CONSTRAINT-AWARE MULTI-AGENT TASK SCHEDULING FOR AI-ENABLED ONE-PERSON COMPANIES IN DIGITAL CONTENT SERVICES

Shanning Fu[1], Ya Han[2]

[1]National Research University "Tashkent Institute of Irrigation and Agricultural Mechanization Engineers"
Tashkent, Uzbekistan | Email: 1915105973@qq.com
ORCID: 0009-0007-3280-7991
[2]Shandong Hongjitang Pharmaceutical Group Co., Ltd.
Jinan, China | Email: 15853899107@163.com
ORCID: 0009-0007-0384-5745

KEYWORDS

AI-enabled one-person company, multi-agent system, constraint-aware scheduling, digital content services, small-scale experiment, human evaluation

ABSTRACT

AI-enabled one-person companies use large language models and software agents to organize content planning, generation, review, and delivery with limited human resources. However, additional roles and model calls may raise cost and propagate errors without producing a stable quality gain. This paper presents a constraint-aware multi-agent task-scheduling prototype for digital content services and compares four execution configurations under controlled conditions. Six frozen tasks were processed by G1 one-shot generation, G2 same-role staged generation, G3 a fixed multi-agent workflow, and G4, which combined constraint-aware role selection, deterministic validator feedback, and bounded final-node revision. Each configuration was repeated three times per task, producing 72 formal runs and 406 API calls at a total cost of USD 0.221318. G4 achieved the highest automatic pass rate (72.22%) and validation score (0.9689), but also the highest mean call volume and cost. After Holm correction, the pass-rate difference remained significant only between G3 and G4; G4 validation scores exceeded G2 and G3 but not G1. Two non-author evaluators independently rated all 72 anonymized outputs. Averaged-score reliability was ICC(2,2)=0.7288, while human mean scores did not differ significantly across groups. Automatic validation was not significantly associated with human ratings. Because G4 changed several mechanisms together, the experiment does not identify their independent effects. The findings support a limited improvement in compliance with encoded rules, not a comprehensive human-perceived quality advantage.

1. Introduction

Generative artificial intelligence has expanded large language models from single-turn text generators into components that can maintain role instructions, pass intermediate results, use tools, and revise outputs. ReAct, Reflexion, AutoGen, and MetaGPT illustrate reasoning-action loops, verbal feedback, configurable multi-agent conversation, and role-based standard workflows [1-5]. These approaches make it possible for one operator to coordinate several software roles, but they also introduce communication overhead, repeated context, and error propagation. A larger number of agents therefore does not automatically imply better output quality [6, 22, 23].

In several Chinese local entrepreneurship policies, the term one-person company (OPC) is used to describe a lightweight business model in which one core operator combines large models, software agents, digital platforms, and external services [7-10]. In this paper, an AI-enabled OPC is an operational concept rather than a nationally standardized legal form. The operator remains responsible for goals, fixed facts, budget, risk decisions, and final release, while the AI system performs standardized information-processing and content-production tasks.

Digital content services are suitable for a controlled experiment because their workflows can be decomposed into requirement analysis, fact organization, planning, writing, editing, quality review, risk review, and delivery. They include machine-checkable constraints such as

(cc) BY

length, required modules, dates, prices, fixed facts, and prohibited expressions, as well as human-judged properties such as factual reliability, language quality, audience fit, and direct usability [11-16, 24, 25]. The coexistence of these two quality layers makes the domain useful for testing both deterministic validation and human review.

The practical problem is not whether a model can generate text, but how a resource-constrained operator should organize generation. One-shot production is inexpensive but may omit constraints. A staged or multi-role workflow can make responsibilities clearer, yet every additional call increases cost and creates another opportunity for context loss or contradiction. A revision loop may repair visible rule violations, but repeated rewriting can also change facts or style. The appropriate configuration therefore depends on the task, the encoded constraints, and the operator's willingness to pay for additional checking.

The study does not rank general-purpose agent frameworks. Table 1 summarizes its narrower position: prior work mainly examines reasoning, feedback, dialogue, role-based workflows, or task-resource matching, whereas this small digital-content experiment integrates a frozen task graph, feasibility rules, cost logging, retained failures, and post-hoc two-reviewer evaluation. The elements are not individually new; the contribution is their controlled integration and comparison.

Table 1.

Positioning relative to related approaches.

Approach	Workflow structure	Reassignment or revision	Evaluation and cost evidence
ReAct [2]	Single-agent reasoning-action loop	No explicit reassignment	Cost not central; task-specific evaluation
Reflexion [3]	Single-agent feedback and memory	Iterative verbal feedback	Cost not central; task-specific evaluation
AutoGen [4]	Configurable multi-role conversation	Configurable, not core	Application-dependent cost and evaluation
MetaGPT [5]	Fixed role-based workflow	Limited in original SOP	Cost not central; application-dependent evaluation
Task-allocation studies [17-21]	Multi-executor capability-resource matching	Supported in several models	Resource objectives; not content-quality focused
Present prototype	Shared-model roles on a frozen DAG	G4 final-node reassignment and revision	API cost and two-reviewer evaluation

Note. The comparison refers to the original cited publications and does not cover all later extensions or application-specific implementations.

The objective is deliberately limited: to compare four execution configurations under the same six tasks, base model, generation parameters, and run environment. The study examines three questions. First, do additional roles, stages, and bounded revision improve compliance with encoded rules? Second, what API-call and monetary costs accompany these configurations? Third, are automatic differences supported by two non-author evaluators who assess broader content quality? The work does not propose a new foundation model or a global optimization algorithm, and its conclusions are restricted to the present small-scale experiment.

2. Theoretical Background

Large-language-model agents combine role instructions, task context, tools, and output control. Role specialization and feedback can organize complex work, but their value depends on task boundaries, information transfer, validation rules, and stopping conditions [1-6, 22, 23]. Because all roles share one frozen model,

(cc) BY

role count represents workflow structure rather than independent intelligence. Capability matching, temporal dependency, and resource constraints from task-allocation research provide the basis for feasibility filtering and rule-based scoring [17-21].

Three limited expectations guide the comparison: stage decomposition may reduce requirement omissions, repeated transfer may accumulate contradiction or context loss, and validator-linked bounded revision may improve explicit rule compliance without necessarily improving factual judgment, style, or direct usability. A DAG fixes dependencies; hard conditions determine feasibility, soft objectives rank feasible candidates, and task rules support final validation. Table 2 summarizes the main constraints.

Table 2.

Main constraints and implementation.

Constraint type	Examples	Implementation	Effect
Dependency and capability	Planning after requirements; delivery after review; writing, fact, and compliance skills	DAG readiness; capability threshold 0.60	Blocks premature execution and removes unsuitable roles
Risk and task rules	Sensitive-content clearance; modules, facts, dates, length, and prohibited terms	Role permissions; frozen validator	Excludes unauthorized roles and produces issue lists
Budget and revision	Technical retry and final-node revision ceilings	Frozen timeout and attempt counters	Prevents uncontrolled calling
Soft objectives	Capability, expected quality, expected latency, expected cost, and load	Weighted candidate score	Ranks feasible roles
Audit	Candidate scores, API usage, events, and outputs	SQLite, logs, and output folders	Supports post-run verification

Digital entrepreneurship research emphasizes the recombination of resources and organizational capabilities [11, 12, 24, 25], while digital content production must also address copyright, factual, privacy, and promotional risks [13-15]. Because deterministic checks and human judgment measure different quality layers [16], the experiment records rule compliance, cost, and two-reviewer evaluation.

3. Materials and Methodology

3.1 System architecture, task graph, and software roles

The Python/SQLite prototype targets text-based digital content services and uses a client compatible with the DeepSeek API. Its components cover experiment control, task and constraint management, scheduling, role execution, model service, automatic validation, and audit storage. It does not process real customer private data or replace human judgment in high-risk decisions.

(cc) BY

ISSN: 3030-3346

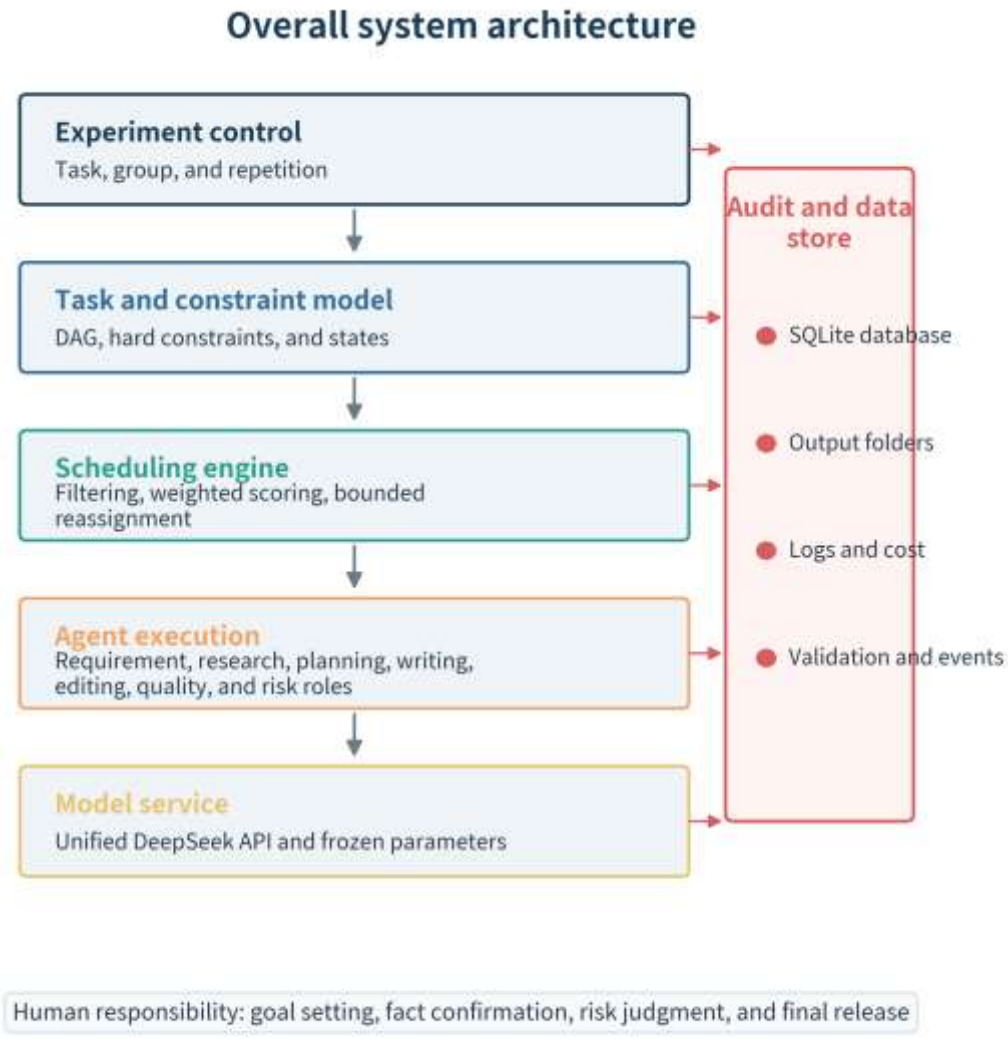


Figure 1. Overall architecture of the prototype system.

Experiment control reads the frozen task, group, and repetition, and the task layer releases DAG nodes whose prerequisites are complete. G1-G3 follow predefined workflows, whereas G4 applies feasibility filtering and role scoring. Each node records inputs, outputs, dependencies, capabilities, risk, state, and attempts; calls, cost, outputs, and events are stored in run-specific files and SQLite.

Software roles cover requirements, facts or research, planning, writing, editing, quality, risk, and final integration. All call the same frozen model and differ through prompts, capability values, permissions, and node responsibilities; the experiment therefore compares division of work and feedback structure rather than independent models.

3.2 Scheduling rules and experimental configurations

For G4, the scheduler first removes candidates that fail hard feasibility conditions: capability for the required node function below 0.60, risk clearance below node risk, workload at or above 1.0, explicit exclusion after failure on the same node, or exhausted retry allowance. For candidate role a and node n, the remaining candidates are scored by capability match C, expected quality Q, historical success H, time efficiency T, cost efficiency K, load suitability L, and risk fit R. The frozen rule score is shown in Equation (1).

$$S(a,n) = 0.26C + 0.22Q + 0.15H + 0.12T + 0.10K + 0.08L + 0.07R \quad (1)$$

(cc) BY

All terms are normalized to [0,1]. C is the candidate role's frozen capability value for the function required by node n; Q and H are frozen profiles, and H was not updated during formal runs. T and K are expected latency and cost profiles frozen before scheduling, not realized values from the current call; L is load suitability. R depends only on node risk level. Because role clearance is enforced first, R is constant across feasible roles for the same node and does not change their relative ranking. All profiles were frozen before the experiment; the full matrix is retained in the reproducibility materials. The score is a transparent engineering rule and does not guarantee a global optimum.

Six frozen Chinese digital content tasks were used: a product introduction page, a product-launch package, a social-media promotion, a seven-day multi-platform plan, a knowledge article, and an online-event promotion package. Task complexity and constraint density differed, but all tasks required Chinese textual deliverables. Each task was executed under four configurations and repeated three times, producing 6 x 4 x 3 = 72 runs.

Table 3a.

Frozen task and constraint profile.

Task	Deliverables / fixed facts	Other encoded constraints	Gate node
T01 Product introduction	5 / 5	4 boundary rules; no citation	No
T02 Product launch	7 / 5	5 boundary rules; no citation	Yes
T03 Social media	4 / 5	4 boundary rules; no citation	No
T04 Multi-platform plan	4 / 5	5 boundary rules; no citation	Yes
T05 Knowledge article	6 / 4	4 boundary rules; S1-S4 each ≥1 citation	Yes
T06 Event promotion	7 / 5	7 boundary rules; no citation	Yes

Note. Deliverables/fixed facts are frozen task-metadata counts; one entry may generate multiple issues. Gate nodes used AUTO_TEST approval in batch runs and did not involve real-time human judgment.

Table 3b.

Four experimental configurations.

Group	Execution workflow	Baseline calls	Validation and revision
G1	One-shot generation for the complete task	1	Final rule check; no model revision
G2	One writing role across analysis, planning, drafting, and self-check	4	Final rule check; no reassignment
G3	Fixed DAG with fixed specialist-role mapping	8	Quality/risk comments; one finalization
G4	Constraint filtering and weighted dynamic role assignment	8	Up to two final-node revisions after failure

(cc) BY



Figure 2. Four execution configurations and the balanced experiment.

G1 generated the complete deliverable in one call. G2 used the same writing role across four sequential stages. G3 used a fixed eight-call multi-role workflow. G4 began with an eight-call graph but could add up to two final-node revisions after a failed deterministic check. In G4, upstream nodes were not rerun during final revision. The failed output, issue list, and valid context were sent to a replacement final role for targeted correction.

All groups used the same frozen model, task facts, temperature, output limit, and run environment. Formal execution took place on 27 July 2026 from 03:43 to 09:05 UTC, and the controls are summarized in Table 4. All 406 calls returned deepseek-v4-flash with finish reason stop. The recorded system fingerprint was fp_8b330d02d0_prod0820_fp8_kvcache_20260402. Formal outputs were not deleted or manually converted from failure to success; both SUCCEEDED and COMPLETED_WITH_ISSUES runs were retained as valid completed observations.

Table 4.

Main experimental controls.

Item	Frozen setting
Tasks and repetitions	6 tasks × 4 groups × 3 repetitions = 72 runs
Model	deepseek-v4-flash; thinking off
Generation parameters	temperature=0.5; max_tokens=6000
Technical retry	180-s timeout; up to 2 interface retries
Automatic metrics	Pass, score, API calls, and cost
Human evaluation	Two non-author evaluators; independent ratings; 8 dimensions; anonymized outputs

3.3 Automatic validation, cost, and statistical analysis

Automatic validation checked only frozen deterministic rules, including structure, count, length, fixed facts, required modules, and prohibited expressions. If m deterministic issues were found, the validation score was calculated using Equation (2).

$$V = max(0,1 - 0.08m) \quad (2)$$

A score of 1 indicated no encoded issue. Validation score is a deterministic monotonic transformation of the raw issue count m and is not an independent quality measure. The deduction of 0.08 per issue was a frozen engineering parameter selected before the formal runs and was not statistically optimized. No run reached the zero floor, so ranking by validation score was exactly

(cc) BY

the reverse of ranking by raw issue count. API calls and cost were recorded from provider usage fields. Cost used the price snapshot dated 26 July 2026 and separated cached input, uncached input, and output tokens. Absolute monetary values should not be generalized to future pricing.

The 18 matched task-by-repetition blocks comprised three repeated runs for each of six tasks; they are not 18 independent task types. Automatic pass status was compared using Cochran's Q and pairwise exact McNemar tests. Validation score, call count, and cost were compared with Friedman tests, Kendall's W, and pairwise Wilcoxon signed-rank tests; rank-biserial correlations were retained for supported score comparisons. Holm correction controlled multiple comparisons. Because the task set and repetitions were small, statistical inference is exploratory. Call-count differences were largely imposed by the frozen group design and are interpreted primarily descriptively.

3.4 Blind human evaluation and reproducibility

After output freezing, two non-author evaluators independently rated 72 anonymized outputs with the same written rubric and without discussion. Neither evaluator participated in system development, prompt design, group assignment, or statistical analysis. R1 had backgrounds in digital technology and AI, economics, and business intelligence; R2 had training in Chinese materia medica, pharmaceutical QA/pharmacovigilance, and AI-content production. Eight dimensions used a five-point scale plus a four-level delivery judgment. Evaluators were blinded to group labels, automatic metrics, calls, and costs, although workflow structure might be inferred from text. The composite was the mean of sixteen ratings; agreement used weighted kappa and the two-way random-effects, absolute-agreement intraclass correlation coefficients ICC(2,1) and ICC(2,2).

The formal experiment retained 72 final outputs, 72 run logs, the frozen SQLite database, configurations, statistical workbooks, and file hashes. The recorded database preserves execution timestamps, model identifiers, the system fingerprint, token usage, and cost fields. The shareable code package excludes API keys, virtual environments, caches, and pre-test databases. Thirty-six human-gate records in the runtime database have reviewer code AUTO_TEST because batch execution used automatic approval; they prove that the gate node executed but are not evidence of real-time human review. The non-author R1 and R2 ratings were conducted only after output freezing.

4. Research Results

4.1 Execution integrity and group-level results

All 72 runs produced final outputs and logs. There were 24 SUCCEEDED results and 48 COMPLETED_WITH_ISSUES results, with no program-level execution error. The study recorded 406 API calls and a total cost of USD 0.221318. The absence of program errors means the workflow completed technically; it does not mean that all content met the deterministic rules or was ready for direct delivery.

G4 achieved the highest automatic pass rate and validation score but also the highest call volume and cost; G1 was cheapest, whereas G3 used eight calls per run but passed only 11.11%. Exact results appear in Tables 5a-5b and Figure 3.

Table 5a.

Quality and encoded-issue results of the four configurations.

Group (n=18)	Pass rate, % (95% Wilson CI)	Mean validation score	Mean issues	Median issues (IQR)
G1	27.78 (12.50–50.87)	0.8933	1.33	1.00 (0.25–2.00)
G2	22.22 (9.00–45.22)	0.8400	2.00	2.00 (1.00–3.00)
G3	11.11 (3.10–32.80)	0.8444	1.94	1.50 (1.00–2.00)
G4	72.22 (49.13–87.50)	0.9689	0.39	0.00 (0.00–0.75)

(cc) BY

Table 5b.

API-call and cost results of the four configurations.

Group	Mean calls	Mean cost (USD)
G1	1.0000	0.000313
G2	4.0000	0.001620
G3	8.0000	0.004798
G4	9.5556	0.005565

Note. Group mean costs are rounded to six decimals; the exact cumulative cost was USD 0.22131796.

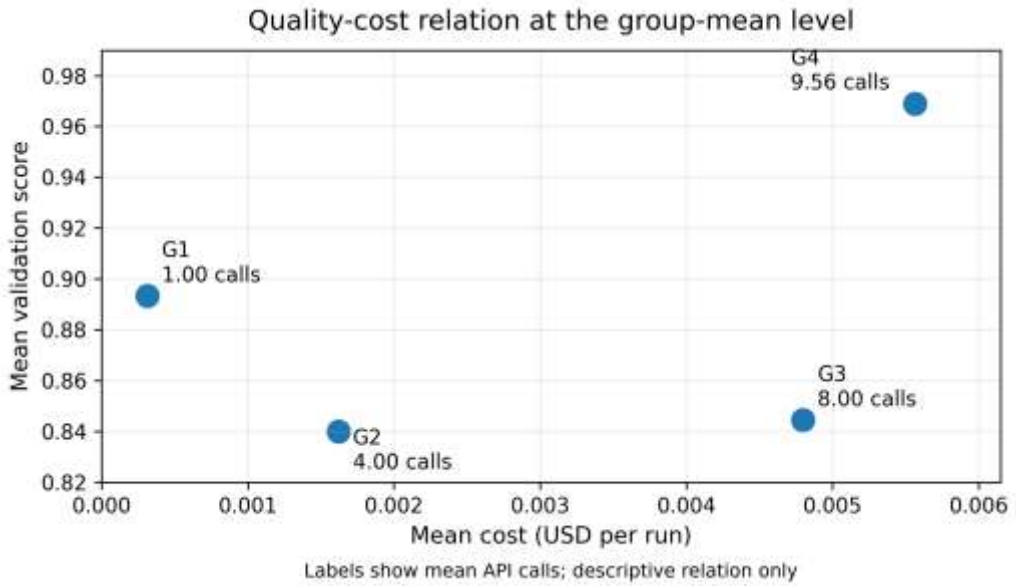


Figure 3. Quality-cost relation at the group-mean level.

4.2 Task-level heterogeneity

Results varied substantially by task. T02 product launch passed 8 of 12 runs with a mean score of 0.9600, whereas T05 knowledge article passed 1 of 12 with a mean score of 0.8267.

Table 6.

Automatic pass outcomes by task and configuration.

Task	G1	G2	G3	G4
T01 Product introduction	0/3	0/3	0/3	3/3
T02 Product launch	2/3	2/3	1/3	3/3
T03 Social media	2/3	0/3	1/3	2/3
T04 Multi-platform plan	0/3	2/3	0/3	2/3
T05 Knowledge article	0/3	0/3	0/3	1/3
T06 Event promotion	1/3	0/3	0/3	2/3

(cc) BY

Configuration rankings also changed by task: G2 exceeded G3 on T04, and G1 and G2 exceeded G3 on T02. Fixed multi-role staging was therefore not a uniform improvement.

4.3 Statistical comparison and bounded revision

Overall tests were significant for pass rate, validation score, call count, and cost. After Holm correction, the pass-rate difference remained significant only between G3 and G4. For validation score, G4 significantly exceeded G2 and G3, but not G1. Table 7 reports Kendall's W and supported score-comparison effect sizes. Because baseline calls were preset by configuration, call-count and cost tests mainly confirm design-imposed separation and are not interpreted as newly discovered behavioral effects.

Table 7.

Statistical comparison of automatic indicators.

Metric	Overall test	p value	Supported pairwise result after Holm correction
Pass rate	Cochran's Q=16.800; exact McNemar pairs	<0.001	G4 > G3 only (adjusted p=0.006)
Validation score	Friedman χ^2=16.257; W=0.301	0.001	G4 > G2 (|r_rb|=0.857); G4 > G3 (|r_rb|=1.000); G4 vs G1 not significant
API calls	Friedman χ^2=53.461; W=0.990	<0.001	All six pairs significant
Run cost	Friedman χ^2=51.000; W=0.944	<0.001	All six pairs significant

Among 18 G4 runs, 16 triggered final-node revision; 11 ultimately passed and 5 exhausted the allowed attempts. All reassignments occurred at the final node, so the evidence supports only bounded final-delivery revision.

At the group-mean level, G1 and G4 formed the quality-cost Pareto frontier. G1 offered the lowest cost and a higher validation score than G2 and G3. G4 offered the highest validation score at the highest cost. This frontier is descriptive and does not imply that either configuration is optimal for every task or every quality criterion.

4.4 Blind human evaluation

Both reviewers completed all 72 ratings. Reliability was ICC(2,1)=0.5733 for a single reviewer and ICC(2,2)=0.7288 for averaged ratings; neither group human scores nor automatic-human relations were significant (Tables 8a-8b).

Table 8a.

Descriptive human composite scores by configuration.

Group	Mean	SD	Median	IQR
G1	3.6493	0.4851	3.5625	3.2500–3.9844
G2	3.5451	0.5099	3.5313	3.1875–3.6875
G3	3.7847	0.4649	3.6563	3.5625–4.0938
G4	3.7049	0.4095	3.6250	3.5156–3.7813

(cc) BY

Table 8b.

Blind human evaluation and automatic-human relations.

Item	Estimate	p value / interval / agreement	Interpretation
Averaged-rating reliability	ICC(2,2)=0.7288	95% CI 0.5782–0.8139	Reliability estimate for the averaged ratings
Group human scores	Friedman χ^2=2.809; W=0.052	0.422	No significant overall difference
Auto score–human score	Spearman ρ=-0.0924	0.440	No significant association
Auto pass–human score	Point-biserial r=0.0688	0.566	No significant association
Delivery-grade agreement	Quadratic-weighted κ=0.6408	50/72 exact agreements	Relatively high categorical agreement

Dimension means showed higher completeness (4.6806), logic and structure (4.5139), and audience fit (4.3403), but lower factual accuracy (2.5417), direct usability (2.6181), and risk/compliance (2.8333). Figure 4 presents overall dimensions rather than group advantages.

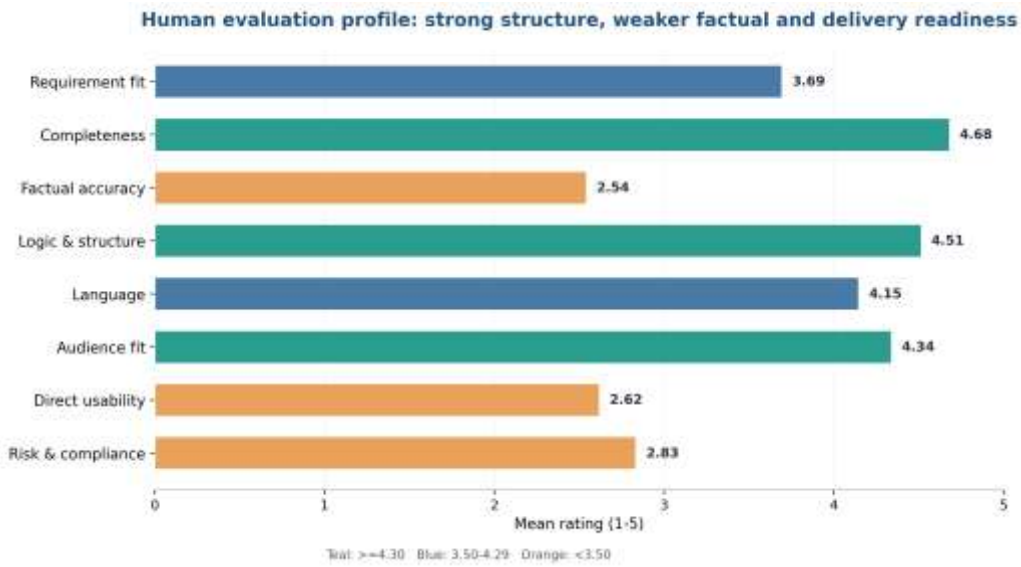


Figure 4. Mean human ratings across eight evaluation dimensions.

R1 and R2 classified only four and three outputs, respectively, as directly deliverable, showing that high rule compliance did not equal delivery readiness.

5. Discussion

5.1 Interpreting G3 and G4

G4 combined role selection, validator feedback, and bounded revision and achieved higher encoded-rule compliance, but support was selective: pass rate differed only from G3, validation score differed from G2 and G3, and the G1 difference was not significant. The design cannot isolate each mechanism; G3 also shows that more roles and calls are not themselves measures of quality.

Final-node issue lists helped 11 of 16 triggered G4 runs pass, while 5 exhausted the attempts. The configuration can repair some visible rule violations but cannot guarantee factual reliability, appropriate expression, or direct usability.

(cc) BY

5.2 Task heterogeneity and evaluation boundaries

Configuration effects depended on task structure. Tasks with fixed modules could benefit from deterministic feedback, whereas the knowledge article's citation, structure, and length constraints remained difficult. Automatic validation checks explicit rules, while human evaluation also covers facts, language, audience, risk, and delivery readiness; their nonsignificant association is therefore not contradictory.

5.3 Practical implications, limitations, and future research

Simple, low-risk tasks can start with G1 plus a human checklist; constraint-dense work may justify a validator-linked G4-type workflow, but high-risk content still requires human verification.

Limitations include six tasks, three repetitions, one model, and two non-author evaluators. Costs depend on a dated price snapshot, automatic rules cover only encoded constraints, AUTO_TEST was not real-time human judgment, and G4 tested final-node revision only. Results do not generalize to other models, languages, industries, or high-risk services; future work can test broader tasks, models, and factorial designs.

6. Conclusion

This paper developed a constraint-aware multi-agent scheduling prototype and evaluated four execution configurations in a small-scale digital content experiment. The combined G4 configuration—role selection, validator feedback, and bounded final-node revision—obtained the highest rule-based pass rate and validation score, but required the most calls and cost. The design does not isolate the independent effect of any one G4 component. Statistical support was partial rather than universal, and the two-reviewer human evaluation found no significant group difference. Automatic validation and human ratings were not significantly associated, showing that encoded constraints and broader content quality require separate assessment. One-shot generation can remain economical for simple tasks, whereas validator-linked bounded revision may improve rule compliance for more constrained work. Human verification remains necessary for facts, risk, and direct usability. Conclusions are limited to the six frozen tasks, one model configuration, and the present evaluation design.

Data and Code Availability

Cleaned code, the frozen role capability and permission matrix, frozen tasks/configurations, de-identified statistics, and reproducibility documentation are available from the first author upon reasonable request. API keys, blind-review mappings, and restricted materials are excluded; frozen records retain model identifiers, fingerprint, usage, timestamps, and hashes.

Declarations

Author Contributions. Shanning Fu: conceptualization, methodology, software, data, analysis, visualization, and original draft. Ya Han: validation and review/editing. Both authors approved the manuscript.

Acknowledgments. The authors thank the two non-author evaluators, R1 and R2, for independently rating the anonymized outputs.

Funding. This research received no external funding.

Conflict of Interest. The authors declare no conflict of interest.

Ethics Statement. No customer data, clinical intervention, or sensitive personal data were used; the two evaluators voluntarily assessed de-identified AI-generated outputs under a written protocol.

References

1. Wooldridge M., Jennings N.R. Intelligent agents: theory and practice // The Knowledge Engineering Review. 1995. Vol. 10, No. 2. P. 115–152.
2. Yao S., Zhao J., Yu D., et al. ReAct: Synergizing Reasoning and Acting in Language Models // International Conference on Learning Representations (ICLR). 2023. [Electronic resource]. URL:

(cc) BY

https://openreview.net/forum?id=WE_vluYUL-X (accessed: 01.08.2026).
3. Shinn N., Cassano F., Berman E., et al. Reflexion: Language Agents with Verbal Reinforcement Learning // Advances in Neural Information Processing Systems. 2023. Vol. 36. [Electronic resource]. URL: https://papers.nips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html (accessed: 01.08.2026).
4. Wu Q., Bansal G., Zhang J., et al. AutoGen: Enabling Next-Generation LLM Applications via Multi-Agent Conversation // Proceedings of the First Conference on Language Modeling (COLM). 2024. [Electronic resource]. URL: https://openreview.net/forum?id=BAakY1hNKS (accessed: 01.08.2026).
5. Hong S., Zhuge M., Chen J., et al. MetaGPT: Meta Programming for a Multi-Agent Collaborative Framework // International Conference on Learning Representations (ICLR). 2024. [Electronic resource]. URL: https://openreview.net/forum?id=VtmBAGCN7o (accessed: 01.08.2026).
6. Guo X., Zhang M., Ma J. A review of agent construction based on large language models // Communication Technology. 2024. Vol. 57, No. 9. P. 873–879. (In Chinese).
7. Suzhou Industrial Park Administrative Committee. Measures for Accelerating the Development of a Globally Leading AI-OPC Entrepreneurship Zone. Su Yuan Guan [2025] No. 40. 28 November 2025. (In Chinese). [Electronic resource]. URL: https://www.sipac.gov.cn/kjhxxhj/kjzc/202511/100fff909fb44586882b0a90d04c5f0b.shtml (accessed: 01.08.2026).
8. Shenzhen Municipal Bureau of Industry and Information Technology. Action Plan for Building a Leading AI-OPC Entrepreneurship Ecosystem in Shenzhen, 2026-2027. Shen Gong Xin [2026] No. 2. 14 January 2026. (In Chinese). [Electronic resource]. URL: https://gxj.sz.gov.cn/gkmlpt/content/12/12602/post_12602272.html (accessed: 01.08.2026).
9. People's Government of Yangpu District, Shanghai. Measures for Supporting the Development of an OPC Super-Individual Community. Yang Fu Fa [2026] No. 4. 13 May 2026. (In Chinese). [Electronic resource]. URL: https://www.shanghai.gov.cn/tszczq-cxcyqj8/20260520/4317e33bbb6c4faaae22b59626a9436c.html (accessed: 01.08.2026).
10. Hangzhou Municipal Administration for Market Regulation. Measures Supporting One-Person Company (OPC) Innovation and Entrepreneurship. Hang Shi Jian [2026] No. 43. 15 April 2026. (In Chinese). [Electronic resource]. URL: https://zfgb.hangzhou.gov.cn/upload/default/bigfile/2026/05/13/20260513_4c5a80bed00443268bcc250c13286413.pdf (accessed: 01.08.2026).
11. Feng S., Fu L., Song L. Growth mechanism of digital content entrepreneurship from the perspective of entrepreneurial capability: a longitudinal case study of FD Reading // China Business and Market. 2022. Vol. 36, No. 10. P. 80–91. (In Chinese).
12. Yan C., Liu J. Digital entrepreneurship research: conceptual clarification, integrated framework, and future prospects // Journal of North University of China (Social Science Edition). 2024. Vol. 40, No. 2. P. 19–30. (In Chinese).
13. Chen C. Copyrightability of artificial-intelligence-generated content: taking ChatGPT-generated content as an example // Science-Technology & Publication. 2023. No. 8. P. 95–103. (In Chinese).
14. Wang P. Compliance dilemmas and adjustment of knowledge production under AIGC publishing // Editorial Friend. 2024. No. 3. P. 14–22. (In Chinese).
15. Wang Y., Cao Y. Technical logic and system construction of an intelligent book-topic planning system empowered by AIGC // Publishing Research. 2024. No. 2. P. 31–37. (In Chinese).
16. Wan P., Gu X. Human–AI collaborative evaluation supported by generative AI: practice models and explanatory cases // Modern Distance Education. 2024. No. 2. P. 33–41. (In Chinese).
17. Smith R.G. The contract net protocol: high-level communication and control in a distributed problem solver // IEEE

(cc) BY *dtai.uz*

Transactions on Computers. 1980. Vol. C-29, No. 12. P. 1104–1113.
18. Gerkey B.P., Mataric M.J. A formal analysis and taxonomy of task allocation in multi-robot systems // The International Journal of Robotics Research. 2004. Vol. 23, No. 9. P. 939–954.
19. Choi H.L., Brunet L., How J.P. Consensus-based decentralized auctions for robust task allocation // IEEE Transactions on Robotics. 2009. Vol. 25, No. 4. P. 912–926.
20. Yan J., Li X., Liu B. Multi-agent cooperative task allocation considering temporal constraints // Control and Decision. 2015. Vol. 30, No. 11. P. 1999–2003. (In Chinese).
21. He L., Li B. A multi-objective cloud workflow scheduling algorithm satisfying Pareto optimality // Computer Applications and Software. 2019. Vol. 36, No. 5. P. 289–297. (In Chinese).
22. Li Y., Xu F., Xie G., et al. Review of the development and applications of multi-agent technology // Computer Engineering and Applications. 2018. Vol. 54, No. 9. P. 13–21. (In Chinese).
23. Liu J., Li N., Liang X., et al. Chinese academic question answering and applications based on multi-agent systems // Journal of Kunming University of Science and Technology (Natural Sciences). 2025. Vol. 50, No. 3. P. 58–69. (In Chinese).
24. Zhang K., Jiang M. Current status and research trends of digital entrepreneurship at home and abroad: a bibliometric analysis // Journal of Xi'an University of Technology. 2023. Vol. 39, No. 2. P. 227–240. (In Chinese).
25. Zhang J., Tu Y., Jin X. Digital business model research: review and prospects // Science & Technology Progress and Policy. 2022. Vol. 39, No. 13. P. 151–160. (In Chinese).

(cc) BY