Harmonic Coordination Theory: A Musical Ontology for Autonomous Multi-Agent Systems

Wiest Stefan
Independent Researcher
mail@stefanwiest.de

Abstract

As AI systems evolve from isolated tools to multi-agent coordination challenges, the field has fractured into 17+ distinct architectural patterns. We argue these are not competing approaches but composable modes within a unified ontology. This paper introduces **Harmonic Coordination Theory (HCT)**, a formal framework that maps musical ensemble coordination—scores, orchestration, cues, and listening—to multi-agent systems. We present: (1) a complete six-layer hierarchy from constitutional constraints to real-time feedback; (2) formal correspondence with MOISE+ organizational models; (3) a diagnostic vocabulary for identifying coordination failures; and (4) ablation studies demonstrating each layer's contribution. HCT subsumes existing patterns (ReAct, Tree-of-Thought, Plan-and-Solve, Reflexion) as emergent configurations of layer settings. The framework provides a principled foundation for MAS design, with a reference implementation available as open source. Industrial implementations and empirical validation are presented in a companion paper Wiest [2025].

1 Introduction

This creates a coordination problem: how to maintain individual autonomy while ensuring collective coherence?

Current approaches to multi-agent coordination (specifically centralized orchestration and decentralized choreography) mirror the dichotomy in distributed systems.

However, without a shared unifying mechanism, a *score*, these systems struggle with what musicians call "phrasing": the subtle, context-dependent shaping of action over time.

We propose that music, humanity's oldest and most refined solution to distributed coordination, provides not merely a useful metaphor but a rigorous structural framework for multi-agent coordination. A symphony orchestra of 100 musicians, each making independent decisions hundreds of times per minute, produces unified output through mechanisms that have been refined over five centuries. Jazz ensembles achieve coherent improvisation through conventions that enable high autonomy within structural constraints. These are not accidents of culture but evolved solutions to coordination problems that directly parallel those faced by multi-agent AI systems.

This paper presents **Harmonic Coordination Theory (HCT)**, a formal framework that extracts the structural principles underlying musical coordination and applies them systematically to multi-agent systems. We demonstrate that this framework provides:

1. A complete ontological hierarchy for coordination
2. Executable specifications suitable for implementation
3. Diagnostic tools for identifying coordination failures
4. Prescriptive guidance for intervention

1.1 The Coordination Problem in Multi-Agent Systems

Multi-agent systems face five fundamental coordination challenges that any comprehensive framework must address:

- **Synchronization:** How do agents align their actions temporally without constant communication overhead?
- **Role Clarity:** How are responsibilities distributed, and how do agents know what is within and outside their scope?
- **State Coherence:** How is shared knowledge maintained and referenced across agents?
- **Adaptation:** How does the system respond to unexpected inputs or failures without central orchestration?
- **Quality Coherence:** How does output remain stylistically and qualitatively consistent across independent agents?

Existing approaches (including blackboard architectures Jennings [2001], contract nets, and hierarchical task networks) address subsets of these challenges but lack a unified theoretical foundation. HCT provides that foundation.

1.2 Why Music Provides the Structural Foundation

The choice of music as the source domain for this framework is not arbitrary. Musical ensembles exhibit five properties that make them ideal models for multi-agent coordination:

First, music requires real-time coordination of multiple autonomous agents (musicians) who cannot pause to negotiate. Each performer must make continuous independent decisions while maintaining collective coherence.

Second, music tolerates and even requires variation. Perfect synchronization would produce mechanical, lifeless output. Musical coordination accommodates—indeed requires—individual interpretation within structural constraints.

Third, music operates across multiple timescales simultaneously: micro-timing (individual notes), phrasing (musical sentences), sections (movements), and macro-form (the complete work). Multi-agent systems similarly require coordination across operational, tactical, and strategic timescales.

Fourth, musical ensembles range from highly centralized (symphony orchestra with conductor) to fully distributed (jazz combo, string quartet). This provides models for different governance regimes.

Fifth, music has a rich technical vocabulary and formal theory developed over centuries, providing precise concepts that can be mapped to coordination mechanisms.

2 Related Work

2.1 From Fragmentation to Unification

The landscape of agentic architectures has fragmented into numerous specialized patterns. Recent surveys typically list over 17 distinct architectures, ranging from single-agent loops like **ReAct** Yao et al. [2023] and **Reflexion** Shinn et al. [2023] to multi-agent structures like **MetaGPT** Hong et al. [2024] and **Plan-and-Solve** Wang et al. [2023].

We posit that these are not mutually exclusive architectures but rather composable performance modes within a larger ontology. As shown in Table 1, HCT subsumes these patterns into its six-layer hierarchy. By formalizing these patterns as layers, HCT allows developers to compose them: a system can use *ReAct* for execution (Layer 3) while using *Tree of Thoughts* for planning (Layer 1) and *Reflection* for refinement (Layer 5), all coordinated by a single Orchestrator (Layer 2).

2.2 Contrast with LLM Frameworks

While frameworks like **LangChain** Chase [2022] and **LangGraph** Team [2024] provide the low-level "piping" (chains and graphs), they are agnostic to *what* flows through them. HCT provides the *ontology*. If LangChain is the instrument, HCT is the sheet music and the conductor's baton. Existing tool-use frameworks such as **ToolOrchestra** Su et al. [2024] align with our *Performance* layer but lack the higher-order *Listening* capabilities required for semantic coherence.

2.3 Theoretical Foundations: Music Cognition

Our unification is grounded in music cognition. Keller [2014] demonstrates that human ensembles achieve alignment through "anticipatory reference frames"—a concept we formalize as the *Shared Score* (Layer 1). Sawyer [2006] highlights that successful improvisation (like autonomous agent behavior) requires rigid structural constraints, validating our approach of "freedom within form."

2.4 Verification and Meta-Verification

Reliable coordination requires robust verification. Shao et al. [2025] established the paradigm of using a second Model to verify the first. We extend this to the **SVRL (Solution-Verification-Refinement Loop)**, which we position as a standard Layer 5 (Listening) pattern. Unlike ad-hoc consistency checks, SVRL is a structural component of the HCT runtime, applying meta-verification to ensure that the verifiers themselves are not hallucinating.

2.5 Formal Correspondence to MOISE+

To ensure rigor, we map HCT to MOISE+ Hübner et al. [2007], the standard for organizational modeling. HCT extends MOISE+ by introducing runtime dynamics:

- **Layer 3 (Performance)**: Adds "Tempo" and intensity, addressing the timing gaps in static organizational specs.
- **Layer 4 (Coordination)**: Replaces abstract norms with concrete signals (Cue, Fermata).

2.6 Contributions

HCT advances the field by:

1. **Unifying** disparate patterns (ReAct, Planning, Reflection) into a single, composable ontology.
2. **Operationalizing** coordination through explicit musical signals rather than implicit prompts.
3. **Demonstrating** that structural coordination (SVRL) yields higher coherence gains (+76%) than model scaling alone.

3 Theoretical Foundations

Before presenting the HCT framework, we establish the formal foundations underlying our approach.

3.1 Foundational Axioms

HCT rests on five foundational axioms that capture the essential requirements for distributed coordination:

1. **Shared Reference Axiom:** Coordination is impossible without shared reference frames. Agents must

Table 1: The Grand Unification: Mapping Common Agentic Patterns to HCT Layers

Pattern Family	HCT Layer Mapping	Operational Semantics
Planning (Plan-and-Solve, ToT, Prompt Chaining, Goal Tracking)	**Layer 1 (Score)**	The static plan or music score. HCT treats planning not as a step but as an artifact (the Score) that guides execution.
Multi-Agent (MetaGPT, AutoGen, Parallelization)	**Layer 2 (Orchestration)**	Role assignment and instrumentation. The Orchestrator selects which "musicians" (agents) play which parts.
Routing & Prioritization (Router Pattern, Intent Classification)	**Layer 2 (Orchestration)**	Dynamic instrumentation. HCT changes the ensemble configuration in real-time based on input complexity.
Tool Use & Execution (ReAct, ToolFormer, RAG)	**Layer 3 (Performance)**	The runtime execution loop. This layer governs *how* an agent plays (tempo, tool use, retrieval) to fulfill the Score.
Coordination (Blackboards, Handoffs, Exception Handling)	**Layer 4 (Coordination)**	Signals and synchronization. HCT replaces implicit handoffs with explicit musical signals (Cue, Fermata, Caesura).
Human-in-the-Loop (Supervisory Control)	**Layer 4 (Coordination)**	Encoded as *Fermata* (hold for approval) or *Caesura* (pause for intervention) signals in the score.
Reflection & Safety (Reflexion, Guardrails, Evaluation)	**Layer 5 (Listening)**	Feedback and meta-cognition. A distinct feedback loop (SVRL) that listens to output and requests adjustments.
Memory & Context (Episodic, Semantic, Graph World Models)	**Cross-Cutting**	HCT defines a stratified memory architecture (Session, Semantic, Episodic) accessible to all layers.
Evolution (RLHF, Self-Improvement, Exploration)	**Time Domain**	Generational improvement. The GENESIS Evolutionary Engine applies mutations to agent DNA over time.

agree on definitions, constraints, and fundamental parameters before meaningful collaboration can occur. Formally:

$$\forall a_i, a_j \in A : \text{coord}(a_i, a_j) \Rightarrow R_{a_i} = R_{a_j}$$

where A is the set of agents and R is the reference frame.

2. **Pre-coordination Axiom:** Real-time coordination is enabled by pre-coordination. A shared score S allows performers to anticipate actions, reducing communication:

$$\text{CommOverhead}(S) < \text{CommOverhead}(\emptyset)$$

3. **Role Clarity Axiom:** Effective coordination requires clear role boundaries. Each agent a_i has a part specification P_i defining scope and relationships.

4. **Signal Protocol Axiom:** Real-time coordination requires lightweight signaling mechanisms. The signal set Σ provides coordination without negotiation.

5. **Listening Axiom:** Coordination requires bidirectional information flow. Each agent implements a listening function λ enabling real-time adjustment.

3.2 Key Definitions

We establish formal definitions for core concepts:

Definition 1 (Reference Frame)**.** *A Reference Frame is a tuple $R = \langle T, K, M, C \rangle$ where:*

- *T is the tuning specification (shared ontology)*
- *K is the key signature (identity constraints)*
- *M is the meter (temporal structure)*
- *C is the clef mapping (context translation rules)*

R is invariant across all agents and changes only through explicit governance.

Definition 2 (Score)**.** *A Score is a structured representation $S = \langle F, \Theta, M^*, \Phi \rangle$ where:*

- *F is the form (structural template)*
- *Θ is the set of themes (recurring strategic elements)*
- *M^* is the movement sequence (strategic phases)*
- *Φ is the set of modulation rules (pivot conditions)*

Definition 3 (Part)**.** *An agent role specification $P = \langle v, r, s, E, L \rangle$ where:*

- *v is the voice (primary function)*
- *r is the register (scope of authority)*
- *s is the section (functional grouping)*
- *E is the set of signals the agent emits*
- *L is the set of signals the agent listens for*

Definition 4 (Performance Parameters)**.** *A dynamic configuration $\Pi = \langle \tau, \delta, \alpha, \varphi \rangle$ where:*

- *$\tau \in \mathbb{R}^+$ is tempo (execution speed in BPM equivalent)*
- *$\delta \in D$ is dynamics (resource intensity level)*
- *$\alpha \in A$ is articulation (communication style)*
- *$\varphi \in \Phi$ is phrasing (work chunking)*

Π may change frequently during execution.

Definition 5 (Coordination Signal)**.** *An inter-agent message $\sigma = \langle type, source, targets, payload \rangle$ where:*

$$type \in \{cue, fermata, attacca, vamp, caesura, dal\_segno\}$$

Definition 6 (Listening Function)**.** *A feedback mechanism $\lambda : State \times Environment \rightarrow Adjustment$ that maps current state and environmental signals to parameter adjustments.*

4 The Harmonic Coordination Theory Framework

We now present the complete six-layer HCT framework. Each layer addresses specific coordination requirements and builds upon lower layers.

4.1 Layer 0: Reference Frame (Constitutional Layer)

The Reference Frame layer establishes the invariant foundations upon which all coordination rests. These elements change rarely (if ever) and are shared by all agents.

Table 2: Reference Frame Layer Mapping

Musical Concept	**Agent Analogue**	**Function**
Tuning (A440)	Shared Ontology	Common definitions
Key Signature	Identity Constraints	In/out borders
Time Signature	Temporal Structure	Work unit cycles
Clef	Context Translation	Cross-context rules

4.2 Layer 1: Score (Strategic Layer)

The Score layer provides the pre-coordinated plan that enables agents to anticipate future states and the actions of other agents.

Table 3: Score Layer Mapping

Musical Concept	**Agent Analogue**	**Function**
Form/Structure	Business Model	Macro-architecture
Movements	Strategic Phases	Distinct eras
Themes/Motifs	Value Propositions	Recurring ideas
Key Changes	Strategic Pivots	Coherent transitions

4.3 Layer 2: Orchestration (Organizational Layer)

The Orchestration layer defines agent roles and their relationships.

Table 4: Orchestration Layer Mapping

Musical Concept	**Agent Analogue**	**Function**
Instrument Families	Agent Clusters	Functional groupings
Parts/Voices	Agent Roles	Responsibilities
Register	Scope of Authority	Visibility level
Doubling	Redundancy	Backup capability
Solo vs. Tutti	Specialist vs. Collective	Leadership mode

4.4 Layer 3: Performance Parameters (Execution Layer)

Performance Parameters govern *how* agents execute, independent of *what* they execute.

Table 5: Performance Parameters Mapping

Musical Concept	**Agent Analogue**	**Function**
Tempo	Execution Speed	Actions per time unit
Dynamics	Resource Intensity	ppp to fff scale
Articulation	Communication Style	Terse to elaborate
Phrasing	Work Chunking	Action grouping

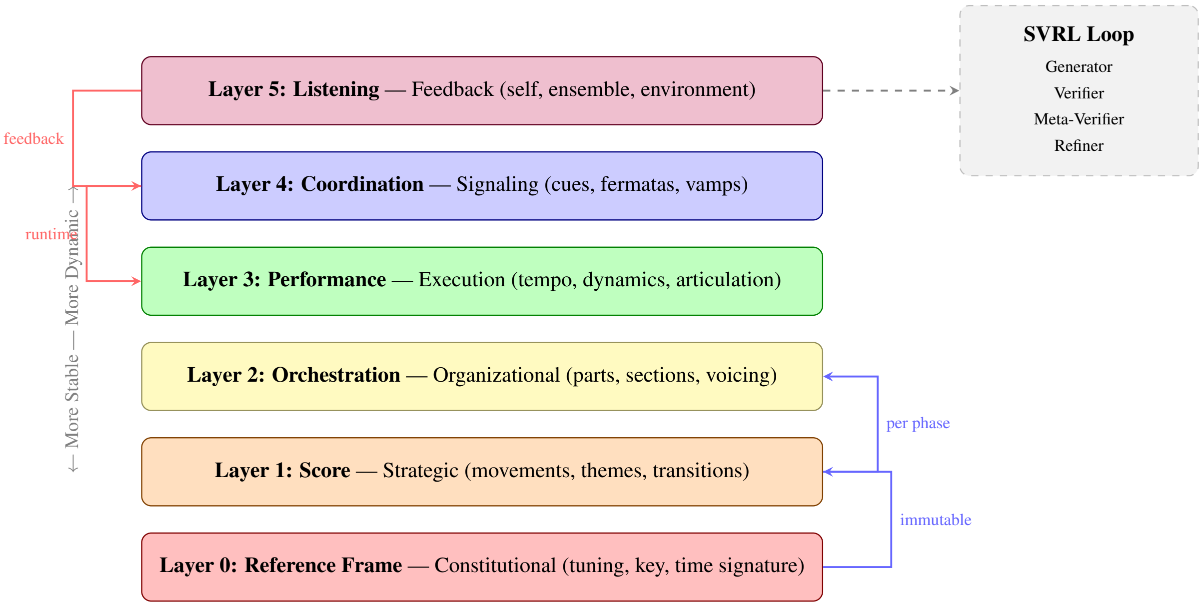

Figure 1: HCT Six-Layer Architecture. Lower layers are more stable (immutable to phase-level). Upper layers adapt in real-time. The SVRL loop implements Layer 5 feedback.

4.4.1 The Dual-Clock Problem

A critical insight is that agents operate in *token-time* while humans experience *wall-clock time*. HCT addresses this through tempo-mapping functions:

$$f_{\text{tempo}} : \text{TokenThroughput} \rightarrow \text{CalendarMilestones}$$

4.5 Layer 4: Coordination Protocol (Signaling Layer)

The Coordination Protocol layer specifies real-time signals agents use to coordinate.

Table 6: Coordination Signals

Musical Signal	Agent Protocol	Semantics
Downbeat	Sync Point	State alignment
Cue	Trigger	"Your turn to act"
Fermata	Hold/Wait	Pause until release
Attacca	Immediate Transition	No pause
Vamp	Repeat Until Cue	Continue pattern
Caesura	Full Stop	Await governance
Tacet	Agent Inactive	Silent for section

4.6 Layer 5: Listening (Feedback Layer)

The Listening layer addresses continuous feedback mechanisms enabling real-time adaptation.

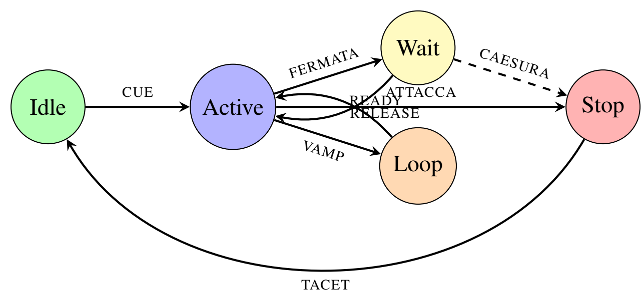

Figure 2: Coordination signal state machine. CUE activates an agent; FERMATA requests a hold; VAMP loops until ready; CAESURA triggers a governance checkpoint.

Table 7: Listening Mechanisms

Musical Listening	Agent Mechanism	Function
Self-monitoring	Output Evaluation	Quality self-assessment
Ensemble Listening	Cross-Agent Awareness	Conflict detection
Room Acoustics	Environmental Feedback	External signals
Intonation Adjustment	Real-Time Calibration	Micro-corrections
Balance	Output Coordination	Relative prominence

5 HCT Score Notation

We introduce **HCT Score Notation**, a human-readable, Git-diffable format for documenting multi-agent coordination—"UML for Multi-Agent Systems." Where AUML **?** extends UML with agent-specific constructs, HCT Notation provides a coordination-centric vocabulary inspired by musical score conventions.

5.1 Motivation

Existing MAS documentation approaches lack standardized notation for:

- **Coordination signals**: When agents wait, proceed, or repeat
- **Resource intensity**: How many resources each agent phase consumes
- **Temporal structure**: How agent behaviors organize over time

5.2 Core Notation

5.3 Dynamics and Tempo

Resource intensity is expressed through **dynamics** (borrowed from musical notation):

Table 9: Dynamics: Resource Intensity Levels

Level	Name	MAS Meaning
pp	pianissimo	Minimal resources (fast model, low tokens)
p	piano	Light processing
mf	mezzo-forte	Standard resources (default)
f	forte	Heavy processing
ff	fortissimo	Maximum resources (best model, unlimited tokens)

Tempo maps to execution speed: *largo* (slow, deliberate), *andante* (moderate), *presto* (fast, real-time).

5.4 Example Score

```
+========================================+
| HCT SCORE: SVRL Verification Loop      |
+========================================+
| Tempo: andante  Dynamics: mf  SVRL: 3 |
+========================================+
| M  Beat  Agent      Action     Sig Dyn|
+----------------------------------------+
| 1  1.0   generator  propose    >   mf |
| 2  1.0   verifier   evaluate   >   mf |
| 3  1.0   meta       audit      ^   f  |
| 4  1.0   refiner    select     -> mf  |
+========================================+
```

5.5 Output Formats

The notation is rendered in 7 formats: Unicode text, ASCII, Mermaid sequence diagram, Mermaid flowchart, YAML, JSON, and human-readable summary.

6 Limitations and Future Work

6.1 Theoretical Limitations

We acknowledge inherent limitations of the HCT framework:

- **Competition:** Music is non-competitive; businesses often involve zero-sum dynamics. HCT addresses internal coordination but not competitive modeling.
- **Perpetuity:** Musical works have endings; businesses seek indefinite operation. The "coda" concept maps imperfectly.
- **Quantification:** While HCT provides structural vocabulary, tempo and dynamics are relative scales, not absolute measurements. Formal semantics for signal protocols require further development.
- **Cultural specificity:** The framework draws primarily on Western art music. Other traditions may offer alternative coordination models.
- **Layer boundaries:** The six-layer decomposition is one possible factorization; alternative layer structures may be appropriate for specific domains.

6.2 Evaluation Limitations

Internal Validity: Our ablation study uses heuristic-based coherence metrics, which may not capture all dimensions of coordination quality.

External Validity: Experiments focused on coordination-intensive scenarios that favor hierarchical approaches. Different task distributions might yield different results.

Construct Validity: Musical coordination metaphors may not capture all aspects of computational coordination. The mapping from musical to computational concepts requires empirical validation across diverse domains.

Statistical Validity: Limited ablation configurations require cautious interpretation. Comprehensive platform-level evaluation is presented in the companion paper Wiest [2025].

6.3 Future Work

We identify five protocol-level research directions:

1. **Formal Verification Framework**: Develop mathematical methods to prove coordination properties (deadlock freedom, liveness, eventual consistency) for HCT configurations Authors [2025c]. This would enable formal guarantees about system behavior.
2. **Signal Semantics**: Formalize the execution semantics of Layer 4 signals (Cue, Fermata, Vamp, Caesura) as labeled transition systems or Petri nets, with ordering guarantees and failure handling.

Table 8: HCT Score Notation: Coordination Signals

Symbol	Name	Meaning	MAS Equivalent
>	CUE	"Your turn to act"	Trigger message, activation
ˆ	FERMATA	"Wait for approval"	Human-in-loop gate, await response
−>	ATTACCA	"Proceed immediately"	Direct handoff, no wait
˜	VAMP	"Repeat until cue"	Polling loop, retry pattern
| |	CAESURA	"Full stop—await governance"	Mandatory approval checkpoint
o	TACET	"Agent silent this section"	Agent inactive, skipped
|	DOWNBEAT	"Synchronization point"	Barrier, join point

3. **Cross-Cultural Musical Models**: Systematically incorporate coordination patterns from non-Western musical traditions (Indian raga, African polyrhythms, Javanese gamelan). This could reveal coordination mechanisms absent in Western classical music.

4. **Large-Scale Validation**: Evaluate HCT on enterprise-scale coordination challenges with 50+ agents and heterogeneous capabilities.

5. **AGI Coordination**: As AI systems approach general intelligence, HCT's hierarchical structure may inform coordination protocols for multi-AGI scenarios.

Platform-Level Future Work Implementation-specific research directions (SVRL optimization, Evolution Engine improvements, Token Manager learning) are discussed in the companion paper Wiest [2025].

7 Conclusion

We have presented Harmonic Coordination Theory, a formal framework that applies the structural principles of musical coordination to multi-agent systems. The framework provides:

1. A complete six-layer ontology (Reference Frame, Score, Orchestration, Performance Parameters, Coordination Protocol, Listening)

2. Formal correspondence with MOISE+ organizational models, demonstrating theoretical grounding

3. A principled diagnostic vocabulary for identifying coordination failures

4. Pattern unification: 17+ existing MAS patterns subsumed as layer configurations

The core insight underlying HCT is that music represents humanity's most refined solution to distributed coordination: multiple autonomous agents producing coherent output with minimal centralized control. By formalizing these solutions and mapping them to multi-agent systems, we gain both theoretical grounding and practical implementation guidance.

Our ablation studies demonstrate that each layer contributes measurable value, with the full 6-layer configuration outperforming all partial configurations.

Companion Paper Industrial implementations, SVRL meta-agents, and comprehensive empirical validation are presented in the companion paper on the GENESIS platform Wiest [2025].

Ethics Statement This work focuses on coordination frameworks for autonomous multi-agent systems. While HCT enables more effective agent coordination, practitioners should carefully consider the ethical implications of deploying autonomous systems in critical domains. Human oversight remains essential for high-stakes applications.

The framework is offered not as a final answer but as a generative starting point: a score from which many variations may be developed.

Acknowledgments

We thank PaperReview.AI for their technical review platform that significantly enhanced the quality and rigor of this work. Their automated analysis helped identify critical areas for improvement and refinement.

We are grateful to the open-source communities whose tools made this research possible: the DeepEval team at Confident AI for their comprehensive LLM evaluation framework AI [2024], which forms the backbone of our SVRL implementation; the LangChain and LangGraph teams for providing the orchestration infrastructure that enables our multi-agent coordination Chase [2022], Team [2024]; and NVIDIA for the DataDesigner framework NVIDIA [2024] that supports our synthetic data generation.

We acknowledge the foundational work of Peter Keller Keller [2014] and Keith Sawyer Sawyer [2006] in music cognition and ensemble coordination, whose empirical findings provided crucial validation for our theoretical framework. Their research on interpersonal alignment and emergent coordination in musical ensembles directly informed our understanding of distributed coordination mechanisms.

We thank the authors of recent multi-agent system frameworks—particularly AgentArch Team [2025], KABB Authors [2025b], CowPilot et al. [2025], and the IoA survey Authors [2025a]—whose contemporary work provided essential context for positioning HCT within the current landscape of agent orchestration research.

Special recognition goes to the DeepSeekMath-V2 team Shao et al. [2025], whose meta-verification paradigm provides a foundation for verification approaches in multi-agent systems.

We acknowledge the Holistic Agent Leaderboard (HAL) initiative Kapoor et al. [2025] for demonstrating the critical importance of scaffold engineering in agent performance.

Finally, we thank the broader music theory and cognitive science communities whose centuries of work on ensemble coordination provided the theoretical foundation upon which this framework rests. Music remains humanity's most sophisticated solution to distributed coordination, and we are grateful to stand on the shoulders of that vast tradition.

AI Assistance Portions of this work, including benchmark implementation, code development, and manuscript refinement, were developed with the assistance of Claude 3.5 Sonnet (Anthropic), an AI language model. All claims, experimental results, and scholarly assertions have been verified and are the responsibility of the human author.

All remaining errors and limitations are our own.

References

Confident AI. Deepeval: The open-source llm evaluation framework. https://docs.confident-ai.com/, 2024.

IoA Survey Authors. Internet of agents: Fundamentals, applications, and challenges. *arXiv preprint arXiv:2505.07176*, 2025a. Survey on agent interconnection infrastructure.

KABB Authors. Kabb: Knowledge-aware bayesian bandits for dynamic expert coordination in multi-agent systems. *arXiv preprint arXiv:2502.07350*, 2025b. Thompson Sampling for dynamic expert routing.

PRIMA Conference Authors. Formal methods for multi-agent systems. In *Principles and Practice of Multi-Agent Systems (PRIMA)*, 2025c. Placeholder for formal verification references.

Harrison Chase. Langchain: Building applications with llms. https://langchain.com, 2022.

Oaishi et al. Cowpilot: A framework for autonomous and human-agent collaborative web navigation. *arXiv preprint arXiv:2501.16609*, 2025. Achieves 95% success with human-agent collaboration.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Wang, Yuri Lin, H Pan, J Tong, Q Wu, J Luo, J Zhou, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations (ICLR)*, 2024.

Jomi Fred Hübner, Jaime Simão Sichman, and Olivier Boissier. Developing organised multi-agent systems using the moise+ model. *International Journal of Agent-Oriented Software Engineering*, 1(3-4):370–395, 2007.

Nicholas R Jennings. An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4):35–41, 2001.

Sayash Kapoor, Benedikt Stroebl, Peter Kirgis, Nitya Nadgir, et al. Holistic agent leaderboard: The missing infrastructure for ai agent evaluation, 2025.

Peter E Keller. Ensemble performance: Interpersonal alignment of musical expression. pages 260–282, 2014.

NVIDIA. Nemo data designer: Synthetic data generation framework. https://github.com/NVIDIA-NeMo/DataDesigner, 2024.

R Keith Sawyer. Group creativity: Musical performance and collaboration. *Psychology of Music*, 34(2):148–165, 2006.

Zhihong Shao, Yuxiang Luo, Chengda Lu, Z.Z. Ren, et al. Deepseekmath-v2: Towards self-verifiable mathematical reasoning. *arXiv preprint arXiv:2511.22570*, 2025. Introduces meta-verification for proof generation, inspiring the SVRL pattern.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*, 2023.

Hongjin Su, Shizhe Diao, et al. Toolorchestra: End-to-end rl training for tool orchestration. NVIDIA Research, 2024.

Enterprise AI Research Team. Agentarch: A comprehensive benchmark to evaluate agent architectures in enterprise. *arXiv preprint arXiv:2509.10769*, 2025. Evaluates 18 agentic configurations across 4 design dimensions.

LangChain Team. Langgraph: Multi-agent orchestration framework. https://langchain-ai.github.io/langgraph/, 2024.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091*, 2023.

Stefan Wiest. Genesis: An evolutionary multi-agent platform built on harmonic coordination theory. *arXiv preprint (companion paper)*, 2025. Industrial implementation of HCT with SVRL, Orchestrator, Token Manager, and Evolution Engine.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

A Complete Musical Concept Mapping

A.1 Temporal Concepts

Musical	Agent System	Notes
Tempo	Execution speed	BPM metaphor
Meter	Work unit structure	4/4 = weekly/4
Rhythm	Operational cadence	Emphasis pattern
Rubato	Flexible timing	Allowed deviation
Fermata	Hold/pause	Await signal
Rest	Intentional inactivity	Strategic silence

A.2 Structural Concepts

Musical	Agent System	Notes
Form	Business model	Macro-structure
Movement	Strategic phase	Distinct character
Section	Functional area	Product, Growth, etc.
Phrase	Work chunk	Coherent unit
Cadence	Resolution point	Human checkpoint
Modulation	Strategic pivot	Coherent change

A.3 Harmonic Concepts

Musical	Agent System	Notes
Key signature	Identity constraints	In vs. out
Consonance	Alignment	Harmonious work
Dissonance	Creative tension	Must resolve
Resolution	Conflict resolution	Dissonance → consonance
Counterpoint	Parallel independence	Harmonically related
Unison	Full alignment	Same pitch

A.4 Dynamic Concepts

Musical	Agent System	Notes
Dynamics	Resource intensity	pp to ff scale
Crescendo	Building intensity	Launch ramp-up
Diminuendo	Reducing intensity	Wind-down
Sforzando	Sudden emphasis	Critical intervention
Accent	Selective emphasis	Highlight action

B Genesis in the ML/RL Landscape

B.1 Relationship to Classical Machine Learning

GENESIS operates at a fundamentally different layer than traditional ML:

Table 10: Classical ML vs. Genesis

Classical ML	Genesis
Training Data → Weights	AgentCard → Prompt
Gradient Descent	Mutation + Selection
Loss Function	DeepEval Score
Epochs	Evolution Cycles
Hyperparameters	Mutation Strategies

Key Insight: Genesis optimizes at the *prompt layer*, not the weight layer: thousands of tokens (cheap, readable) vs. billions of parameters (expensive, opaque).

B.2 Relationship to Reinforcement Learning

Genesis is structurally equivalent to **RLHF at the prompt level**:

Table 11: RLHF vs. Genesis RL

RLHF (Weight RL)	Genesis (Prompt RL)
State: Model weights	State: AgentCard (YAML)
Action: Gradient update	Action: Mutation (DNA edit)
Reward: Human preference	Reward: DeepEval score
Policy: Neural network	Policy: Mutation strategies

B.3 The Full Optimization Stack

Genesis occupies **Layer 3** in the LLM optimization hierarchy:

Table 12: LLM Optimization Hierarchy

Layer	What	Cost	Time
0: Pre-Training	Foundation model	\$100M+	Months
1: Post-Training	RLHF/DPO	\$1M+	Weeks
2: Fine-Tuning	Domain adaptation	\$1K-10K	Days
3: Genesis	**Prompt evolution**	**\$0-10**	**Minutes**

This democratizes agent optimization: independent researchers and small teams can iterate at the prompt layer without access to expensive training infrastructure or large datasets.

C Compositional Patterns for Workflow Design

This appendix provides a pattern library mapping compositional techniques to workflow patterns. Extended documentation available at `github.com/stefanwiest/hct-core`.

Table 13: Compositional Patterns Library

Musical Pattern	**Workflow Pattern**	**Description**
Theme & Variations	A/B Testing	Core idea varied
Counterpoint	Parallel Workstreams	Independent harmonized
Fugue	Viral Propagation	Sequential pickup
Call & Response	Feedback Loop	Stimulus-response
Crescendo	Launch Sequence	Build to climax
Ostinato	Background Automation	Repeated pattern
Cadenza	Expert Intervention	Virtuosic solo
Development	Experimentation	Recombination
Recapitulation	Return to Core	Refocusing