MACHINE PIDGIN

SPEAR/0.2 and Exact On-Task Adherence

A paired four-model synthetic pilot

RESEARCH NOTE 001
2 August 2026 / Open research note / Not peer reviewed

Abstract

Machine Pidgin studies how human purposes, constraints, and authority can survive interpretation by systems more capable than the person specifying a task. This pilot compares ordinary prose with SPEAR/0.2, a compact interpretation contract and structured task format, on 16 held-out synthetic tasks across four OpenRouter-hosted model tiers. Exact on-task adherence increased from 46/64 observations (71.9%) in prose to 57/64 (89.1%) with SPEAR, an absolute difference of 17.2 percentage points. GPT-5.6 Sol rose from 14/16 to 16/16; the largest lift was GPT-4o mini, from 5/16 to 11/16. Mean expected-field agreement rose from 80.1% to 92.1%. The structured condition cost more because it used more prompt tokens. These results support further study of explicit authority, precedence, canonical vocabulary, stop conditions, and output checking, but do not establish preservation of latent human intent or alignment in open-world settings.

1. Research question

How do you communicate constructively with an intelligence much more capable than you without losing your agenda? A familiar asymmetry makes the problem concrete: a three-year-old has fewer words, less context, and less power than an adult, but the child's purpose still matters. Advanced AI could widen that gap dramatically. Competent execution of the wrong interpretation may be more consequential than incompetent execution.

The narrow question in this note is testable: holding task facts constant, does a SPEAR interpretation prompt and field structure change how often models return the exact requested result and output contract?

2. Intervention

The control condition used an ordinary-language task plus a system instruction to return one JSON object. The treatment condition expressed the same facts with SPEAR fields and a system-level interpretation contract.

SPEAR/0.2 makes the following elements explicit:

- TASK and typed OBJECTS;
- AUTHORITY, including allowed actions, approval requirements, and prohibitions;
- preserved and ignored distinctions;
- OBJECTIVE and non-compensatory HARD constraints;
- rule, exception, source, and tie-break PRECEDENCE;
- canonical VOCABULARY for machine-readable labels;
- UNCERTAINTY and clarification value;
- OUTPUT shape, EVALUATION tests, STOP conditions, and a final CHECK.

The protocol is an intervention under study, not a standard and not a complete alignment method.

3. Methods

3.1 Tasks and split

The corpus contains eight development tasks and 16 held-out tasks. Tasks cover constrained selection, ordering, scheduling, integer allocation, unit conversion, access authority, record reconciliation, source precedence, nested exceptions, stop gates, value of information, task-sufficient abstraction, and exact failure reporting.

Each task has a prose form, a SPEAR form, and preregistered expected JSON. The final on-task outcome requires exact normalized equality. A recursive leaf score records partial agreement with required values. JSON validity is scored separately. No model grades another model.

3.2 Models and settings

Four operational tiers were sampled through OpenRouter:

1. OpenAI GPT-4o mini;
2. OpenAI GPT-5.6 Luna;
3. OpenAI GPT-5.6 Terra;
4. OpenAI GPT-5.6 Sol.

GPT-5.6 models used low reasoning effort. GPT-4o mini used provider defaults. The runner requested a fixed seed and JSON response format. There was one final response per task, condition, and model, producing 128 API calls and 64 paired observations.

3.3 Reproducibility

The public record contains task prompts, expected answers, runner, raw responses, model and provider metadata, token usage, reported cost, latency, errors, task-file hash, row-level CSV, and an audit of invalidated pilots. The OpenRouter key is loaded only from an environment variable and is not stored in the artifacts.

4. Results

Model	Ordinary prose	SPEAR/0.2	Difference
GPT-4o mini	5/16 (31.3%)	11/16 (68.8%)	+37.5 points
GPT-5.6 Luna	14/16 (87.5%)	15/16 (93.8%)	+6.3 points
GPT-5.6 Terra	13/16 (81.3%)	15/16 (93.8%)	+12.5 points
GPT-5.6 Sol	14/16 (87.5%)	16/16 (100.0%)	+12.5 points
All observations	**46/64 (71.9%)**	**57/64 (89.1%)**	**+17.2 points**

The relative increase in exact successes was 23.9% over the prose baseline. Mean expected-field agreement rose from 80.1% to 92.1%. Of the 15 model-task pairs with different binary outcomes, 13 favored SPEAR and two favored prose.

Provider-reported held-out cost was $0.0364 for prose and $0.0571 for SPEAR, $0.0935 total. The intervention therefore improved exact adherence in this sample while increasing prompt cost. All 128 responses were valid JSON and no API requests failed in the held-out run.

5. Development audit and protocol revision

The first development run made SPEAR/0.1 look worse than prose across all model tiers. The project retained that result and inspected it rather than selecting only a favorable run.

Audit found an impossible scheduling answer key, two incorrect held-out arithmetic keys caught before held-out calls, a canonical source label that differed across paired prompt conditions, and a Sol new-account rate limit. It also found genuine model failures involving exception precedence, authority synonyms, arithmetic feasibility, and exact output labels.

SPEAR/0.2 responded by adding AUTHORITY, PRECEDENCE & VOCABULARY, STOP, and CHECK. Benchmark defects were corrected before the held-out run. Invalidated development records remain public with reasons.

6. Interpretation

The strongest supported claim is narrow: in a small synthetic task suite emphasizing exact constraints and machine-readable outputs, SPEAR/0.2 was associated with more exact answers than ordinary prose containing the same task facts. The largest absolute benefit occurred in the weakest sampled model, but positive differences also appeared in each GPT-5.6 tier. Sol reached the ceiling under SPEAR.

The result does not show that structure becomes more valuable monotonically with model capability. It does not show that SPEAR recovers unexpressed values, prevents deception, resolves political disagreement, or grants legitimate authority. It shows that an inspectable contract can help models honor certain expressed distinctions.

7. Threats to validity

The protocol author also designed the tasks and scorer. The final sample has only 16 tasks and one observation per condition. Model families are related. Exact JSON scoring treats a semantically correct answer with the wrong key or canonical phrase as failure. This strictness is appropriate for interoperable software contracts but not a complete measure of usefulness. The SPEAR condition used more tokens and a stronger interpretive system instruction, so the study does not isolate typography, field names, instruction content, or attention allocation.

The tasks are synthetic and short. They do not include long-horizon tool use, adversarial users, hidden objectives, multi-party value conflict, dynamic environments, distribution shift, or real institutional consequences. Aggregate observations share tasks and model families and should not be treated as fully independent trials.

8. Next experiments

1. Independent replication with blinded task authors and preregistered analysis.
2. A factorial study separating structured fields, interpretation prompt, examples, and final checking.
3. Human authoring-time and repair-cost measurement.
4. Long-horizon tool-use tasks with explicit authority, reversible checkpoints, and unauthorized-action traps.
5. Multi-party specifications with conflicting values, minority reports, and appeal.

6. Adversarial paraphrases, missing fields, infeasible constraints, and specification gaming.
7. Cross-vendor model families and repeated stochastic samples.
8. Natural tasks contributed by researchers who did not design SPEAR.

9. Artifacts and disclosure

The task corpus, runner, result JSON, tidy CSV, protocol prompt, and audit log are released in the public Machine Pidgin repository. The founding theoretical preprint remains a separate document.

This note and the experiment code were prepared with AI assistance under human direction. Models produced the evaluated responses. Constantine Goltsev remains the named human project founder; independent authorship and peer review are invited.

References

1. Goltsev, C. *Task-Optimal Pidgin: Information-Theoretic Specification Across Human-AI Expressiveness Gaps*. Draft preprint, 2026.
2. Machine Pidgin Project. *SPEAR/0.2 Quick Reference*. 2026.
3. OpenRouter. *Chat Completions API* and *Models API* documentation. Accessed 2 August 2026.