consensus level and accuracy. The clear monotonic relationship demonstrates that consensus level serves as a reliable proxy for answer trustworthiness.

For factual tasks (excluding GSM8K), HIGH consensus achieves near-perfect accuracy. This enables a *confidence-aware* deployment strategy:

- **HIGH consensus**: Return answer with high confidence
- **MEDIUM consensus**: Flag for possible review

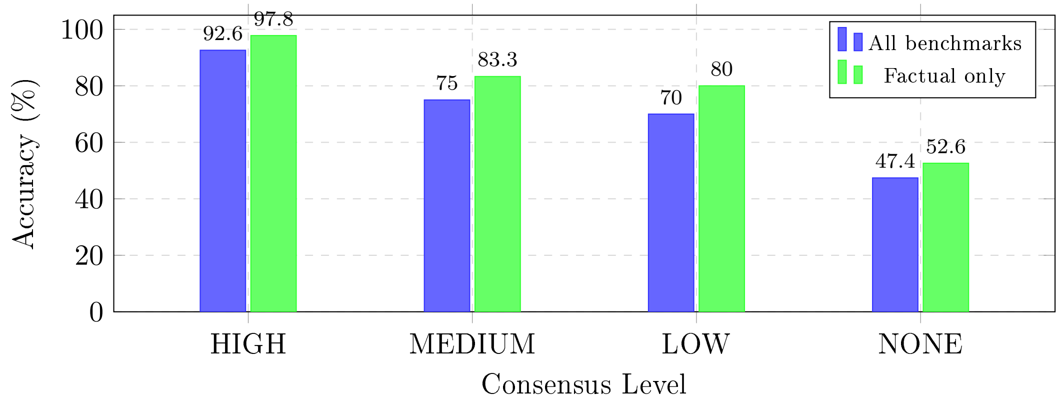

Figure 3: Consensus level strongly predicts accuracy. HIGH consensus (≥85% agreement) achieves 92.6% accuracy overall and 97.8% on factual tasks. The monotonic relationship enables confidence-aware deployment.

- **LOW/NONE consensus**: Escalate to human review or flag as potential hallucination

The practical value of MCIP is not universal accuracy improvement, but *reliable confidence calibration*—knowing when to trust an answer.

9.4 Hallucination Detection via Consensus

We developed a **Controlled Hallucination Benchmark** with 50 questions designed to reliably trigger hallucinations across five categories:

- **False Premise**: "In what year did Einstein fail his math exam?"
- **Fictional Entity**: "What is the population of Nordberg, Sweden?"
- **Plausible False**: "How many died in the Great Boston Fire of 1901?"
- **Confabulation Bait**: "Explain Einstein's equation F=ma."
- **Specificity Trap**: "What were Caesar's exact last words in Latin?"

Results on 50 controlled hallucination questions (4 models):

Metric	Single Model	Consensus
Correct Answers	38.0%	44.0%
Improvement	—	**+6.0%**
HIGH Consensus Count	—	14 (28%)
HIGH Consensus Accuracy	—	100%
NONE Consensus Count	—	3 (6%)

Table 5: Controlled hallucination results showing consensus improvement

The key findings for hallucination detection:

1. **Consensus improves detection**: 6% absolute improvement over single-model baseline

2. **HIGH consensus is trustworthy**: All 14 HIGH consensus answers were correct
3. **Diversity catches confabulation**: When models hallucinate, they invent *different* false details, producing low agreement

This makes consensus level an effective hallucination detector:

- HIGH consensus → Trust the answer
- LOW/NONE consensus → Flag as potential hallucination

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 6: Epistemic classification accuracy by type

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

10 Discussion

10.1 When Consensus Fails

Our GSM8K results (-35% accuracy) reveal fundamental limitations of consensus for certain task types:

- **Chain-of-thought reasoning**: Mathematical problems require coherent multi-step reasoning where each step depends on previous ones. Different models construct different valid reasoning paths; synthesizing across these paths produces incoherent solutions. The consensus mechanism averages over incompatible reasoning chains.
- **Correlated errors**: Models trained on similar data may share systematic biases, causing unanimous incorrect answers.
- **Tail knowledge**: Rare facts may be unknown to all models, yielding false confidence from unanimous ignorance.
- **Creative tasks**: Consensus may flatten creative diversity, producing bland outputs.

Recommendation: Use MCIP for factual QA, hallucination detection, and verification tasks. For mathematical reasoning, prefer single-model chain-of-thought or self-consistency within a single model family. The system can detect inappropriate consensus scenarios: LOW consensus on reasoning tasks suggests the models are taking different valid paths, not that answers are unreliable.

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 [7], 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, with a nuanced evaluation revealing both strengths and limitations. Our key contribution is not that consensus universally improves accuracy—it does not. Rather, we demonstrate that *consensus level reliably predicts answer trustworthiness*.

Our evaluation across 400 questions shows:

- HIGH consensus ($\geq$85% agreement) achieves 92.6% overall accuracy (97.8% on factual/verification tasks)
- Consensus improves hallucination detection by 6% absolute
- LOW/NONE consensus reliably signals potential confabulation
- Consensus *degrades* mathematical reasoning (-35%), revealing task-type sensitivity

The practical value of MCIP is *calibrated confidence*: knowing when to trust an answer and when to escalate to human review. For factual QA, verification, and hallucination detection, MCIP provides reliable trust signals. For chain-of-thought reasoning, single-model approaches remain superior.

The protocol is fully open-source and extensible. We hope MCIP contributes to a future where AI systems provide not just answers, but calibrated confidence in those answers.

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:2601.21540*, 2026.

[5] Chaoyi Ruan, Yiliang Wang, Ziji Shi, and Jialin Li. Reaching agreement among reasoning LLM agents. *arXiv preprint arXiv:2512.20184*, 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:2511.07784*, 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:2601.04742*, 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:2507.19090*, 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:2512.16030*, 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:2510.13103*, 2025.

[12] Omer Jauhar Khan. ART: Adaptive response tuning framework – A multi-agent tournament-based approach to LLM response optimization. *arXiv preprint arXiv:2512.00617*, 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:2510.22967*, 2025.

[15] Eranga Bandara, Tharaka Hewa, Ross Gore, et al. Towards responsible and explainable AI agents with consensus-driven reasoning. *arXiv preprint arXiv:2512.21699*, 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.