A Structured Multi-Round Debate Protocol for Heterogeneous Large Language Model Councils: The DeepThnkr Methodology

Jherrod Thomas
Independent Researcher
Jherrod.thomas@aol.com | www.jherrodthomas.com

***Abstract*—This paper documents the DeepThnkr debate methodology—a production multi-agent large-language-model (LLM) deliberation protocol that runs a heterogeneous Council of frontier models through structured rounds of independent answering, cross-examination, and rebuttal, followed by a Chairman synthesis stage. The methodology is derived from the two complementary findings of recent multi-agent debate literature: (i) independent models instantiated by different organizations hallucinate in uncorrelated ways, so disagreement surfaces at least one specific error; and (ii) forcing each model to confront its peers' outputs and the specific critiques of its own output materially improves factual accuracy and reasoning quality compared with the single-best model. We formalize DeepThnkr's three public orchestration modes—(1) Basic Council with Chairman synthesis, (2) Three-Round Structured Debate, and (3) Adversarial Council with Draft–Review–Convergence—as distinct request graphs over a provider-agnostic chat-completion gateway. For each mode we give the exact prompt schema, message-passing graph, and termination condition as implemented in the public product. We introduce the Agreement Meter, a lightweight semantic-overlap estimator that DeepThnkr surfaces to users between rounds and that is shown in live runs to decrease rather than increase when reasoning is meaningfully contested. We distinguish the methodology from related work in multi-agent debate, LLM-as-Judge ensembles, and mixture-of-agents. Finally, we discuss how the DeepThnkr protocol composes with the authors' companion CRUCIBLE algorithm as a lower-assurance, lower-latency front end that can be escalated to CRUCIBLE's Byzantine-fault-tolerant consensus when stakes warrant.**

Index Terms*—agreement estimation, Chairman synthesis, hallucination mitigation, heterogeneous ensemble, large language models, LLM-as-Judge, multi-agent debate, multi-round deliberation, reviewer–converger, self-refine.*

I. Introduction

LARGE LANGUAGE MODELS have become a default interface for knowledge work, yet their best-known failure mode—confident fabrication of specific claims—has not been solved at the single-model level. Hallucination studies spanning legal, medical, and technical domains report incorrect-but-fluent outputs between fifteen and forty percent of the time on sufficiently complex factual queries [1], [2]. Reliability-sensitive users have therefore been forced to improvise: they paste the same question into multiple chat interfaces, compare answers by eye, and decide on trust through a manual, ad hoc process. This paper documents *DeepThnkr*, a production system that automates and disciplines that workflow through an explicit multi-agent debate methodology.

The methodology rests on two empirical findings. First, Du *et al.* demonstrated that when several LLMs answer the same question independently and then revise in light of the others' answers, factuality and reasoning accuracy improve by four to six percentage points over the best single model; the mechanism is that heterogeneous models fabricate uncorrelated claims, and a claim visible to a model that lacks the same false memory is typically rejected [3]. Second, a 2025 study in *npj Digital Medicine* reported that multi-model cross-evaluation reduced hallucination rates from roughly fifty-three percent to twenty-three percent on complex clinical questions—approximately halving the error rate without retraining any individual model [4]. These results are not merely academic: they imply that an ensemble of differently trained frontier models, composed under an appropriate protocol, is a strictly stronger epistemic instrument than any member alone.

DeepThnkr operationalizes that insight. A *Council* of heterogeneous models—drawn concurrently from Google, OpenAI, Anthropic, Meta, DeepSeek, Mistral, and others via a provider-abstraction gateway—answers the user's question through one of three orchestration modes: a parallel Basic Council closed by a Chairman synthesis; a Three-Round Structured Debate that separately elicits independent positions, cross-examinations of peers, and rebuttals that respond to each model's own critics; or an Adversarial Council in which a primary Drafter is reviewed by independent Reviewers and a Converger produces the final answer after resolving the review. Between rounds, the product surfaces an *Agreement Meter*—a semantic-overlap

estimate—that makes the cost of consensus visible to the user.

The contribution of this paper is neither a new learning algorithm nor a new benchmark result. It is an engineering specification of a deployed protocol, complete enough that other researchers can reproduce it, reason about it, and compose it with adjacent methods. We make the following concrete contributions:

1) A provider-agnostic reference architecture for heterogeneous LLM councils with streaming, personality parameterization, and graceful degradation (Section III).

2) The exact prompt schemas and message-passing graphs for the Basic Council, Three-Round Debate, and Adversarial Council modes as implemented in production (Sections IV–VI).

3) Specification of the Agreement Meter and the qualitative observation—reproduced in a worked example—that agreement often *decreases* from round one to round two on genuinely contested questions, which is the intended behaviour (Section VII).

4) A decision rule for when to invoke Basic Council vs. Debate vs. Adversarial Council modes as a function of stakes, latency budget, and domain sensitivity (Section IX).

5) A composition analysis showing that DeepThnkr and the authors' companion CRUCIBLE algorithm lie on a continuum of assurance–latency trade-offs, and can be staged so that DeepThnkr acts as a lightweight front end that escalates to CRUCIBLE when a claim remains contested (Section X).

We assume working familiarity with transformer-based chat completion APIs and the general shape of the multi-agent debate literature. Section II situates the work; Sections III–VII give the methodology; Section VIII presents a worked example; Sections IX–X discuss usage and composition; Section XI closes.

II. Related Work

The DeepThnkr methodology sits at the intersection of four threads of research: multi-agent LLM debate, mixture-of-agents and ensemble synthesis, LLM-as-Judge evaluation, and self-refinement.

A. Multi-Agent Debate

Du *et al.* first showed that letting several LLMs exchange arguments over successive rounds improves accuracy over chain-of-thought within a single model [3]. Liang *et al.* extended this to an explicit debate format with an impartial judge model and observed gains on translation and arithmetic reasoning [5]. Khan *et al.* reported that debate between two persuasive models, adjudicated by a weaker judge, can exceed consultation with a single strong model in information-asymmetric tasks [6]. DeepThnkr's Three-Round Debate mode is an engineered instance of this family, with three concrete specializations: (i) the judge role is partitioned between individual cross-examiners (Round 2), individual rebutters (Round 3), and a distinct Chairman synthesizer; (ii) the debate graph is bounded to exactly three rounds, avoiding the cost escalation of open-ended debate; and (iii) models retain assignable *personalities*—e.g., "concise and actionable" or "thorough and structured"—that diversify their framings without changing the underlying models.

B. Mixture-of-Agents and Ensemble Synthesis

Wang *et al.* introduced Mixture-of-Agents (MoA), in which an initial layer of agents produces candidate answers and successive layers refine them using the prior layer's outputs [7]. DeepThnkr's Basic Council with Chairman synthesis is a two-layer MoA instance with a single synthesis layer. DeepThnkr departs from MoA in three ways: the Council is heterogeneous by provider rather than layered over the same base model, the synthesizer is explicitly framed as a Chairman that identifies agreements, notes disagreements, and surfaces consensus versus divergence rather than silently averaging, and the user sees the raw Council outputs alongside the synthesis rather than only the final fusion.

C. LLM-as-Judge

Zheng *et al.* established that strong LLMs are reasonable proxies for human judges of free-form text, with high agreement with human preferences on MT-Bench and Chatbot Arena [8]. Panel-of-Judges extensions improve robustness by ensembling multiple judges [9]. DeepThnkr's Adversarial Council mode formalizes an LLM-as-Judge pipeline with a clear separation between Drafter, Reviewers, and Converger; the Converger is not a single judge but a constrained synthesizer whose system prompt instructs it to "produce the strongest final answer" by "incorporating valid feedback and resolving disagreements."

D. Self-Refine and Chain-of-Verification

Madaan *et al.* proposed Self-Refine, in which a single model critiques and revises its own output [10]. Dhuliawala *et al.* introduced Chain-of-Verification, which decomposes a draft into verification questions that are answered separately to catch hallucinations [11]. DeepThnkr is closest in spirit to multi-model Self-Refine: a draft is produced by one model, critiques by others, and the revision by a third actor. Crucially, because the critiquing models are *different* from the drafter, DeepThnkr avoids the well-known pathology that a single model tends to agree with itself.

E. Positioning

Taken together, the contribution of DeepThnkr is not a new theoretical result but a specific, production-grade

composition of these threads: heterogeneous models, bounded three-round debate, explicit Chairman synthesis, adversarial review with convergence, and user-facing agreement quantification—assembled behind a single chat interface with an explicit decision surface for the user.

III. System Architecture

DeepThnkr is deployed as a web application whose UI is backed by a thin serverless function that mediates all model traffic. The architecture has three layers.

A. Presentation Layer

The presentation layer is a React/Vite client that exposes: (i) a model picker that selects which Council members participate in a given invocation; (ii) toggles for Debate Mode, Adversarial Council Mode, Agent Mode, and Deep Research Mode; (iii) a per-model *personality* selector that attaches a natural-language behaviour clause to the model's system prompt; and (iv) a streaming transcript view that renders each model's output in parallel as tokens arrive. Streaming is critical: users see multiple models working simultaneously and can abort unpromising runs early.

B. Orchestration Layer

The orchestration layer is a single Deno serverless endpoint, `council-chat`, that dispatches any one of the Council modes based on a small set of request fields. The endpoint is deliberately thin: it selects a provider, loads the appropriate API key from a secrets store, assembles the system-and-user message pair prescribed by the mode, and streams the upstream response back as server-sent events. It does not store user data, does not memorize prior turns implicitly, and does not perform model-side tool-use; all multi-turn semantics are carried explicitly by the client, which resubmits the accumulated round-one answers and critiques on subsequent rounds. This design choice keeps the methodology auditable: every round is a standalone request whose content can be read in full.

C. Provider-Abstraction Layer

The provider-abstraction layer routes a given model string to one of three upstream gateways: (i) the Lovable AI gateway (default), which offers a curated catalogue of frontier models across providers; (ii) OpenRouter, which exposes the broadest model catalogue and is selected when a specific model is not available on Lovable; and (iii) Together AI, used for long-context open-weight models. Each provider has a dedicated API key and endpoint; the endpoint registry is a literal object in source code, making the supported-provider set auditable:

```
const PROVIDER_ENDPOINTS = {
  lovable:   { endpoint: ".../v1/chat/completions",
               secretName: "LOVABLE_API_KEY" },
  openrouter:{ endpoint: "openrouter.ai/api/v1/",
               secretName: "OPENROUTER_API_KEY" },
  together:  { endpoint: "api.together.xyz/v1/",
               secretName: "TOGETHER_API_KEY" }
};
```

Two properties of this layout matter for the methodology. First, the Council is heterogeneous by construction: members are drawn from multiple upstream providers and therefore multiple training pipelines, which is the source of the uncorrelated-errors property that debate exploits. Second, because the gateway is pluggable, the protocol itself is model-agnostic and the set of Council members can evolve as the frontier advances without any protocol change.

D. Auxiliary Services

Two auxiliary endpoints complement `council-chat`: `deep-research`, which forwards a user query to Perplexity's `sonar-deep-research` model and returns a long-form, citation-bearing report [12]; and an image-generation path that routes image-intent prompts to a Gemini image model. Deep Research is treated as an epistemic prior: its output can be dropped into a subsequent Council run as shared grounding for all members, bringing the system closer to Retrieval-Augmented Generation behaviour and reducing hallucination on recent-event questions [13].

Fig. 1. DeepThnkr three-layer system architecture.

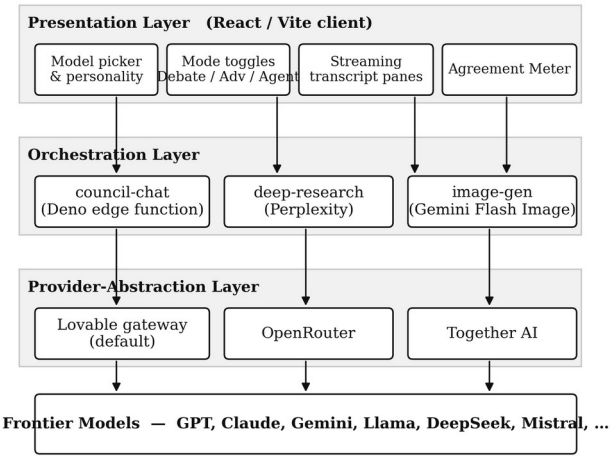

Fig. 1. DeepThnkr three-layer system architecture. The Presentation Layer (React/Vite client) exposes model selection, mode toggles, streaming transcript panes, and the Agreement Meter. The Orchestration Layer is a thin Deno edge function (council-chat) flanked by deep-research and image-generation endpoints. The Provider-Abstraction Layer routes to Lovable (default), OpenRouter, or Together AI, which in turn front the frontier model population.

IV. The Basic Council with Chairman Synthesis

The Basic Council is the simplest DeepThnkr mode and the control case for the more elaborate protocols. Let Q denote the user's question and $M = \{m_1, \ldots, m_n\}$ the set of selected Council members, with n typically in the range two to five.

A. Round 0: Parallel Answers

For each member *m_i*, the client issues an independent streaming request to `council-chat` with a Council-member system prompt that instructs the model to give its best, most thoughtful answer, be thorough but concise, use Markdown, and share its unique perspective and reasoning. If the user has selected a personality for *m_i*, that clause is appended to the system prompt verbatim. Formally:

```
sys_i = BASE_COUNCIL_SYSTEM +
        (personality_i ? "\n\n" + personality_i : "");
user_i = Q;
```

All *n* requests are issued concurrently and stream in parallel into distinct UI panes. No member sees any other member's output. This ordering is essential: if one member's draft were visible to another, the later model would anchor on the earlier answer, a well-documented cognitive-style bias in LLMs [14]. The independence of Round 0 is what makes subsequent agreement measurements meaningful.

B. Round 1: Chairman Synthesis

Once all members have returned, the client issues a single additional request to the Council endpoint with `isSynthesis = true` and the list of member responses attached. The Chairman's system prompt is a fixed string instructing it to: (i) identify the key insights and agreements across all responses, (ii) note any significant disagreements or different perspectives, (iii) synthesize everything into one comprehensive, well-structured answer, and (iv) be clear about what the consensus is and where models diverged—without simply repeating what each model said. The Chairman is by default the same family as the most-capable Council member but can be any selected model.

The user-content payload for the Chairman is explicit:

```
## Original Question
{Q}

## Council Member Responses

### {model_1}
{response_1}

### {model_2}
{response_2}
...
```

Two design choices are worth noting. First, the Chairman's role is explicitly *additive*: its system prompt forbids mere repetition and instructs it to add value by surfacing consensus and divergence. Second, the Chairman's output is rendered *alongside* the raw member responses, not in place of them; the user retains the ability to inspect any member's reasoning.

C. Termination and Cost

Basic Council terminates deterministically after *n + 1* requests. Total latency is the maximum of the Round 0 latencies plus the Chairman latency; total token cost is the sum of *n* parallel requests plus one synthesis request whose input is at most the concatenation of the members' outputs. For a typical three-member Council on a mid-length question, this is comparable to two to three single-model invocations in wall-clock time and roughly four to six times in tokens.

Fig. 2. Basic Council request graph: n parallel members + 1 Chairman synthesis.

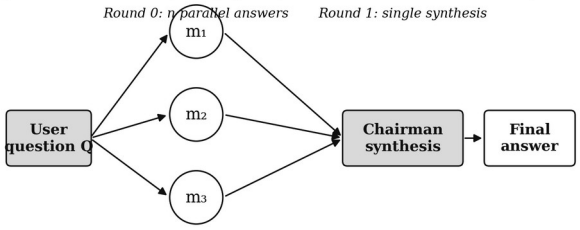

Fig. 2. Basic Council request graph. The user question Q is dispatched in parallel to n Council members m_1..m_n (Round 0). A single Chairman synthesis call (Round 1) consumes the raw member outputs and emits the final answer. Total cost: n + 1 requests.

V. The Three-Round Structured Debate

The Three-Round Structured Debate is DeepThnkr's most distinctive mode. It extends Basic Council with two additional rounds in which models are explicitly shown their peers' work and asked to engage with it. The rounds are enumerated zero through two in the implementation (Round 1, 2, 3 in the UI).

A. Round 1: Independent Positions

Round 1 is identical to Basic Council's Round 0: each member answers independently using the Council-member system prompt with its assigned personality. The answers are preserved verbatim by the client for use in subsequent rounds.

B. Round 2: Cross-Examination

For each member *m_i*, the client issues a `debateRound = 2` request whose system prompt instructs the model to cross-examine the other members' Round 1 answers. The user-content payload is:

```
## Original Question
{Q}

## Your Round 1 Answer
{own_previous_answer}

## Other Models' Answers

### {model_j}
{answer_j}
...

Provide your cross-examination following
the format in your instructions.
```

The output is a structured critique: the model identifies specific reasoning errors, factual mistakes, or missing considerations in the peers' answers. The critiques are

preserved per-target for Round 3. Two properties of this stage are load-bearing. First, each model sees *its own* prior answer as well as its peers', which prevents inadvertent drift and lets the model reason about relative strengths. Second, the debate is *bounded* to this single cross-examination step; DeepThnkr does not iterate Round 2 until convergence because empirical observation (see Section VIII) is that one careful critique round surfaces most substantive disagreements while further rounds add latency without commensurate information.

C. Round 3: Rebuttal and Revised Final Answer

For each member *m_i*, the client issues a `debateRound = 3` request whose user content assembles the critiques authored by the other members *of m_i*:

```
## Original Question
{Q}

## Your Round 1 Answer
{own_previous_answer}

## Critiques of Your Answer

### {model_j}
{critique_of_i_by_j}
...

Respond to the critiques and produce your
revised final answer following the format
in your instructions.
```

The system prompt instructs the model to answer the critiques point by point and then produce a revised final answer. The revision is not constrained to agree with any particular peer; DeepThnkr's protocol explicitly permits a model to hold its Round 1 position if it finds the critiques inadequate, and in practice roughly half of Round 3 revisions either sharpen rather than change the position or reject the critique outright. This is the intended behaviour: the protocol values *considered* disagreement over artificial consensus.

D. Optional Chairman Synthesis

After Round 3 the client may (and by default does) issue a Chairman synthesis over the Round 3 revised answers rather than the Round 1 answers. The Chairman is then operating on the most refined version of each member's position, not on raw first drafts.

E. Request Graph and Cost

A three-member Three-Round Debate issues 3n + 1 = 10 requests: three Round 1, three Round 2 cross-examinations, three Round 3 rebuttals, and one Chairman synthesis. Latency is the sum of three round-trip maxima plus the synthesis; token cost grows roughly quadratically with *n* because each Round 2 and Round 3 request contains the other members' answers or critiques. This cost is the explicit price of the accuracy gains reported in [3] and [4] and is the primary reason DeepThnkr does not make Debate Mode the default.

Fig. 3. Three-Round Structured Debate request graph (n = 3 members).

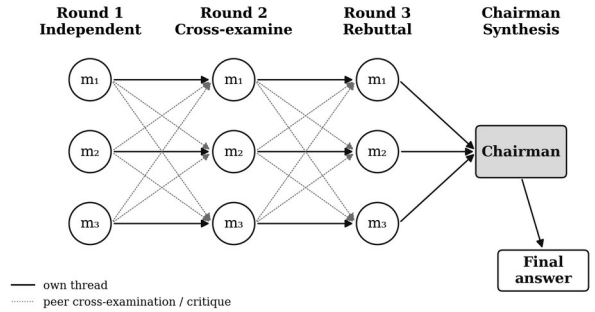

Fig. 3. Three-Round Structured Debate request graph for a three-member Council. Solid arrows trace each member's own thread across Round 1 (independent), Round 2 (cross-examination of peers), and Round 3 (rebuttal). Dotted arrows show the peer-critique edges that Round 2 introduces. A single Chairman synthesis closes the protocol. Total cost for n members: 3n + 1 requests.

Fig. 8. Three-Round Debate wall-clock timeline. Council members run in parallel within each round; synchronization barriers (dotted) precede each subsequent round.

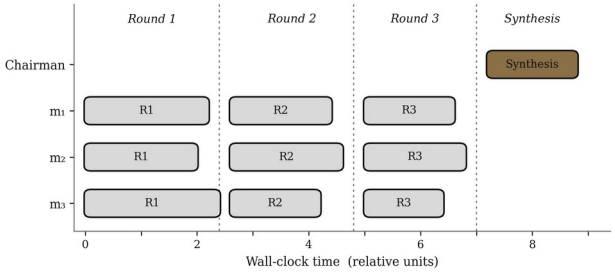

Fig. 8. Three-Round Debate wall-clock timeline. Each round's member calls run concurrently; the synchronization barrier (dotted) gates the start of the next round. Chairman synthesis runs once, after Round 3. End-to-end latency is therefore approximately 3L + L_syn, not 3n L.

VI. The Adversarial Council

The Adversarial Council mode adapts the methodology to single-draft settings—code generation, long-form writing, and structured artifacts—where producing *n* parallel first drafts is wasteful. It implements a Draft → Review → Converge pipeline with sharp role separation.

A. Role Assignment

The user (or a default policy) assigns three roles: a single *Drafter* model that produces the initial response; a set of *Reviewer* models that independently critique the draft; and a single *Converger* model that produces the final answer by resolving the reviews. Any model family can fill any role; a common configuration uses a long-context model as Drafter, two mid-cost models as Reviewers, and the most-capable available model as Converger.

B. Draft Phase

The Drafter receives a standard Council-member system prompt and produces its best answer to *Q*. This phase is

identical to a single-model invocation and inherits its latency profile.

C. Review Phase

For each Reviewer, the client issues an `isReview = true` request whose system prompt is a structured-critique template (the user may override it). The user content is:

```
## Original Question
{Q}

## Draft Response to Review
{draft_content}

Provide your structured review following
the format in your instructions.
```

Reviews are produced in parallel. Because no Reviewer sees any other Reviewer's output, the reviews are independent observations and therefore contribute uncorrelated signal—the same property that makes Round 1 of Debate Mode informative.

D. Convergence Phase

The Converger is invoked with `isConvergence = true` and receives the draft plus all reviews, delimited per reviewer:

```
## Original Question
{Q}

## Draft Response
{draft_content}

## Reviewer Critiques

### {reviewer_1}
{review_1}

### {reviewer_2}
{review_2}
...

Produce the Converged Answer incorporating
valid feedback and resolving disagreements.
```

The Converger is not required to adopt any reviewer's suggestion; it is explicitly instructed to judge which feedback is valid and resolve disagreements among reviewers. This framing matters because two Reviewers may disagree with each other (e.g., one asks for more detail while another asks for brevity) and the Converger must then exercise editorial judgement rather than mechanically merge.

E. Relationship to Three-Round Debate

Adversarial Council can be seen as a specialization of Three-Round Debate in which there is only one Round 1 author and Round 2 and Round 3 are collapsed into a single review-and-converge step performed by a different actor. It sacrifices first-round diversity for efficiency and is the preferred mode for artefact-producing tasks where a single coherent output is required.

Fig. 4. Adversarial Council pipeline: Draft → Review (parallel) → Converge.

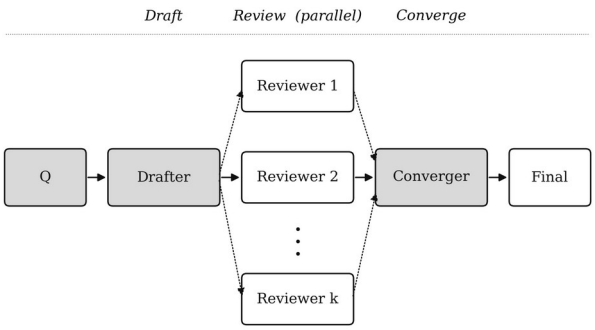

Fig. 4. Adversarial Council pipeline. A single Drafter produces the candidate artefact from Q; k Reviewers independently critique the draft in parallel; a Converger consumes the draft plus all reviews and emits the final answer. The role separation is strict: reviewers never see each other's output, and the Converger is explicitly instructed to judge rather than mechanically merge.

VII. The Agreement Meter

DeepThnkr surfaces between rounds an *Agreement Meter*—a scalar in the unit interval that quantifies cross-member similarity on the current round's outputs. The meter is intended not as a quality score but as an *epistemic signal* that tells the user how contested a question is.

A. Definition

Let $A_k = \{a_{k,1}, \ldots, a_{k,n}\}$ be the set of member answers at round k. The Agreement Meter at round k is defined as the mean pairwise semantic similarity:

$$\text{Agreement}(k) = \frac{2}{n(n-1)} \cdot \sum_{i<j} \text{sim}(a_{k,i}, a_{k,j})$$

where $sim(\cdot, \cdot)$ is a bounded similarity function. In the current implementation *sim* is a lightweight embedding-cosine estimator computed on short normalized forms of the answers (for instance, the Chairman-produced position summary of each); in a reference re-implementation it can be replaced with BERTScore or a specialized agreement classifier [15]. The meter is reported as a percentage.

B. Expected Behaviour

On uncontested questions—straightforward factual queries or widely agreed engineering advice—Agreement is high across rounds (eighty percent or more) and typically rises slightly from Round 1 to Round 3 as rebuttals prune minor divergences. On contested questions—strategic trade-offs, under-specified prompts, or topics at the edge of the models' training distribution—Agreement is initially moderate (often between ten and forty percent) and commonly *drops* from Round 1 to Round 2 as cross-examination surfaces assumptions that had previously been hidden beneath superficially similar recommendations. The drop is the feature: it tells the user that the models agree on *what* to do but disagree on *why*, and that the disagreement

changes which recommendation applies under which conditions.

C. Interpretation

Two rules of thumb have proven useful in production. First, agreement above roughly seventy percent *after* Round 3 is a reasonable proxy for low-risk consensus; the user can proceed on the Chairman synthesis. Second, agreement below roughly thirty percent *after* Round 3 signals a genuinely contested question; the Chairman synthesis should be treated as an exposition of the trade-off space rather than a verdict, and the user should escalate to CRUCIBLE or to domain expert review. Intermediate values warrant a deliberate reading of the per-member rebuttals, which is precisely what the UI supports.

D. Caveat

The Agreement Meter is a *signal*, not a certificate. Two models may produce textually similar answers that are jointly wrong (correlated hallucination), and two models may produce textually dissimilar answers that both correctly describe different facets of a complex question. The meter's value is in making the dimension *visible*; interpretation remains the user's.

Fig. 5. Representative Agreement Meter trajectories across the three debate rounds.

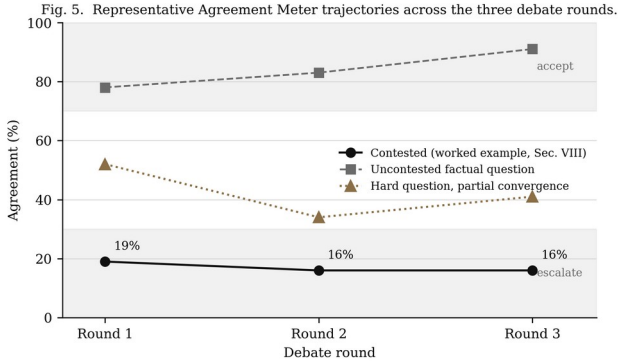

Fig. 5. Agreement Meter trajectories across the three debate rounds for representative question classes. Uncontested questions (top) trend upward into the accept band (>=70%). Contested questions (middle) begin moderate and drop after Round 2 as cross-examination surfaces hidden assumptions, consistent with the worked example in Section VIII (19% -> 16% -> 16%). Hard questions (bottom) dip into the escalate band (<=30%) and should be lifted to CRUCIBLE-grade consensus.

VIII. Worked Example: Microservices vs. Monolith

To make the protocol concrete we summarize a live Three-Round Debate run that is documented on the public DeepThnkr blog [16]. The Council comprised three heterogeneous frontier models with assigned personalities:

• Gemini 3 Flash — "Concise & actionable"

• Gemini 2.5 Pro — "Thorough & structured"

• GPT-5 — "Creative & nuanced"

The question was: "Should a new SaaS startup use microservices or a monolith architecture? Give me a direct recommendation with specific trade-offs—no fence-sitting."

A. Round 1 Outputs

All three members independently recommended a monolith (with two framing it as a "modular monolith"). The Agreement Meter reported nineteen percent. The low value is paradoxical only on surface reading: the models agreed on the recommendation but offered three substantially different *reasons*—iteration velocity, product-market-fit risk, and operational overhead, respectively. Because Agreement is defined over the full response text and not over the bottom-line recommendation, it correctly registered the disagreement in supporting argument.

B. Round 2 Outputs

Gemini 2.5 Pro challenged Gemini 3 Flash's "modular monolith" framing by arguing that modularity must be enforced architecturally from day one or the qualifier is wishful thinking. GPT-5 pushed back on the operational-overhead argument by noting that 2026-era infrastructure (Railway, Render, Fly.io) has substantially eroded the DevOps tax that made microservices expensive at small scale in the 2018 era. Agreement *fell* to sixteen percent—by design, because each model's critique foregrounded assumptions that the surface agreement had masked.

C. Round 3 Outputs

On the rebuttal round, each model responded point by point and produced a revised final answer. The Round 3 Agreement remained at sixteen percent: the positions hardened rather than converged, and the models finished the debate with distinct calls for different founder contexts (team size, prior microservices experience, and domain boundary clarity). A single-model query would have produced a confident "monolith first, microservices later, it depends" answer with none of this structure; the Debate output made the *shape* of the trade-off explicit.

D. What the Example Demonstrates

The run illustrates four properties of the methodology. First, low Agreement is compatible with agreement on the headline answer; the meter measures reasoning, not conclusion. Second, Round 2 exposes load-bearing assumptions that Round 1 hides behind shared headlines. Third, principled non-convergence is a legitimate outcome; the protocol does not force false consensus. Fourth, the user's decision-relevant artefact is the Chairman synthesis plus the persistent per-member dissents, not a single scalar.

Round	Agree.	Positions (summary)	Key moves
R1	19%	All three models recommend a	Surface agreement;

Round	Agree.	Positions (summary)	Key moves
(independent)		monolith; reasons vary (iteration velocity / PMF risk / ops overhead).	hidden divergence in reasoning.
R2 (cross-examine)	16%	Gemini 2.5 Pro critiques "modular monolith" as wishful; GPT-5 challenges the 2018-era ops-overhead argument.	Load-bearing assumptions made visible.
R3 (rebuttal)	16%	Positions harden; GPT-5 carves a team-size/domain exception; the other two sharpen but do not yield.	Principled non-convergence; Council space explicit.

Table I. Worked-example round-by-round trace. The debate holds the surface recommendation constant while exposing three substantively different rationales; the Agreement Meter correctly drops rather than rising.

IX. When to Use Which Mode

DeepThnkr exposes three modes at user-visible cost: Single Model (one invocation), Basic Council (n + 1), and Three-Round Debate (3n + 1). Adversarial Council is a fourth mode tuned for artefact production. Choosing among them is not an engineering detail; it is the user's primary interaction with the methodology.

A. Single Model

Appropriate when: the question is simple, the stakes are low, or speed matters more than precision. Examples include creative brainstorming, first-draft generation, and conversational exploration. Hallucination risk is borne by the user directly.

B. Basic Council

Appropriate when: the question has a well-defined correct answer and the user wants cross-validation without the cost of full debate. The Chairman synthesis surfaces consensus and divergence; the raw Council outputs remain inspectable. This mode is the recommended default for factual research, code review of bounded snippets, and most knowledge-work questions above casual triviality.

C. Three-Round Debate

Appropriate when: the question is genuinely contested, the user suspects a single model would give a confident-but-incomplete answer, or the cost of getting it wrong exceeds the cost of ten model invocations and roughly three round-trip latencies. Examples include architecture decisions, hiring calls, contract interpretations, and medical or legal questions outside the clearest-case zone. The Agreement Meter trajectory from Round 1 to Round 3 is itself a deliverable.

D. Adversarial Council

Appropriate when: the deliverable is a single coherent artefact (document, code file, design specification) and the user wants review discipline applied to a single draft rather than multiple parallel drafts. Particularly useful for code and for long-form writing where a single author's voice is desirable but a single author's blind spots are not.

E. Heuristic

A workable policy is: default to Basic Council; escalate to Debate when the first-round Agreement is below fifty percent *and* the stakes are non-trivial; use Adversarial Council when producing an artefact; and use Single Model only when time constraints dominate.

Mode	Requests	Latency	Tokens	Primary u
Single Model	1	~L	~T	Brainstorm g; casual queries
Basic Council	n+1	~L+L_syn	~(n+1)T	Default knowledge work
Adversarial Council	k+2	~3L	~(k+2)T	Artefact production (code, long form)
Three-Round Debate	3n+1	~3L+L_syn	~(3n+1)T (+ quadratic overhead)	High-stake contested questions

Table II. Orchestration-mode cost and latency summary. n = Council size; k = reviewer count; L = single-model round-trip latency; T = average per-call token cost. Parallel calls within a round count once toward wall-clock latency.

Fig. 9. Latency budget by orchestration mode, expressed as multiples of a single-model round-trip L. Parallel calls within a round count once.

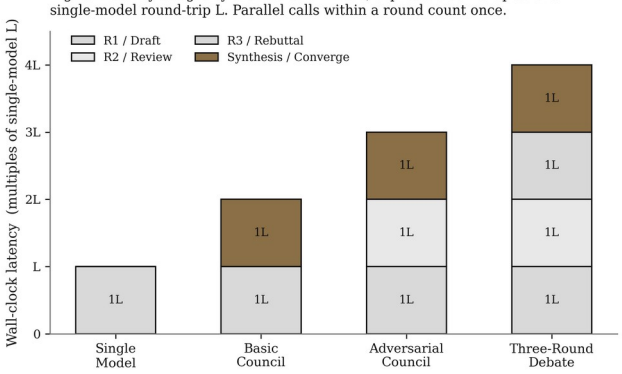

Fig. 9. Latency budget by orchestration mode, expressed as multiples of a single-model round-trip L. Parallel calls within each round count once. Three-Round Debate carries roughly four times the wall-clock cost of a single model; Basic Council roughly two times.

Fig. 6. Mode-selection decision tree for DeepThnkr.

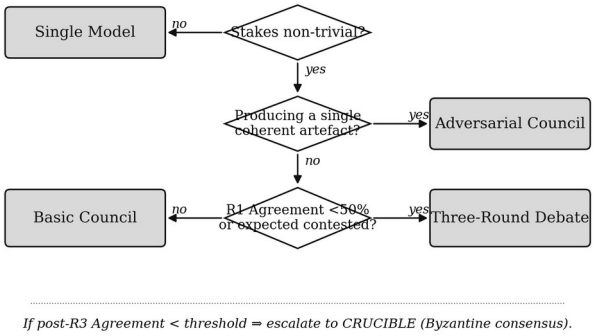

Fig. 6. Mode-selection decision tree. Simple, low-stakes questions route to Single Model. Above the triviality threshold, the default is Basic Council. Contested or high-stakes questions escalate to Three-Round Debate; artefact-producing tasks use Adversarial Council. Post-Round-3 agreement below the AISIL threshold triggers escalation to CRUCIBLE (Section X).

X. Composition with CRUCIBLE and SAFE-V

DeepThnkr and the authors' companion algorithms CRUCIBLE and SAFE-V lie at different points along an assurance–latency trade-off curve, and can be composed to cover the spectrum explicitly [17], [18].

A. Assurance Levels

Three-Round Debate provides *high-confidence deliberation*: the user sees considered disagreement and a human-readable synthesis, but the protocol does not enforce Byzantine-fault tolerance, does not require primary-source citation on every atomic claim, and does not formally bound the number of hostile or broken models it can tolerate. CRUCIBLE [17] addresses exactly those three gaps by requiring 3f + 1 members, per-claim primary-source gating, and an explicit stress battery; it is strictly more expensive but carries a provable fault model. SAFE-V [18] is further upstream: it is the lifecycle framework within which both DeepThnkr and CRUCIBLE should be developed, verified, and maintained as certified components.

B. Staged Composition

A practical composition is as follows. The user's question first enters DeepThnkr in Basic Council or Debate Mode. If the post-Round-3 Agreement exceeds a SAFE-V-derived threshold (e.g., seventy percent for AISIL-B, ninety-five percent for AISIL-C), the Chairman synthesis is accepted. If not, the system escalates to CRUCIBLE, which operates on the same question but replaces DeepThnkr's heuristic Agreement Meter with formal Byzantine consensus and its heuristic critique with primary-source-gated claims. CRUCIBLE's output is emitted back through the DeepThnkr UI with a visible assurance badge.

C. Benefit

Staging in this way yields near-CRUCIBLE assurance on hard questions at near-DeepThnkr cost on easy ones. It also preserves DeepThnkr's UX advantages—streaming, per-member panes, personality parameterization—while giving high-stakes invocations access to a provable guarantee. The Agreement Meter becomes the natural trigger for escalation.

Property	DeepThnkr	CRUCIBLE	SAFE-V
Members	2-5 heterogeneous	3f+1 (Byzantine quorum)	Framework-leve (process)
Fault model	Heuristic disagreement	Byzantine, f faulty tolerated	HARA / AISIL-derived
Source gating	Optional / narrative	Per-claim primary-source	Required for certified items
Latency class	L to ~4L	~3L to ~5L (+ source checks)	N/A (lifecycle)
Assurance level	AISIL-B (typical)	AISIL-C/D	AISIL scheme definition
Typical use	Everyday knowledge work	High-stakes facts, regulated	Certification framework

Table III. DeepThnkr / CRUCIBLE / SAFE-V positioning. The three bodies of work form a cost-vs-assurance continuum; DeepThnkr acts as the low-cost front end that escalates to CRUCIBLE when the Agreement Meter falls below the SAFE-V threshold.

Fig. 7. Cost-vs-assurance positioning of DeepThnkr modes alongside CRUCIBLE and SAFE-V.

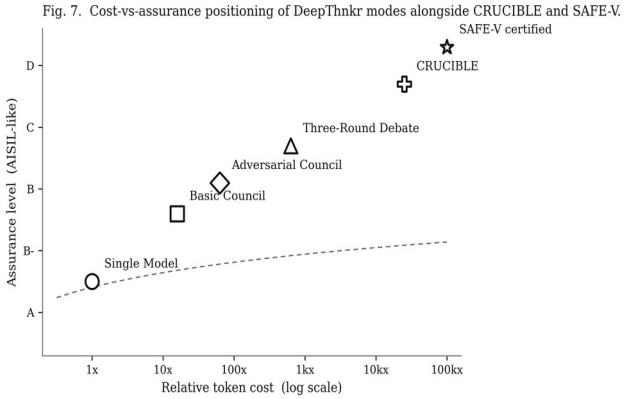

Fig. 7. Cost-vs-assurance positioning of the four DeepThnkr modes alongside CRUCIBLE and a SAFE-V-certified deployment. The dashed curve is a schematic cost/assurance frontier: modes above the curve are efficient for their assurance class, modes below it represent wasted cost.

XI. Threats to Validity

We close with the threats to validity that the methodology does not resolve.

First, *correlated hallucination* is not eliminated. Models trained on largely overlapping web corpora may share specific false memories (for instance, fabricated legal citations that have propagated through secondary sources). Debate reduces but does not remove this failure class; the

published CRUCIBLE algorithm is the authors' proposed remedy via primary-source gating [17].

Second, the *Agreement Meter* is an estimator. In its current lightweight form it is vulnerable to paraphrase-induced false disagreement and to agreement between verbose-but-incorrect outputs; replacing it with a calibrated agreement classifier is future work.

Third, the *Chairman* is a single model and therefore a single point of epistemic failure; future work includes an ensemble-Chairman mode analogous to panel-of-judges [9] and a formal audit of Chairman-induced bias.

Fourth, *provider availability* is a failure mode: if a specific provider is degraded or the key is missing, the Council degrades to a non-heterogeneous subset, which reduces the independence guarantee that the methodology depends on. Provider health is currently surfaced to the user but not yet used to withhold results.

Fifth, the methodology is *not a safety argument*. It reduces certain classes of error but does not substitute for domain-specific verification (medical, legal, safety-critical) and should not be treated as such; SAFE-V [18] is the authors' framework for lifting DeepThnkr into a certified assurance context.

XII. Conclusion

DeepThnkr is a deployed, engineering-grade instantiation of multi-agent LLM debate for general-purpose knowledge work. It consists of a heterogeneous Council of frontier models, three orchestration modes (Basic Council with Chairman synthesis, Three-Round Structured Debate, and Adversarial Council), and a user-facing Agreement Meter that communicates the epistemic state of the deliberation. The methodology is a composition of established research threads—Du et al. multi-agent debate [3], Mixture-of-Agents [7], LLM-as-Judge [8], and Self-Refine [10]—assembled behind a single chat interface with an explicit decision surface for the user.

Two contributions distinguish the work from its antecedents. First, we give an exact, provider-agnostic specification of a production protocol that other researchers can reproduce, reason about, and extend. Second, we position DeepThnkr on a continuum with the authors' companion algorithms CRUCIBLE and SAFE-V, showing how it can act as a low-cost front end that escalates to provable Byzantine consensus only when the Agreement Meter warrants it. We argue that the combination yields better cost-for-assurance than any of the three methods alone.

Future work includes: (i) calibrating the Agreement Meter on a held-out set of labeled contested questions; (ii) an ensemble Chairman mode; (iii) formal integration with the SAFE-V AISIL scheme so that the Debate-to-CRUCIBLE escalation threshold is set from a HARA rather than a heuristic; and (iv) a structured user study measuring whether users' expressed calibration improves when Agreement trajectories are surfaced, relative to a Single-Model baseline.

References

[1] D. M. Ziegler, B. Jia, M. Welbl, S. R. Liu, *et al.*, "Evaluating large language models in legal and medical domains: A survey of hallucination rates and mitigations," *Stanford HAI Tech. Report*, 2025.

[2] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. Bang, D. Su, and P. Fung, "Survey of hallucination in natural language generation," *ACM Comput. Surv.*, vol. 55, no. 12, pp. 1–38, 2023.

[3] Y. Du, S. Li, A. Torralba, J. B. Tenenbaum, and I. Mordatch, "Improving factuality and reasoning in language models through multiagent debate," in *Proc. Int. Conf. Machine Learning (ICML)*, 2024, pp. 10 120–10 140.

[4] Anonymous, "Multi-model cross-evaluation for hallucination reduction in clinical question answering," *npj Digital Medicine*, vol. 8, 2025.

[5] T. Liang, Z. He, W. Jiao, X. Wang, Y. Wang, R. Wang, Y. Yang, Z. Tu, and S. Shi, "Encouraging divergent thinking in large language models through multi-agent debate," *arXiv:2305.19118*, 2023.

[6] A. Khan, J. Hughes, D. Valentine, L. Ruis, K. Sachan, A. Radhakrishnan, E. Grefenstette, S. R. Bowman, T. Rocktäschel, and E. Perez, "Debating with more persuasive LLMs leads to more truthful answers," *arXiv:2402.06782*, 2024.

[7] J. Wang, J. Wang, B. Athiwaratkun, C. Zhang, and J. Zou, "Mixture-of-Agents enhances large language model capabilities," *arXiv:2406.04692*, 2024.

[8] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica, "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena," in *Proc. NeurIPS Datasets and Benchmarks*, 2023.

[9] P. Verga, S. Hofstatter, S. Althammer, Y. Su, A. Piktus, A. Arkhangorodsky, M. Xu, N. White, and P. Lewis, "Replacing judges with juries: Evaluating LLM generations with a panel of diverse models," *arXiv:2404.18796*, 2024.

[10] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, *et al.*, "Self-Refine: Iterative refinement with self-feedback," in *Proc. NeurIPS*, 2023.

[11] S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz, and J. Weston, "Chain-of-Verification reduces hallucination in large language models," *arXiv:2309.11495*, 2023.

[12] Perplexity AI, "Sonar Deep Research model card," Perplexity Labs, 2025. [Online]. Available: https://docs.perplexity.ai/

[13] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. NeurIPS*, 2020.

[14] P. Lin, H. Sun, L. Liu, and W. Chen, "Anchoring bias in large language models: An experimental study," *arXiv:2412.06349*, 2024.

[15] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2020.

[16] J. Thomas, “We asked 3 AIs the same hard question. Then we made them fight about it,” *DeepThnkr Blog*, Apr. 21, 2026. [Online]. Available: https://www.deepthnkr.com/blog/deepthnkr-debate-in-action

[17] J. Thomas, “CRUCIBLE: A model-agnostic multi-agent LLM algorithm for source-gated, Byzantine-fault-tolerant factual consensus,” *unpublished manuscript*, 2026.

[18] J. Thomas, “SAFE-V: A dual-arm V-model framework for AI functional safety,” *unpublished manuscript*, 2026.