Local Interface Descriptions Do Not Compose

John Komkov, Independent Researcher

Abstract

We identify a class of compositionally relevant predicates — hidden conventions — that are not recoverable from local interface descriptions alone. The coherence fee, defined as the rank deficiency of the observable coboundary matrix, measures the number of independent convention dimensions lost when tools are composed. We prove that hiddenness is a graph-relative predicate: the same tool has different hidden sets depending on its composition partner (Theorems 1–2). In a controlled synthetic benchmark, frontier LLMs recover this non-local object with 88% accuracy under structured relational prompts but collapse to 46% under flat representations, reverting to lexical heuristics. The effect is representation-conditioned: it requires both relational framing and convention-specific task language, each contributing +33 percentage points. Cross-model replication reveals divergent failure modes — higher precision but fragile recall (Claude) versus robust recall but 3× more false positives (GPT-4o). Neither achieves stable exact computation. Bulla, an algebraic diagnostic layer, computes the non-local object exactly across all conditions.

---

1. Introduction

When AI agents compose tools from multiple servers, they face a problem that no amount of schema reading can solve: some fields carry conventions that are invisible in the schema but critical for correct composition. A `path` field in a filesystem server and a `path` field in a GitHub server may follow different normalization conventions (POSIX vs URL-encoded), but nothing in either schema reveals this.

We call these fields *hidden*. Their identification is the bottleneck for safe tool composition — not because the conventions themselves are complex, but because determining *which fields are hidden* requires reasoning about the composition graph, not just the individual schemas.

This paper makes three contributions:

1. **Theorem (graph-relativity).** We formalize hiddenness as a graph-relative predicate: the same field in the same tool may or may not be a blind spot depending on the composition partner. Local schema inspection cannot determine this; the coherence fee quantifies what is missing. (§3)
2. **Experiment (representation-conditioned access).** In a synthetic benchmark where ground truth varies with composition partner, the same `data_loader` tool gets different hiddenness judgments depending on its partner — and frontier LLMs recover this non-local variation with 88% accuracy under structured relational prompts but collapse to 46% under flat representations. Renaming a field from `direction` to `path` increases identification from 0% to 58% ($p = 0.008$), holding composition structure fixed: the models are tracking surface tokens, not compositional structure. (§4)
3. **System (algebraic diagnostic).** Bulla computes the coherence fee and hidden set exactly via the coboundary matrix, providing the stable global computation that models lack. (§5)

1.1 Related Work

Schema integration and ontology matching. For twenty-five years, the schema-matching and ontology-alignment communities have pursued a program of local enrichment: making individual schemas richer, combining matchers more sophisticatedly, and iterating alignment procedures until pairwise correspondences converge. Rahm and Bernstein [1] defined the taxonomy of schema-matching approaches — element-level vs. structure-level, schema-based vs. instance-based — that has governed the field since 2001. Euzenat and Shvaiko [2] extended this to the ontology layer, proposing richer alignments between richer ontologies as the path to interoperability. The shared premise is that composition is a byproduct of pairwise correctness: match each pair well enough and the global system will cohere. Our result identifies a structural limitation of this framing. When the composition graph has nontrivial cycle structure, pairwise correctness does not entail global coherence: the obstruction is a rank deficiency in the observable coboundary that persists regardless of how expressive each local description becomes.

Prompt sensitivity and representation effects. A growing literature documents that LLM performance varies substantially under meaning-preserving surface changes. Sclar et al. [3] show up to 76-point accuracy swings across format perturbations and frame this as spurious variance to be averaged over. Min et al. [4] identify three structural features of in-context demonstrations — label space, input distribution, sequence format — whose presence suffices to activate pre-existing competence, independent of label correctness. Both treat the prompt as noise around a latent function. Our claim is categorically different: for certain non-local computations, there exists no surface variant in the sampled equivalence class that makes the computation accessible unless the representation preserves specific relational structure. In their frame, format is noise around a stable competence. In ours, representation is a precondition that either opens or closes the computation. Noise sensitivity is not structural access; variance is not support.

Assume-guarantee reasoning and session types. The formal-methods community has long known that bilateral reasoning is incomplete for global properties. Jones [5] and Misra and Chandy [6] established this for shared-state and message-passing concurrency, respectively; Benveniste et al. [7] give the modern meta-theoretic treatment. Honda, Yoshida, and Carbone [8] responded by introducing multiparty session types — global protocol specifications that project to locally checkable endpoint obligations, the closest existing analogue to our setting. Their projection machinery is a constructive solution: given a global type, verify each endpoint locally. We provide the complementary impossibility result: when components are LLM agents with unspecifiable internal semantics, the bilateral certificates that assume-guarantee reasoning produces cannot close the sheaf-cohomological defect. We identify the specific obstruction in the tool-composition setting, give it a computable characterization (the coherence fee), and measure it empirically.

---

2. Definitions

2.1 Composition Graph

A *composition* is a tuple $G = (\mathcal{T}, E, \delta)$: - $\mathcal{T} = \{T_1, \ldots, T_n\}$: tools, each with internal state $S(T_i)$ and observable schema $\sigma(T_i) \subseteq S(T_i)$ - E: edges, one per shared convention dimension between tool pairs - $\delta : C^0 \to C^1$: the coboundary operator (signed incidence matrix)

A field $f \in S(T_i)$ is *hidden* if $f \notin \sigma(T_i)$ — its convention is not fully determined by the schema.

2.2 Coboundary and Fee

The coboundary δ has two projections: - δ_{obs}: restricted to observable columns (what callers can inspect) - δ_{full}: all columns including hidden fields

The **coherence fee** is:

$$\text{fee}(G) = \text{rank}(\delta_{\text{full}}) - \text{rank}(\delta_{\text{obs}}) = h^1_{\text{obs}} - h^1_{\text{full}}$$

where $h^1 = |E| - \text{rank}(\delta)$ counts independent cycles. The fee counts invisible cycles: semantic couplings that exist in the full composition but vanish when hidden fields are projected away.

2.3 Blind Spots

An edge $e = (T_i, T_j)$ on dimension d is a *blind spot* if at least one endpoint field is hidden. Blind spot fields are the hidden endpoints of blind-spot edges. They are the fields whose conventions could silently diverge between servers.

2.4 Worked Example

Consider two tools composed along a single shared dimension (path_convention):

- `file_reader`: fields {`path`, ***`content`***, ***`query`***}, *where* ***`path`*** *is hidden (marked)*
- `data_loader`: fields {`path`, ***`timestamp`***, `data`, `source`}, where `path` and `timestamp` are hidden

The composition has one edge on path_convention. The coboundary matrix δ has one row (the edge) and columns for each field. The full and observable projections are:

$$\delta_{\text{full}} = \begin{pmatrix} +1 & -1 \end{pmatrix} \quad \text{(columns: file\_reader.path, data\_loader.path)}$$

$$\delta_{\text{obs}} = \begin{pmatrix} \end{pmatrix} \quad \text{(empty — both endpoints are hidden)}$$

$\text{rank}(\delta_{\text{full}}) = 1$, $\text{rank}(\delta_{\text{obs}}) = 0$. The coherence fee is 1: one semantic coupling that exists in the full composition but vanishes under observable projection. The blind spot field is `path` — the dimension whose convention could silently diverge. Note that `data_loader.timestamp` is hidden but is *not* a blind spot here, because `file_reader` has no field on the date_format dimension. Whether `timestamp` becomes a blind spot depends on the composition partner (Theorem 1).

---

3. Non-Locality Theorems

Theorem 1 (Graph-Relativity of Hiddenness)

The following formalizes a systems intuition — that a field's compositional relevance depends on context, not just its own schema — and gives it a precise graph-theoretic characterization that makes it computable.

Formal statement. For any tool T with hidden field f, there exist compositions G_1, G_2 containing T such that f is a blind spot in G_1 but not in G_2.

Constructive proof. Let: - T_A = `file_reader` with hidden field `path` (path_convention) - T_B = `data_loader` with hidden field `path` (path_convention) - T_C = `event_logger` with hidden field `timestamp` (date_format)

In $G_1 = \{T_A, T_B\}$: both tools share the path_convention dimension. The edge creates a blind spot on `path`. $\text{fee}(G_1) = 1$.

In $G_2 = \{T_A, T_C\}$: no shared dimension (path_convention $\neq$ date_format). No edge, no blind spot. $\text{fee}(G_2) = 0$.

T_A's schema is identical in both compositions. Only the partner differs. □

Theorem 2 (Local Equivalence, Global Divergence)

There exist pairs of compositions sharing a tool where the fee differs.

Proof. Immediate from Theorem 1: $\text{fee}(G_1) = 1 \neq 0 = \text{fee}(G_2)$. □

Corollary

No algorithm inspecting only the schema of a single tool can determine whether a field is a blind spot. Hiddenness identification requires access to the composition graph.

Theorem 3 (Diagnostic Sufficiency)

Once the hidden set is externally identified, the minimum disclosure set has size exactly $\text{fee}(G)$, and specification is algebraically trivial.

Proof sketch. The minimum disclosure set is the basis of the quotient matroid M/O. By matroid theory, all bases have cardinality $\text{rank}(\delta_{\text{full}}) - \text{rank}(\delta_{\text{obs}}) = \text{fee}(G)$. □

---

4. Experiments

4.1 Ground Truth Methodology

All experiments use Bulla as the ground-truth oracle. A field's hidden/observable status is determined by a multi-signal heuristic classifier that combines three independent sources: (i) regex patterns matching field names against semantic dimensions (e.g., `path|filepath|dir_path` → path_convention; `timestamp|datetime|created_at` → date_format), (ii) JSON Schema structural signals (format annotations, enum values, numeric ranges), and (iii) description keyword matching. Confidence is assigned in tiers: *declared* (two or more independent signals agree), *inferred* (one strong signal), or *unknown* (weak signals only). Only declared and inferred fields participate in composition construction; unknown-confidence fields are recorded for auditability but do not affect the coboundary matrix.

Edges between tools are constructed by intersecting each tool's classified dimensions: if tool T_i has a field classified as dimension d and tool T_j also has a field classified as d, an edge is created on dimension d linking those fields. The coherence fee and blind spots are then computed exactly via Bulla's coboundary rank computation using rational arithmetic.

For the synthetic ecology benchmark (§4.2), six single-tool servers are constructed with field names chosen to trigger specific classifier dimensions. Each server exposes exactly one tool to eliminate

intra-server edges, ensuring that all edges arise from inter-server composition. The ground truth is computed by running Bulla's full `from_tools_list()` → `diagnose()` pipeline, yielding deterministic fee and blind-spot assignments for each composition.

Independent validation of ground truth. To break the circularity between Bulla-as-classifier and Bulla-as-oracle, we independently annotated 20 real MCP compositions (22 distinct servers) stratified across fee bands: 4 at fee = 0, 4 at fee = 1, 4 at fee = 2–3, and 8 at fee ≥ 10 (including the maximal filesystem + github pair at fee = 22). For each composition, we loaded both server manifests and compared raw JSON Schema constraints — `type`, `format`, `enum`, `pattern`, `minimum`/`maximum`, and description text — across all tool pairs, *without invoking Bulla's dimension classifier*.

Each of Bulla's 573 predicted blind spots across the 20 compositions was independently adjudicated as CONFIRMED (field genuinely hidden or schemas incompatible) or FALSE_POSITIVE (compatible schemas, no convention risk). We additionally checked all same-named field pairs *not* flagged by Bulla for type or format mismatches that would constitute missed blind spots.

Metric	Value
Compositions annotated	20 (22 servers, fee 0–22)
Bulla predictions evaluated	573
Independently confirmed	573
False positives	0
Missed blind spots	9
Precision	**1.000**
Recall	**0.985**

All 9 missed spots are *homonym collisions* — same-named fields with incompatible types that Bulla's dimension classifier does not map to the same semantic dimension: `title` (string in Google Tasks vs rich-text array in Notion, 4 pairs), `filter` (MongoDB query object vs Todoist filter string, 1 pair), and `head` (integer line count in filesystem vs git branch name in GitHub, 4 pairs). These are genuine runtime hazards that fall outside Bulla's current dimension taxonomy; they represent a category of structural mismatch (type-level homonymy) orthogonal to the convention-level blind spots the coherence fee measures.

The independent annotation data, including per-composition verdicts and evidence strings, is available at `calibration/data/registry/report/independent_annotation.json`.

4.2 Synthetic Ecology Benchmark

Design. Six single-tool synthetic servers with field names triggering known classifier dimensions. Eight pairwise compositions, of which two have nonzero fee and six have fee = 0. Ground truth computed by Bulla. Three prompt conditions (structured, bare, flat) × 3 repeats. Models: Claude Sonnet 4 (`anthropic/claude-sonnet-4` via OpenRouter, temperature 0.0) and GPT-4o (`openai/gpt-4o` via OpenRouter, temperature 0.0). All API calls made April 2026.

Key property. The same tool (`file_reader`, `data_loader`) appears in multiple compositions with different ground-truth blind spots.

Results (Claude Sonnet 4). 95% bootstrap CIs on exact match (10,000 resamples, N=24 trials per condition).

Condition	Exact Match [95% CI]	Precision	Recall	Non-locality
Structured	88% [75–100%]	55%	100%	Correct variation
Bare	50% [29–71%]	10%	100%	Partially correct
Flat	46% [25–67%]	12%	50%	Collapses

Under the structured prompt, the model correctly varies its answer for the same tool depending on composition partner: `data_loader + event_logger` → identifies `timestamp`; `data_loader + price_fetcher` → correctly returns empty. Under the flat prompt, this non-local reasoning disappears: `file_reader + data_loader` → misses `path` entirely (0/3).

False-positive signature. Under flat/bare prompts, the dominant false positive is `timestamp` (flagged 5–6 times in compositions where it is NOT a blind spot). This is the lexical proxy in action: absent structural reasoning, the model substitutes "this field name sounds convention-problematic."

4.3 Context Ablation

Design. 12 real filesystem compositions, 4 prompt conditions isolating relational framing, task language, and contextual cues.

Condition	`path` ID rate	Δ
Full (structured, named, "hidden conventions")	100%	—
Anonymous (structured, anonymous, stripped)	92%	−8%
No grouping (flat list, named, "hidden conventions")	67%	−33%
Neutral task (structured, named, "integration issues")	67%	−33%

Finding. Two factors gate the effect: relational framing (two-server grouping, +33pp) and convention-specific task language ("hidden conventions", +33pp). Server names and tool descriptions contribute negligibly (−8pp). The vocabulary phenomenon is representation-conditioned, not name-conditioned.

4.4 Vocabulary Phenomenon (Real Corpus)

Design. 60 pairwise compositions from 38 real MCP servers (GitHub stars ≥ 100), yielding 103 hidden field instances. Field-level identification scoring against Bulla ground truth.

Path-family fields: 59% identification rate. Non-path fields: 4%. Fisher's exact OR = 34.5, $p < 2.4\times10^{-6}$.

Within path-family: `path` (85%) » `paths` (28%). Same concept, morphological variant. Surface-form gradient, not reasoning gradient.

4.5 Lexical Intervention (Causal)

Design. 12 minimal pairs. Hold composition graph fixed, intervene only on field names. Three conditions: baseline, swap (rename canonical↔obscure), mask (neutral placeholders).

Condition	Obscure field ID rate
Baseline ("direction")	0%
Neutral ("value")	12%
Renamed to "path"	75%

McNemar exact binomial (baseline vs swap): $p = 0.0078$. The name change causes the identification change.

Neutral-token ablation. To rule out the alternative that "path" simply activates more attention weight than "direction" due to token frequency rather than convention relevance, we added a neutral condition: renaming the target field to "value" — a common English word with no convention association. Identification rises to only 12% (1/8), compared to 75% (6/8) when renamed to the convention-specific "path." The effect is vocabulary-specific, not frequency-driven: the model's lexical proxy targets convention-associated tokens, not merely common ones.

Prompt contingency. This effect appears under the structured two-server prompt. Under a flat prompt, the effect disappears entirely. The causal intervention is representation-conditioned.

4.6 Cross-Model Divergence

Real corpus. Same compositions, same fields, same algebraic status:

Field	Claude Sonnet 4	GPT-4o
`path`	42%	25%
`direction`	0%	88%

Identification is model-specific: a function of training distribution, not compositional structure.

Synthetic ecology replication (GPT-4o, N=8×3×3). 95% bootstrap CIs on exact match.

	Claude Sonnet 4	GPT-4o
Structured exact match	**88%** [75–100%]	42% [21–63%]
Structured precision	**55%**	17%
Structured recall	100%	100%
Flat exact match	46% [25–67%]	33% [17–54%]
Flat recall	**50%**	**100%**

Both models achieve 100% recall under structured prompts — they always find the real blind spots. But the failure modes diverge: - **Claude** is more precise (55% vs 17%) but more fragile: loses recall under flat prompts (50%). - **GPT-4o** is more robust: maintains 100% recall even under flat prompts. But it is dramatically noisier, flagging `content`, `timestamp`, `amount`, `currency` as false positives across nearly all compositions.

The false-positive profiles differ by model: - Claude structured FPs: `data`(2), `message`(2) - GPT-4o structured FPs: `content`(7), `timestamp`(5), `amount`(4), `currency`(3)

Different training distributions produce different precision/robustness tradeoffs on the same non-local predicate. Neither model achieves the exact, stable computation that Bulla provides.

---

5. Bulla: The Algebraic Diagnostic Layer

Bulla computes the coherence fee and hidden set via the coboundary matrix. Key properties:

1. **Exact.** Uses rational arithmetic (`Fraction`), guaranteeing field-independent rank computation. Total unimodularity of the coboundary ensures rank is the same over any field.
2. **Stable.** The fee is a topological invariant of the composition graph. It does not depend on prompt format, model choice, or field naming.
3. **Complete.** Bulla identifies all blind spots, computes the minimum disclosure set, and reports leverage scores (per-field indispensability) via the witness Gram matrix.
4. **Framing-invariant.** Unlike LLM identification, Bulla's output is the same regardless of how the composition is presented. On all 8 synthetic compositions $\times$ all 3 prompt conditions, Bulla returns the correct answer.

The engineering implication is direct: any agent stack that composes tools across organizational boundaries needs a diagnostic layer that sits above schemas and below planning, computing the composition graph's hidden structure before the planner encounters it at runtime. Bulla provides this layer.

---

6. Discussion

6.1 Representation-Conditioned Access

The deeper finding is not that prompts matter. It is that the model needs the composition *represented in a form that preserves relational structure*. Flat arrays erase the ecology; structured pair prompts expose it. This is not ordinary prompt sensitivity — it is a claim about what information the representation must carry for non-local reasoning to be possible.

The 88% structured accuracy with 100% recall suggests the model has partial access to the global object: it can sense the compositional seam, but cannot specify it cleanly (precision = 55%). It has access without exact representation. Bulla supplies what is missing: not more access, but exact, stable computation.

6.2 The False-Positive Signature

When structural reasoning is unavailable (flat prompt), the model's dominant false positive is `timestamp` — a field that sounds convention-problematic but is not a blind spot in the tested composition. This is the lexical proxy in direct observation: absent the global object, the model substitutes token-level priors about which field names "sound dangerous."

6.3 Limitations

We characterize current frontier models under direct prompting. The non-locality theorems constrain any approach working from bilateral schema inspection alone, but do not rule out approaches

with access to runtime behavior, extended context, or explicit composition-graph reasoning. The "two-channel model" (lexical prior × contextual redundancy) is an interpretive framework consistent with the data, not an experimentally isolated mechanism. The heuristic classifier that establishes ground truth achieves 1.000 precision and 0.985 recall against independent schema-level annotation on a stratified 20-composition sample (§4.1), but does not claim exhaustive coverage of all possible convention dimensions. The 9 missed cases — all type-level homonym collisions — suggest a complementary detection layer for structural type mismatches that is orthogonal to the convention-level analysis presented here.

---

7. Conclusion

Local interface descriptions do not determine compositional hiddenness. The missing object — the hidden convention ecology — is a graph-relative predicate that requires global computation. Current frontier LLMs recover this object only under supportive relational framing and otherwise revert to lexical heuristics. Bulla computes it directly and stably.

The graph-relativity formalization (§3) is model-independent: it constrains any approach that works from bilateral schema inspection, whether performed by an LLM, a rule-based system, or a human developer reading documentation. The empirical finding (§4) adds that current frontier models do not circumvent this constraint — their access to the non-local predicate is representation-conditioned and unstable. The coherence fee quantifies the gap between what bilateral inspection can certify and what global coherence requires. Algebraic diagnostic layers like Bulla provide one path to closing that gap; whether richer context, runtime observation, or explicit graph reasoning can provide others remains an open question.

Reproduction

All experiments are reproducible from the Bulla repository. API calls require provider keys. Temperature is set to 0.0 for all runs.

```
cd bulla

# Synthetic ecology benchmark (8 compositions × 3 conditions × 3 repeats)
python -m calibration.harness.run_synthetic_ecology --api-key $KEY --repeats 3

# Context ablation (4 conditions × 12 compositions)
python -m calibration.harness.run_context_ablation --api-key $KEY --max-cases 12

# Vocabulary phenomenon (60+ compositions)
python -m calibration.harness.run_familiarity_probe --full --api-key $KEY

# Lexical intervention (12 minimal pairs × 3 conditions)
python -m calibration.harness.run_lexical_intervention --api-key $KEY --max-cases 12

# Cross-model replication
python -m calibration.harness.run_synthetic_ecology --api-key $KEY --model openai/gpt-4o --repe
```

```
# Independent ground-truth annotation (no API key needed)
python calibration/scripts/independent_annotation.py
```

Data directory: `calibration/data/agent_confusion/`

Appendix A: Superseded Experiments

Several intermediate probes were run during development and superseded by the experiments reported above:

- A **second field-family forward intervention** (state→path, state→filepath) did not replicate the direction→path effect (+8% and 0% respectively, N=12). This is consistent with prompt contingency: the effect requires both the canonical token and supportive relational framing.
- A **token ladder** (12 field names in the same structural slot) showed uniformly near-zero identification under a flat prompt format, revealing the prompt-sensitivity that motivated the context ablation experiment.
- **Pilot runs** (N=20) of the vocabulary phenomenon and specification-gap experiments were superseded by full-corpus runs (N=60+).

These probes informed the experimental design but are not included in the curated result set.

References

[1] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, 2001.

[2] J. Euzenat and P. Shvaiko. *Ontology Matching*, 2nd edition. Springer-Verlag, 2013.

[3] M. Sclar, Y. Choi, Y. Tsvetkov, and A. Suhr. Quantifying language models' sensitivity to spurious features in prompt design. In *ICLR*, 2024.

[4] S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *EMNLP*, 2022.

[5] C. B. Jones. Tentative steps toward a development method for interfering programs. *ACM TOPLAS*, 5(4):596–619, 1983.

[6] J. Misra and K. M. Chandy. Proofs of networks of processes. *IEEE TSE*, SE-7(4):417–426, 1981.

[7] A. Benveniste, B. Caillaud, D. Nickovic, R. Passerone, J.-B. Raclet, P. Reinkemeier, A. Sangiovanni-Vincentelli, W. Damm, T. A. Henzinger, and K. G. Larsen. Contracts for system design. *Foundations and Trends in EDA*, 12(2–3):124–400, 2018.

[8] K. Honda, N. Yoshida, and M. Carbone. Multiparty asynchronous session types. In *POPL*, pages 273–284, 2008.