VIBE PATENTING: AGENTIC AI FOR PROFESSIONAL PATENT DRAFTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Professional patent drafting requires more than long-form legal text generation: an agent must extract inventive concepts from heterogeneous technical evidence, plan a protection strategy, coordinate claims, the specification, embodiments, and figures, and repeatedly check support and consistency. We present **Vibe Patenting**, an agentic AI framework for this end-to-end professional workflow. The system combines a structured Problem–Insight–Solution–Effect (PISE) invention representation with specialized patent skills, realized either as reusable modules orchestrated by a meta-agent or as an integrated domain-specialized patent agent. Across more than one hundred generated drafts from ten technical reports, stronger models and greater inference-time reasoning generally receive higher LLM-judge scores, and agentic configurations outperform general-purpose chat in our system-level comparisons. We also place a separate patent-QA LLM judge inside a multi-round revision loop: structured judge feedback continues to improve judge-assessed quality after generic revision begins to saturate. Independent evaluation by a professional patent attorney confirms a substantial aggregate quality gain after QA-guided revision while revealing strong metric dependence in judge agreement and calibration. A second, independently instructed QA evaluator further shows that revision gains can transfer beyond the optimizing judge even as evaluator-specific preferences emerge. These results position professional patent drafting as a testbed for agentic AI built from structured domain representations, reusable workflows, inference-time reasoning, and external evaluation.

1 INTRODUCTION

Large language models are rapidly evolving from one-shot text generators into *agents* that plan, invoke tools, maintain intermediate state, critique outputs, and execute multi-stage workflows Yao et al. (2023b); Wu et al. (2023); Hong et al. (2024). This shift matters especially for professional knowledge work, where success requires factual and linguistic competence together with procedural structure, domain-specific abstractions, artifact coordination, and quality control. We ask *how a capable general-purpose model should be organized into an effective professional drafting agent*.

We study this question through **Vibe Patenting**, which transforms scientific and technical materials into a coordinated patent-draft package with minimal manual drafting. Patent drafting is a demanding agentic task: a system must identify protectable concepts rather than summarize a paper; reason about breadth and fallback positions; draft supported claims and embodiments; generate consistent figures and specification text; and preserve terminology and technical fidelity. Recent work has begun to automate individual patent components and, increasingly, complete applications Jiang et al. (2025d); Knappich et al. (2025); Shea & Yu (2025); Wang et al. (2024b); Jiang et al. (2026). Our complementary systems question is how professional patent knowledge should be represented, modularized, orchestrated, and evaluated within an agent.

Patent drafting is especially useful as a professional-agent testbed because local fluency is insufficient. A plausible claim may lack written-description support, a specification may describe mechanisms that never reach the claims, and a figure may be visually coherent yet fail to support the protection strategy. Revisions create additional long-range dependencies: changing claim scope can require corresponding changes to embodiments, terminology, figures, and support analysis. Quality is therefore a property

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

of a *coupled artifact package*, not any single passage, motivating explicit invention state, artifact interfaces, and external checks rather than a longer one-shot prompt.

Our framework introduces an explicit intermediate invention state, *Problem–Insight–Solution–Effect (PISE)*, that organizes technical evidence by problem, inventive recognition, technical mechanism, and effect. Specialized skills reuse this state for invention and prior-art analysis, protection strategy, embodiment expansion, claims, specification, figures, support checking, and revision. We study two realizations: a *meta-agent* that composes reusable patent skills from professional workflow knowledge, and a *custom patent agent* that integrates workflow logic, reference materials, and skills into a dedicated domain agent. We compare both with general-purpose chat.

Professional agents also need credible evaluation. We therefore use a separately invoked patent-QA judge that scores complete drafts along five professional dimensions and returns structured critique. This judge serves both as a scalable experimental evaluator and as an optimization signal for multi-round revision. Because optimizing an automated evaluator can exploit its idiosyncrasies, we compare repeated judge scores with independent scores from a patent attorney rather than equating judge improvement with professional improvement.

Our experiments examine three complementary axes of professional-agent capability: *model capability*, *inference-time reasoning effort*, and *agentic scaffolding*. Across more than one hundred drafts from ten technical reports, stronger models and greater reasoning generally receive higher judge scores, and domain-specific agent structure also scores favorably. Because the end-to-end configurations differ in several implementation choices, we treat them as system-level comparisons rather than component ablations. Judge-guided revision continues after generic self-revision tends to saturate and can partly trade revision depth for single-pass reasoning strength. Aggregate attorney ratings independently confirm the round-4 gain, while dimension-level agreement remains heterogeneous.

Our contributions are:

- We formulate professional patent drafting as an end-to-end agentic task that transforms heterogeneous technical evidence into mutually constrained claims, specification, figures, and supporting analyses.
- We develop a structured agent design centered on PISE and specialized patent skills, and compare general-purpose chat, modular meta-agentic orchestration, and an integrated custom patent agent.
- We study how model capability, inference-time reasoning, and professional agentic scaffolding interact in patent-drafting quality and computation.
- We evaluate a separate patent-QA judge as a multi-round revision signal and validate it against professional patent attorney scores, showing that revision gains transfer in aggregate while detailed evaluator preferences can diverge.

2 Related Work

Patent drafting and legal document generation. Patent NLP has moved from isolated language tasks toward long-form and agentic drafting. Claim-centric work studies LLM claim generation and revision Jiang et al. (2025d;c), while Patent-CE and PatentScore develop expert-aligned multidimensional evaluation Jiang et al. (2025b); Yoo et al. (2025). PatentFormer and PatentVision generate specifications from claims and drawings Wang et al. (2024a); Yang et al. (2026), and PAP2PAT benchmarks paper-to-patent generation Knappich et al. (2025). Most closely related to our setting, AutoSpec decomposes specification drafting into agentic subtasks Shea & Yu (2025), AutoPatent uses planner, writer, and examiner agents for full-length generation Wang et al. (2024b), and Dis2Pat introduces disclosure-to-patent generation with a multi-agent baseline Jiang et al. (2026). We instead study end-to-end professional agent design around explicit invention state, reusable skills, coordinated patent artifacts, and independently validated QA-guided revision.

Professional and legal agents. General frameworks such as ReAct, AutoGen, and MetaGPT emphasize tool use, multi-agent coordination, and explicit workflow structure Yao et al. (2023b); Wu et al. (2023); Hong et al. (2024). Legal work increasingly follows this pattern: LAW orchestrates domain tools and agents for contract analysis Watson et al. (2025), LegalAgentBench evaluates

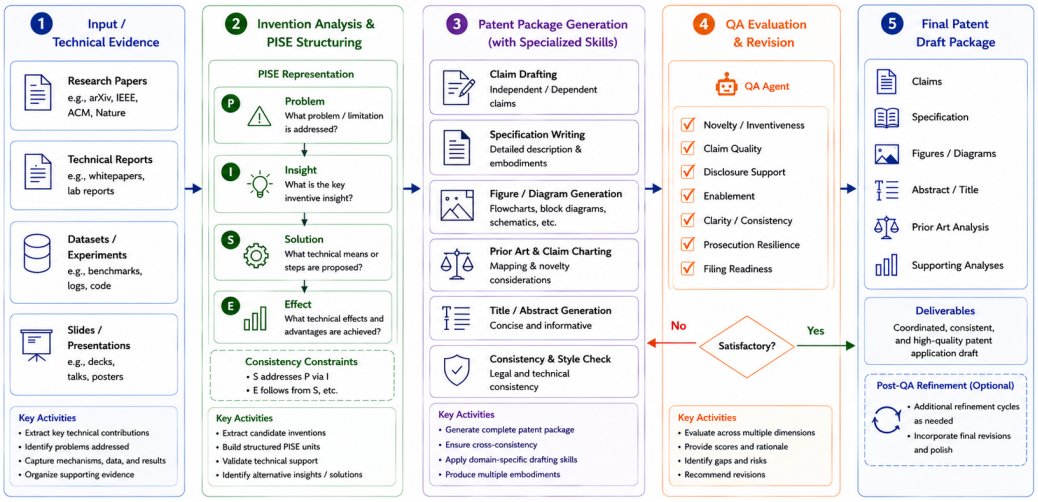


Figure 1: Vibe Patenting framework. The agent structures technical evidence; patent-specific skills generate coordinated claims, specification, figures, and analyses; and a QA judge evaluates the package and can trigger revision.

tool-using legal agents Li et al. (2025b), and surveys identify planning, memory, tools, and procedural execution as core challenges Liu et al. (2026). Structured legal-document generation also benefits from planning before section-level drafting Nigam et al. (2026), motivating our view of patent drafting as a professional workflow rather than a single prompt.

LLM judges and iterative refinement. LLM-as-a-judge methods enable scalable rubric-based evaluation Zheng et al. (2023); Liu et al. (2023), but reliability, bias, and calibration remain concerns Zeng et al. (2024); Tan et al. (2024); Li et al. (2025a). Self-Refine and Reflexion use language feedback for iterative improvement Madaan et al. (2023); Shinn et al. (2023). We combine these ideas by using a patent-specific judge as evaluator and revision signal, then independently compare its scores with professional patent attorney ratings. Appendix A gives a broader survey.

3 Vibe Patenting: Building a Professional Patent-Drafting Agent

Figure 1 summarizes the end-to-end system. Given technical source material $\mathcal{X}$, the objective is a coordinated package rather than a single piece of prose: $\mathcal{D} = \{\mathcal{C}, \mathcal{S}, \mathcal{F}, \mathcal{A}\}$, where $\mathcal{C}$ denotes claims, $\mathcal{S}$ the specification, $\mathcal{F}$ patent figures, and $\mathcal{A}$ supporting analyses. These artifacts are mutually constrained: claims need disclosure support, figures should depict disclosed structures or processes, embodiments should provide fallback positions, and terminology should remain consistent. We therefore draft through shared-state professional operations rather than independent generations.

We treat the intermediate products as *artifact contracts*. PISE and the protection strategy form upstream state; claims consume that state to define protection targets; the specification and figures make those targets technically explicit; and support analysis and QA consume the complete package. The workflow has a forward drafting order but also feedback dependencies: a later claim change creates new consistency obligations for disclosure, figures, terminology, and fallback support. This operationalizes statefulness: the agent must preserve and reconcile inspectable artifacts across stages rather than merely retain chat history.

3.1 Structured Invention State and Professional Skills

The first design principle is to separate *invention understanding* from patent prose. The agent maps source evidence into a PISE representation $\mathcal{Z}_{\text{PISE}} = (P, I, S, E)$, where P is the technical problem or limitation, I the key inventive insight, S the mechanism realizing it, and E the resulting technical

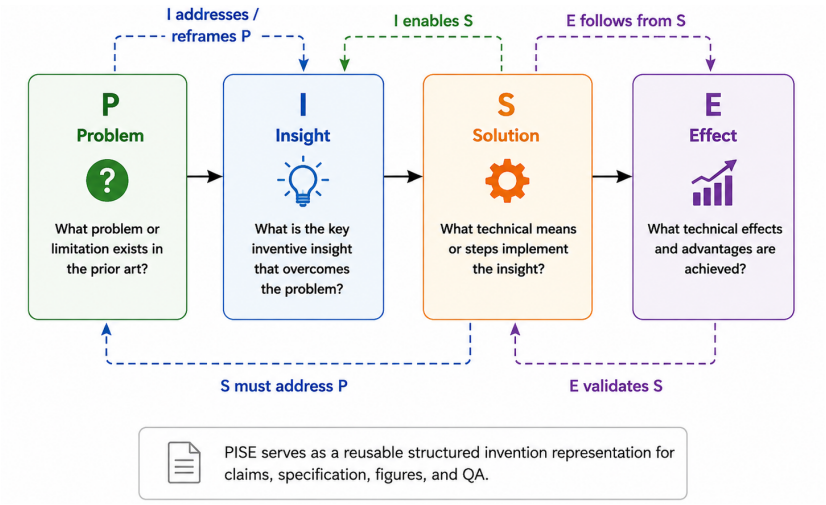


Figure 2: PISE invention state: problem, insight, solution, and technical effect share one state reused by strategy, claims, specification, figures, and QA.

effect. PISE is relational rather than four independent summaries: S should address P through I, and claimed effects should be attributable to S (Figure 2).

A source may support multiple inventive concepts, so we allow a hierarchy $\{\mathcal{Z}_{\text{PISE}}^{(k)}\}_{k=1}^{K}$ rather than forcing an entire work into one tuple. The agent can form a broad PISE, decompose it into mechanism- or embodiment-level units, and explore extensions by asking what problem is addressed, what consequence follows from a solution, and what related implementation can be derived. This supplies downstream claims and embodiments with multiple abstraction levels while preserving a common invention-centered state; Appendix C gives a schematic PISE tree.

PISE is deliberately not a claim template. It preserves invention semantics separately from the legal expression chosen for a particular draft. A broad parent unit can inform an independent-claim concept, while narrower mechanism, embodiment, or hardware-specific units supply dependent limitations and fallback disclosure. The same unit can feed specification and figure skills differently. This separation lets later stages revise claim wording or presentation while reusing a stable invention-centered representation rather than rediscovering the technical contribution.

The second design principle is to expose professional drafting operations as reusable skills with explicit inputs and outputs. A typical trajectory is

$$\mathcal{X} \to \mathcal{Z}_{\text{PISE}} \to \mathcal{R} \to \{\mathcal{C}, \mathcal{S}, \mathcal{F}, \mathcal{A}\} \to \mathcal{Q}, \tag{1}$$

where $\mathcal{R}$ denotes protection-strategy and embodiment decisions and $\mathcal{Q}$ a structured QA report. The implementation expands this trajectory into technical and prior-art-oriented analysis, invention discovery, protection strategy, embodiment expansion, claim/specification drafting, figure generation, support checking, QA, and revision. Later skills reuse the same PISE state and earlier artifacts rather than restarting from the raw source; Appendix B gives implementation details.

Stateful orchestration and artifact contracts. Table 1 is useful only if the stages remain coupled. Both agentic implementations persist and pass intermediate artifacts forward: strategy depends on PISE; claims consume strategy and embodiments; specification expands the same concepts into enabling disclosure; figures follow the claims and specification; and support checking evaluates the complete package. The result is a stateful professional process whose inspectable interfaces let a human or another agent intervene without regenerating the package.

3.2 Three Agent Architectures

Figure 3 compares three allocations of professional structure. **Generalist chat** receives the technical source and a direct drafting request, leaving decomposition and coordination implicit. The **meta-**

Table 1: Representative skill stages and artifact interfaces. Later stages reuse shared PISE state and prior artifacts rather than restarting from the source.

Skill stage	Primary operation	State/artifact produced
Technical analysis	Extract mechanisms, evidence, assumptions, and candidate contributions	Structured technical findings
Invention discovery + PISE	Reorganize evidence around problem, insight, solution, and effect	One or more PISE units
Prior-art-oriented analysis	Identify likely novelty/obviousness pressure points and differentiators	Risk and distinction analysis
Protection strategy	Select protection hierarchy, independent-claim concepts, and fallback directions	Claim/protection plan
Embodiment expansion	Generate alternative structures, procedures, deployments, and substitutions	Embodiment/fallback set
Coordinated drafting	Draft claims and enabling specification from the same strategy and state	Claims + specification
Figure generation	Depict important structures, processes, and relationships consistently with the text	Patent figures
Support + QA + revision	Check terminology, support, figure alignment, and filing readiness; revise weaknesses	Support analysis + QA report + revised draft

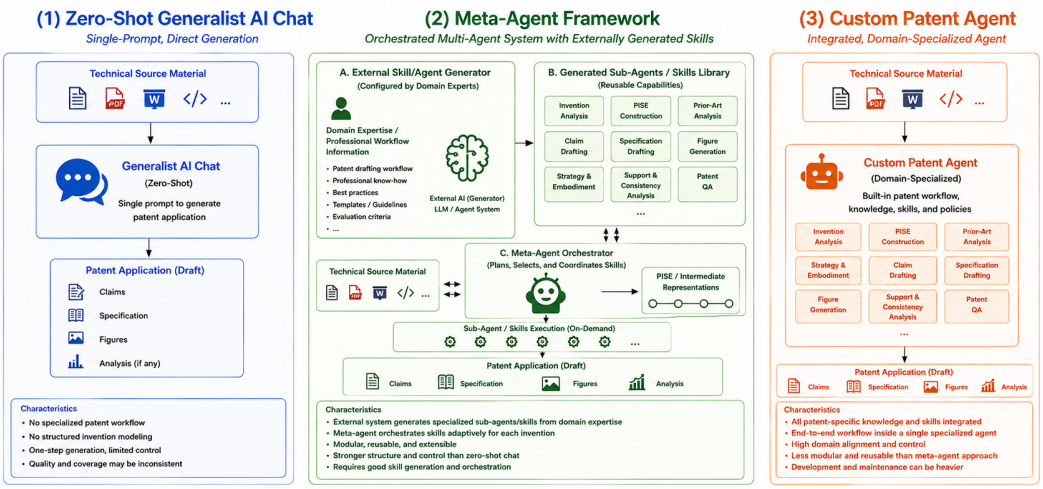


Figure 3: Patent-drafting architectures: generalist chat is implicit; the meta-agent composes reusable skills and artifacts; the custom agent integrates them in a domain configuration.

agentic framework externalizes patent knowledge into reusable skills selected and sequenced by a general orchestrator that maintains intermediate artifacts. The **custom patent agent** integrates workflow logic, professional instructions, reference materials, and specialized skills into a dedicated configuration. Thus professional structure is largely implicit in chat, externalized for the meta-agent, and more deeply embedded in the custom-agent environment (Appendix B).

3.3 External Patent-QA Judge and Revision Loop

A separate LLM-based patent-QA judge scores each completed draft on five dimensions: *Prosecution Resilience*, *Claim Strength*, *Disclosure Strength*, *Figure Quality*, and *Filing Readiness*. Resilience covers prosecution risks such as eligibility, novelty, and obviousness; Claim Strength covers scope, fallback positions, enforceability, and design-around resistance; Disclosure Strength covers written description, enablement, alternatives, and consistency; Figure Quality covers substantive visual

coverage and alignment; and Filing Readiness captures residual defects in the complete package. The judge uses a rubric-anchored 1–10 scale and returns scores plus structured critique. Each draft is evaluated independently five times; reported dimensions are repeat means. Their mean is a descriptive overall composite, while the five dimensions remain primary outcomes. Appendix D gives the rubric.

The same judge can be inserted into an iterative loop. At revision round t,

$$Q_t = f_{\text{judge}}(D_t), \qquad D_{t+1} = f_{\text{rev}}(\mathcal{X}, D_t, Q_t). \tag{2}$$

We compare against a generic revision control $D'_{t+1} = f_{\text{rev}}(\mathcal{X}, D_t)$ using the same drafting configuration without the QA report, isolating structured evaluator feedback from repeated rewriting. Appendix E gives loop details and prompts.

Finally, we distinguish two human roles. A *skilled human drafter* (with experience as a co-inventor on hundreds of patent filings) provides contextual generation reference points, not a controlled human–AI contest. A separate *professional patent attorney evaluator* independently scores drafts for judge validation.

4 Experiments and Results

Experimental setup. We evaluate more than one hundred patent drafts from ten scientific reports spanning multiple technical domains. Drafting configurations include GPT-5.6 Sol chat at several reasoning levels, GPT-5.4–5.6 agents, a domain-specialized custom patent agent, and a skilled-human reference. Each completed draft is independently evaluated five times by the LLM-based QA judge; repeats estimate evaluator variability rather than independent inventions. The judge assigns integer 1–10 scores on five dimensions whose mean is reported as a descriptive overall composite.

Evaluation hierarchy and unit of analysis. Repeated QA executions are averaged *within* a draft and never treated as independent patent observations. Configuration plots average available draft means, while attorney correlations use matched draft–condition means. Professional validation covers nine patents: the QA1–attorney analysis uses 28 matched AI-draft conditions; QA2 rescored six complete configurations (54 patent/configuration observations); and the main three-way attorney comparison uses 27 matched `agent`/`pro`/`instant` observations. This hierarchy separates evaluator repeatability, patent variation, and configuration-level comparison.

4.1 Scaling Professional Patent-Drafting Agents

Figure 4 (Left) compares end-to-end GPT-5.6 Sol drafting configurations. Agentic configurations score above general-purpose chat, and the domain-specialized custom agent has the highest observed mean. Because configurations also differ in instructions, state handling, tools, and workflow integration, this is a system-level comparison rather than a causal ablation. The skilled-drafter bar is a contextual generation reference, distinct from the evaluating patent attorney.

Figure 4 (Right) compares overall judge score with observed end-to-end generation time. Quality generally increases with observed generation time (Pearson $r = 0.63$), yet similarly timed configurations can score quite differently, suggesting that workflow organization matters in addition to inference budget. The relationship is also dimension dependent: configuration-level correlations with observed generation time are approximately 0.79 for Figure Quality and 0.68 for Disclosure Strength, but only 0.02 for Claim Strength (Appendix H.3). Thus inference time is not a uniform proxy for professional quality.

Figure 5 gives the dimension-wise comparison. For GPT-5.6 Sol chat, overall score rises from 5.53 (Instant) to 5.97 (Medium), 6.36 (High), and 6.58 (Extra-High), showing a clear reasoning-effort trend despite nonmonotonic axes. The GPT-5.6 agent reaches 6.15 overall and 6.80 at Extra-High. The custom agent reaches 7.02 on two patents with strong Disclosure and Figure scores; with two unbalanced cases, it is illustrative rather than population-level evidence.

4.2 Judge-Guided Iterative Revision

We test the LLM judge as an optimization signal using a 2×2 design: Instant versus Extra-High reasoning, each with structured QA feedback or generic revision. As shown in Fig. 6(a), generic

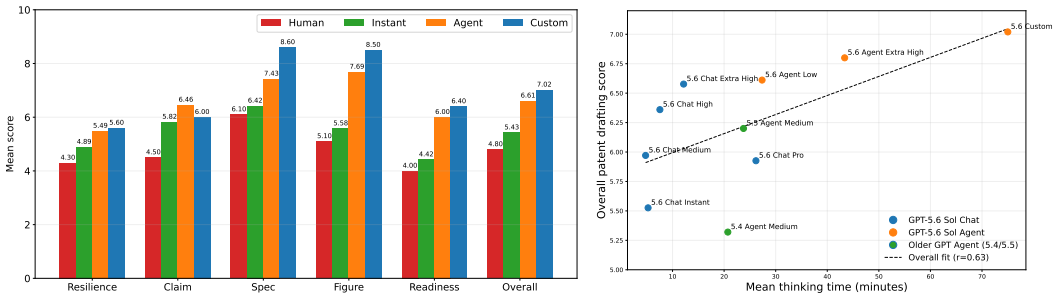


Figure 4: **Left:** Judge-assessed quality by drafting configuration. **Right:** Overall judge score versus mean end-to-end generation time.

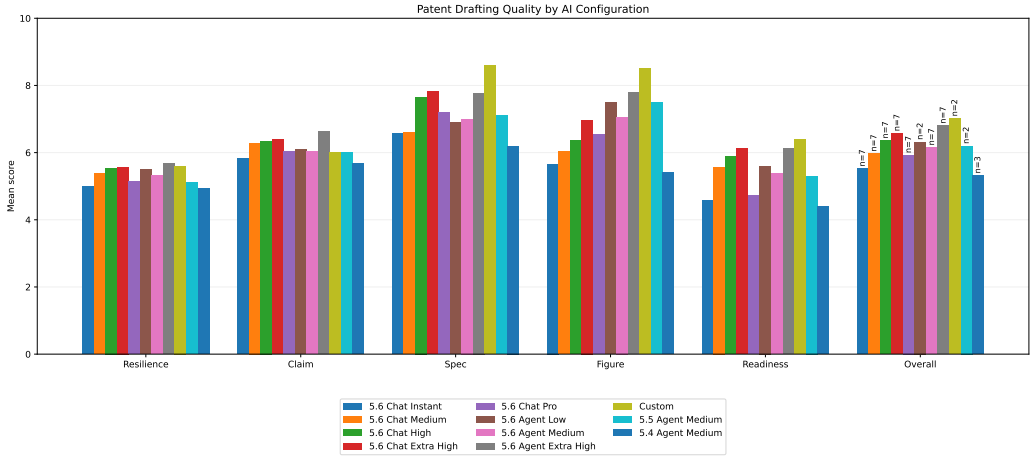


Figure 5: Five-axis and overall judge scores across model, reasoning, and agent configurations. Bars average repeated QA runs and available drafts; coverage varies.

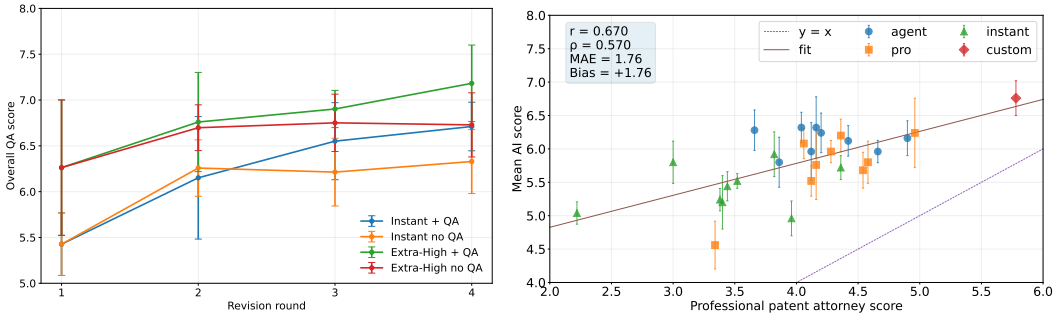
revision improves sharply at first but tends to saturate, whereas judge-guided revision continues through round 4: Instant rises from 5.43 to 6.71 with QA guidance versus 6.33 under generic revision, and Extra-High rises from 6.26 to 7.18 with QA versus 6.73 under generic revision.

Iterative feedback also closes the initial reasoning gap: Instant begins 0.84 points below Extra-High but reaches 6.71 with QA by round 4, nearly matching Extra-High generic revision (6.73). Under this judge, structured feedback can trade revision depth for single-pass inference strength, although judge-score gains need not equal professional-quality gains. Appendix H.5 shows uneven five-axis gains, so the overall mean is a summary rather than the sole stopping rule; per-axis scores and structured critique should guide the next revision.

4.3 Agreement with Professional Evaluation

We next compare the LLM judge with independent professional judgment. A patent attorney scored matched drafts on the same five dimensions; each score is matched to the mean of five independent LLM-QA evaluations. Figure 6(b) and Table 2 show a moderate overall association (Pearson $r = 0.670$, $p < 0.001$; Spearman $\rho = 0.570$, $p = 0.002$) across 28 matched AI-draft conditions from nine patents. Agreement is strongly metric dependent: Figure Quality is strongest ($r = 0.819$), followed by Disclosure Strength ($r = 0.533$), while several dimensions remain weak.

Correlation and calibration capture different judge properties: Claim Strength has essentially no relative association ($r = -0.077$) but small absolute error (MAE 0.35) and little bias (-0.04), whereas Figure and Disclosure better track relative expert judgments yet are systematically higher than attorney scores. Thus an LLM judge can support ranking or optimization without numerical



(a) Judge-guided iterative revision **(b)** LLM judge vs. patent attorney

Figure 6: LLM judge as evaluator and optimization signal. **(a)** Four-round overall scores for Instant and Extra-High, with and without structured QA. **(b)** QA1 versus attorney scores for 28 matched AI-draft conditions from nine patents.

Table 2: Agreement between repeated QA1 scores and professional patent attorney scores across 28 matched AI-draft conditions from nine patents. Bias is QA1 minus attorney score. Given the modest sample and repeated conditions within inventions, p-values are descriptive.

Metric	Pearson r (p-value)	Spearman ρ (p-value)	MAE	Bias
Prosecution Resilience	0.054 (0.786)	0.089 (0.651)	1.40	+1.40
Claim Strength	-0.077 (0.695)	-0.045 (0.822)	0.35	-0.04
Disclosure Strength	0.533 (0.004)	0.481 (0.010)	3.33	+3.33
Figure Quality	**0.819** (< 0.001)	**0.767** (< 0.001)	2.58	+2.58
Filing Readiness	0.305 (0.115)	0.270 (0.165)	1.53	+1.53
Overall	**0.670** (< 0.001)	0.570 (0.002)	1.76	+1.76

calibration to professional ratings. AgentEx and round-4 Instant+QA (Instant×4) remain outside the 28-point correlation because only aggregate attorney means are available; Section 4.4 combines those aggregates with QA2 for independent transfer and evaluator-specific analyses.

4.4 Three-Way Agreement with an Independent QA2 Evaluator

We rescore the drafts with a separately configured **QA2** evaluator. QA1 is the original Instant-reasoning optimization judge; QA2 uses separate instructions, Extra-High reasoning, and the attorney's five-axis rubric and scoring principles. Because both rubric/instructions and inference effort differ, this is an evaluator-system comparison rather than a causal ablation. Across 27 matched `agent`/`pro`/`instant` conditions, QA1 and QA2 agree strongly ($r = 0.797$, $\rho = 0.771$), while their pooled attorney correlations are similar (QA1: $r = 0.598$, $\rho = 0.520$; QA2: $r = 0.603$, $\rho = 0.558$). Figure 7 shows that this pooling masks substantial condition- and axis-level structure; for TuneComp and LatentLLM, `pro` denotes the Extra-High attorney-scored baseline.

Agreement is strongly heterogeneous. By condition, both judges track the attorney for `pro` (QA1 $r = 0.796$; QA2 $r = 0.703$) and each other ($r = 0.861$); for `instant`, QA2–attorney reaches $r = 0.868$ versus 0.332 for QA1, with QA1–QA2 $r = 0.620$; and for `agent`, neither judge tracks the attorney despite QA1–QA2 $r = 0.628$. By axis, QA2 improves attorney agreement for Prosecution Resilience ($r = 0.628$ versus 0.038), Disclosure Strength, and Filing Readiness, whereas QA1 remains stronger for Figure Quality (0.784 versus 0.673). QA1–QA2 agreement peaks for Figure Quality ($r = 0.862$) and Disclosure Strength (0.770) but is weak for Resilience and Claim Strength, showing how pooled trends can mask professional-preference differences.

Across all 54 complete patent/configuration observations for `agent`, `pro`, `instant`, Instant×4, AgentEx, and Extra-High×4, QA1 and QA2 correlate at Pearson $r = 0.760$ and Spearman $\rho = 0.740$; every configuration also has positive five-axis profile correlation ($r = 0.683$–0.827). Their disagreement is therefore structured: the judges share a broad quality notion but weight dimensions and draft conditions differently. Revision further provides a transfer test: Instant→Instant×4 improves

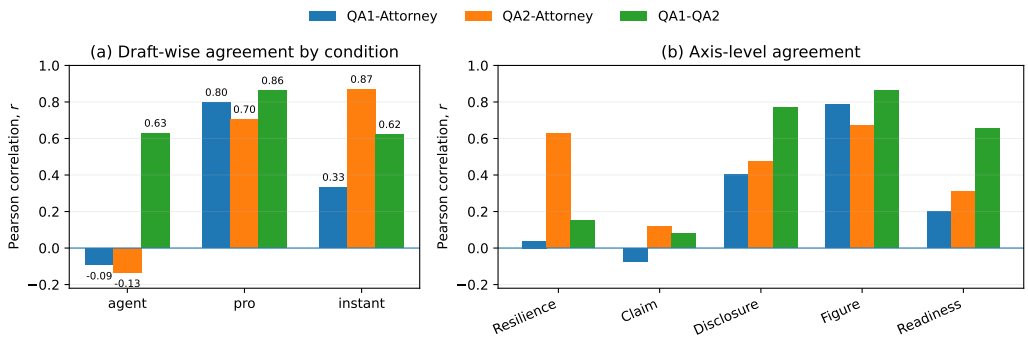


Figure 7: Pearson agreement by **(a)** draft condition and **(b)** quality axis for QA1, attorney-aligned QA2, and the attorney.

by $+1.28$ (QA1), $+0.58$ (QA2; 8/9 patents), and $+1.58$ (attorney), yet AgentEx exceeds Instant$\times$4 by only $+0.06/+0.65/+0.68$ under QA1/QA2/attorney. This is consistent with transferable improvement plus evaluator-specific optimization, where repeated QA1 revision narrows its perceived gap to a separately generated AgentEx configuration. Appendix H.8 gives the full three-way and aggregate analyses.

5 Limitations and Responsible Use

This is an initial study, not a claim of autonomous professional equivalence. The invention set is modest, coverage is uneven, and the custom-agent result uses two patents. Chat, meta-agent, and custom-agent systems differ in instructions, state handling, tools, and workflow integration, so they do not isolate PISE or any individual skill. Professional validation uses one attorney, while AgentEx/Instant$\times$4 have only aggregate attorney means. QA1 and QA2 differ in rubric/instructions and reasoning effort, so their comparison is not causal. The judges show dimension-dependent bias and are evaluation signals rather than ground truth; frontier products may also evolve.

Patent drafting also involves inventorship, legal obligations, patentability and claim-strategy judgments, filing approval, and prosecution. We therefore envision *researchers validate technical content $\rightarrow$ AI structures, drafts, and checks $\rightarrow$ professionals review and approve*. Confidential pre-filing material requires controls for retention, access, logging, and external tools (Appendix I).

The highest-value follow-ups are causal rather than merely larger: hold the base model, reasoning effort, source reports, and outputs fixed while ablating PISE or individual skills; separately vary evaluator rubric/instructions and reasoning effort; add multiple attorneys to measure human–human and human–AI agreement; and preserve invention-level clustering so repeated conditions are not treated as independent inventions.

6 Conclusion

We presented *Vibe Patenting* as a testbed for professional agentic AI: technical evidence becomes a shared PISE state, specialized skills coordinate strategy and patent artifacts, and external QA drives revision. System-level comparisons suggest complementary benefits from stronger models, greater inference-time reasoning, and explicit workflow structure; controlled ablations remain necessary for component-level causality. QA-guided revision scores higher under QA1, independently configured QA2, and the patent attorney, confirming transfer beyond the optimizing judge; yet QA1 nearly ties AgentEx with revised Instant$\times$4, while QA2 and the attorney retain a much larger AgentEx advantage. Automated judges can thus provide useful feedback while retaining evaluator-specific preferences, motivating independent professional validation. More broadly, professional-agent quality depends jointly on *representation*, *orchestration*, *inference allocation*, and *evaluation design*—what state is explicit, how dependent artifacts are reconciled, where reasoning is spent, and which signals are optimized and independently checked.

AI USE STATEMENT

OpenAI ChatGPT assisted with writing, editing, formatting, and basic coding. Figures 1, 2, 3, 8, and 10 are AI-generated conceptual illustrations; LLMs and VLMs are also evaluated. The authors reviewed all AI-assisted content and take responsibility for the final submission.

REFERENCES

Daniil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624:570–578, 2023. doi: 10.1038/s41586-023-06792-0. URL https://doi.org/10.1038/s41586-023-06792-0.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6:525–535, 2024. doi: 10.1038/s42256-024-00832-8. URL https://doi.org/10.1038/s42256-024-00832-8.

Alireza Ghafarollahi and Markus J. Buehler. SciAgents: Automating scientific discovery through bioinspired multi-agent intelligent graph reasoning. *Advanced Materials*, 37(22):e2413523, 2025. doi: 10.1002/adma.202413523. URL https://doi.org/10.1002/adma.202413523.

Ali E. Ghareeb, Benjamin Chang, Ludovico Mitchener, Angela Yiu, Caralyn J. Szostkiewicz, Dmytro Shved, Gavin J. Gyimesi, Jon M. Laurent, Samantha M. Wright, Muhammed T. Razzak, Andrew D. White, Silvia C. Finnemann, Michaela M. Hinks, et al. A multi-agent system for automating scientific discovery. *Nature*, 655:497–505, 2026. doi: 10.1038/s41586-026-10652-y. URL https://doi.org/10.1038/s41586-026-10652-y.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Petar Sirkovic, Artiom Myaskovsky, Grzegorz Glowaty, Felix Weissenberger, Alessio Orlandi, et al. Accelerating scientific discovery with Co-Scientist. *Nature*, 655:487–496, 2026. doi: 10.1038/s41586-026-10644-y. URL https://doi.org/10.1038/s41586-026-10644-y.

Mourad Gridach, Jay Nanavati, Khaldoun Zine El Abidine, Lenon Mendes, and Christina Mack. Agentic AI for scientific discovery: A survey of progress, challenges, and future directions. *arXiv preprint arXiv:2503.08979*, 2025. URL https://arxiv.org/abs/2503.08979.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. A survey on LLM-as-a-judge. *The Innovation*, 7(6):101253, 2026. doi: 10.1016/j.xinn.2025.101253. URL https://doi.org/10.1016/j.xinn.2025.101253.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Juergen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations*, 2024.

Zheng Jia, Shengbin Yue, Wei Chen, Siyuan Wang, Yidong Liu, Zejun Li, Yun Song, and Zhongyu Wei. Ready Jurist One: Benchmarking language agents for legal intelligence in dynamic environments. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10351–10376, 2026. doi: 10.18653/v1/2026.acl-long.471.

Lekang Jiang and Stephan M. Goetz. Natural language processing in the patent domain: a survey. *Artificial Intelligence Review*, 58:214, 2025. doi: 10.1007/s10462-025-11168-z.

Lekang Jiang, Chengzu Li, and Stefan Goetz. Enriching patent claim generation with European Patent Dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 7734–7751, 2025a. doi: 10.18653/v1/2025.findings-emnlp.408.

Lekang Jiang, Pascal A. Scherz, and Stefan Goetz. Towards better evaluation for generated patent claims. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025b. URL https://aclanthology.org/2025.acl-long.190/.

Lekang Jiang, Pascal A. Scherz, and Stefan Goetz. Patent-CR: A dataset for patent claim revision. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2025c. URL https://aclanthology.org/2025.naacl-long.116/.

Lekang Jiang, Caiqi Zhang, Pascal A. Scherz, and Stefan Goetz. Can large language models generate high-quality patent claims? In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1272–1287, 2025d. doi: 10.18653/v1/2025.findings-naacl.70.

Lekang Jiang, Wenjun Sun, and Stephan Goetz. Benchmarking patent drafting from inventor-style disclosures. *arXiv preprint arXiv:2608.21249*, 2026. URL https://arxiv.org/abs/2608.21249.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing fine-grained evaluation capability in language models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=8euJaTveKw.

Valentin Knappich, Anna Hätty, Simon Razniewski, and Annemarie Friedrich. PAP2PAT: Benchmarking outline-guided long-text patent generation with patent-paper pairs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 9524–9554, 2025. doi: 10.18653/v1/2025.findings-acl.496.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. From generation to judgment: Opportunities and challenges of LLM-as-a-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2757–2791. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.emnlp-main.138. URL https://aclanthology.org/2025.emnlp-main.138/.

Haitao Li, Junjie Chen, Jingli Yang, Qingyao Ai, Wei Jia, Youfeng Liu, Kai Lin, Yueyue Wu, Guozhi Yuan, Yiran Hu, Wuyue Wang, Yiqun Liu, and Minlie Huang. LegalAgentBench: Evaluating LLM agents in legal domain. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2322–2344, 2025b. doi: 10.18653/v1/2025.acl-long.116.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. Generative judge for evaluating alignment. In *International Conference on Learning Representations*, 2024. URL https://gair-nlp.github.io/auto-j/.

Shuang Liu, Ruijia Zhang, Ruoyun Ma, Yujia Deng, Lanyi Zhu, Jiayu Li, Zelong Li, Zhibin Shen, and Mengnan Du. LLM agents in law: Taxonomy, applications, and challenges. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15768–15792, 2026. doi: 10.18653/v1/2026.acl-long.718.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2511–2522. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.153. URL https://aclanthology.org/2023.emnlp-main.153/.

Chris Lu, Cong Lu, Robert Tjarko Lange, Yutaro Yamada, Shengran Hu, Jakob Foerster, David Ha, and Jeff Clune. Towards end-to-end automation of AI research. *Nature*, 651(8107): 914–919, 2026. doi: 10.1038/s41586-026-10265-5. URL https://doi.org/10.1038/s41586-026-10265-5.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-2019.

Shubham Kumar Nigam, Deepak Patnaik Balaramamahanthi, Noel Shallum, Kripabandhu Ghosh, and Arnab Bhattacharya. Structured legal document generation in India: A model-agnostic wrapper approach with VidhikDastaavej. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference*, pp. 6661–6673, 2026. doi: 10.63317/2wswvwyqxt4k.

Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Michael Moor, Zicheng Liu, and Emad Barsoum. Agent Laboratory: Using LLM agents as research assistants. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 5977–6043, 2025. doi: 10.18653/v1/2025.findings-emnlp.320. URL https://aclanthology.org/2025.findings-emnlp.320/.

Ryan Shea and Zhou Yu. AutoSpec: An agentic framework for automatically drafting patent specification. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 12829–12840, 2025. doi: 10.18653/v1/2025.findings-emnlp.687.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0377.

Shreya Shukla, Nakul Sharma, Manish Gupta, and Anand Mishra. PatentLMM: Large multimodal model for generating descriptions for patent figures. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20488–20496, 2025. doi: 10.1609/aaai.v39i19.34257.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Y. Tang, Alejandro Cuadron, Chenguang Wang, Raluca Ada Popa, and Ion Stoica. JudgeBench: A benchmark for evaluating LLM-based judges. *arXiv preprint arXiv:2410.12784*, 2024. URL https://arxiv.org/abs/2410.12784.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. Replacing judges with juries: Evaluating LLM generations with a panel of diverse models. *arXiv preprint arXiv:2404.18796*, 2024. URL https://arxiv.org/abs/2404.18796.

Juanyan Wang, Sai Krishna Reddy Mudhiganti, and Manali Sharma. Patentformer: A novel method to automate the generation of patent applications. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, 2024a. URL https://aclanthology.org/2024.emnlp-industry.101/.

Qiyao Wang, Shiwen Ni, Huaren Liu, Shule Lu, Guhong Chen, Xi Feng, Chi Wei, Qiang Qu, Hamid Alinejad-Rokny, Yuan Lin, and Min Yang. AutoPatent: A multi-agent framework for automatic patent generation. *arXiv preprint arXiv:2412.09796*, 2024b. URL https://arxiv.org/abs/2412.09796.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

Koki Wataoka, Tsubasa Takahashi, and Ryokan Ri. Self-preference bias in LLM-as-a-judge. In *NeurIPS 2024 Workshop on Safe Generative AI*, 2024. URL https://arxiv.org/abs/2410.21819.

William Watson, Nicole Cho, Nishan Srishankar, Zhen Zeng, Lucas Cecchi, Daniel Scott, Suchetha Siddagangappa, Rachneet Kaur, Tucker Balch, and Manuela Veloso. LAW: Legal agentic workflows for custody and fund services contracts. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pp. 583–594, 2025. URL https://aclanthology.org/2025.coling-industry.50/.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

Ruo Yang, Sai Krishna Reddy Mudhiganti, and Manali Sharma. PatentVision: A multimodal method for drafting patent applications. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track)*, 2026. doi: 10.18653/v1/2026.eacl-industry.29.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*, 2023a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023b. URL https://arxiv.org/abs/2210.03629.

Yongmin Yoo, Qiongkai Xu, and Longbing Cao. PatentScore: Multi-dimensional evaluation of LLM-generated patent claims. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025. doi: 10.18653/v1/2025.emnlp-main.1564.

Yongmin Yoo, Qiongkai Xu, and Longbing Cao. PatentMind: A multi-aspect reasoning graph for patent similarity evaluation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 14947–14964, 2026. doi: 10.18653/v1/2026.findings-acl.735.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. Evaluating large language models at evaluating instruction following. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.07641.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2306.05685.

A Related Work

A.1 Patent NLP, Generation, and Revision

Patent processing has long included classification, retrieval, summarization, and generation tasks; a recent survey by Jiang and Goetz provides a broad taxonomy of text-based and multimodal patent NLP and emphasizes the distinctive technical and legal structure of patent documents Jiang & Goetz (2025). Within generation, early LLM work focused on individual components rather than complete applications. Jiang et al. Jiang et al. (2025d) study claim generation from patent descriptions and report that modern general-purpose LLMs can produce strong first independent claims but degrade on dependent claims and still require professional revision. The European Patent Dataset expands claim-generation research across jurisdictions and granted European patents Jiang et al. (2025a). Patent-CR formulates claim revision using rejected application claims paired with granted claims, showing that legality-aware revision remains difficult and that automated and professional evaluations can disagree Jiang et al. (2025c).

A second line of work addresses the long specification. PatentFormer generates patent specifications from claims, drawing text, and drawing descriptions Wang et al. (2024a). PatentLMM focuses on multimodal generation of descriptions for patent figures using a patent-specific vision-language architecture and the PatentDesc-355K dataset Shukla et al. (2025). PatentVision further combines claims and patent drawings to generate complete specifications with multimodal models Yang et al. (2026). These studies demonstrate that figures and document structure carry information not captured by text-only generation.

Research has increasingly moved toward end-to-end patent drafting. PAP2PAT constructs 1.8k patent–paper pairs and studies outline-guided long-form generation from research papers, finding that LLMs can use technical source material but still struggle with the level of detail required by patent descriptions Knappich et al. (2025). AutoSpec decomposes specification drafting from claims into manageable subtasks executed by open-source models augmented with custom tools, and evaluates the system with patent attorney input Shea & Yu (2025). AutoPatent introduces the Draft2Patent benchmark and a multi-agent framework with planner, specialized writer, and examiner roles for

generating full-length patents from technical drafts Wang et al. (2024b). Most recently, Dis2Pat targets inventor-style, de-legalized disclosures and associated figures; its Patent-MAF baseline uses a manager, specialized drafters, and a polisher to produce a complete application Jiang et al. (2026). These works are especially close to our setting and make clear that patent drafting is naturally an orchestration problem, not merely a long-text generation problem.

Our work differs primarily in the object of study and system organization. We treat the full professional drafting process as a testbed for *agentic AI*: technical evidence is first reorganized into an explicit Problem–Insight–Solution–Effect (PISE) invention state, then reused by modular patent skills for strategy, embodiments, claims, specification, figures, consistency analysis, and QA. We compare zero-shot chat, a meta-agent that dynamically composes reusable skills, and a domain-specialized custom agent, and we study how this agentic structure interacts with model capability, reasoning effort, and multi-round QA-guided revision. Rather than claiming a first patent-drafting agent, the contribution is a systematic study of how professional workflow structure and external evaluation can be assembled and measured end to end.

A.2 Patent Evaluation and LLM-as-a-Judge

Evaluating patent text is difficult because lexical overlap does not capture legal structure, technical support, or drafting quality. Patent-CE introduces expert-annotated comparisons of generated claims and PatClaimEval, a multidimensional evaluation method aligned with patent-expert judgments Jiang et al. (2025b). PatentScore similarly decomposes claim quality across structural, semantic, and legal dimensions and reports strong agreement with expert annotations Yoo et al. (2025). PatentMind extends expert-aligned multidimensional reasoning to patent-similarity assessment by separately modeling technical features, application domains, and claim scope Yoo et al. (2026). These studies motivate domain-specific decomposition instead of generic text metrics; our evaluation extends the scope from claims or pairwise similarity to a complete draft package containing claims, specification, figures, and supporting analysis.

More broadly, LLM-as-a-judge has become a common approach for evaluating open-ended generation. MT-Bench and Chatbot Arena showed that strong LLM judges can correlate with human preferences while exhibiting position, verbosity, and self-enhancement biases Zheng et al. (2023). G-Eval demonstrated benefits from rubric-based prompting and explicit evaluation steps Liu et al. (2023). Evaluator-focused models such as Prometheus and Auto-J aim to improve rubric-conditioned scoring and natural-language critique Kim et al. (2024); Li et al. (2024). Subsequent work has concentrated on judge reliability: LLMBar and JudgeBench expose failures on adversarial instruction following and objectively verifiable reasoning tasks Zeng et al. (2024); Tan et al. (2024); diverse evaluator panels can reduce some single-model biases Verga et al. (2024); and self-preference remains a documented concern Wataoka et al. (2024). Recent surveys synthesize these issues around calibration, consistency, bias, and deployment Li et al. (2025a); Gu et al. (2026).

Our judge is therefore used with two safeguards in the experimental design. First, each draft is evaluated repeatedly and scores are aggregated across five dimensions. Second, the judge is separately compared with evaluation by a professional patent attorney. This allows us to distinguish optimization against the judge from evidence that the judge tracks professional judgment; the latter is empirically metric dependent in our study.

A.3 Professional and Legal Agentic Workflows

General-purpose LLM agents increasingly combine reasoning, actions, tools, memory, and workflow decomposition. ReAct interleaves reasoning with external actions Yao et al. (2023b); AutoGen supports conversations among customizable agents Wu et al. (2023); and MetaGPT explicitly encodes human standard operating procedures into multi-agent collaboration Hong et al. (2024). These ideas are particularly relevant to professional work, where successful execution depends not only on domain knowledge but also on ordering, intermediate artifacts, quality checks, and procedural constraints.

Legal AI has begun to adopt this agentic perspective. LAW uses modular, domain-specific tools and text agents for complex custody and fund-services contract analysis and reports large gains over an off-the-shelf LLM baseline on some tasks Watson et al. (2025). LegalAgentBench evaluates legal agents using real-world corpora and external tools, emphasizing both task success and progress

through intermediate steps Li et al. (2025b). Ready Jurist One evaluates agents in dynamic legal environments and separately measures task outcomes and procedural compliance Jia et al. (2026). A 2026 survey of LLM agents in law organizes the field around planning, memory, tools, legal-domain applications, evaluation, and reliability challenges Liu et al. (2026). Complementary work on structured legal document generation proposes a model-agnostic wrapper that first plans document sections and then generates each section with retrieval-based prompts, with both LLM-based and expert evaluation Nigam et al. (2026).

Vibe Patenting adopts a similar principle of explicit professional workflow, but the generated object is a mutually constrained patent package rather than a query answer or single legal document. Claims, specification, figures, embodiments, and analyses share terminology and support requirements, so we maintain an intermediate invention representation and pass structured artifacts between specialized stages.

A.4 Agentic AI for Scientific and Technical Knowledge Work

Patent drafting begins from technical evidence, so scientific-agent systems provide another relevant precedent. ChemCrow augments an LLM with chemistry tools for planning and execution Bran et al. (2024), while Coscientist combines literature search, code, and laboratory automation Boiko et al. (2023). SciAgents uses multi-agent reasoning and knowledge graphs for materials hypothesis generation Ghafarollahi & Buehler (2025). Agent Laboratory automates literature review, experimentation, and report writing Schmidgall et al. (2025); The AI Scientist spans idea generation, experimentation, analysis, manuscript writing, and automated review Lu et al. (2026); and newer systems such as Co-Scientist and Robin use specialized agents, reflection, ranking, and experimental feedback in iterative scientific workflows Gottweis et al. (2026); Ghareeb et al. (2026). A recent survey summarizes this broader trend toward agentic scientific discovery Gridach et al. (2025).

Our work addresses a downstream professional transformation of scientific knowledge rather than scientific discovery itself. The common design lesson is nevertheless similar: explicit decomposition, intermediate representations, domain-specific tools or skills, and external feedback can provide structure that is absent from monolithic generation.

A.5 Test-Time Scaling and Iterative Refinement

A complementary line of research improves LLM performance by allocating additional computation at inference time. Self-consistency aggregates multiple reasoning trajectories Wang et al. (2023), Tree of Thoughts searches over intermediate states Yao et al. (2023a), and test-time scaling studies show that the value of additional inference computation depends on task difficulty and allocation strategy Snell et al. (2024). These results motivate our comparison across reasoning-effort settings and our view that stronger models, more inference computation, and better agent design are potentially complementary rather than mutually exclusive.

Iterative linguistic feedback provides another form of test-time improvement. Self-Refine repeatedly critiques and revises a model's output using self-generated feedback Madaan et al. (2023), while Reflexion uses verbal feedback and reflective memory to improve subsequent agent behavior Shinn et al. (2023). Our QA-guided loop separates the generator and evaluator roles: a patent-specific judge evaluates the complete package under a stable rubric, and the resulting report is supplied to the drafting agent for the next revision. The control condition uses the same drafting model and repeated revision without access to that structured QA report.

B Vibe Patenting: Agentic Framework

We develop *Vibe Patenting*, an agentic AI framework that transforms scientific and technical source materials into a coordinated patent draft package. Unlike zero-shot prompting, in which an LLM directly converts source text into patent prose, our framework explicitly decomposes patent drafting into invention understanding, protection-oriented reasoning, document generation, and quality assurance. The system accepts heterogeneous technical evidence—such as research papers, technical reports, slides, experimental results, and source code—and produces claims, a specification, patent figures, and supporting analysis reports. Figure 1 illustrates the overall workflow.

B.1 From Technical Evidence to Patent Package

Let $\mathcal{X}$ denote a collection of technical source materials describing a candidate invention. The objective is to produce a patent package $\mathcal{D} = \{\mathcal{C}, \mathcal{S}, \mathcal{F}, \mathcal{A}\}$, where $\mathcal{C}$ denotes claims, $\mathcal{S}$ the written specification, $\mathcal{F}$ patent figures, and $\mathcal{A}$ auxiliary analyses such as invention characterization, prior-art-oriented observations, embodiment expansion, support analysis, and quality assessment.

This formulation differs from ordinary long-form generation because the output components are mutually constrained. Independent claims should capture the inventive concept while maintaining support in the written description; dependent claims should provide meaningful fallback positions; embodiments should broaden implementation coverage; figures should correspond to the terminology and structures described in the specification; and all components should remain consistent with the technical evidence. We therefore treat patent generation as a coordinated reasoning problem rather than independent document generation. The overall process consists of four stages:

1. **Technical evidence ingestion:** analyze the source material and extract technical contributions, assumptions, mechanisms, experimental evidence, and candidate inventive concepts.
2. **Invention structuring:** organize the candidate invention through the PISE representation described below.
3. **Patent package generation:** invoke specialized patent-oriented skills for protection strategy, claim drafting, specification drafting, embodiment expansion, figure generation, and supporting analysis.
4. **Quality assurance and revision:** evaluate the complete draft using an external QA process and, when necessary, feed the resulting analysis back to the drafting agent.

B.2 PISE: A Structured Invention Representation

A central component of both agentic implementations is the *Problem–Insight–Solution–Effect (PISE)* representation. Given technical evidence $\mathcal{X}$, the agent first constructs $\mathcal{Z}_{\mathrm{PISE}} = (P, I, S, E)$, where:

- P (**Problem**) describes the technical limitation, unmet need, or deficiency being addressed;
- I (**Insight**) captures the key technical recognition or inventive idea that enables a solution;
- S (**Solution**) describes the technical mechanism, structure, procedure, or combination that realizes the insight; and
- E (**Effect**) identifies the technical consequences, improvements, capabilities, or measurable advantages produced by the solution.

PISE is intended to make latent invention structure explicit before patent text is generated. In particular, the four elements are constrained semantically rather than being independent summaries. The solution should address the problem through the identified insight, and the stated effects should be technically attributable to the solution. These relations can be expressed abstractly as $I \Rightarrow S, S \models P, S \Rightarrow E$, where the notation represents semantic consistency rather than formal logical implication.

A technical work may contain multiple related inventive concepts. In such cases, the system can construct a hierarchy or collection of PISE units, $\mathcal{Z} = \{\mathcal{Z}^{(1)}_{\mathrm{PISE}}, \ldots, \mathcal{Z}^{(K)}_{\mathrm{PISE}}\}$, allowing the agent to separate a broad inventive concept from narrower implementations and alternative embodiments. This structured representation becomes the shared reasoning state used by downstream drafting modules.

The motivation for PISE is that papers and technical reports are generally organized to communicate scientific results, whereas patents must organize the same technical knowledge around protectable inventive concepts. PISE provides an intermediate abstraction between these two document structures.

B.3 Professional Patent-Drafting Skills

After constructing the invention representation, the agent invokes patent-oriented reasoning and generation skills. The workflow includes technical insight and invention analysis; prior-art-oriented analysis; strength/weakness assessment; protection and claim strategy; alternative embodiment generation; independent and dependent claim drafting; specification drafting; patent figure generation; support and consistency analysis; and quality review and revision.

These skills are not intended to act as independent text generators. They operate on a shared representation of the invention and on artifacts produced by earlier stages. For example, claim drafting uses the PISE representation together with the selected protection strategy; specification drafting expands the same concepts into enabling embodiments; and figure generation converts important structures and relationships into graphical form consistent with the specification.

The resulting computation can be summarized as $\mathcal{X} \xrightarrow{f_{\text{inv}}} \mathcal{Z}_{\text{PISE}} \xrightarrow{f_{\text{strategy}}} \mathcal{R} \xrightarrow{f_{\text{draft}}} \mathcal{D}$, where $\mathcal{R}$ represents intermediate patent strategy and embodiment decisions. This decomposition enables the system to perform explicit invention-oriented reasoning before committing to detailed patent language.

B.4 Professional Agent Implementations

We investigate three implementations of professional AI patent drafting, illustrated in Figure 3 in the main paper. The discussion below provides additional implementation detail beyond the concise main-text comparison.

Generalist chat. The simplest approach uses a modern general-purpose chat LLM to convert technical materials directly into a patent draft in a zero-shot manner. Any task decomposition is implicit rather than imposed through explicit patent-specific workflow scaffolding.

Meta-agentic framework. The second implementation uses a general-purpose agent to construct professional skills from expert knowledge and orchestrate them within a patent-specific workflow. The agent receives a high-level patent-generation task and dynamically invokes reusable skills for invention analysis, PISE construction, claim drafting, specification drafting, figure generation, and related subtasks. It autonomously passes intermediate artifacts to later workflow stages.

Custom patent agent. The third implementation packages professional patent-drafting behavior into a dedicated domain agent. Patent-specific workflow instructions, drafting guidelines, templates, reference materials, and specialized skills are directly available to the agent. The agent is instructed to perform invention analysis, construct the PISE representation, develop a protection strategy, and coordinate the generation of the complete patent package. This architecture places substantial domain knowledge inside the agent configuration. It therefore provides a strong form of professional scaffolding and serves as our most specialized system.

Conceptually, the custom agent encodes professional behavior primarily through a dedicated agent configuration, whereas the meta-agentic system externalizes more of this behavior into modular skills and workflow components through the use of a generalist AI agent. This distinction allows us to examine whether professional capability can be obtained through reusable agentic scaffolding rather than only through a strongly customized domain agent.

B.5 End-to-End Workflow

Given technical source material $\mathcal{X}$, both agentic implementations execute a workflow of the general form

$$\mathcal{X} \rightarrow \mathcal{Z}_{\text{PISE}} \rightarrow \mathcal{R} \rightarrow \{\mathcal{C}, \mathcal{S}, \mathcal{F}, \mathcal{A}\} \rightarrow \mathcal{Q}, \tag{3}$$

where $\mathcal{Z}_{\text{PISE}}$ is the structured invention representation, $\mathcal{R}$ denotes protection-strategy and embodiment decisions, $\mathcal{C}$ denotes claims, $\mathcal{S}$ the specification, $\mathcal{F}$ the figures, $\mathcal{A}$ supporting analyses, and $\mathcal{Q}$ the QA report.

Table 3 summarizes the major skills used in the system.

B.6 Custom Patent Agent

The custom implementation integrates the patent-specific workflow, professional instructions, reference materials, templates, and specialized skills into a dedicated domain agent. The agent is explicitly instructed to reason about the technical material before drafting and to generate a coordinated patent package rather than independent document fragments.

Table 3: Representative professional skills used by the Vibe Patenting framework.

Skill	Purpose	Primary Input	Output
Technical analysis	Identify contributions, mechanisms, assumptions, and evidence	Paper, report, slides, or code	Structured technical findings
Invention discovery	Identify potentially patentable concepts and abstractions	Technical findings	Candidate inventions
PISE construction	Represent each invention as Problem–Insight–Solution–Effect	Candidate invention + evidence	PISE representation
Prior-art-oriented analysis	Identify likely novelty and obviousness pressure points	PISE + technical context	Prior-art-oriented analysis
Strength/weakness analysis	Identify protection opportunities, gaps, and design-around risks	PISE + analysis	Strategic assessment
Patent strategy	Determine protection hierarchy and independent-claim concepts	PISE + strategic assessment	Claim/protection strategy
Embodiment expansion	Generate alternative implementations and fallback positions	PISE + technical context	Embodiment set
Claim drafting	Generate independent and dependent claim hierarchy	PISE + strategy + embodiments	Claims
Specification drafting	Generate enabling disclosure consistent with claims	Claims + PISE + embodiments	Specification
Figure generation	Generate patent-oriented system, process, and embodiment diagrams	PISE + claims + specification	Patent figures
Support and consistency analysis	Check terminology, claim support, and figure/spec alignment	Complete draft	Support analysis
Patent QA	Score and critique complete patent package	Draft + figures	Multi-axis scores + report
Revision	Address identified QA weaknesses	Draft + QA report	Revised draft

The custom agent therefore has access to:

- patent-drafting workflow instructions;
- structured PISE reasoning;
- patent strategy and embodiment-expansion procedures;
- claim and specification drafting skills;
- patent-figure generation procedures;
- support and quality-analysis skills; and
- revision procedures for addressing QA feedback.

The custom implementation represents a strongly domain-specialized configuration in which professional knowledge is embedded directly into the agent environment.

B.7 Meta-Agentic Implementation

The meta-agentic implementation separates the general orchestration agent from the patent-specific skills. The LLM agent first constructs professional skills and workflow logic from expert knowledge. A general-purpose agent then dynamically invokes reusable skills for invention analysis, PISE construction, claim generation, specification drafting, figure generation, and QA.

This design externalizes more of the professional knowledge into reusable skills and intermediate artifacts:

$$\text{Meta-Agent} + \{\text{Patent Skills}\} + \mathcal{Z}_{\text{PISE}} \rightarrow \text{Patent Package}. \tag{4}$$

The distinction between the two implementations is therefore not whether professional structure is used—both systems use it—but where that structure is represented. The custom agent encodes more professional behavior in a dedicated agent configuration, whereas the meta-agentic system composes modular skills through a more general orchestrator.

C PISE Representation and Examples

C.1 PISE Structure

PISE represents an invention using

$$\mathcal{Z}_{\text{PISE}} = (P, I, S, E), \tag{5}$$

where P, I, S, and E denote Problem, Insight, Solution, and Effect, respectively.

The purpose of PISE is not simply to summarize a technical paper. Instead, it reorganizes the source material according to invention-oriented semantics:

- the **Problem** identifies the relevant technical limitation;
- the **Insight** captures the inventive recognition that enables a new approach;
- the **Solution** identifies the technical mechanism implementing that insight; and
- the **Effect** describes the resulting technical consequence or improvement.

Figure 2 shows the PISE representation and its semantic relations. Specifically, the agent asks what problem or limitation exists in the prior art, what inventive insight overcomes it, what technical mechanisms implement that insight, and what technical effects or advantages follow.

The following simplified example illustrates the representation format. The text is schematic and is intended to show structure rather than reproduce any confidential invention.

> **Problem:** Existing inference-time compression methods may rely on fixed transformations that do not adapt to the structure of the particular input or activation instance.
>
> **Insight:** The compressibility of a tensor can depend strongly on how its dimensions or indexed components are organized, and this organization can be adapted using information available at inference time.
>
> **Solution:** Determine an input-dependent ordering or transformation of tensor components, apply the resulting transformation, and compress the transformed representation using a structured low-complexity model.
>
> **Effect:** The transformed representation can exhibit lower effective complexity, enabling reduced storage or computation while preserving inference quality.

The structured invention model can then be reused across downstream artifacts:

$$\mathcal{Z}_{\text{PISE}} \rightarrow \begin{cases} \text{claim scope}, \\ \text{fallback limitations}, \\ \text{embodiments}, \\ \text{specification}, \\ \text{figures}, \\ \text{QA criteria}. \end{cases} \tag{6}$$

C.2 Multiple PISE Units

A technical work may support multiple inventive concepts. In such cases, the system maintains a set

$$\mathcal{Z} = \left\{ \mathcal{Z}_{\text{PISE}}^{(1)}, \ldots, \mathcal{Z}_{\text{PISE}}^{(K)} \right\}. \tag{7}$$

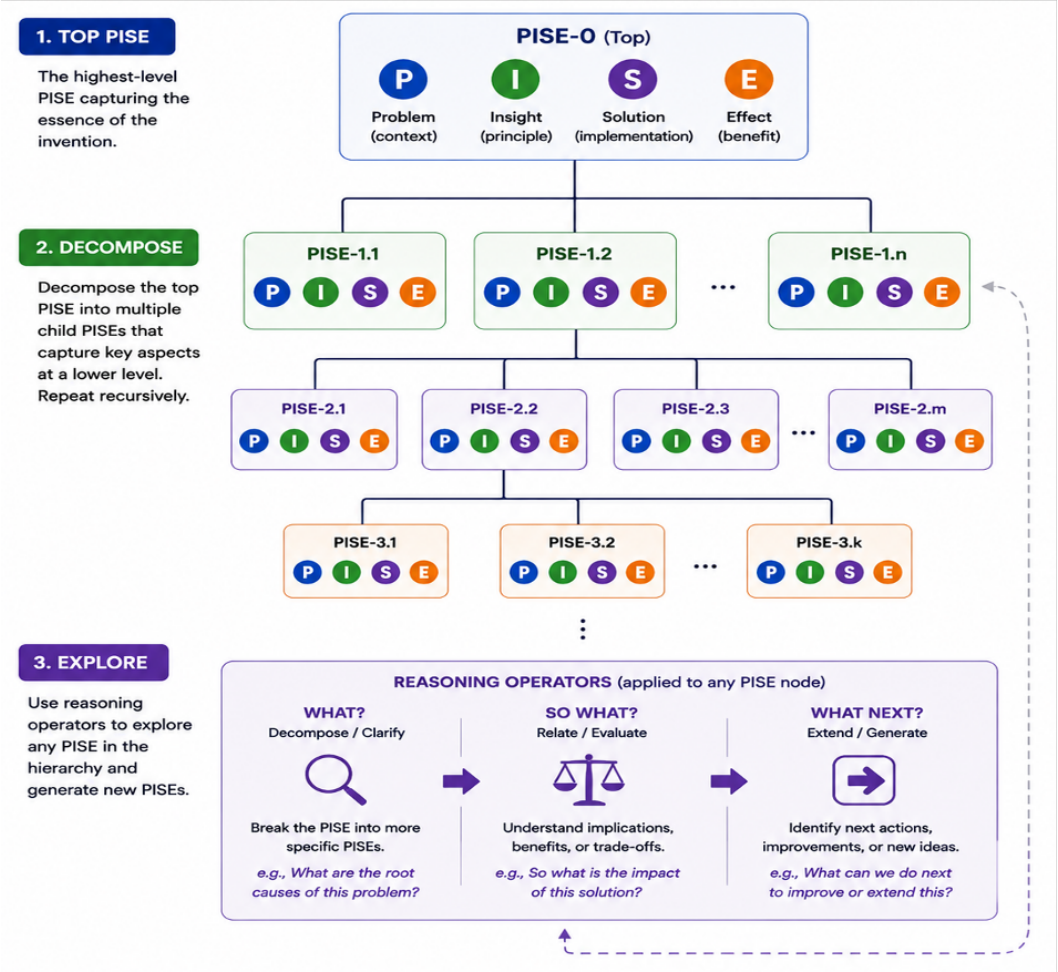


Figure 8: PISE hierarchy for constrained exploration: extract a core PISE, expand hierarchical branches, and explore new nodes through reasoning.

These units may represent a broad parent invention, narrower implementation mechanisms, alternative embodiments, hardware-specific realizations, or complementary inventions. The drafting workflow can selectively combine these units when constructing independent claims and fallback positions.

Figure 8 illustrates the hierarchy. From source material, the agent extracts PISE units, summarizes them in a unified top-level PISE, and explores new nodes by reasoning about root causes, downstream effects, and possible extensions.

D AI Judge for Patent Quality Assurance (QA)

D.1 Patent QA Skill

We built a reusable patent QA skill for rigorous pre-filing quality assurance of U.S. utility patent drafts under USPTO practice. It produces a structured report with integer scores from 1 to 10 on five axes and supports both draft-based and web-assisted review. It attempts to identify defects that could reduce claim value, create prosecution difficulty, weaken § 112 support, impair enforceability, or make the package unready to file. It always analyzes the actual draft and figures before scoring without inferring quality from polish alone.

The review workflow follows this sequence:

1. Inventory the supplied files and identify the primary patent draft, claim set, complete patent figure set, and any supplemental technical sources.
2. Read the entire patent draft, including claims, specification, abstract, tables, comments, tracked changes, placeholders, and embedded figures when accessible.
3. Inspect every patent figure visually at useful resolution. Do not rely only on extracted text or alt text.
4. If a supplemental technical report, invention disclosure, experimental report, or academic-paper draft is supplied, read it carefully as a separate technical source, including its figures, tables, equations, appendices, and experimental details.
5. Build a working model of the invention using all supplied technical information, while preserving a strict distinction between what the patent application itself discloses and what appears only in supplemental material.
6. Cross-check claims against the patent specification and patent figures. For each material independent-claim limitation, locate written-description support, enabling disclosure, and relevant patent-figure support where appropriate. Do not count report-only material as patent-application support.
7. Cross-check the patent draft against supplemental technical sources for omitted embodiments, mechanisms, variants, parameters, terminology, experimental evidence, technical advantages, and inconsistencies that matter to claim scope or filing value.
8. Analyze prosecution and claim risks under current U.S. utility patent practice, including § 101/102/103/112 issues, prior art, legal guidance, and possible publication or public-disclosure concerns.
9. Perform or recommend web-assisted review when appropriate, keeping intrinsic patent-draft findings, supplemental-source findings, and web-assisted findings distinct.
10. Review the patent figure set for substantive coverage, clarity, consistency, reference numerals, and filing-quality issues.
11. Review filing-readiness defects: unresolved drafting notes, inconsistent terminology, broken dependencies or antecedent basis, missing sections/figures, mismatched numbering, unsupported statements, report-only technical content that should be incorporated before filing, and other substantive or formal issues visible from the supplied package.
12. Prioritize findings by severity, then assign scores only after the qualitative analysis is complete and apply the scoring rubric consistently.
13. Create the final structured QA report.

The review standards are as follows:

Claims Review every independent claim closely and the dependent-claim strategy as a set. Evaluate at least:

- scope relative to the disclosed inventive contribution;
- unnecessary narrowing and accidental overbreadth;
- functional/result-oriented language and structural or procedural support;
- antecedent basis, dependency, clarity, and internal claim consistency;
- claim categories and coverage of commercially relevant implementations;
- fallback positions and useful dependent limitations;
- enforceability concerns, divided-infringement or actor issues when relevant;
- design-around opportunities and whether the claims protect the actual inventive center;
- support for each material limitation in the specification and figures;
- terminology drift between claims, specification, and figures.

Do not reward breadth by itself. Broad claims unsupported by disclosure or exposed to known art are weak claims.

Disclosure Evaluate § 112 support from the patent application itself across the intended claim scope. Supplemental technical sources may reveal what is missing, but they do not cure an omission unless the substance is included in the application before filing. Evaluate at least:

- written description for claimed combinations and alternatives;
- enablement across the claimed scope without undue experimentation;
- implementation detail proportionate to predictability of the technology;
- embodiments, alternatives, ranges, optionality, substitutions, and fallback disclosure;
- definitions and consistent use of important terms;
- support for functional language and any means-plus-function concerns;
- consistency among summary, detailed description, claims, abstract, and figures;
- whether likely amendments during prosecution would have clear original support.

Distinguish missing disclosure that cannot safely be added after filing from ordinary editorial improvements. Treat the former as much more serious.

Figures Evaluate both substantive coverage and drawing quality. Check whether the figures:

- cover each important embodiment, architecture, flow, state, component relationship, or variation that materially supports the claims;
- use consistent FIG. numbers, reference numerals, labels, arrows, and terminology;
- correspond to the written description and brief description of the drawings;
- avoid unexplained elements, orphan numerals, missing numerals, and conflicting labels;
- are legible and understandable without guessing;
- provide enough visual disclosure to support key structural/functional relationships;
- appear suitable for filing or need formal drawing cleanup.

Do not equate visual neatness with substantive figure completeness.

D.2 Scoring Rubric

Score calibration is as follows:

Score	General meaning
10	Exceptional and essentially filing-ready on this axis; no material deficiency found. Reserve for rare drafts.
9	Filing-ready on this axis with only minor, non-substantive cleanup.
8	Strong; limited meaningful improvements remain, but no major weakness.
7	Generally solid but meaningful revisions are advisable before filing.
6	Material weaknesses exist and should be addressed before filing.
5	Multiple material weaknesses or one major weakness; not comfortably filing-ready.
4	Significant deficiencies affect value, support, prosecution, or completeness.
3	Major systemic deficiencies create substantial prosecution or validity risk.
2	Severe deficiencies across core aspects of the axis.
1	Fundamentally deficient; substantial reconstruction is required.

D.3 Patent Quality Dimensions for QA Scoring

Our QA framework evaluates a patent draft along five dimensions:

1. **Prosecution Resilience**: withstanding prosecution challenges, including prior art, novelty, obviousness, eligibility, and other § 101/102/103 risks;
2. **Claim Strength**: quality and breadth of the claims, including independent-claim scope, fallback positions, enforceability, and design-around resistance;
3. **Disclosure Strength**: strength of § 112 support, including written description, enablement, embodiments, alternatives, terminology, and internal consistency;
4. **Figure Quality**: completeness and clarity of the figure set, including coverage of key embodiments/variations, readability, consistency, and support for the specification and claims; and
5. **Filing Readiness**: overall readiness for filing, considering substantive gaps, drafting issues, claims, specification, and figures.

Each dimension is scored on a 1–10 scale. The meta-agent used a separate QA-evaluation skill, and we report the mean score across five independent QA runs. This multidimensional evaluation enables us to compare general-purpose chat models, reasoning-effort settings, agentic systems, model generations, and QA-guided revision under a common quality framework. We additionally compare the AI assessments with evaluations from a professional patent attorney.

E QA Judge-Guided Revision

Generating a complete patent package in a single pass can leave weaknesses such as insufficient support, overly narrow or broad claims, terminology inconsistencies, missing embodiments, or inadequate figure coverage. We therefore introduce a QA-guided revision loop in which the drafting and evaluation functions are separated.

Let $\mathcal{D}_t$ denote the patent draft produced at revision round t. An independent QA agent evaluates the draft and produces a structured report $\mathcal{Q}_t = f_{\mathrm{QA}}(\mathcal{D}_t)$, where $\mathcal{Q}_t$ contains numerical assessments together with identified weaknesses and revision suggestions.

The drafting agent then receives the current patent draft, the corresponding QA report, and the original technical context: $\mathcal{D}_{t+1} = f_{\mathrm{rev}}\left(\mathcal{X}, \mathcal{Z}_{\mathrm{PISE}}, \mathcal{D}_t, \mathcal{Q}_t\right)$. The process can be iterated for several rounds: $\mathcal{D}_1 \to \mathcal{Q}_1 \to \mathcal{D}_2 \to \mathcal{Q}_2 \to \cdots \to \mathcal{D}_T$.

Figure 9 illustrates this external revision mechanism.

This setup differs from simply prompting the drafting agent to "improve" its own output. The QA report provides a stable domain-specific interface between evaluation and generation, allowing the same evaluation procedure to guide different models and agent configurations. It also enables us to experimentally measure how patent quality evolves as additional revision rounds are allocated.

E.1 Agent Instructions and Prompting

This section documents the principal prompting interfaces used in our experiments. We distinguish between the minimal baseline prompts used for generic LLM generation and the richer instructions available to the professional agentic systems.

E.1.1 Instruction Prompting

Each agent receives the source technical material together with a concise request to generate a patent draft. The prompt intentionally does not expose the full professional workflow or modular patent skills used by the proposed systems. For the common drafting task, all agents receive the same task-level request:

```
Use this technical report to create a full patent draft
package, including prior-art-oriented analysis, strengths
```

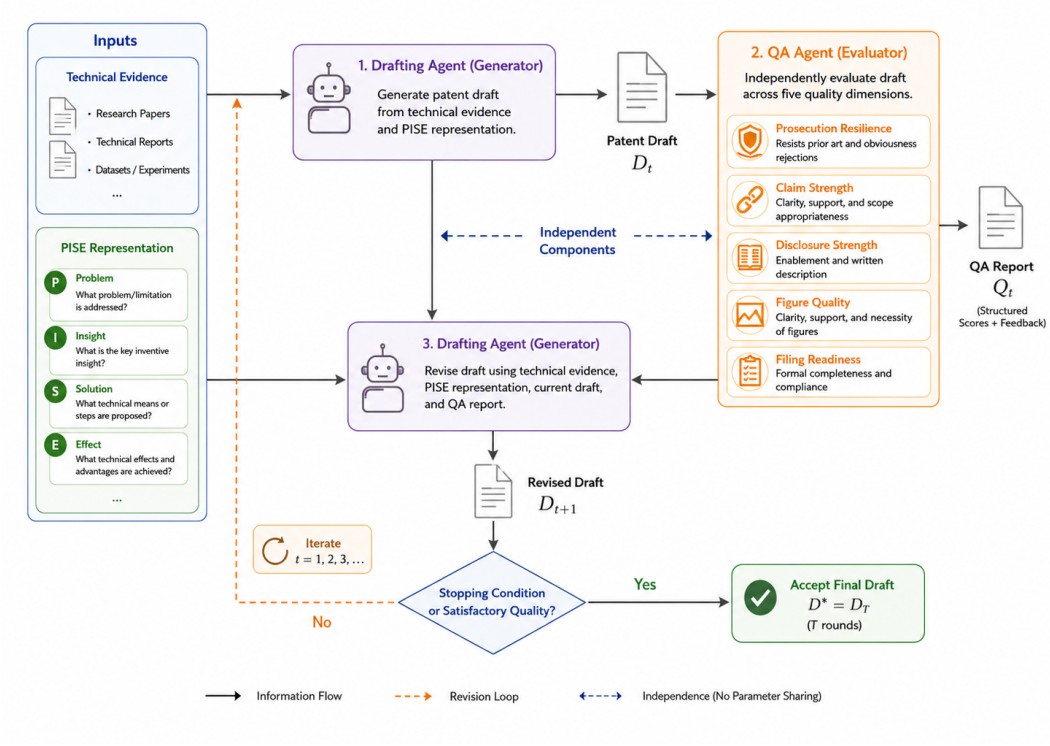


Figure 9: External QA-guided revision loop. A drafting agent generates from technical evidence and structured invention state; a separate QA process returns scores and feedback; the agent revises from that report over repeated rounds.

```
and weaknesses, embodiment expansion, 3 independent claims,
17 dependent claims, specification sections, a patent
draft markdown file, a patent draft DOCX file, direct
high-resolution black-and-white patent figure images, the
corresponding figure generation prompt markdown file, and a
patent-analysis note.
```

The same task-level request is held fixed across reasoning-effort settings whenever possible, so that the principal differences arise from the agent, model, or inference configuration.

E.1.2 Patent QA Prompt

The QA agent receives a completed patent draft and, when available, associated patent figures. It evaluates five dimensions on a 1–10 scale and provides structured justification and recommendations. The task-level instruction invokes the dedicated patent-QA evaluator described above. Proprietary system instructions and skill internals are not reproduced; the scientific interface is the completed draft and figures as input, followed by five-axis scores, findings, and revision recommendations as output.

E.1.3 QA Judge-Guided Revision Prompt

At revision round $t + 1$, the drafting agent receives the previous draft $\mathcal{D}_t$ together with the QA report $\mathcal{Q}_t$. The task-level request is augmented with an instruction to use the QA report:

```
Use this technical report to create a full patent draft
package, including prior-art-oriented analysis, strengths
and weaknesses, embodiment expansion, 3 independent claims,
17 dependent claims, specification sections, a patent
draft markdown file, a patent draft DOCX file, direct
```

```
high-resolution black-and-white patent figure images, the
corresponding figure generation prompt markdown file, and
a patent-analysis note.  NOTE: please revise and improve
the initial draft package attached according to the quality
analysis report given.
```

We intentionally separate the QA report from the drafting process so that the same feedback interface can be used across different model and agent configurations.

E.1.4 QA JUDGE-UNGUIDED GENERIC REVISION PROMPT

At revision round $t+1$, the drafting agent receives the previous draft $\mathcal{D}_t$ without using the QA report $\mathcal{Q}_t$. The control uses the same task-level request but omits the QA report:

```
Use this technical report to create a full patent draft
package, including prior-art-oriented analysis, strengths
and weaknesses, embodiment expansion, 3 independent claims,
17 dependent claims, specification sections, a patent
draft markdown file, a patent draft DOCX file, direct
high-resolution black-and-white patent figure images, the
corresponding figure generation prompt markdown file, and a
patent-analysis note.  NOTE: please revise and improve the
initial draft package attached.
```

This control isolates repeated revision from access to structured QA feedback.

F EXAMPLES OF GENERATED PATENT ARTIFACTS

This section provides representative outputs generated by the agentic workflow. The purpose is not to evaluate the legal merit of an individual patent application, but to illustrate the diversity and coordination of artifacts produced from a single technical input.

F.1 REPRESENTATIVE PATENT FIGURES

The framework generates patent-oriented figures rather than directly copying figures from the source publication. Depending on the invention, generated figures may include:

- overall system architectures;
- algorithmic process diagrams;
- component-level architectures;
- training or inference workflows;
- hardware embodiments;
- alternative implementations; and
- interaction diagrams among system components.

Figure 10 shows representative outputs.

F.2 REPRESENTATIVE CLAIM GENERATION

A generated patent package includes a hierarchy of independent and dependent claims. The agent first determines broad protection concepts from the PISE representation and then introduces narrower limitations as fallback positions.

Illustrative independent claim excerpt:

```
14.  A non-transitory computer-readable medium storing
instructions that, when executed by one or more processors,
cause the one or more processors to perform operations
comprising:  enumerating candidate tensor-encoding
```

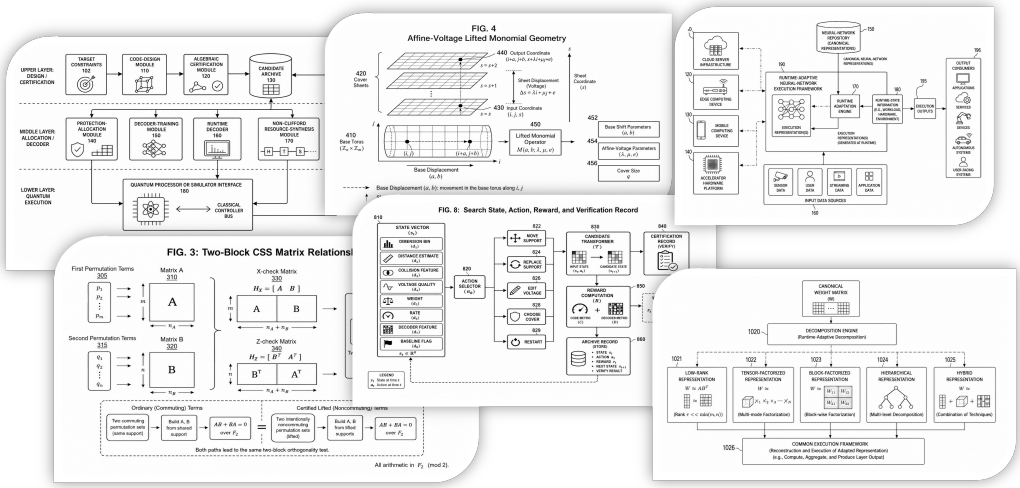


Figure 10: Representative patent figures generated from technical source material, reorganizing or expanding content into patent-oriented system, process, and embodiment views.

configurations, each candidate tensor-encoding configuration specifying a folding of a source tensor, a value mapping, a reduction across one or more slice modes, a permutation scope, a tensor-decomposition topology, one or more tensor ranks, and a permutation encoding; for each of a plurality of the candidate tensor-encoding configurations: generating a score field by applying the value mapping to values of the source tensor and applying the reduction across the one or more slice modes; deriving, from the score field, a candidate reversible permutation within the permutation scope; applying the candidate reversible permutation to multiple slices of the source tensor to obtain a candidate ordered tensor; forming candidate tensor cores by decomposing the candidate ordered tensor according to the tensor-decomposition topology and the one or more tensor ranks; determining a reconstruction-error measure and an encoded-storage measure that includes storage for the candidate tensor cores and storage for the candidate reversible permutation under the permutation encoding; selecting a selected tensor-encoding configuration from the plurality of the candidate tensor-encoding configurations based on the reconstruction-error measures and the encoded-storage measures; producing a compressed representation comprising selected tensor cores, selected permutation data, and a descriptor of the selected tensor-encoding configuration; and reconstructing at least a portion of the source tensor, or performing a consumer computation corresponding to the at least a portion, using the selected tensor cores, the selected permutation data, and the descriptor.

Illustrative dependent claim excerpt:

16. The non-transitory computer-readable medium of claim 14, wherein at least one candidate tensor-encoding configuration specifies a group sorting operation that leaves a within-group order unchanged or a sequential-axis sorting operation that sorts different subsets of tensor modes in sequence.

The claim-generation skill is coordinated with specification drafting so that important claim elements are supported by corresponding embodiments and terminology.

F.3 Representative Specification Expansion

The specification-generation stage expands the core invention beyond the narrow implementation appearing in the technical source. Typical expansions include:

- alternative architectures;
- alternative parameterizations;
- software and hardware embodiments;
- centralized and distributed implementations;
- optional processing stages;
- alternative ordering of operations; and
- variations addressing foreseeable design-arounds.

G Patent Analysis and QA Report Examples

In addition to the patent draft itself, the system produces analysis artifacts intended to expose the reasoning behind the generated protection strategy.

G.1 Patent Analysis

The AI agent produces a patent analysis report during drafting to improve quality within an internal review loop. A representative analysis report contains sections such as:

- Invention analysis (problem/solution/concepts/effect/evidence/missing facts);
- Prior-art analysis (pressure point/weakness/eligibility/enablement);
- Strengthening and distinction strategy (tree architecture/fallback positions/claim support crosswalk);
- Embodiment expansion lists;
- Claim support analysis (coverage/limitation/support/concern);
- Figure and description alignment review;
- Summary and recommended plan.

G.2 QA Report

The QA report is generated by a separate LLM judge. A representative QA report contains sections such as:

- Executive assessment;
- Priority issues before filing;
- Claim analysis (tree architecture/fallback positions/claim support crosswalk);
- Disclosure and § 112 analysis (description/enablement/definiteness/terminology);
- Figure analysis (coverage/quality/consistency);
- Web-assisted research findings;
- Recommended revision plan;
- Scoring and summary.

The QA agent evaluates the complete draft along the five dimensions used in the experiments. Table 4 illustrates a representative scoring report.

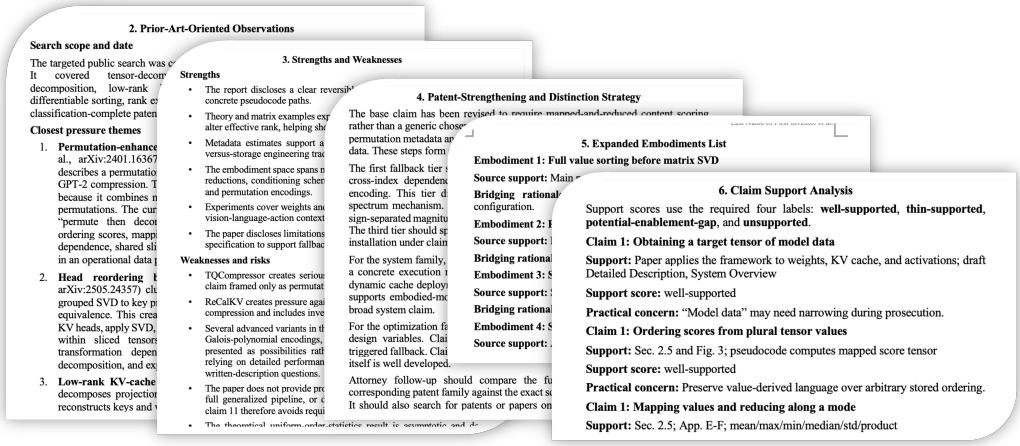


Figure 11: Sample of patent analysis report for patent drafting.

Table 4: Example QA report scores and rationales.

Dimension	Score	Rationales
Prosecution Resilience	6	The draft has plausible technical distinctions, but the art landscape contains variational state preparation, hardware-efficient brickwork/alternating ansätze, and resource-aware compilation. A targeted screen was performed, but not an exhaustive search; obviousness risk remains material.
Claim Strength	6	Claim 1 captures a concrete architectural distinction, but the three independent claims protect different centers and the dependent claims need stronger combination ladders. Claim 18 carries a support/provenance concern.
Disclosure Strength	6	Core fixed-topology embodiments are well described and broadly supported. Drafting-origin admissions for adaptive growth and learned prediction, plus limited quantitative detail for the key empirical effects, are material weaknesses.
Figure Quality	8	All 20 figures were present, legible, and substantially consistent with the specification. Improvements are mainly evidentiary/formal: schematic empirical plots, unexplained FIG. 18 numbers, and a few dense drawings.
Filing Readiness	5	The package should not be filed unchanged because inventor-possession/provenance issues and placeholders remain, and claim strategy should be tightened. The core invention is captured, so reconstruction is not required.

H Extended Experimental Results

H.1 Model, Reasoning, and Agent Scaling

Figure 5 in the main paper compares the five quality dimensions and overall score for the evaluated configurations. For GPT-5.6 Sol chat mode, increasing reasoning effort generally improves drafting quality. The overall score increases from 5.53 for Instant to 5.97 for Medium, 6.36 for High, and 6.58 for Extra-High. This trend supports inference-time scaling for the professional drafting task, although performance is not strictly monotonic across all available configurations; in particular, the observed Pro score is 5.93.

Agentic execution also produces strong results. The GPT-5.6 agent obtains an overall score of 6.15, while its Extra-High reasoning configuration reaches 6.80. The GPT-5.4 and GPT-5.5 agent configurations obtain overall scores of 5.32 and 6.20, respectively, providing additional evidence that underlying model capability contributes substantially to professional-task performance.

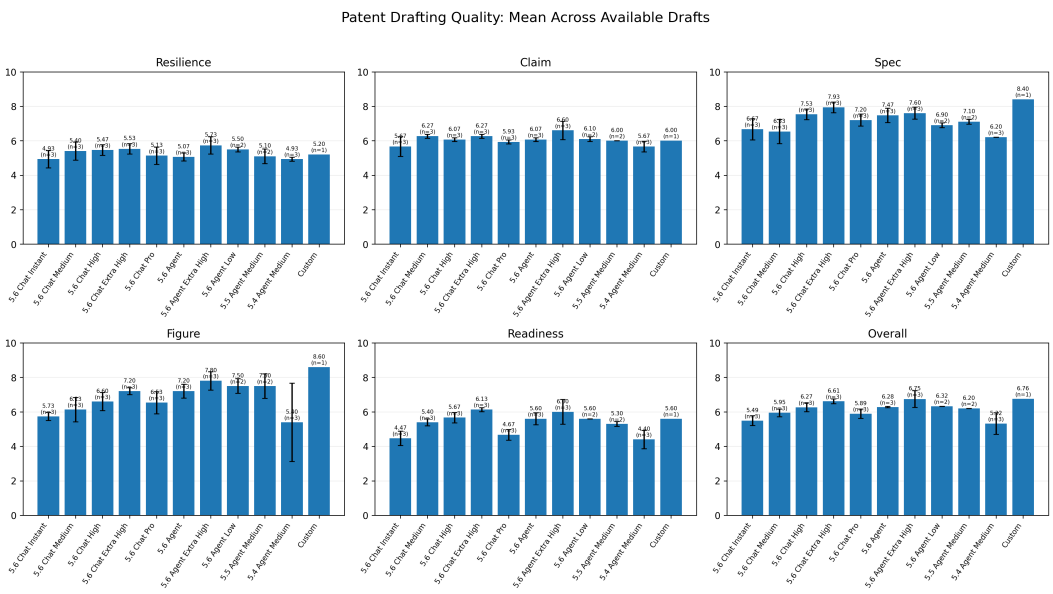


Figure 12: Extended quality comparison across model, reasoning-effort, and agentic configurations; error bars summarize variability across available drafts.

The custom patent agent obtains an overall score of 7.02 across the two available patents, with particularly strong Disclosure and Figure scores. Because the custom-agent configuration is available for only two patents, compared with seven for several other configurations, we treat this result as illustrative rather than as a fully balanced comparison.

Taken together, the results do not suggest that agentic structure replaces model or inference-time scaling. Rather, they indicate that *model capability, inference-time reasoning, and domain-specific agentic scaffolding are complementary mechanisms for improving professional drafting performance*.

H.2 Per-Dimension Model and Agent Results

Figure 12 reports the five individual quality dimensions and overall score for the evaluated model and agent configurations.

H.3 Observed Generation-Time Scaling

We next examine the computational cost of the different configurations using observed end-to-end generation time during patent generation. Figure 13 reports mean generation time with one standard deviation over repeated patent-drafting runs.

Chat-mode inference is substantially faster than full agent execution. For example, mean drafting times are approximately 5.3, 4.8, 7.6, and 12.1 minutes for Instant, Medium, High, and Extra-High chat configurations, respectively. GPT-5.6 Pro requires approximately 26.2 minutes, while the GPT-5.6 agent with Medium reasoning requires 28.3 minutes and the Extra-High agent approximately 41.5 minutes.

H.4 Quality Versus Observed Generation Time by Dimension

The relationship between drafting time and quality differs across evaluation dimensions. Figure 14 shows separate configuration-level scatter plots.

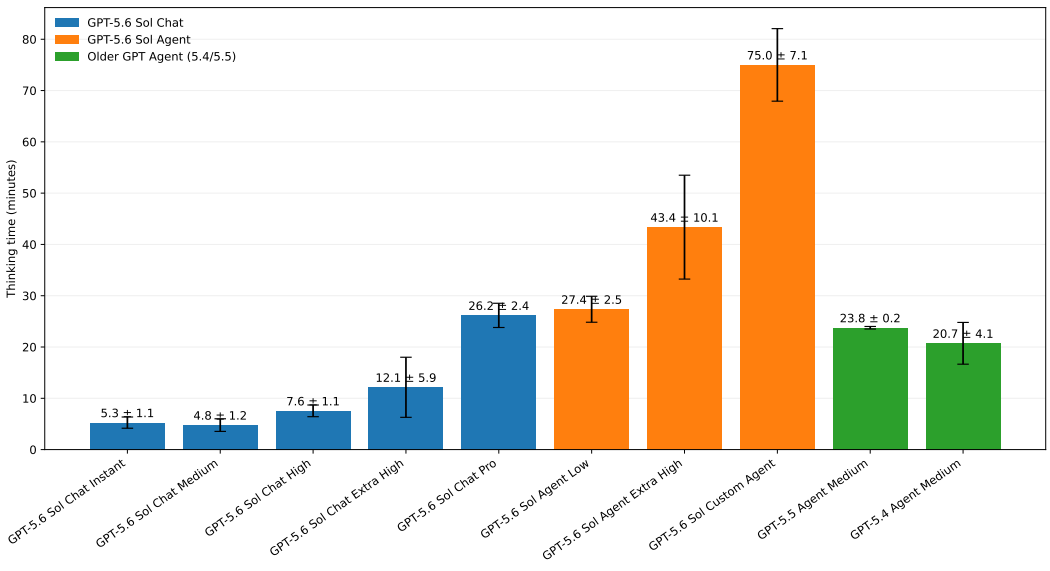


Figure 13: Mean observed patent-drafting generation time by model and agent configuration; error bars show one standard deviation across available runs.

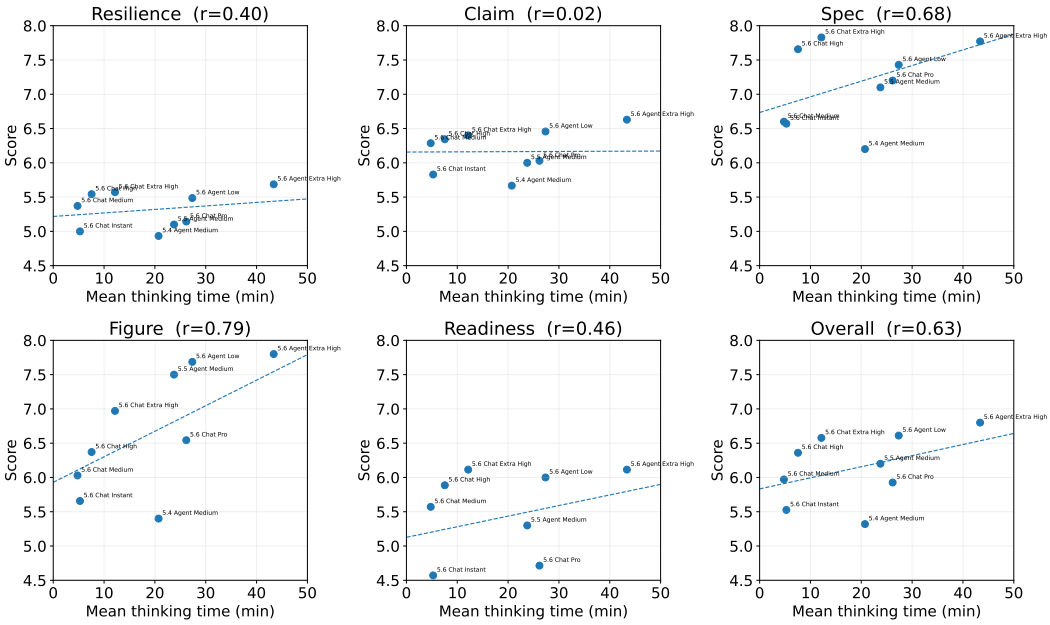


Figure 14: Patent-drafting quality versus mean observed generation time for five quality dimensions and the overall score; each point is one model/reasoning/agent configuration.

In our current configuration-level measurements, the Pearson correlations between observed generation time and the five dimensions are approximately

$$r_{\text{Resilience}} = 0.40, \tag{8}$$
$$r_{\text{Claim}} = 0.02, \tag{9}$$
$$r_{\text{Disclosure}} = 0.68, \tag{10}$$
$$r_{\text{Figure}} = 0.79, \tag{11}$$
$$r_{\text{Readiness}} = 0.46, \tag{12}$$

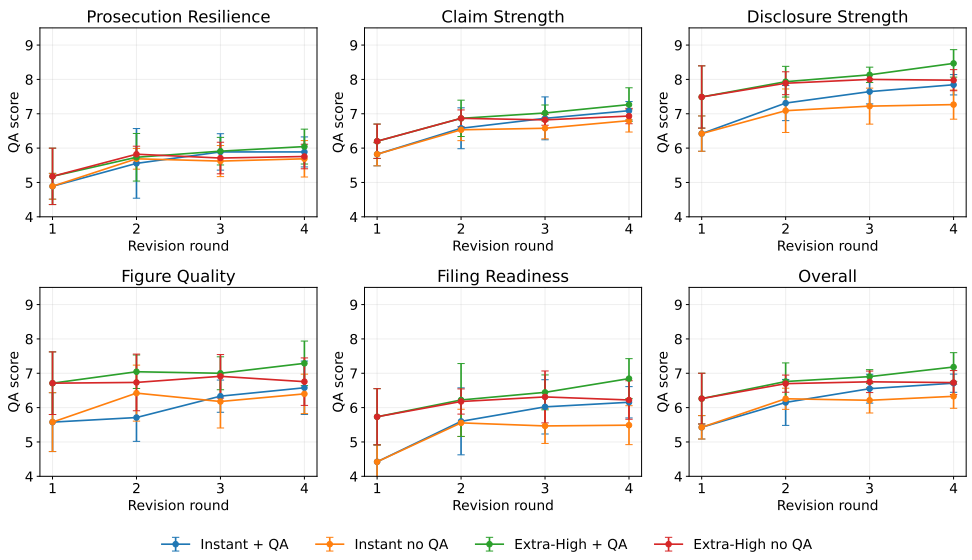


Figure 15: Per-dimension QA-guided revision trajectories for Instant and Extra-High, with and without structured QA feedback; error bars summarize variation across inventions.

with an overall correlation of approximately $r = 0.63$. These values are descriptive because the number of configurations is small and the observations are not randomized allocations of inference compute.

H.5 Per-Dimension QA-Guided Revision

Figure 15 provides the full five-axis revision trajectories that complement the overall-score trends in Fig. 6(a). The QA-guided trajectories generally continue improving through later rounds, but the size and variance of the gain differ substantially by dimension. This heterogeneity is operationally important because a scalar stopping rule can hide unresolved weaknesses even while the overall score increases.

H.6 Observed Generation Time Over Revision

Figure 16 shows the accumulated observed generation time over patent revision with and without QA judge guidance. Instant reasoning is approximately 3.3 times faster than Extra-High reasoning.

H.7 AI–Expert Agreement

Figure 17 shows the complete QA1-versus-attorney comparison. Figure 18 plots Pearson r and Spearman ρ by quality dimension. The 28-point set contains the 27 `agent`/`pro`/`instant` observations from nine patents plus the SLOP custom-agent observation. For TuneComp and LatentLLM, the `pro` condition is instantiated with the Extra-High baseline rather than AgentEx.

The resulting mapping gives an overall Pearson correlation of $r = 0.670$ and Spearman correlation of $\rho = 0.570$. Figure Quality has the strongest relative agreement ($r = 0.819$), followed by Disclosure Strength ($r = 0.533$), whereas Claim Strength has low rank discrimination despite small absolute error. This distinction motivates reporting both correlation and calibration-oriented metrics such as MAE and bias.

H.8 Three-Way Agreement with a Second QA Evaluator

We next introduce a second automated evaluator, **QA2**. QA1 is the original patent-QA evaluator used for optimization and uses Instant reasoning. QA2 is separately instructed, uses Extra-High reasoning, and follows the same five-axis rubric and scoring principles supplied to the professional

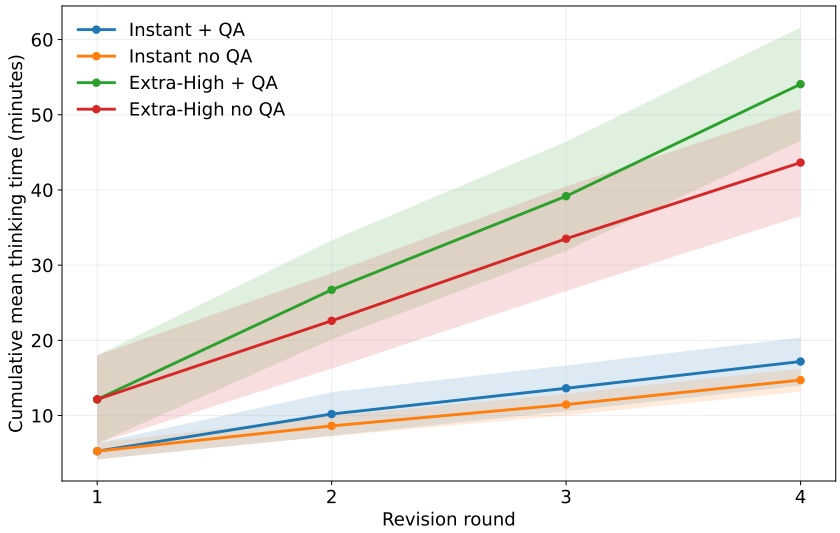


Figure 16: Observed end-to-end generation time across revision rounds: mean and standard deviation.

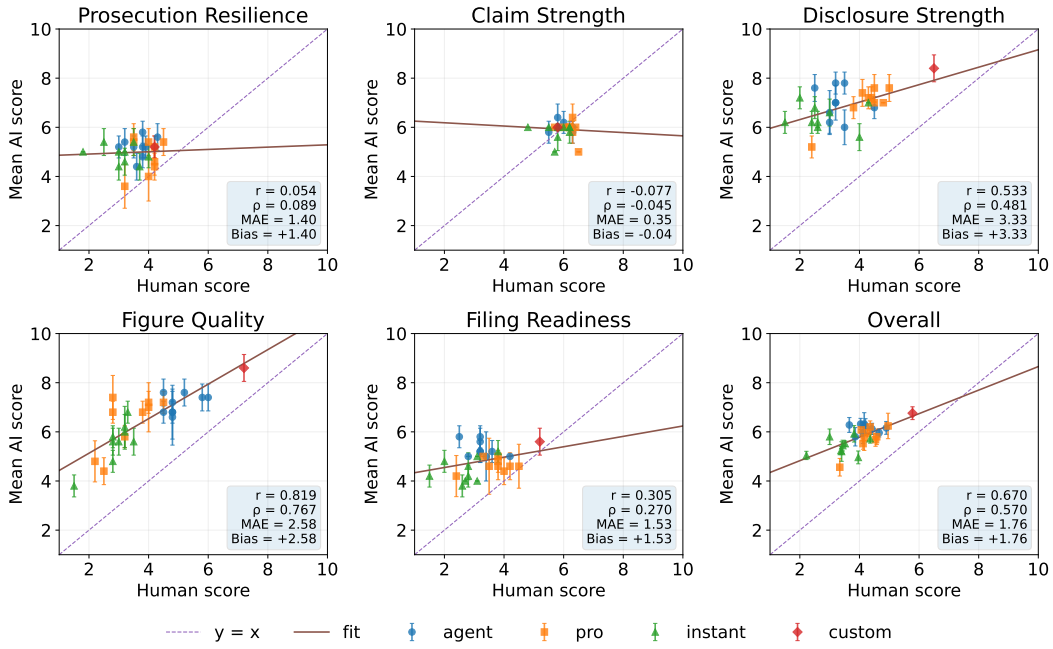


Figure 17: QA1 versus professional patent attorney scores for five quality dimensions and overall score across 28 matched conditions from nine patents; QA1 is averaged over five runs.

attorney. Thus QA2 is intentionally closer to the professional evaluation protocol, but the comparison changes both evaluator instructions/rubric alignment and inference effort; it should be interpreted as a comparison between evaluator systems rather than a causal ablation of either factor.

For a three-way matched analysis we use the 27 `agent`/`pro`/`instant` patent-condition pairs that have scores from QA1, QA2, and the attorney. The SLOP custom-agent point used in Table 2 is excluded because QA2 was not run on that condition. Pooled across these 27 common drafts, QA1–attorney and QA2–attorney correlations are similar in Pearson correlation ($r = 0.598$ and 0.603), while QA2 has somewhat higher rank agreement ($\rho = 0.558$ versus 0.520). QA1 and QA2 themselves agree strongly ($r = 0.797$, $\rho = 0.771$), despite QA2 scoring about 0.40 points higher on

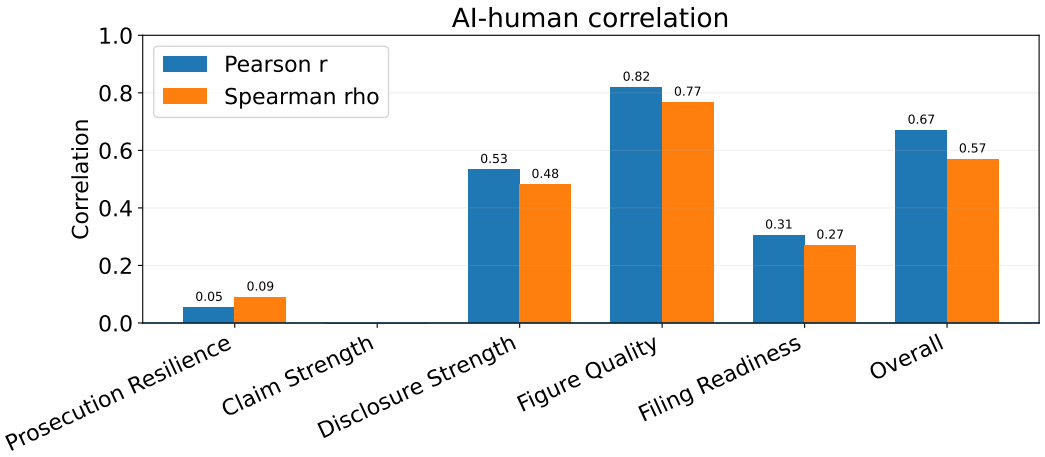


Figure 18: QA1–attorney correlation by quality dimension for the nine-patent validation set.

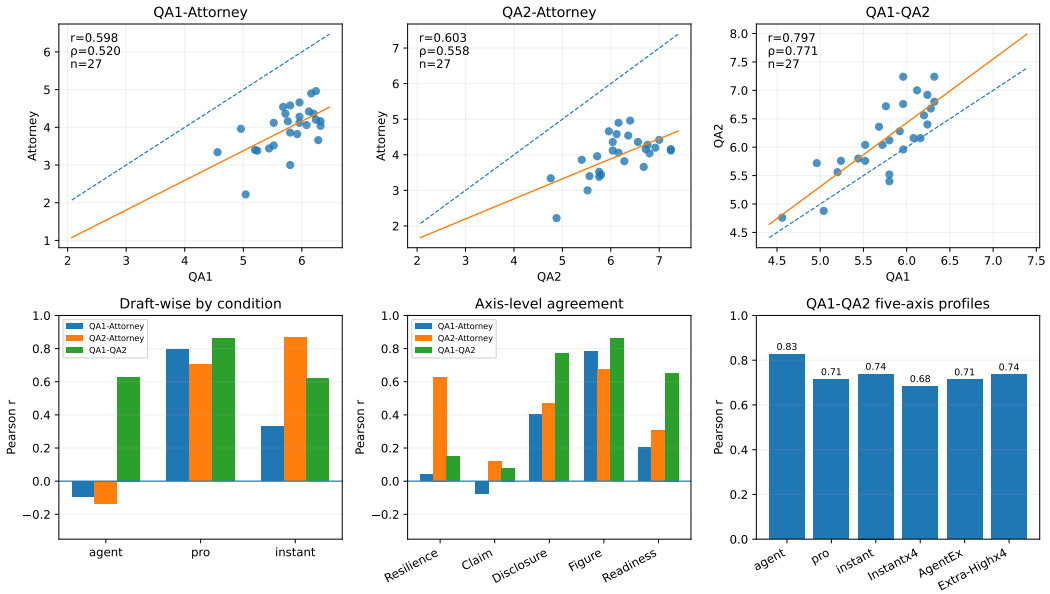


Figure 19: Three-way evaluator agreement. **Top:** pooled overall-score relationships on 27 common `agent`/`pro`/`instant` pairs. **Bottom:** Pearson r by condition, by quality axis, and between QA1–QA2 five-axis profiles for six complete configurations.

average. The main paper promotes the condition-wise and axis-level views in Fig. 7; Fig. 19 below retains the pooled scatter plots and configuration-profile view for completeness.

The pooled result masks substantial condition dependence. For `agent`, neither automated evaluator tracks the attorney well ($r = -0.092$ for QA1 and -0.133 for QA2), although QA1 and QA2 themselves remain positively associated ($r = 0.628$). For `pro`, QA1–attorney agreement is $r = 0.796$, QA2–attorney is $r = 0.703$, and QA1–QA2 is $r = 0.861$. The largest QA2 improvement occurs for `instant`: QA1–attorney is only $r = 0.332$, whereas QA2–attorney rises to $r = 0.868$; QA1–QA2 remains $r = 0.620$. Hence the more attorney-aligned evaluator is not uniformly better, but it much more closely reproduces the attorney's patent-to-patent ordering for the `instant` condition.

Axis-level behavior is similarly heterogeneous. QA2 substantially improves attorney agreement for Prosecution Resilience ($r = 0.628$ versus 0.038 for QA1), and modestly improves Disclosure Strength (0.472 versus 0.404) and Filing Readiness (0.310 versus 0.201). QA1 remains stronger for

Table 5: Three-way agreement on the 27 common agent/pro/instant patent-condition pairs. QA1 uses Instant reasoning; QA2 uses Extra-High reasoning and the attorney-aligned rubric/scoring principles. Bias is the first evaluator minus the second.

Comparison	Pearson r	Spearman ρ	MAE	Bias
QA1-Attorney	0.598	0.520	1.79	+1.79
QA2-Attorney	0.603	0.558	2.19	+2.19
QA1-QA2	0.797	0.771	0.46	-0.40

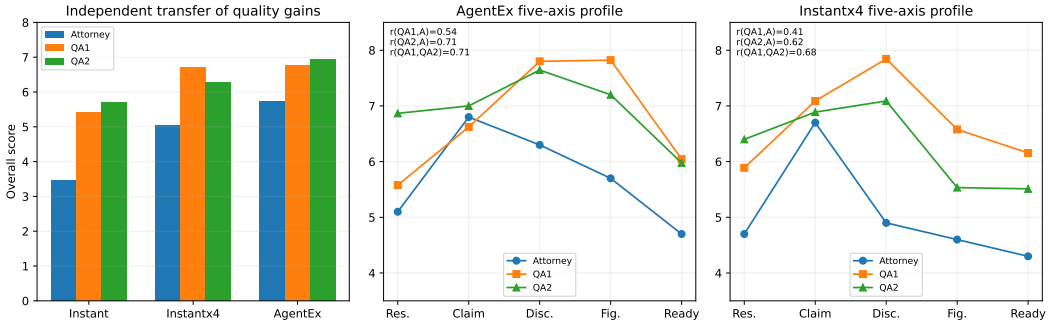


Figure 20: Aggregate three-evaluator validation. **(a)** Overall scores for Instant, Instant×4, and AgentEx. **(b–c)** Five-axis AgentEx and Instant×4 profiles. Attorney values for the latter two are nine-patent aggregate means because per-patent scores were unavailable.

Figure Quality (0.784 versus 0.673), while Claim Strength is weak for both. QA1 and QA2 agree most strongly on Figure Quality ($r = 0.862$) and Disclosure Strength (0.770), but agree weakly on Resilience (0.149) and Claim Strength (0.080). This shows that two LLM evaluators can share an overall scoring trend while encoding materially different professional preferences by axis.

A complementary five-axis view compares the aggregate quality profiles produced by QA1 and QA2. Their descriptive profile correlations are positive for all six complete configurations: $r = 0.827$ (`agent`), 0.714 (`pro`), 0.738 (`instant`), 0.683 (Instant×4), 0.713 (AgentEx), and 0.735 (Extra-High×4). Across all 54 patent/configuration observations for these six complete conditions, QA1 and QA2 have Pearson $r = 0.760$ and Spearman $\rho = 0.740$. The two judges therefore broadly agree, while their disagreements are structured rather than random.

H.9 Aggregate External Validation of AgentEx and Instant×4

The attorney additionally supplied configuration-level five-axis means for AgentEx and Instant×4, where Instant×4 denotes the final output after four QA1-guided Instant revision rounds. The per-patent attorney matrix was not supplied for these two conditions, so this analysis remains separate from the 28 matched per-draft observations used in Fig. 6(b), Fig. 17, and Table 2.

The three evaluators agree that QA-guided revision improves Instant drafts in aggregate. QA1 rises from 5.43 to 6.71 (+1.28), QA2 rises from 5.70 to 6.28 (+0.58), and the attorney rises from 3.46 to 5.04 (+1.58). At the per-patent level QA2 scores Instant×4 above Instant for 8 of 9 patents; the paired mean improvement is +0.58 ($p = 0.009$ by a two-sided paired t-test). This transfer to both QA2 and the attorney shows that revision gains are not confined to the optimizing QA1 evaluator.

The detailed profiles nevertheless expose evaluator-specific effects. For AgentEx, five-axis Pearson correlations are $r = 0.545$ for QA1–attorney, 0.714 for QA2–attorney, and 0.713 for QA1–QA2. For Instant×4 they are 0.410, 0.622, and 0.683, respectively; the QA2–attorney Spearman profile correlation is $\rho = 0.900$. Because these profile correlations use only five axes, they are descriptive rather than inferential.

A particularly informative comparison is AgentEx versus Instant×4. QA1 nearly ties the two configurations (AgentEx advantage +0.06), whereas QA2 gives AgentEx a +0.65 advantage and the attorney gives +0.68. QA2 and the attorney agree on the direction of all five dimension-wise

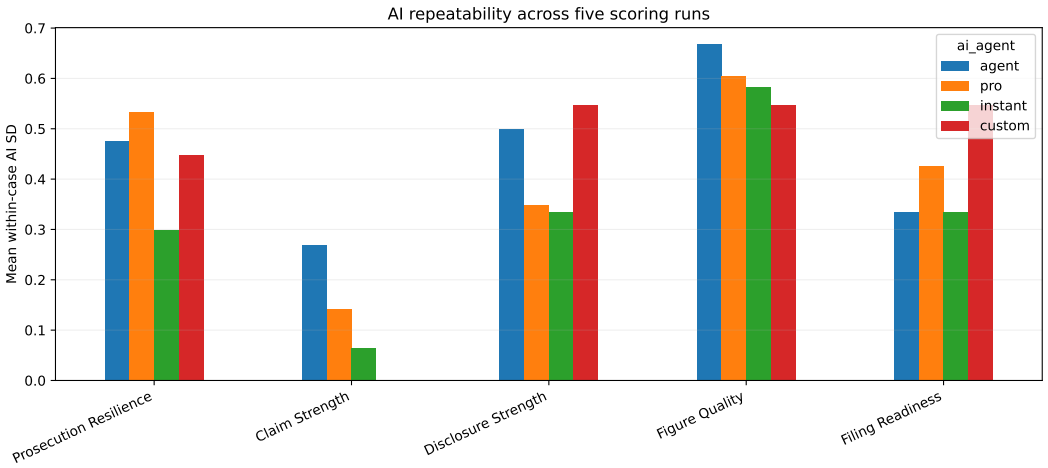


Figure 21: Run-to-run variability of repeated QA1 evaluation across patent-quality dimensions.

AgentEx–Instant×4 differences, while QA1 favors AgentEx only on Figure Quality. This pattern is consistent with evaluator-specific optimization: QA1-guided revision yields genuine transferable improvement, yet repeated optimization against QA1 can additionally move drafts toward QA1-specific preferences. The data do not establish preference overfitting causally, because QA2 differs from QA1 in both instructions/rubric alignment and reasoning effort.

H.10 QA Agent Repeatability

Because each draft in the original QA1 study is evaluated five times, we can also characterize the stability of QA1. For draft i and quality dimension k, we compute

$$\sigma_{i,k} = \mathrm{Std}\left(q_{i,k}^{(1)}, \ldots, q_{i,k}^{(5)}\right). \tag{13}$$

Low $\sigma_{i,k}$ indicates that repeated QA executions return similar judgments, whereas high variability identifies dimensions or drafts for which the evaluator is less stable.

Figure 21 shows the mean within-class standard deviation across QA1 runs. The standard deviation is generally below 0.5 points on the integer-valued scale, indicating relatively stable QA1 scores.

H.11 Summary of Findings

Our experiments yield several complementary observations. First, judge-assessed patent quality generally improves with stronger model capability, greater inference-time reasoning, and domain-specific agentic scaffolding, although additional computation alone does not guarantee proportional gains. The skilled human drafter provides a separate generation baseline, distinct from the professional patent attorney used for expert evaluation.

Second, iterative revision substantially improves the LLM judge's scores. Generic revision provides considerable initial gains but tends to saturate, whereas QA-guided revision continues to improve through the evaluated revision rounds. This effect is particularly pronounced for the low-reasoning

Table 6: Three-evaluator aggregate validation. Overall scores are nine-patent means. The three r columns are descriptive Pearson correlations across the five quality-axis means only.

Configuration	QA1	QA2	Attorney	$r_{1,A}$	$r_{2,A}$	$r_{1,2}$
Instant	5.43	5.70	3.46	0.265	0.618	0.738
Instant×4	6.71	6.28	5.04	0.410	0.622	0.683
AgentEx	6.77	6.94	5.72	0.545	0.714	0.713

generator. After repeated judge-guided refinement, low-reasoning generation approaches the judge-assessed quality achieved by substantially more expensive high-reasoning generation, suggesting that revision depth and structured evaluator feedback can partially compensate for inference strength.

Third, comparison with the professional patent attorney shows that LLM-judge agreement depends strongly on both quality dimension and draft condition. QA2, which uses the attorney-aligned rubric and Extra-High reasoning, substantially improves attorney agreement for Instant drafts and Prosecution Resilience, but does not dominate QA1 on every condition or axis. QA1 and QA2 themselves remain strongly associated overall, showing that evaluator disagreement is structured rather than arbitrary. Correlation, calibration, repeatability, and evaluator choice therefore capture distinct properties of automated professional assessment.

Finally, aggregate professional validation separates transferable revision gains from evaluator-specific optimization. Instant$\times$4 improves over Instant under QA1, QA2, and the attorney, so QA1-guided revision does not merely raise its own reward. Yet QA1 nearly ties Instant$\times$4 with AgentEx, whereas QA2 and the attorney independently recover almost the same AgentEx advantage ($+0.65$ and $+0.68$). This combination is consistent with genuine quality improvement plus an additional QA1-specific preference effect. Together, these findings support LLM judges as scalable feedback mechanisms while preserving the need for independent evaluators and professional validation.

I Limitations and Responsible Use

I.1 Experimental Limitations

Our experiments are intended as an initial controlled study of professional agentic drafting and have several limitations.

First, the number of technical inventions is modest, and not every model or agent configuration was evaluated on every patent. We therefore distinguish balanced evaluations from illustrative results and explicitly report the number of contributing cases where appropriate.

Second, expert human evaluation is currently based on a single professional patent attorney. Patent quality involves subjective professional judgment, and future evaluation should include multiple practitioners to characterize inter-expert variability and establish an appropriate human performance range.

Third, the underlying frontier models and their reasoning modes are rapidly evolving. Configuration names such as Instant, High, Extra-High, and Pro identify product-level inference settings rather than stable computational algorithms. The precise behavior, resource allocation, and latency of such systems may change over time.

Fourth, both automatic QA evaluators are LLM-based systems. Although we compare them with professional evaluation, each exhibits dimension- and condition-dependent bias. QA2 also differs from QA1 in both instructions/rubric alignment and reasoning effort, so the observed agreement differences cannot be attributed causally to either change in isolation. Neither evaluator should be interpreted as an objective ground-truth metric.

Finally, our present evaluation measures draft quality before prosecution. Long-term patent value depends on factors that cannot be established from an initial application alone, including examiner search results, prosecution history, allowed claim scope, enforceability, litigation outcomes, and commercial relevance.

I.2 Professional Oversight

Vibe Patenting is intended to automate substantial portions of *patent production*, not to eliminate professional judgment from the intellectual-property process. Important decisions remain outside the scope of autonomous drafting, including:

- determining whether an invention should be patented;
- verifying technical accuracy with the inventors;
- determining inventorship;

- assessing legal obligations and disclosure requirements;
- making final patentability and claim-strategy judgments;
- approving the application for filing; and
- conducting prosecution before a patent office.

A useful operating model is therefore

$$\text{Researchers create and validate} \rightarrow \text{AI structures and drafts} \rightarrow \text{Professionals review and approve.} \tag{14}$$

The objective is to reduce repetitive drafting and coordination effort so that researchers and patent practitioners can spend more time on technical truth, strategic judgment, and portfolio decisions.

I.3 Confidentiality and Data Governance

Patent drafting frequently involves unpublished and commercially sensitive technical information. Deployment of an automated patent-drafting system therefore requires appropriate controls over data retention, model access, logging, external tool usage, and disclosure of confidential materials. Organizations should ensure that the AI infrastructure used for drafting is compatible with their confidentiality, security, and intellectual-property policies.

I.4 Future Work

Several directions follow naturally from the present study:

- evaluation on a substantially larger and more diverse collection of inventions;
- evaluation by multiple patent practitioners and measurement of human–human agreement;
- comparison with commercial patent-drafting systems where reproducible access is available;
- controlled ablation of individual professional skills and PISE components;
- evaluation using open-weight models and reproducible inference budgets;
- optimization of the quality–compute trade-off through adaptive reasoning allocation;
- learned stopping criteria for QA-guided revision;
- longitudinal evaluation using prosecution outcomes and allowed claim scope; and
- extension of structured professional-agent workflows to other scientific, engineering, legal, and business tasks.