Higher Categorical Semantics of Coordinator-Executor Prompt Optimization: Typed Transport, Coherent Assembly, and Interaction Bounds

Anonymous

November 1, 2026 · Version 1.0

Abstract

We give a higher categorical semantics for the separation of shared coordination and local execution in multi-agent prompt optimization, taking CEPO as the motivating architecture. The literal prompt parameterization is embedded in a family of executor categories over a category of coordinator contracts. Unstraightening and a fiber product organize admissible systems, contract migration, and local refinement. A separate operadic interpretation describes composition of executions; a finite stochastic specialization supplies observable laws and total variation bounds. Internalization in an infinity-topos identifies coherent descent as the condition for assembling compatible local specifications. We then distinguish semantic equivalences from optimization arrows and prove improvement only under explicit acceptance hypotheses. A Boolean-cube decomposition isolates interactions between independently selected executor changes, with a two-agent counterexample to unconditional improvement after recombination. A sequential acceptance rule and a supermodularity condition provide distinct sufficient remedies. An exact-arithmetic companion checks finite instances and counterexamples. This is a mathematical reconstruction and extension, with no replication of the source experiments and no claim of machine-checked higher categorical proofs.

Keywords: multi-agent systems; prompt optimization; cocartesian fibrations; infinity-operads; infinity-topoi; descent; stochastic semantics; block optimization.

1 Problem, attribution, and scope

CEPO, due to Kim, Lee, Chung, Yoon, and Kim [1], makes shared coordination and agent-specific execution distinct prompt parameters. We study the mathematical consequences of this separation. The architecture and empirical evidence belong to those authors; the categorical interpretation and conditional results below are developed here.

The source supplies a shared coordinator prompt, executor prompts, and role-specific guidance to an underlying optimizer. Its default schedule first selects a coordinator and then updates selected executors. Both roles use a system-level objective. Equation (4) and Algorithm 1 display executor contexts using the other executors' initial prompts; Appendix B.2 also describes sequential executor optimization. We distinguish frozen-context recombination from current-context acceptance rather than assume these are equivalent implementations. The source reports tests on eleven benchmarks in four domains; we do not reproduce them.

Our contribution is a precise separation of four questions: which prompt tuples are syntactically assembled; which implementations satisfy a coordinator contract; which compositions preserve

observable behavior; and which accepted changes improve a specified score. Shared text settles only the first question. The remaining questions need semantics, evidence, and hypotheses.

The categorical ingredients are standard: straightening and unstraightening, operadic composition, and descent in an infinity-topos [2, 3]. We do not present those general results as new. The contribution is their specialization to coordinator-executor optimization, together with an explicit analysis of update interactions and the limits of possible guarantees.

Conventions. We fix universes so that the infinity-categories considered below are small when needed; Cat_∞ may be taken in a larger universe. Limits and fibers are homotopy coherent. The core $\mathcal{A}^{\simeq}$ contains all objects and only equivalences of $\mathcal{A}$. A numerical score is defined on objects invariant under specified equivalences; it is not silently a functor from every update arrow to a discrete real-valued space. All stochastic examples have finite state spaces and a finite execution horizon. Infinite traces, unbounded recursion, and general measurable kernels require additional constructions.

2 The literal parameterization and its limits

Let Σ^* be the set of finite strings over a token alphabet. Let $C_0 \subseteq \Sigma^*$ and $E_{i,0} \subseteq \Sigma^*$ be allowed coordinator and executor strings. For $N \geq 1$, put

$$P_0 = C_0 \times \prod_{i=1}^{N} E_{i,0}, \qquad D(c, e_1, \ldots, e_N) = (c \parallel e_1, \ldots, c \parallel e_N).$$

Here $\parallel$ is literal concatenation, and the tuple retains the component boundary. Algebraically, D factors through the diagonal $\Delta_N : C_0 \to C_0^N$, followed by coordinatewise assembly. A coordinator edit therefore changes one stored value replicated into each invocation.

Proposition 2.1 (What shared parameterization guarantees)**.** *For every $p = (c, e_1, \ldots, e_N)$, all assembled prompts have the designated prefix c. If $C_0 = E_{i,0} = \Sigma^*$ and the empty coordinator is allowed, D is surjective onto $(\Sigma^*)^N$. In general D is not injective.*

Proof. The prefix assertion is the definition of concatenation. Given $(s_i)_i$, take c empty and $e_i = s_i$, proving surjectivity. Noninjectivity already occurs for $N = 1$: (a, bc) and (ab, c) both assemble to abc. Repeating these executors gives the same example for arbitrary N. □

Consequently, unrestricted decomposition need not reduce expressivity. Any claimed reduction in the search domain must state restrictions on component lengths, grammars, admissibility, or the optimizer. Recovering the decomposition from a concatenated prompt requires an explicit serialization, such as length-prefixing or an escaped separator. None of this proves that a model obeys the shared instruction, or interprets it identically in different local contexts.

For example, two agents may receive the same requested output schema and still emit incompatible records. A typed contract addresses this possibility by recording evidence of admissibility. It does not identify a natural-language instruction with a proof of compliance.

3 Executor families over coordinator contracts

Definition 3.1 (Contract-indexed system)**.** A contract-indexed coordinator-executor system consists of an infinity-category $\mathcal{C}$ and functors

$$F_i : \mathcal{C} \longrightarrow \mathrm{Cat}_\infty, \qquad 1 \leq i \leq N.$$

An object c is a coordinator contract; an object of $F_i(c)$ is an executor implementation with its declared admissibility evidence. A morphism in $F_i(c)$ is a declared refinement or transformation within that contract. A morphism $u : c \to c'$ has a specified migration functor $F_i(u)$. The functoriality of F_i includes the coherence data for successive migrations.

The existence of these functors is a hypothesis on the semantics. Not every prompt edit has a sound migration. A practical model can include only certified migrations in $\mathcal{C}$, or replace total migration by a separately specified partial construction. Arbitrary edits must not be admitted merely to obtain a cocartesian fibration.

Let $q_i : \mathcal{E}_i \to \mathcal{C}$ be the cocartesian fibration obtained by unstraightening F_i, and define

$$\mathcal{E} = \mathcal{E}_1 \times_{\mathcal{C}} \cdots \times_{\mathcal{C}} \mathcal{E}_N, \qquad q : \mathcal{E} \longrightarrow \mathcal{C}. \tag{1}$$

An object is a shared contract together with compatible executors. The compatibility here means that each executor lies over the same contract; cross-executor behavioral constraints must already be included in the contract/evidence or imposed by a further admissibility subcategory. Such a subcategory inherits the chosen transport only if it is closed under that transport; the product theorem below does not automatically extend to arbitrary additional joint constraints.

Theorem 3.2 (Assembly and typed transport)**.** *The projection q is the cocartesian fibration classified by $F = \prod_i F_i$. In particular,*

$$\mathcal{E}_c \simeq \prod_i F_i(c), \qquad u_!(e_1, \dots, e_N) \simeq (F_1(u)e_1, \dots, F_N(u)e_N).$$

For a specified base arrow $u : c \to c'$, the space of arrows over u from $(c, \mathbf{e})$ to $(c', \mathbf{e}')$, namely the homotopy fiber of the mapping-space projection over u, is

$$\mathrm{Map}^u_{\mathcal{E}}((c, \mathbf{e}), (c', \mathbf{e}')) \simeq \prod_i \mathrm{Map}_{F_i(c')}(F_i(u)e_i, e'_i). \tag{2}$$

Each such arrow factors as cocartesian transport followed by a vertical arrow, with a contractible space of choices of factorization data. Successive transports satisfy $v_!u_! \simeq (vu)_!$ with all higher coherence inherited from F.

Proof. Straightening identifies cocartesian fibrations over $\mathcal{C}$ and cocartesian-edge-preserving maps with $\mathrm{Fun}(\mathcal{C}, \mathrm{Cat}_\infty)$, by the dual of the cartesian version in [2, §3.2]. Products on the functor side are pointwise, so their unstraightening is the fiber product (1). Its fibers and transport are therefore as stated. For a chosen cocartesian lift of u, its defining mapping-space property identifies arrows over u with vertical arrows from the transported object. The fiber is a product, giving (2). The universal property makes the choice of cocartesian lift and subsequent factorization contractible. Coherent composition follows from functoriality of F. □

Example 3.3 (Literal prompts as a constant family)**.** Take $\mathcal{C}$ to be the free category on a graph of allowed coordinator edits and let each F_i be constant at a category of executor strings and declared local edits. Transport keeps executor text fixed. At the level of object tuples, this recovers Section 2. If a contract change alters an input schema, the constant-family model is generally inappropriate: an adapter or a newly certified executor is needed.

Proposition 3.4 (Independent local transformations)**.** *Fix c. For endofunctors $R_i : F_i(c) \to F_i(c)$, let $\widehat{R}_i$ act in coordinate i of $\mathcal{E}_c$. If $i \neq j$, then $\widehat{R}_i\widehat{R}_j \simeq \widehat{R}_j\widehat{R}_i$, with the canonical product coherence.*

Proof. Both composites have the same projection to every factor: R_i in factor i, R_j in factor j, and the identity elsewhere. The product universal property supplies the natural equivalence and its coherence. □

This is a statement about fixed transformations of separate factors. An optimizer that reads the whole system and chooses R_i from its score is not generally a product endofunctor of this form. Section 7 gives a concrete failure of commutation for such choices.

4 Coherent execution and observable stochastic laws

4.1 A compositional semantics, specified separately

Fix a finite acyclic typed workflow. Let $\mathcal{W}$ be the symmetric monoidal category freely generated by its typed port signature and invocation/routing operations, with explicit copying and deletion where permitted. Objects are finite port tuples, and morphisms are generated acyclic circuits; this allows operations with several input and output ports. Wires include persistent state and shared random seeds whenever those can affect later behavior. A finite unrolling represents a workflow with a declared finite loop bound. Feedback is not supplied for free by an ordinary symmetric monoidal structure.

Let $\mathcal{V}^{\otimes}$ be a symmetric monoidal infinity-category. An implementation is a strong symmetric monoidal functor, represented by an infinity-operad map preserving cocartesian edges,

$$A : \mathrm{N}^{\otimes}\mathcal{W} \longrightarrow \mathcal{V}^{\otimes}.$$

The notation denotes the operadic nerve of the symmetric monoidal category. Write $\mathrm{Fun}^{\otimes}(\mathcal{W}, \mathcal{V})$ for the infinity-category of these functors. A higher execution semantics is a coherent assignment

$$\sigma : \mathcal{E}^{\simeq} \longrightarrow \mathrm{Fun}^{\otimes}(\mathcal{W}, \mathcal{V})^{\simeq}. \tag{3}$$

This assignment, including the action on equivalences, is part of the model, not a consequence of prompt concatenation. It permits certified equivalences of implementations and coherent equivalences between their witnesses. Noninvertible optimization moves are handled separately in Section 6.

Proposition 4.1 (Contextual invariance)**.** *Suppose two interpretations of the generators of a workflow extend to equivalent strong symmetric monoidal functors on $\mathcal{W}$, with the equivalence compatible with their typed interfaces. Evaluation of any fixed workflow operation gives equivalent morphisms in $\mathcal{V}$. If an observation map on these equivalences takes values in a discrete space of laws, the observed laws are equal.*

Proof. Evaluation of an operation is functorial in the symmetric monoidal functor and its equivalences, or in the corresponding operadic data [3, §2.1]. It therefore takes the given equivalence to an equivalence of the evaluated morphisms. A path in a discrete space has equal endpoints, proving the second assertion. □

Local equalities without compatibility data do not automatically give a monoidal natural equivalence. Conversely, a finite stochastic interpretation may live entirely in an ordinary category: its nerve does not acquire nontrivial higher mapping spaces just because we use infinity-categorical notation.

4.2 Finite stochastic specialization

In FinStoch, objects are finite sets and a morphism $k : X \to Y$ is a stochastic matrix $k(y \mid x)$. Composition is matrix multiplication, tensor is independent product, and deterministic copying/deletion are explicit maps. This is a standard Markov-category model [4]. Correlated random variables must be outputs of a shared source, rather than be replaced by the independent tensor of their marginal laws.

For a parameter p and input z, let

$$K_p(o \mid z), \qquad o \in O,$$

be the complete observed execution law after composing the workflow. The observation o may include answer, trace, cost, and next persistent state. Omitting state is allowed only if no future context observes it. Given a fixed input distribution μ and reward $r(z, o) \in [0, 1]$, define

$$J_{\mu,r}(p) = \sum_z \mu(z) \sum_o K_p(o \mid z) r(z, o). \tag{4}$$

Equivalent observed kernels have equal scores for this μ, r. Equal scores for one reward do not imply equal kernels. The score is constant on the equivalences certified by (3) only when the observation map preserves those equivalences as hypothesized.

Theorem 4.2 (Finite-horizon substitution bound)**.** *Consider two finite stochastic workflows with the same typed schedule and at most H invocation slots. Include the full relevant history in the state at each slot; stopped executions use an absorbing state. At slot h, suppose the old and new transition kernels differ uniformly over states by at most ε_h in total variation. Then*

$$\sup_z \mathrm{TV}(K_p(\cdot \mid z), K_{p'}(\cdot \mid z)) \le \min\left(1, \sum_{h=1}^{H} \varepsilon_h\right), \qquad |J_{\mu,r}(p) - J_{\mu,r}(p')| \le \min\left(1, \sum_h \varepsilon_h\right).$$

Proof. For distributions α, β and kernels k, k', contraction and convexity give

$$\mathrm{TV}(\alpha k, \beta k') \le \mathrm{TV}(\alpha k, \beta k) + \mathrm{TV}(\beta k, \beta k') \le \mathrm{TV}(\alpha, \beta) + \sup_s \mathrm{TV}(k_s, k'_s).$$

Starting with equal initial laws and inducting through the invocation slots yields the sum bound. Taking an observation is a deterministic kernel and cannot increase total variation. For a function valued in $[0, 1]$, its expectation difference is at most total variation; integrate over μ. Every total variation distance is also at most one. □

The uniform hypothesis includes histories of probability zero under one version. Without a common state model or control of distribution shift at changed states, local benchmark discrepancies do not establish the bound. The theorem proves robustness to certified kernel changes; it does not estimate those discrepancies from prompt text.

5 Internal language and coherent descent

5.1 An infinity-topos model of admissible specifications

Let $\mathcal{X}$ be an infinity-topos. We now use *spaces* of specifications and equivalence witnesses, as distinct from the noninvertible refinements retained in $\mathcal{E}$. Let $C \in \mathcal{X}$, let $E_i \to C$ be objects of its slice, and put

$$P = E_1 \times_C \cdots \times_C E_N \longrightarrow C. \tag{5}$$

The internal dependent-type notation for its total space is

$$\sum_{c:C} \prod_{i=1}^{N} E_i(c).$$

An identification between coordinators requires transport of the executor witnesses. It is not componentwise equality of untyped strings. Taking $\mathcal{X} = \mathcal{S}$ gives a space-valued model; taking a sheaf infinity-topos permits dependence on contexts of observation. No computability or effective presentation is asserted for an arbitrary $\mathcal{X}$.

To relate this to Section 3 in the space-valued case, restrict F_i to $\mathcal{C}^{\simeq}$ and take the core of each value. These form space-valued families; their unstraightenings and fiber product recover the core-level compatible tuples. Keeping noninvertible refinements requires internal category objects or the external Cat_∞-valued model, not just dependent types interpreted as spaces.

Theorem 5.1 (Descent of compatible executor families)**.** *Let $a : U \to C$ be an effective epimorphism, with $U_n = U^{\times_C (n+1)}$ its Cech nerve. Pullback gives an equivalence*

$$\mathcal{X}_{/C} \simeq \lim_{[n]\in\Delta} \mathcal{X}_{/U_n}. \tag{6}$$

Under this equivalence the pullbacks of P are the products of the pullbacks of E_i. Thus local executor families assemble to a global compatible family precisely when they constitute full descent data. The space of realizations of a specified descent datum is contractible.

Proof. The Cech nerve of an effective epimorphism has colimit C. Descent for objects in an infinity-topos identifies the slice over that colimit with the limit of the slices, giving (6); see [2, §6.1.3]. Each pullback functor preserves finite limits, hence the product in (5). Equivalence (6) is fully faithful and essentially surjective, so the homotopy fiber over a fixed descent object, including its realization equivalence, is contractible. □

This uses Cech descent, not hyperdescent; hypercompleteness is not required. The local data must include equivalences on double overlaps, their compatibility on triple overlaps, and the required higher coherence. Pairwise agreement alone is insufficient when the specifications retain nontrivial automorphisms.

Example 5.2 (A concrete coherence obstruction)**.** In spaces take the effective epimorphism $U = \{0, 1, 2\} \to *$. Let each local specification be a two-element set regarded as an object of the groupoid of such sets and bijections. Write s for its nonidentity swap, so $s^2 = \mathrm{id}$. Set $g_{01} = g_{12} = g_{02} = s$, with inverse transitions and identity diagonal transitions. Every pair is related by a bijection, but on the triple overlap the descent equation requires

$$g_{02} = g_{12} g_{01} = \mathrm{id},$$

which is false. In this 1-groupoid there is no higher homotopy identifying these distinct bijections. These chosen pairwise witnesses therefore do not assemble. Other coherent witnesses could assemble the same local objects.

For distributed agents, an overlap can represent a shared message schema or an overlapping view of a contract. The theorem explains what a sound assembly certificate would contain. It does not assert that a language model has produced such a certificate, or that ordinary text agreement supplies one.

6 Proposal arrows, acceptance, and numerical guarantees

6.1 Updates are directed and generally not equivalences

Let $\mathscr{P}$ be a set of represented admissible systems drawn from the preceding semantics, with a specified score $J : \mathscr{P} \to [0,1]$. Let G be a graph of optimizer proposals. A role label specifies either a coordinator migration with its required executor adaptations, or a vertical executor change at a fixed coordinator. Role-specific meta-instructions may influence the proposal mechanism. They are not a proof that a generated candidate lies in the corresponding admissible class; that requires a checker or an explicit hypothesis.

The free category on accepted proposal edges has a nerve, providing a directed infinity-category of optimization histories. If every accepted edge satisfies $J(p) \le J(p')$, the score extends to a functor from that category to the ordered category $([0,1], \le)$. In general it cannot extend to a discrete space of scores, since a discrete-space-valued functor would force equality along every edge. This distinguishes performance improvement from semantic equivalence.

Theorem 6.1 (Current-context finite acceptance)**.** *At step t, let $B_t \subseteq \mathscr{P}$ be a nonempty finite candidate set containing the actual current system p_t. Evaluate all candidates using the same exact score J and select*

$$p_{t+1} \in \arg\max_{p \in B_t} J(p).$$

Then $J(p_{t+1}) \ge J(p_t)$. If each strict acceptance gains at least $\delta > 0$, at most $\lfloor (1 - J(p_0))/\delta \rfloor$ strict acceptances occur. On a finite reachable state set, a strictly improving sequence has at most $|\mathscr{P}| - 1$ accepted edges.

Proof. The maximum exists because B_t is finite and its score is at least that of its member p_t. Summing the strict gains and using $J \le 1$ gives the δ bound. Strict increase forbids repeating a state, proving the finite-state bound. □

These bounds count strict acceptances, not proposal calls, rejections, or equal-score changes. Finite termination of a complete optimization procedure also needs a stopping rule. A terminal state is a blockwise optimum only if all allowed block alternatives have been evaluated without improvement. It need not be globally optimal. No maximum over arbitrary natural-language prompts is assumed to exist.

Proposition 6.2 (Acceptance with certified evaluation error)**.** *Suppose $|\widehat{J}(p) - J(p)| \le \epsilon$ holds simultaneously for the current and proposed systems. Accept only if*

$$\widehat{J}(p') \ge \widehat{J}(p) + 2\epsilon.$$

Then $J(p') \ge J(p)$. More generally an empirical margin m certifies a true gain of at least $m - 2\epsilon$.

Proof. $J(p') \ge \widehat{J}(p') - \epsilon \ge \widehat{J}(p) + m - \epsilon \ge J(p) + m - 2\epsilon$. □

6.2 What a validation estimate can certify

Proposition 6.3 (Finite-class held-out bound)**.** *Let $\mathscr{H}$ be a class of K systems specified independently of a held-out sample of n independent examples and execution random variables. Each system's per-example reward lies in $[0,1]$, with mean $J(p)$ and empirical mean $\widehat{J}_n(p)$. With probability at least $1 - \alpha$, simultaneously for all $p \in \mathscr{H}$,*

$$|\widehat{J}_n(p) - J(p)| \le \sqrt{\frac{\log(2K/\alpha)}{2n}}.$$

If $\hat{p}$ maximizes the empirical mean over $\mathscr{H}$, its true score is within twice this bound of the best true score in $\mathscr{H}$.

Proof. Hoeffding's inequality [5] gives tail probability at most $2e^{-2n\epsilon^2}$ for a fixed system. A union bound over K systems and the stated choice of ϵ prove the simultaneous bound. Comparing the empirical maximizer to a true maximizer and applying the error bound twice gives the final assertion. □

Selection may be adaptive *within* an independently fixed finite class because the event is uniform over that class. If prompts are generated using feedback from the same held-out data, the number of adaptively evaluated candidates cannot simply be substituted for K. One needs an independent final evaluation, a uniform bound for a predeclared larger class, or an appropriate adaptive-data-analysis method [6]. Empirical monotonicity on a reused validation set is a different assertion from population improvement.

7 Interactions between executor changes

Fix the coordinator and let $b = (e_1^0, \dots, e_N^0)$ be the executor baseline. Suppose candidate e_i^1 is selected while the other executors remain at their baseline values. For $A \subseteq [N]$, let p^A use the new executor precisely at coordinates in A, and write $j(A) = J(p^A)$.

Example 7.1 (Standalone improvements can conflict)**.** For two executor choices, define the exact score table

	e_2^0	e_2^1
e_1^0	1/2	3/4
e_1^1	3/4	0

Each independent executor selection strictly improves the baseline, but their recombination has score zero. This is realizable as a finite stochastic system: conditional on the two executor flags, output a Bernoulli success with the displayed probability and reward success by one. Thus the counterexample is an execution model, not just an unconstrained numerical assignment.

The example refutes an unconditional inference from individually favorable frozen-context scores to favorable joint performance. It does not refute the empirical CEPO results, and makes no claim about which context was actually used by every implementation of the source algorithm. It establishes why that implementation detail matters for a proof.

Theorem 7.2 (Exact interaction decomposition)**.** *For $S \subseteq [N]$, define its baseline mixed difference*

$$\Delta_S j = \sum_{T \subseteq S} (-1)^{|S|-|T|} j(T).$$

For every $A \subseteq [N]$,

$$j(A) - j(\varnothing) = \sum_{i \in A} \Delta_{\{i\}} j + \sum_{\substack{S \subseteq A \\ |S| \geq 2}} \Delta_S j. \tag{7}$$

If $\Delta_S j \geq -\beta_S$ for $|S| \geq 2$, with $\beta_S \geq 0$, then

$$j(A) - j(\varnothing) \geq \sum_{i \in A} \big(j(\{i\}) - j(\varnothing)\big) - \sum_{\substack{S \subseteq A \\ |S| \geq 2}} \beta_S.$$

Proof. Expand $\sum_{S\subseteq A}\Delta_S j$ and interchange the finite sums. The coefficient of $j(T)$ is

$$\sum_{T\subseteq S\subseteq A}(-1)^{|S|-|T|}=(1-1)^{|A\setminus T|},$$

which is one for $T=A$ and zero otherwise. Subtract the empty-set term $\Delta_\varnothing j=j(\varnothing)$ and separate singletons. The bound follows by replacing each higher-order term by its lower bound. □

This is Boolean-lattice Mobius inversion applied to executor interventions, not a new general inversion theorem. It specifies the missing interaction evidence: positive singleton gains alone do not control the higher terms. In Example 7.1, the singleton gains are $1/4$ each and the pair term is -1, yielding the actual change $-1/2$.

Proposition 7.3 (A sufficient condition for safe recombination)**.** *Suppose j has increasing marginal returns: for $A\subseteq B$ and $i\notin B$,*

$$j(A\cup\{i\})-j(A)\le j(B\cup\{i\})-j(B).$$

If each $j(\{i\})\ge j(\varnothing)$, then $j(A)\ge j(\varnothing)$ for every A. More precisely, the joint gain is at least the sum of the corresponding singleton gains.

Proof. Order the members of A arbitrarily and telescope the gain as they are inserted. Increasing marginal returns bounds each insertion gain below by that member's marginal at the empty set. Sum these bounds. □

Increasing marginal returns is the supermodularity condition on this Boolean lattice. Additive objectives satisfy it with equality. It is a substantive property of the composite objective, not a consequence of shared coordination or operadic associativity. Instead of estimating all interactions, one can evaluate the recombined candidate against the incumbent, or apply Theorem 6.1 after each executor change in the actual current context.

Example 7.4 (Context-dependent optimization need not commute)**.** Let two binary executor coordinates have values

$$j(00)=0,\quad j(10)=2/3,\quad j(01)=2/3,\quad j(11)=1/3.$$

A coordinate best-response update chooses the higher score, holding the other coordinate fixed. Starting at 00, updating coordinate 1 then 2 gives 10; reversing their order gives 01. The fixed replacement maps commute as in Proposition 3.4, but the choices made after contextual scoring do not.

Example 7.5 (Coordinator-first is not universally optimal)**.** For a coordinator flag c and executor flag e, take

$$J(0,0)=1/2,\quad J(1,0)=2/5,\quad J(0,1)=3/5,\quad J(1,1)=1.$$

Starting at $(0,0)$, one exact coordinator block update retains $c=0$, then one executor update reaches score $3/5$. Executor-first followed by coordinator reaches one. Another coordinator pass after the first schedule also reaches one. Hence a one-pass coordinator-first preference is a data-dependent design choice, not a universal categorical theorem.

8 An implementable, typed acceptance procedure

The formalization suggests the following finite-horizon procedure. A candidate is represented as a typed record containing its coordinator, executors, interface evidence, and execution configuration. A migration candidate must carry the required transport/adaptation evidence; a local candidate must retain the current coordinator. In the literal model these checks can be just syntactic, while stronger models require a checker for the chosen contract language.

1. Select a role and generate a finite batch of candidates with the relevant guidance. Retain the actual current system as an explicit candidate.
2. Check admissibility of the complete assembled systems. Reject a candidate whose proof obligations have not been discharged by the declared checker.
3. Evaluate each candidate under a recorded context: model identifier, prompt bytes, topology, input distribution or dataset, randomization scheme, reward, evaluator version, and finite horizon.
4. Compare the full assembled candidate to the incumbent. Use exact acceptance when available, or a stated empirical/error-certified rule. Persist the context and score with the chosen system.
5. After any independently proposed batch, evaluate the recombined tuple before accepting it. If using sequential updates, score each update with all already accepted changes in place.
6. Stop when the budget or stopping criterion is reached. Report the selected system and its evaluation record, without claiming an unproved global optimum.

In a finite stochastic reference implementation, a proof-carrying interface may be represented by a decidable schema check plus explicit finite matrices. A production LLM service usually does not expose its exact kernel. The matrix model then serves as a specification and counterexample language; empirical confidence statements require an independently justified statistical protocol.

Record separation is mathematically consequential. If the model version, reward, or evaluation distribution changes between comparisons, the same symbol J no longer denotes one fixed function. The monotonicity proof must either restart in the new context or use a transport theorem bounding the change, such as Theorem 4.2 when its hypotheses apply.

9 Validation, limitations, and claims

The companion `verify.py` uses only the Python standard library and rational arithmetic for the numerical identities. It enumerates small score tables, checks current-context monotonicity and the two-coordinate supermodular sufficient condition, verifies the displayed counterexamples, checks Boolean-cube inversion on bounded finite instances, and checks finite stochastic contraction and the substitution inequality. Its JSON report records case counts and results. These computations support finite instances of the arguments and the correctness of the examples; they do not replace the proofs or mechanize infinity-category theory.

The following limitations are part of the statement of the work.

- No categorical construction yields task accuracy merely from separating prompt components. Performance is an additional observable depending on implementation and data.
- The contract-indexed model assumes specified functorial migrations; it is a semantic refinement of the literal architecture, not a fact proved about arbitrary natural-language edits.

- Operadic coherence gives well-defined composition and substitution of certified equivalences. It does not imply statistical independence, additive rewards, or convergence of an optimizer.
- Topos descent concerns coherent local-to-global assembly. It supplies neither an effective method for finding the coherence witnesses nor a guarantee that local agents will generate them.
- Current-context acceptance is monotone for its fixed exact objective. Reused empirical validation, changing environments, and noisy evaluations require separate error control.
- The stochastic proofs assume finite states and a finite horizon. General LLM runtimes and unbounded distributed executions require additional probability and domain-theoretic infrastructure.

The proposed semantics therefore gives a precise specification language for the architectural split, a compositional account of observations, and conditional optimization guarantees. Its falsifiable operational consequence is simple: independently favorable executor updates require an interaction condition or joint revalidation before improvement of their recombination can be certified.

Authorship and automation disclosure. This anonymous manuscript was prepared with OpenAI Codex assistance at the request of a human requester. Generative assistance was used for mathematical exposition, source retrieval, companion programming, and document production. The source architecture and results are attributed to Kim et al. The mathematical results here are supplied with written proofs and finite computational checks; no independent referee certification, Lean verification, or replication of the CEPO benchmark experiments is claimed. The human submitting author is responsible for the final submitted version and its declarations.

Data and code availability. The accompanying publication package contains this manuscript's LaTeX and Markdown sources, its PDF, the exact-arithmetic verification program and report, bibliographic/link provenance, build instructions, and a submission-preparation record. It contains no model credentials, user datasets, or copied benchmark results. The package is prepared for the journal at https://agentic-meta-journal.pages.dev/; a preparation record is not a submission receipt.

References

[1] Jongha Kim, Seohyun Lee, Jonathan Minho Chung, Jinsung Yoon, and Hyunwoo J. Kim. *CEPO: A General Framework for Improving Prompt Optimization in Multi-Agent Systems*. arXiv:2610.34722v1, October 26, 2026. https://arxiv.org/abs/2610.34722v1. The motivating architecture; CC BY 4.0 as indicated on its arXiv HTML version.

[2] Jacob Lurie. *Higher Topos Theory*. Annals of Mathematics Studies 170. Princeton University Press, 2009. In particular, §3.2 (straightening and unstraightening) and §6.1.3 (descent). https://www.math.ias.edu/~lurie/papers/HTT.pdf.

[3] Jacob Lurie. *Higher Algebra*. Author's version, September 18, 2017. In particular, §2.1 (infinity-operads and their algebras). https://www.math.ias.edu/~lurie/papers/HA.pdf.

[4] Tobias Fritz. *A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics*. Advances in Mathematics 370 (2020), 107239. https://arxiv.org/abs/1908.07021.

[5] Wassily Hoeffding. *Probability inequalities for sums of bounded random variables*. Journal of the American Statistical Association 58(301) (1963), 13–30. https://doi.org/10.1080/01621459.1963.10500830.

[6] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. *Generalization in Adaptive Data Analysis and Holdout Reuse*. Advances in Neural Information Processing Systems 28 (2015). https://arxiv.org/abs/1506.02629.