Higher-Categorical Semantics of Coordinator-Executor Prompt Optimization

Anonymous (human-directed, AI-assisted)

1 November 2026

Abstract

We specify coordinator-executor prompt optimization using dependent fiber products and finite Markov kernels. We prove contextual substitution and a quantitative replacement bound, descent of configurations, typed coordinator transport, conditional monotonicity, and finite-class audit guarantees. Exact countermodels exhibit failures of pairwise coherence and independently selected executor updates. These are applications of standard mathematical tools, not universal performance guarantees for language models.

1. Scope and relation to CEPO

Kim, Lee, Chung, Yoon, and Kim introduce CEPO, separating a shared coordinator prompt from agent-specific executor prompts [1, Section 3]. Both layers optimize the same system objective; role-specific meta-instructions guide proposal generation. Their default schedule optimizes the coordinator first. Equation (4) and Algorithm 1 describe executor proposals against initial other executors, whereas Appendix B.2 also describes sequential optimization with other prompts fixed. We distinguish fixed-initial and current-incumbent contexts below rather than infer an implementation from this ambiguity.

Our specification connects shared parameters, contextual execution, coherent gluing, and validated updates. Its countermodels concern defined rules, not measured failures of CEPO. We do not reproduce its experiments. Background is Markov categories [2], compositional learning [3], higher-topos descent [4], and concentration [5].

Throughout, $N \geq 1$ is finite. The executable fragment has finite state spaces, a fixed horizon H, and rational transition probabilities. The mathematical results also allow real probabilities on finite spaces. Source versions and links appear in Section 9.

2. Syntax, dependent roles, and shared parameters

Let $S = \Sigma^*$ be finite strings over a fixed alphabet, with empty string ϵ and concatenation ce. An untyped parameter set is

$$P_{\mathrm{raw}} = S \times S^N, \qquad A(c, e_1, \ldots, e_N) = (ce_1, \ldots, ce_N).$$

Proposition 2.1 (expressivity and nonidentifiability). If all strings and the empty coordinator are allowed, A is surjective and generally noninjective.

Proof. A tuple (s_i) is assembled from $(\epsilon, (s_i))$. For any nonempty string a, the parameters $(a, (\epsilon)_i)$ and $(\epsilon, (a)_i)$ have the same assembly. □

Length limits, admissibility, and proposal distributions can change the search problem. Delimiters can distinguish syntax; none supplies a semantic law for concatenation. We do not assume

$$\mathrm{sem}(ce) = \mathrm{sem}(c) \circ \mathrm{sem}(e).$$

For certified configurations let C be coordination contracts and let $\pi_i : E_i \to C$ specify executor implementations admissible under each contract. An element of the fiber $E_i(c)$ includes any required evidence of admissibility. Define

$$P = E_1 \times_C \cdots \times_C E_N \simeq \sum_{c \in C} \prod_{i=1}^{N} E_i(c).$$

The independent syntax is the special case $E_i = C \times S$. Admissibility is an explicit modeled predicate or certificate, never something guaranteed merely by natural-language role instructions. If the predicate is undecidable, this definition does not provide an algorithm deciding it.

A shared parameter is copied before each invocation. In finite stochastic semantics this is the deterministic kernel $\Delta_C(c) = \delta_{(c,\ldots,c)}$. If an initial coordinator is sampled once from μ, its copies have law $(\Delta_C)_*\mu$, not generally $\mu^{\otimes N}$. For a fair bit and two copies the former is supported on $00, 11$; the latter is uniform on four outcomes. Their total variation distance is $1/2$. Sharing and independently resampling are distinct constructions.

3. Bounded stochastic execution and contextual substitution

For finite nonempty sets X, Y, a kernel $K : X \rightsquigarrow Y$ assigns a probability distribution $K(- \mid x)$. Composition and tensor are

$$(L \circ K)(z \mid x) = \sum_y L(z \mid y) K(y \mid x), \qquad (K \otimes L)(y, y' \mid x, x') = K(y \mid x) L(y' \mid x').$$

Together with deterministic copying and discarding these give **FinStoch** [2]. Tensor expresses a specified conditional independence; it does not assert that interacting agents are independent.

A runtime fixes models, tools, decoding, scheduler, initial law, and horizon. Histories include all state needed by transitions, including tool responses and errors. A scheduler kernel selects an executor or stop; an executor kernel produces an event. Stops are absorbing through H. Composition and final observation define

$$K_p : Q \rightsquigarrow O.$$

For a distribution ν of labeled tasks $z = (q, y)$ and a fixed score $0 \le M(o, y) \le 1$, set

$$J_\nu(p) = \sum_{q,y} \nu(q,y) \sum_o K_p(o \mid q) M(o,y).$$

Runtime sampling is included in K_p. A finite empirical score is a separate object. Shared hidden state must be included in the history or supplied as a joint kernel. We make no termination claim for an unbounded agent loop.

Components are *contextually equivalent* when replacement preserves observable laws in every permitted environment and continuation. Equality of a benchmark score is weaker.

Theorem 3.1 (contextual substitution). Suppose two versions of executor i have identical next-event kernels for every permitted parameter context and history. Then substituting one for the other preserves the law of every bounded transcript, hence every allowed final observation and score, under the same scheduler and remaining components.

Proof. Initial laws coincide. Equal prefix laws give equal scheduler laws. Selected event kernels agree by hypothesis or because the component is unchanged. Summing over prefixes proves equality at the next step. Induction reaches H; postprocessing preserves equality. □

For finite distributions write

$$d_{\mathrm{TV}}(\mu, \eta) = \tfrac{1}{2} \sum_x |\mu(x) - \eta(x)|.$$

Theorem 3.2 (quantitative substitution). Keep the initial law and scheduler fixed. Suppose replacement executor i differs from the original by at most ε_i in total variation at every common history and parameter context. If every permitted run calls executor i at most n_i times, then, for each query,

$$d_{\mathrm{TV}}(K_p(- \mid q), K_{p'}(- \mid q)) \le B := \min\{1, \sum_i n_i \varepsilon_i\}.$$

Consequently $|J_\nu(p) - J_\nu(p')| \le B$.

Proof. Couple initial states identically. While histories agree, couple scheduler choices identically and use a maximal coupling of the chosen event distributions. A first event disagreement at a call to i has conditional probability at most ε_i. A union bound over calls before the first disagreement is at most $\sum_i n_i \varepsilon_i$: every such common prefix extends an original run satisfying the call bounds. The coupling inequality bounds transcript total variation, and observation postprocessing cannot increase it. The expectation difference of a $[0,1]$-valued function is at most total variation. Average over ν. □

Distances estimated on old-system histories do not certify uniform control on new histories. Unbounded horizons need additional summability hypotheses.

4. Local writes, incompatible improvements, and validated updates

Let $R_j \subseteq P \times P$ be an explicitly permitted update relation. Executor updates keep the coordinator and all other executors fixed. Coordinator updates in dependent fibers require transports, as specified in Section 6. Role guidance may propose a candidate; a separate checker establishes membership in P and R_j.

Proposition 4.1 (when executor updates commute). For fixed c, maps $u_i : E_i(c) \to E_i(c)$ induce maps on $\prod_i E_i(c)$ by changing only their own coordinates. Distinct such induced maps commute.

Proof. Both composites apply the same map to each affected coordinate and the identity elsewhere. □

The hypothesis includes a restriction on *reads*. Merely writing one coordinate does not suffice. On two bits, let $U_1(x, y) = (y, y)$ and $U_2(x, y) = (x, x)$. Then

$$U_2U_1(0, 1) = (1, 1), \qquad U_1U_2(0, 1) = (0, 0).$$

An optimizer reading the full team context is not a coordinatewise map of Proposition 4.1.

Counterexample 4.2 (merging individually improving updates). Take a singleton coordinator, two binary executors, and

(e_1, e_2)	$(0,0)$	$(1,0)$	$(0,1)$	$(1,1)$
J	$1/2$	$3/4$	$3/4$	$0.$

Relative to $(0, 0)$, changing either executor improves J by $1/4$. Selecting both changes in the initial context and assembling them gives score 0, a loss of $1/2$. These scores are realized by a finite kernel returning a success bit with the displayed probability. Thus individually valid and improving proposals need not yield a jointly improving team. This counterexample applies to the fixed-initial-context rule, irrespective of how its candidates are generated.

A sufficient repair is a sequential acceptance rule. At the current incumbent p, obtain candidates $B_j(p)$ from training feedback, include p itself, reject inadmissible candidates, and choose

$$p^+ \in \operatorname*{arg\,max}_{q \in B_j(p) \cup \{p\}} J(q).$$

Candidates are finite; ties retain the incumbent. Every evaluated candidate is a *whole-team configuration in the current context*, including the incumbent values of other executors. An alternative repair evaluates the merged team and rolls it back if it fails the same acceptance condition.

Theorem 4.3 (conditional monotonicity and finite termination). With an exact stationary score and nonempty finite candidate sets, the sequential rule makes scores nondecreasing. If P is finite, at most $|P| - 1$ changes are accepted. If, on every visit to block j, all its permitted alternatives are considered, and a complete pass with no change triggers termination, the terminal state has no strictly improving permitted one-block neighbor.

Proof. The current state is a candidate, so the selected score is at least its score. A state change strictly raises the score by the tie rule. No state can recur, proving the bound. In a terminal pass the state never changes, and exhaustive checking of each block rules out every strictly better neighbor. □

This is a sufficient implementation contract, not a theorem about arbitrary prompt optimizers. It yields a one-block optimum, not a global optimum: the table $(1/2, 1/4, 1/4, 1)$ traps the rule at 00 although 11 is better. Infinite prompt spaces need not terminate. Noisy re-evaluation need not preserve the exact-score conclusion.

5. Population claims require an independent audit

Let $\mathcal{H}$ be a nonempty finite set of complete configurations, fixed independently of an audit sample. Conditional on any training randomness, take m independent identically distributed task-and-runtime seeds Z_k, with reward $g(p, Z_k) \in [0, 1]$. Define

$$\widehat{J}(p) = m^{-1} \sum_{k=1}^{m} g(p, Z_k), \quad J(p) = \mathbb{E}g(p, Z), \quad \epsilon = \sqrt{\frac{\log(2|\mathcal{H}|/\delta)}{2m}},$$

where $0 < \delta < 1$. Rewards for different p may be correlated through reuse of the same task seeds.

Theorem 5.1 (finite-class audit). With probability at least $1 - \delta$, simultaneously for all $p \in \mathcal{H}$,

$$|\widehat{J}(p) - J(p)| \leq \epsilon.$$

On this event an empirical maximizer has population regret at most 2ϵ within $\mathcal{H}$. Any sequence of changes inside $\mathcal{H}$ accepted only when

$$\widehat{J}(p^+) - \widehat{J}(p) \geq 2\epsilon$$

is nondecreasing in population score.

Proof. For each fixed p, Hoeffding's inequality [5] bounds deviation probability by $2e^{-2m\epsilon^2}$. A union bound gives δ. For an empirical maximizer $\hat{p}$ and any p, insert $\widehat{J}(p)$ and $\widehat{J}(\hat{p})$ to obtain $J(p) - J(\hat{p}) \leq 2\epsilon$. Similarly,

$$J(p^+) - J(p) \geq \widehat{J}(p^+) - \widehat{J}(p) - 2\epsilon \geq 0.$$

All claims follow on the same uniform event. □

Adaptive selection *within the predetermined class* is allowed. Counting prompts adaptively generated from the audit does not justify this bound. Use an independent final audit or an appropriate adaptive-data-analysis guarantee. Empirical monotonicity need not imply population monotonicity; coordinator-first need not be optimal.

6. Higher-topos semantics and typed coordinator revisions

Let B be a topological space and $\mathcal{X} = \mathrm{Sh}_\infty(B)$ its infinity-topos of sheaves of spaces [4]. The same construction works on a small site with the needed Cech diagrams. Choose objects C and $\pi_i : E_i \to C$ in $\mathcal{X}$, and define their finite homotopy fiber product

$$P = E_1 \times^h_C \cdots \times^h_C E_N.$$

Here $C(U)$ is a space of coordination contracts on context U; the homotopy fiber $E_i(U)_c$ contains executor data and admissibility witnesses. Paths and higher simplices mean specified translations and coherence witnesses. Raw strings are a zero-truncated special case. Distinct prompts or equal benchmark scores do not automatically supply paths in these spaces.

Theorem 6.1 (descent of dependent teams). For an open cover (U_a) of U, restriction induces an equivalence

$$P(U) \simeq \lim_{[n]\in\Delta} \prod_{a_0,\ldots,a_n} P(U_{a_0} \cap \cdots \cap U_{a_n}).$$

Thus *full coherent descent data* determine a global team up to a contractible space of compatible choices.

Proof. The analogous maps for C and each E_i are equivalences because these objects are sheaves of spaces. Mapping out of an open context preserves limits, so $P(U)$ is their homotopy fiber product. Limits commute with limits; taking that finite fiber product through the Cech limit proves the displayed equivalence. Its homotopy fiber over any complete descent datum is contractible. □

Only Cech descent for covers is used; no hypercompleteness is assumed. Arbitrary prompts need not supply descent data.

For a groupoid-valued example, overlap identifications ϕ_{ab} must satisfy $\phi_{bc}\phi_{ab} = \phi_{ac}$ on triple intersections. With automorphism group $\mathbb{Z}/2$, the transition labels $g_{01} = 0$, $g_{12} = 0$, $g_{02} = 1$ on a nonempty triple overlap violate this equation. Pairwise equivalences exist, but they do not constitute descent data. For general space-valued data, triangle witnesses must themselves satisfy the higher coherence conditions encoded by the full Cech limit.

Proposition 6.2 (typed coordinator transport). Let $s, t : R \to C$ describe allowed coordinator revisions. Suppose, for each executor, a morphism over R is supplied:

$$\tau_i : s^*E_i \longrightarrow t^*E_i.$$

Then there is an induced morphism $s^*P \to t^*P$, compatible with restriction to contexts.

Proof. Pullback preserves finite limits. Hence s^*P and t^*P are the respective products in $\mathcal{X}_{/R}$. The maps τ_i induce a map of these products by their universal property. All maps are in the sheaf category, so commute coherently with restriction. □

Without transports a new coordinator may leave old executors ill-typed. Transport certifies admissibility, not score improvement, which needs separate evidence.

The higher semantics includes a specified observation sheaf O and morphism $\mathrm{sem} : P \to O$. Naturality then makes observations commute with restriction and descent. Such a morphism is extra structure to construct and verify; it is not a consequence of concatenating instructions. Equality in a zero-truncated observation object need not imply a certified equivalence of configurations.

One can also choose a zero-truncated ordered score object V, a morphism $J : P \to V$, and update subobjects $R_j \hookrightarrow P \times P$. The internal predicate

$$\mathrm{Opt}(p) \equiv \forall q : P,\ ((\bigvee_j R_j(p,q)) \Rightarrow J(q) \leq J(p))$$

defines a subobject $\mathrm{Opt} \hookrightarrow P$. As an object of $\mathcal{X}$, it satisfies descent. This is a statement about *internal* comparison, with quantification over all local generalized elements. An external global benchmark, total network cost, or cross-context capacity constraint need not define this local predicate; its optimizers need not glue.

We do not assert that **FinStoch** is a topos or that arbitrary infinity-toposes provide probability, integration, or effective optimization. Relating the external stochastic model to the contextual specification requires a chosen, verified observation interpretation.

7. Executable evidence and implementation obligations

The dependency-free Python program uses exact rational arithmetic. It checks 729 kernel associativity triples, 81 data-processing cases, 6,561 two-step replacement cases, shared sampling, update countermodels, and the parity obstruction. Across 625 four-entry score tables over $\{0, 1/4, 1/2, 3/4, 1\}$, 1,250 sequential runs satisfy monotonicity; 20 tables permit the fixed-initial merge regression.

These finite witnesses do not machine-check the theorems, implement higher descent, certify real prompts, estimate kernel distances, or reproduce language-model experiments.

A conforming system records versions, team context, admissibility evidence, runtime settings, data provenance, candidate rules, rewards, and acceptance decisions. It validates coordinator transports and whole-team candidates. Failures are explicit outcomes. Score comparisons use a stable metric and an appropriate treatment of randomness.

8. Conclusion and limitations

Fiber products express shared contracts, kernels express execution, descent expresses coherent gluing, and transports express typed revision. None establishes superior performance by itself.

Baseline-relative improvements can fail when merged; current-incumbent validation has a conditional monotonicity proof. We claim a specification and countermodels, not a new general descent theorem, universal optimizer, or empirical result about CEPO.

9. 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, 26 October 2026. Versioned source; full text. The initiating paper; authorship and its empirical claims remain with its authors.

2. Tobias Fritz. *A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics*. Advances in Mathematics 370, 107239, 2020. DOI; arXiv:1908.07021v8.

3. Brendan Fong, David I. Spivak, and Remy Tuyeras. *Backprop as Functor: A compositional perspective on supervised learning*. LICS, 2019. arXiv:1711.10455v3. Background on compositional update rules; we assert no differentiability of prompts.

4. Jacob Lurie. *Higher Topos Theory*. Annals of Mathematics Studies 170, Princeton University Press, 2009, Chapter 6. Author-maintained text; arXiv:math/0608040.

5. Wassily Hoeffding. *Probability Inequalities for Sums of Bounded Random Variables*. Journal of the American Statistical Association 58(301), 13-30, 1963. DOI.

Authorship, automation, and reproducibility

Anonymous human-directed submission; the directing human researcher is the responsible submitting party. OpenAI Codex assisted with source reading, mathematics, drafting, code, and checks. The source authors did not participate or endorse this work. No proof-assistant verification, independent referee review, or language-model benchmark experiment is claimed. The bundle includes sources, PDF, exact code, verification output, and provenance.