# DraftNEPABench: A Benchmark for Drafting NEPA Document Sections with Coding Agents

**Anurag Acharya[1], Bishal Lakha[1], Rounak Meyur[1], Rohan Nuttall[2], Sarthak Chaturvedi[1], Anika Halappanavar[1], Leah Hare[1], Lin Zeng[1], Mike Parker[1], Sai Munikoti[1], Sameera Horawalavithana[1]**

[1]Pacific Northwest National Laboratory, Richland, WA

{firstname.lastname}@pnnl.gov

[2]OpenAI, San Francisco, CA

rohan@openai.com

## Abstract

Coding agents represent a transformative paradigm in software engineering, enabling automated coding, generation, and debugging through a natural language interface. Recent advancements in large language models (LLMs) and their ability to use external tools have expanded the potential of using these agents beyond software engineering tasks. In this work, we explore the application of coding agents in a noncoding domain: drafting environmental impact statement (EIS) sections. For that, we introduce DraftNEPABench: a challenging benchmark that requires coding agents to compose structured, coherent, and domain-specific drafts grounded in multiple complex regulatory and scientific reference materials. We evaluate various state-of-the-art commercial coding agents on this benchmark and demonstrate their promise in generating EIS documents. Our findings show that while coding agents outperform vanilla retrieval-augmented generation (RAG) setups for these tasks, there is still room for improvement. We highlight the potential and limitations of such agents in high-stakes, complex, and real-world tasks, and point toward future directions.

## 1 Introduction

Large language model (LLM)-based coding agents represent a major advancement in artificial intelligence (AI)-assisted software engineering, leveraging LLMs to autonomously perform tasks such as code generation, debugging, static analysis, and test creation (Liu et al., 2024a; Qian et al., 2023; Zhang et al., 2024a; Yang et al., 2024). Unlike traditional code completion tools, modern agents like Codex CLI (OpenAI, 2025a), Claude Code (Anthropic, 2025a), Gemini CLI (Google, 2025), SWE-Agent (Yang et al., 2024), AutoCoder-Rover (Zhang et al., 2024b), and CodeAgent (Zhang et al., 2024a) integrate with development environments and external tools, enabling iterative solutions for complex, multistep, and repository-level tasks. These extended capabilities enable the agents to tackle a wide range of challenges, redefining the boundaries of automated software development.

More recently, the application of coding agents has extended beyond software development, demonstrating their adaptability to a range of specialized fields (Guo et al., 2024; Gandhi et al., 2025). These agents demonstrate capabilities to autonomously retrieve and synthesize documents from multiple sources, interact with external tools, and iteratively generate both structured and unstructured content (Yang et al., 2024; Zhang et al., 2024a,b). These capabilities have enabled coding agents to generalize beyond traditional software engineering tasks into domains such as healthcare analytics, scientific data analysis, and research automation, where they support database querying, statistical computation, and complex workflow execution through natural language interfaces (Wu et al., 2024; Gandhi et al., 2025).

In this study, we aim to evaluate how effectively coding agents can adapt to regulatory domains such as National Environmental Policy Act (NEPA) reviews. NEPA requires federal agencies to conduct environmental reviews, often including an environmental impact statement (EIS) or other documentation to assess the environmental impacts of a proposed action. Drafting such EIS sections presents complex and challenging tests for the generalizability and adaptability of coding agents in different domains. These drafts are multimodal, multisource, and domain-intensive and must be coherent and legally defensible, integrating scientific data, geospatial analyses, engineering specifications, environmental modeling outputs, and regulatory policy information dispersed across lengthy technical reports, environmental databases, policy documents, and expert assessments. Drafting a high-quality EIS from such heterogeneous sources demands strong expertise in environmental science, engineering, and regulatory compliance.

**Contributions.** In this work, we introduce a novel drafting benchmark called **DraftNEPABench** for evaluating the capabilities of LLM-based agents for drafting EIS sections. The benchmark is curated by subject matter experts (SMEs) to reflect realistic and domain-relevant drafting challenges. We systematically evaluate the performance of the state-of-the-art coding agents on this benchmark, employing multiple LLM judges to assess the quality of the generated drafts. Furthermore, we rigorously validate the judgments of the LLM judges against the expert human judgments provided by SMEs. This multilayered evaluation framework enables a comprehensive understanding of both agent performance

and the trustworthiness of LLM-based evaluation methods for EIS drafting tasks.

## 2 Related Work

NEPA documents must uphold scientific integrity and use reliable data[1]. Although written to be understood by the public, drafting environmental documents for NEPA often requires expertise in both legal and scientific domains. Therefore, in the absence of works directly pertaining to NEPA, we look at existing benchmarks for legal and scientific writing (Section 2.1). We describe the LLM-based applications developed to support related drafting tasks in Section 2.2.

### 2.1 Drafting Benchmarks

**Legal drafting benchmarks.** AI research in the legal domain has mostly focused on evaluating the legal reasoning capabilities of LLMs, with benchmarks like LegalBench (Guha et al., 2023), LawBench (Fei et al., 2023), IL-TUR (Joshi et al., 2024), and LexEval (Li et al., 2024). These benchmarks emphasize classification, comprehension, and reasoning but do not address document generation or drafting. Even in common areas like contract law, benchmarks such as CUAD (Hendrycks et al., 2021a) and ContractNLI (Koreeda and Manning, 2021) primarily focus on clause extraction or reading comprehension, offering little support for drafting or information retrieval tasks. Despite drafting's central role in legal practice, only a few benchmarks evaluate the ability of LLMs to generate legally sound and contextually appropriate drafts. Recent efforts like CaseGen (Li et al., 2025) and JUDGE (Su et al., 2025) have begun addressing this gap by introducing case document generation. However, these benchmarks focus on the general legal domain and provide limited attention to specialized areas (e.g., environmental permitting).

**Scientific drafting benchmarks.** Just as legal drafting benchmarks remain scarce despite the complexity, scientific drafting benchmarks are similarly underdeveloped. While scientific LLM evaluation has expanded—ranging from general-purpose benchmarks like MMLU (Wang et al., 2024) and BIG-bench (Srivastava et al., 2022) to domain-specific efforts like MultiMedQA (Zhou et al., 2024), Chem-LLMBench (Guo et al., 2023), and MATH (Hendrycks et al., 2021b)—they primarily assess factual recall or narrow reasoning skills. More comprehensive frameworks like SciEval (Sun et al., 2024), SciAssess (Cai et al., 2024), and SciEx (Dinh et al., 2024) attempt to evaluate deeper capabilities but still fall short in capturing the long-form and interdisciplinary nature of real-world scientific writing.

[1] 42 U.S.C. 4332(2)(D),(E).

### 2.2 LLMs for Legal and Scientific Drafting

LLMs show strong potential for automating legal drafting across diverse domains, generating documents like case reports and contracts that align with legal standards (Lai et al., 2023). GPT-3.5, in particular, excels at drafting complex legal complaints, including securities cryptocurrency class action lawsuits (Trozze et al., 2024). LLMs have also been explored in patent law, where LLMs assist with drafting claims and adapting content to jurisdictional norms (Jiang and Goetz, 2024). They also assist with drafting contracts and modifying clauses to enhance legal precision (Narendra et al., 2024; Savelka and Ashley, 2023). For more complex needs, retrieval-based tools like ACORD specialize in identifying and ranking precedent clauses from large legal corpora, enabling more accurate and context-aware drafting (Wang et al., 2025b).

LLMs are increasingly used for scientific drafting, particularly for generating initial drafts of sections (e.g., *Related Work* and *Introduction*), where they help articulate the background and streamline citation-heavy writing (Morris, 2023). Researchers generally guide the LLMs with outlines or rough notes, enabling them to produce coherent, grammatically polished paragraphs (Castellanos-Gómez, 2023; Pervez and Titus, 2024; Gao et al., 2023; Agarwal et al., 2024). However, limitations persist in technical accuracy, citation contextualization, and hallucination risks, making human oversight quite essential (Garg et al., 2025; Basile et al., 2025).

Coding agents have been adapted to domains such as health care (e.g., EHRAgent (Shi et al., 2024)), data analysis (e.g., DS-Agent (Guo et al., 2024) and DatawiseAgent (You et al., 2025)), education (e.g., CodeEdu (Zhao et al., 2025), CoderAgent (Zhan et al., 2025), and TRAVER (Wang et al., 2025a)), and research (e.g., ResearchCodeAgent (Gandhi et al., 2025)). While specialized agentic systems have proven to be effective for both legal (Suravarjhula et al., 2025; Shea and Yu, 2025) and scientific (Ghafarollahi and Buehler, 2024) domains, building specialized agentic systems is time- and labor-intensive, and even with the incorporation of domain expertise, custom infrastructure, and significant engineering efforts, such systems can still have quality issues. This makes general-purpose agents (e.g., a coding agent) a practical and scalable alternative for many drafting tasks, offering reasonable performance with decreased development costs.

## 3 Building DraftNEPABench

EIS documents typically contain the following fundamental and interdependent components:

**Purpose and need.** The purpose and need section describes the goal of the proposed action and why the action is necessary.

**Proposed action and alternatives.** A proposed action is an internally or externally generated project, plan, or rulemaking requiring an agency decision. Alternatives are reasonable options to achieve the agency's objective

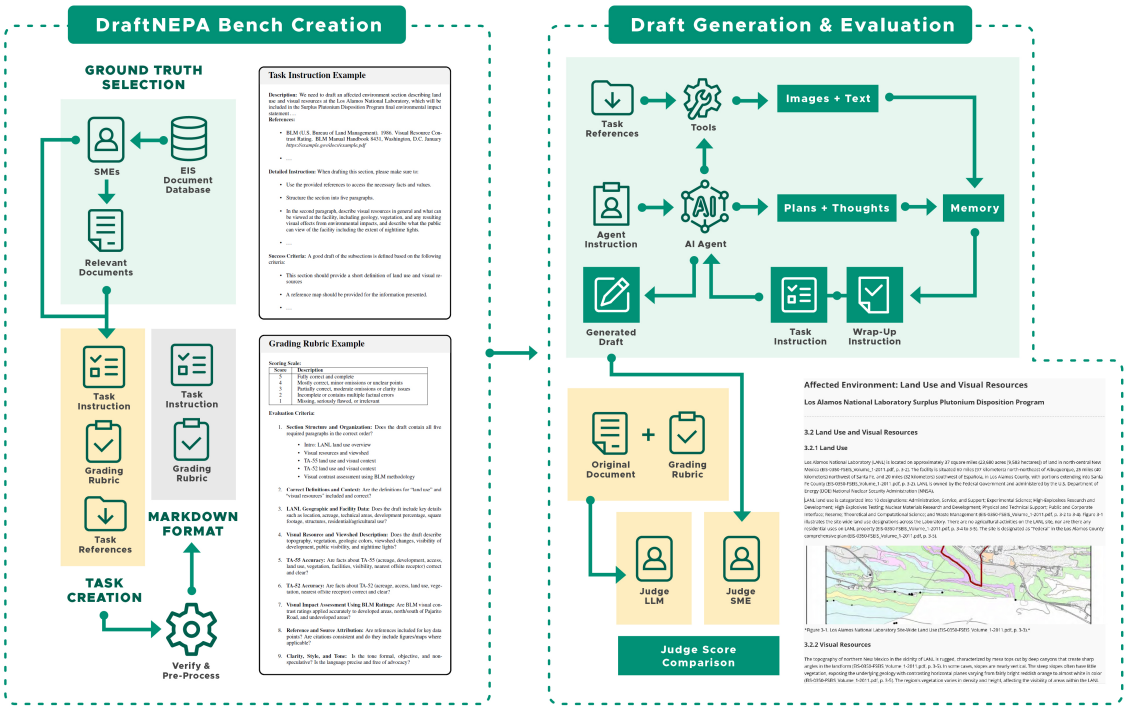


Figure 1: The entire process for DraftNEPABench creation and evaluation.

in place of the proposed action. Alternatives, including a no-action alternative, are determined by the agency, often in coordination with the project sponsor and cooperating agencies, based on technical and economic feasibility and the ability of an alternative to achieve the purpose and need. Each alternative should be described in sufficient detail to analyze environmental impacts, including the physical or process design, the associated activities, and an implementation schedule.

**Affected environment.** The affected environment section provides a description of current environmental conditions in areas that may experience environmental impacts to provide a baseline for assessing the potential impacts of the proposed action. Environmental conditions include both natural and social resources such as ecological, historic and cultural, and economic resources.

**Environmental consequences.** The environmental consequences section describes the foreseeable environmental impacts of the proposed action and alternatives on the affected environment, detailing their scale or significance, duration (temporary or long-term), and mitigation measures. Impacts may be quantitative and qualitative and may require specialized expertise in various disciplines and regulatory processes. Historically, most EIS documents have been prepared by federal agency staff or contractors under agency direction. However, many agencies have developed or are developing procedures for the review of NEPA documents prepared by project sponsors, a practice likely to grow.

Despite the established statutory requirements and familiar section-level conventions, EIS preparation poses recurring challenges in analysis, coordination, and document preparation. Authors must balance scope and detail, ensuring that the purpose and need support reasonable alternatives, while the proposed actions and impact analyses are described with technical specificity. These judgments are drawn from multiple sources, including project sponsors, cooperating agencies, and project-specific studies, which vary in quality and completeness. EIS drafting also requires coordination across interdependent resource sections, consistency with prior documents and evolving regulations, and substantial editorial effort to integrate the acontributions from multiple authors, manage page limits, maintain version control, and ensure accurate citations.

### 3.1 Design Principles and Case Selection

We designed the benchmark to include 102 test cases. First, we began with a pilot pair of cases from two EIS sections and then added 100 cases drawn from five sections, or portions of sections (referred to here as case sections), from 20 different published EISs. We selected case sections from diverse action types (e.g., geothermal development, ecosystem restoration, and disposal of mine waste) led by different agencies (e.g., U.S. Department of Energy). This initial corpus of EISs comprised 18 different lead agencies and 22 different actions.

We chose case sections from the 22 EISs representing the standard components of EIS documents (i.e., purpose and need, alternatives to the proposed federal action, mitigation measures, affected environment, and environmental consequences). Content from the affected environment and environmental consequences sections covered diverse resource areas, with section

counts based on their frequency and depth of analysis. For instance, biological resources and water resources had a higher number of case sections because these resource areas often i) include extensive analysis and ii) are usually further subdivided (e.g., into aquatic and terrestrial for biological resources and into groundwater and surface water for water resources). Fewer cases were drawn from the purpose and need, alternatives, and mitigation measures sections, as these rely heavily on agency and applicant input. For the full list of agencies, selected case sections, and counts, please refer to Appendix B.1.

### 3.2 Subject Matter Experts

SMEs are personnel with specialized expertise in one or more subject areas. For our study, we engaged 19 SMEs with prior EIS drafting experience and determined their level and type of specialized expertise. All SMEs held at least a bachelor's degree, most with advanced degrees, spanning disciplines such as biological sciences, geological sciences, environmental sciences, civil engineering, chemical engineering, radiation health physics, anthropology, and law. This multidisciplinary team ensured a comprehensive knowledge base and intellectual foundation.

### 3.3 Case Creation Procedure

SMEs were asked to choose a case study that matched their expertise and to create two artifacts: *task instruction* and *grading rubric*. The task instruction is task-specific prompt used to generate the draft, while the grading rubric provides a list of scoring criteria and their respective definitions.

#### 3.3.1 Task Instruction

The task instruction comprises four parts. The first part is the *Case Description*, which provides a brief overview of the task (i.e., to draft a subsection related to the specific EIS section) and the case (i.e., the EIS's proposed action). The second part is *References*, which lists the external documents that were used to generate the ground-truth section. References contain either a URL or relative path to the relevant PDF. The agent will need to consult these documents directly to draft the LLM-generated case section. The third part, called *Detailed Instruction*, provides a breakdown of the types of information that the generated draft should include. This part provides information regarding the required structure of the draft, its requested content, and any other instruction necessary for a successful draft. The final part, *Success Criteria*, provides a summary list of prompts reiterating the requirements for a successful LLM-generated case section and ways to make the final outcome better. An example of a task instruction can be seen in Figure 1, as well as in Figure 4 in Appendix A.

#### 3.3.2 Grading Rubric

For each case, the SMEs created a custom grading rubric. These rubrics contain instructions and criteria to judge the generated draft. The instructions provide guidance on how each draft should be scored on a scale of 1 to 5, where 1 indicates poor performance and 5 represents an excellent draft. The evaluation criteria are task specific and vary across different tasks. The rubric also asks the judges to provide the justification for their scores. When the drafts are evaluated, the rubrics are compared with the ground-truth case section with which the generated section is compared.

Since there are more than 350 criteria across all tasks, we generalize them into four key criteria to facilitate aggregate analyses. *Structure* checks that the draft has all required components and is structured as specified. *Clarity* ensures that the draft is objective, formal, and nonspeculative. *Accuracy*[2] examines the case-specific facts and details required in the draft, while *Reference* confirms correct citation and proper formatting. An example of a corresponding task rubric can be seen in Figure 1 and Appendix 6. Section Structure and Organization is categorized as *Structure*; Reference and Source Attribution as *Reference*; Clarity, Style, and Tone as *Clarity*; and the rest as *Accuracy*.

### 3.4 Verification and Preprocessing

Once both artifacts for a task are created, we manually verify that i) the task instruction has all required sections, ii) the external links or the documents used for the references are available, and iii) the grading rubric is complete. Once the artifacts are verified, we convert them into Markdown format, as required for the agent.

### 3.5 Data Statistics

After manual verification, DraftNEPABench comprises 102 case documents. Unlike traditional text-only corpora, these cases are multimodal, combining text, tables, and visual content. These cases originate from 19 distinct government agencies. Each case is, on average, 1,266 words long and has an average of 6 references. This diversity in modality, source, and complexity presents a substantial challenge for LLMs, particularly for long-context reasoning and cross-modal understanding. Full statistics on the number of drafts, the average length, and the number of references are shown in Table 1.

## 4 Experimental Setup

### 4.1 Setup

**Baseline.** We evaluated three state-of-the-art LLMs: GPT-5 (OpenAI, 2025b), Gemini 2.5 Pro (Comanici et al., 2025), and Claude Sonnet 4.5 (Anthropic, 2025b) as the baselines. Because of the context window limit, we employed a retrieval-augmented generation (RAG) setup for the baseline evaluation. First, we split the *Detailed Instruction* section of the tasks into individual queries. To compensate for these individual queries not

[2]It is noted that this is not the standard *accuracy* metric but rather the colloquial term for how *accurate* the draft is.

| Lead Agency | D | L | R |
|---|---|---|---|
| Bureau of Ocean Energy Management | 10 | 1,020 | 7 |
| Federal Aviation Administration | 5 | 1,820 | 8 |
| Federal Energy Regulatory Commission | 10 | 1,485 | 4 |
| Hawaii Department of Land and Natural Resources | 5 | 1,422 | 7 |
| National Nuclear Security Administration | 5 | 1,374 | 6 |
| National Oceanic and Atmospheric Administration | 5 | 1,229 | 8 |
| Tennessee Valley Authority | 5 | 1,255 | 5 |
| U.S. Army Corps of Engineers | 5 | 1,298 | 5 |
| U.S. Bureau of Land Management | 5 | 762 | 6 |
| U.S. Bureau of Reclamation | 5 | 1,434 | 4 |
| U.S. Department of Agriculture | 5 | 938 | 6 |
| U.S. Department of Commerce | 5 | 1,479 | 11 |
| U.S. Department of the Navy | 5 | 1,370 | 4 |
| U.S. Department of Transportation | 5 | 1,259 | 8 |
| U.S. Environmental Protection Agency | 5 | 1,471 | 8 |
| U.S. Fish and Wildlife Service | 5 | 1,237 | 6 |
| U.S. Forest Service | 5 | 778 | 4 |
| U.S. Nuclear Regulatory Commission | 6 | 1,262 | 8 |
| U.S. Department of Energy | 1 | 866 | 6 |

Table 1: Number of drafts (D), average length (L), and average number of references (R) per lead agency.

having the proper context of the task, we used LLMs to rewrite the queries with additional context to create "contextualized queries."[3] Then, using both sets of queries, we retrieved top $k$ relevant passages from the PDF references (converted into a vector database using OpenAI's *text-embedding-3-small* model (OpenAI, 2025c)). Finally, we used the retrieved context and the task instructions to generate the final report. We ran each baseline model $n$ times, resulting in $n$ independent drafts per task.

**Agents.** We evaluated three major coding agents for our drafting tasks—namely, Codex CLI from OpenAI, Gemini CLI from Google, and Claude Code by Anthropic. We use GPT-5 with high reasoning with Codex CLI, Gemini 2.5 Pro with Gemini CLI, and Claude Sonnet 4.5 with Claude Code. We provided unrestricted access to all agents to avoid any human-in-the-loop feedback, and the same instruction prompts were used for all agents to ensure a fair comparison.

**Models for Judge LLM.** To evaluate the generated drafts, we use the three baseline models as judges. We also asked the LLM judges to provide the reason behind their grades for further verification and cross-checking.

### 4.2 Drafting Pipeline

Figure 1 depicts the pipeline for generating the case section draft. The pipeline starts with the *Agent Instruction*, which contains the prompt followed by the agent for all tasks. Based on the instruction, the agent first creates a plan to tackle the task and a scratchpad to log its thoughts and steps during the process. The agent then reads the task-specific prompt called *Task Instruction* and retrieves the relevant text and images from the reference URLs or PDFs in *Task Instruction* using web browsing or PDF parsing tools. Using the information, the agent drafts the required EIS section and then applies a final prompt *Wrap-up Instructions* to verify completeness and generate the final output files in Markdown and HTML formats for subsequent evaluation. This process is repeated for $n$ trials, resulting in $n$ drafts per task.

[3]Please see Appendix B.2 for the exact prompt that we used for this process.

### 4.3 Evaluation Pipeline

We use LLM-as-a-judge to evaluate the generated drafts, as they are highly capable judges that are increasingly aligning with human judgment (Gu et al., 2024).

**Task-specific criteria.** Decomposing evaluation criteria into smaller, interpretable components leads to more consistent and trustworthy judgment (Liu et al., 2024b). Thus, we adopt a fine-grained, task-specific rubric targeting multiple dimensions of quality. We leverage Gemini 2.5 Pro to classify and organize the rubric criteria into the four overarching categories mentioned in Section 3.3.2: *Accuracy*, *Clarity*, *Structure*, and *Reference*. This ensures scalable and consistent evaluation across the tasks while still preserving the task-specific nuances.

**Pointwise scoring.** We use pointwise scoring on a scale of 1–5. A score of 1 indicates major flaws or irrelevant content. A score of 2 represents an incomplete draft with multiple factual errors. A score of 3 is for a partially correct draft with some omissions or a clarity issue, while a score of 4 is for a mostly correct draft with only minor omissions or unclear points. A full score is provided to a fully correct and complete draft.

### 4.4 Human Validation

#### 4.4.1 Validation Procedure

To validate the LLM judges' scores and incorporate SMEs' perspectives, SMEs directly evaluate the generated drafts from each agent. For each task, we generate $k$ drafts and evaluate them using $i$ LLM judges. Since SMEs cannot review all of the drafts because of resource limits, we use LLM scores to select the most promising draft. As outlined in Algorithm 1, we compute each draft's average score per judge, then aggregate these averages across all judges to obtain a final score. We select the highest scoring draft for each task and agent. Those drafts are then independently assessed by SMEs using the corresponding grading rubrics.

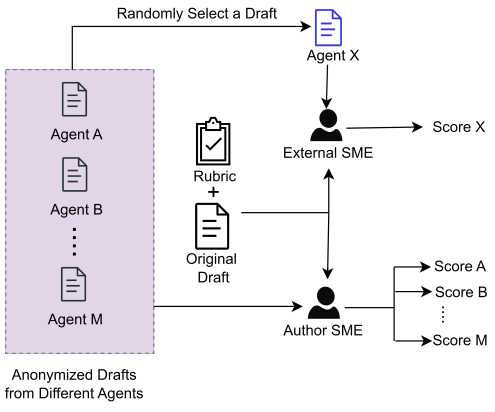


Figure 2: SME validation for generated drafts.

**Algorithm 1:** Selecting the Best Trial per Task and Agent Based on the LLM Judges' Scores

**Input :** $m$: Number of agents
$n$: Number of tasks
$k$: Number of drafts per task
$i$: Number of LLM judges per evaluation
$j$: Number of evaluations per draft
**Output :** $B[a][t]$: Best draft for task $t$ and agent $a$

**for** $a \leftarrow 1$ **to** $m$ **do**
  **for** $t \leftarrow 1$ **to** $n$ **do**
    **for** $d \leftarrow 1$ **to** $k$ **do**
      **for** $l \leftarrow 1$ **to** $i$ **do**
        **for** $e \leftarrow 1$ **to** $j$ **do**
          $S[a][t][d][l][e] \leftarrow$ judge $l$'s score ;
        $J[a][t][d][e] \leftarrow \frac{1}{j}\sum_{l=1}^{j} S[a][t][d][l][e]$ ; `// Avg score over trial`
      $A[a][t][d] \leftarrow \frac{1}{i}\sum_{l=1}^{i} J[a][t][d][l]$ ; `// Avg over judges`
    $B[a][t] \leftarrow \arg\max_d A[a][t][d]$
**return** $B$

We sample $n' = 25$ tasks for SME validation. As in Figure 2, all the drafts are first anonymized to avoid any bias towards a specific agent. Then, each sampled task is validated by the SME who created the corresponding task instruction and grading rubric (i.e., the author SME). While one would typically not use the same SME to grade the task they themselves created, this is necessary in our study as we need SMEs with specific expertise to assess drafts from the specific subsections. To counteract this, we run additional tests to ensure that our author SMEs aren't biased in their evaluations. This effort is detailed in Section 4.4.2.

We then compare the LLM judges' evaluations against the SME scores to assess the alignment and reliability of the LLM judge when replicating expert judgment for NEPA drafting. To quantify the score difference, we use the mean absolute difference (MAD) as in Pires et al. (2025), given as $\text{MAD} = \frac{1}{n'}\sum_{i=1}^{n'}\left|S_{\text{LLM}}^{(i)} - S_{\text{SME}}^{(i)}\right|$. Here, $n'$ is the number of samples graded by both the LLM and SMEs, $S_{\text{LLM}}^{(i)}$ is the score assigned by the LLM for sample $i$, and $S_{\text{SME}}^{(i)}$ is the corresponding score assigned by the SME grader.

#### 4.4.2 Secondary Independent Validation

To ensure reliable and fair evaluation, we use a dual-review validation process. The task author (internal SME), reviews and scores all of the drafts generated by agents, while one of the generated drafts per task is randomly evaluated by an independent (external) SME, as shown in Figure 2. This enables the comparison of the scoring patterns to identify any potential biases that might occur from having authors of the cases evaluate the drafts for these cases. Additionally, a small number of cases were evaluated by external SMEs as an additional robustness check. By analyzing the agreement levels and score distributions across both groups, we can assess the objectivity of the author SME's judgments and take corrective actions if discrepancies are found, thereby strengthening the overall quality. To quantify the difference between the scores, we use the MAD metric.

## 5 Results

### 5.1 Performance Trends

Table 2 reports the performance of the coding agents on DraftNEPABench, with different LLM judges and evaluation critera. For each task, scores are averaged over five runs, and each run is evaluated five times by each judge. The first three models (GPT-5, Gemini 2.5 Pro, and Claude Sonnet 4.5) are RAG baselines, and the latter three (Codex, Gemini CLI, and Claude Code) are coding agents. Boldfaced entries within a judge block highlight the strongest RAG baseline and coding agent for each criterion. Across all baselines, GPT-5 is the strongest model in almost all categories, but coding agents substantially outperform all baselines. Claude Code achieves the highest overall scores for every judge ($3.44 \pm 0.59$ to $4.04 \pm 0.70$). In contrast, GPT-5 receives a score of only $2.52 \pm 0.61$ from its own judge, with lower scores from the Gemini and Sonnet judges. Codex CLI also performs consistently well, while Gemini CLI shows moderate but still clearly superior performance relative to all baselines. These results indicate a significant performance gap between baseline models and coding agents on complex drafting tasks.

**Performance by judge.** Results are consistent across judges despite score variation. For the GPT-5 judge, GPT-5 is the strongest RAG baseline, yet its score ($2.52 \pm 0.61$) remains below the weakest coding agent (Gemini CLI at $2.74 \pm 0.57$). Similar patterns are observed for the Gemini 2.5 Pro and Claude Sonnet 4.5 judges, where Claude Code again achieves the highest scores and Codex CLI consistently follows. Consistent rankings across judges suggests that the observed performance differences reflect substantive capabilities rather than judge-specific preferences.

**Accuracy.** We break down the scores into four evaluation criteria: accuracy, clarity, reference, and structure. Accuracy, the most critical dimension for assessing draft quality, remains relatively low across all models and agents, even for the best-performing systems. While coding agents outperform RAG baselines for all judges, accuracy remains well below the upper bound. For example, Claude Code receives the highest accuracy score from the Gemini 2.5 Pro judge ($3.71 \pm 0.99$), while the strongest RAG baseline, GPT-5, is $2.01 \pm 0.77$ from the GPT-5 judge. These results indicate that the agent-based approaches improve accuracy, but challenges in factual correctness and task alignment still persist.

**Clarity, reference, and structure.** Beyond accuracy, the other criteria provide complementary insights. Clar-

Table 2: Overall performance across different evaluation criteria. The best-performing model/agent is in boldface.

| Judge | Model/Agent | Accuracy | Clarity | Reference | Structure | Overall Score |
|---|---|---|---|---|---|---|
| GPT-5 | GPT-5 | **2.01 ± 0.77** | **4.04 ± 0.90** | **3.05 ± 0.99** | **1.94 ± 1.04** | **2.52 ± 0.61** |
| | Gemini 2.5 Pro | 1.66 ± 0.57 | 3.87 ± 0.88 | 2.42 ± 0.71 | 1.76 ± 0.91 | 2.21 ± 0.50 |
| | Claude Sonnet 4.5 | 1.80 ± 0.74 | 3.94 ± 0.91 | 2.75 ± 0.92 | 1.74 ± 0.98 | 2.33 ± 0.59 |
| | Codex CLI | 2.70 ± 0.68 | 3.90 ± 0.49 | 2.66 ± 0.68 | 3.64 ± 0.96 | 3.04 ± 0.53 |
| | Gemini CLI | 2.40 ± 0.70 | 3.60 ± 0.54 | 2.10 ± 0.67 | 3.43 ± 0.99 | 2.74 ± 0.57 |
| | Claude Code | **3.11 ± 0.84** | **4.23 ± 0.55** | **3.05 ± 0.71** | **3.91 ± 0.91** | **3.44 ± 0.59** |
| Gemini 2.5 Pro | GPT-5 | **1.89 ± 0.91** | **3.95 ± 1.06** | **3.16 ± 1.35** | 1.51 ± 0.87 | **2.43 ± 0.90** |
| | Gemini 2.5 Pro | 1.70 ± 0.68 | 3.89 ± 1.15 | 2.85 ± 1.23 | **1.55 ± 0.85** | 2.28 ± 0.76 |
| | Claude Sonnet 4.5 | 1.70 ± 0.84 | 3.73 ± 1.22 | 2.75 ± 1.23 | 1.35 ± 0.69 | 2.20 ± 0.85 |
| | Codex CLI | 3.24 ± 0.95 | 4.56 ± 0.47 | 3.71 ± 1.14 | 3.97 ± 0.96 | 3.65 ± 0.73 |
| | Gemini CLI | 2.81 ± 0.87 | 4.12 ± 0.65 | 2.58 ± 1.00 | 3.68 ± 0.99 | 3.16 ± 0.68 |
| | Claude Code | **3.71 ± 0.99** | **4.74 ± 0.41** | **4.00 ± 0.99** | **4.28 ± 0.91** | **4.04 ± 0.70** |
| Claude Sonnet 4.5 | GPT-5 | **1.90 ± 0.82** | 3.46 ± 0.78 | **3.03 ± 1.12** | **1.47 ± 0.53** | **2.27 ± 0.71** |
| | Gemini 2.5 Pro | 1.72 ± 0.74 | **3.47 ± 0.81** | 2.66 ± 0.83 | **1.47 ± 0.57** | 2.15 ± 0.63 |
| | Claude Sonnet 4.5 | 1.69 ± 0.73 | 3.44 ± 0.86 | 2.69 ± 0.94 | 1.34 ± 0.49 | 2.13 ± 0.67 |
| | Codex CLI | 2.83 ± 0.68 | 3.84 ± 0.45 | 3.43 ± 0.95 | 2.92 ± 0.89 | 3.10 ± 0.59 |
| | Gemini CLI | 2.60 ± 0.66 | 3.63 ± 0.47 | 2.46 ± 0.78 | 2.83 ± 0.87 | 2.80 ± 0.57 |
| | Claude Code | **3.61 ± 0.92** | **4.35 ± 0.54** | **3.94 ± 0.91** | **3.43 ± 1.08** | **3.76 ± 0.79** |
| Aggregated | GPT-5 | **1.90 ± 0.83** | **4.11 ± 0.75** | **3.04 ± 1.19** | **1.63 ± 0.88** | **2.41 ± 0.76** |
| | Gemini 2.5 Pro | 1.68 ± 0.66 | 4.10 ± 0.72 | 2.61 ± 0.98 | 1.63 ± 0.83 | 2.21 ± 0.64 |
| | Claude Sonnet 4.5 | 1.72 ± 0.75 | 4.02 ± 0.89 | 2.71 ± 1.06 | 1.46 ± 0.75 | 2.22 ± 0.71 |
| | Codex CLI | 2.92 ± 0.81 | 4.20 ± 0.53 | 3.23 ± 1.05 | 3.52 ± 1.05 | 3.27 ± 0.68 |
| | Gemini CLI | 2.62 ± 0.75 | 3.90 ± 0.59 | 2.37 ± 0.85 | 3.38 ± 1.00 | 2.90 ± 0.64 |
| | Claude Code | **3.47 ± 0.94** | **4.54 ± 0.49** | **3.64 ± 1.00** | **3.88 ± 1.02** | **3.75 ± 0.74** |

ity scores are consistently high for both agents and baselines, suggesting that most systems can produce clear and readable drafts. In some cases, the RAG baselines match the agents' performance on clarity. Reference usage exhibits greater variability, with coding agents generally citing relevant information more appropriately. Structure scores further highlight the advantages of agent-based approaches, as agents like Claude Code and Codex CLI consistently produce drafts satisfying the structural requirements of the task. Overall, agentic planning and iterative refinement meaningfully improve reference use and document organization, while factual accuracy remains the primary bottleneck for reliable end-to-end EIS drafting.

### 5.2 Generation Efficiency for Coding Agents

To assess the coding agents' efficiency, we measure the average task completion time (in minutes) and the average total token consumption (in millions of tokens), as summarized in Table 3. Codex CLI is the most efficient, completing tasks in an average of $7.69 \pm 4.86$ minutes while also consuming the fewest tokens ($1.27 \pm 0.81$ million tokens). Claude Code requires more time and tokens on average ($12.39 \pm 4.75$ minutes and $2.40 \pm 0.99$ million tokens) but remains stable. In contrast, Gemini CLI shows high variability in both completion time and token usage ($13.66 \pm 27.14$ minutes and $6.60 \pm 11.77$ million tokens), indicating inconsistent performance and the occasional outlier.

### 5.3 SME Validation

We sampled 25 cases for SME validation to assess the quality of the agent-generated drafts, following the procedure described in Section 4.4. Table 4 reports the average scores assigned by the LLM judges and Author SMEs across the selected tasks. Claude Code received the highest scores from all LLM judges as well as Author SMEs, while Codex CLI generally ranked second and Gemini CLI received lower scores.

Table 3: Average time taken and tokens used by the agents.

| Agent/Model | Time (Minutes) | Token (Millions) |
|---|---|---|
| Codex CLI | 7.69 ± 4.86 | 1.27 ± 0.81 |
| Gemini CLI | 13.66 ± 27.14 | 6.60 ± 11.77 |
| Claude Code | 12.39 ± 4.75 | 2.40 ± 0.99 |

Table 4: Scores given by all LLM judges and Author SMEs.

| Agent/Model | GPT-5 | Gemini 2.5 Pro | Claude Sonnet 4.5 | Author SME |
|---|---|---|---|---|
| Codex CLI | 3.05 ± 0.48 | 3.65 ± 0.69 | 3.06 ± 0.41 | 3.19 ± 0.65 |
| Gemini CLI | 2.73 ± 0.56 | 3.16 ± 0.72 | 2.75 ± 0.57 | 3.43 ± 0.80 |
| Claude Code | **3.41 ± 0.48** | **4.03 ± 0.53** | **3.71 ± 0.68** | **4.10 ± 0.59** |

This overall ranking is consistent across evaluation criteria. Detailed criterion-level Author SME scores are provided in Appendix E. In addition, the alignment between LLM judges and SMEs is quantified in Table 5, which shows moderate to strong agreement in overall system ranking. Taken together, these results indicate that SME assessments are consistent with automated evaluation outcomes at an aggregate level.

### 5.4 SME Alignment

To assess the potential bias from Author SMEs, we compared their scores with those assigned by an Ex-

Table 5: Overall MADs for all LLM judges.

| Agent/Model | GPT-5 | Gemini 2.5 Pro | Claude Sonnet 4.5 |
|---|---|---|---|
| Codex CLI | 0.667 | 0.825 | 0.559 |
| Gemini CLI | 0.839 | 0.713 | 0.854 |
| Claude Code | 0.859 | 0.542 | 0.785 |

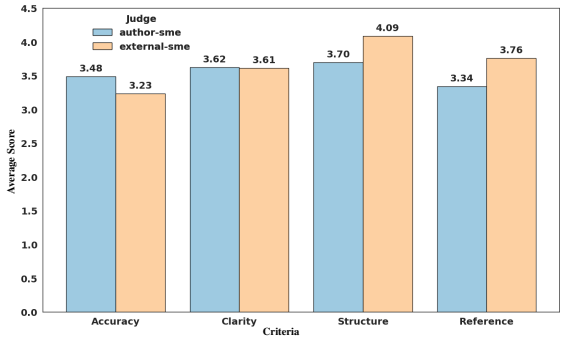


Figure 3: Average criterion-level scores for sampled drafts, comparing Author and External SME evaluations.

ternal SME, using the MAD (Section 4.4.2). Besides the 25 randomly selected drafts described in Section 4, 8 additional drafts from the 25 cases were also evaluated by External SMEs, resulting in a total of 33 cases. We observe moderate to strong agreement across agents. Claude Code exhibits the highest alignment, while Codex CLI shows the lowest alignment. Overall, Author SME evaluations closely track those of an external evaluator and do not exhibit systematic inflation. Figure 3 compares the average scores across evaluation criteria. External SMEs tend to score accuracy more strictly, while Author SMEs are comparatively stricter on structure and reference use; clarity scores are closely aligned. These differences reflect the variation in evaluative emphasis rather than disagreement in overall assessment, supporting the validity of the human evaluation protocol.

## 6 Discussion

**Agentic systems improve long-form drafting but do not resolve accuracy limitations.** We observe that coding agents consistently outperform standalone RAG baselines on long-form, structured drafting tasks, especially organization, reference handling, and overall usability, highlighting the value of agentic planning and iterative refinement. However, accuracy remains a persistent limitation across all models, including the strongest agentic systems. SME feedback highlights recurring issues with factual specificity, data placement, and citation correctness, indicating that improved drafting workflows alone do not guarantee factual grounding.

**Performance varies systematically by model and section type.** We find that qualitative SME assessments reveal consistent behavioral differences across agent implementations, reflecting trade-offs between consistency, specificity, and verbosity. In addition, model performance varies substantially by section type. Less analytically complex sections are generally more amenable to agent-assisted drafting than sections requiring tighter data integration or analytical judgment. These patterns suggest that both model design and task structure play a significant role in determining the output quality. We provide a detailed model-specific analysis in Appendix E.1, and a section-level discussion in Appendix E.2.

**Human and automated evaluations provide complementary signals.** Comparisons across LLM judges, Author SMEs, and External SMEs show moderate to strong agreement in overall system ranking, with some differences in emphasis. We find that the alignment between Author and External SMEs indicates that author involvement as evaluators does not introduce substantial bias. Overall, automated evaluation can provide useful comparative signals for long-form generation but remains complementary to expert human review. Additional discussion of the evaluation alignment and bias considerations is provided in Appendix E.3.

## 7 Conclusion

In this work, we introduced DraftNEPABench, a benchmark for evaluating the ability of LLMs and agent-based systems to draft sections of EISs for NEPA-related tasks. The benchmark captures the key complexities of environmental impact drafting. Our evaluation combines scalable automatic assessment using multiple LLM judges with complementary SME assessments, providing insight into both model performance and limitations. Although commercial off-the-shelf agents were evaluated, they demonstrated promising performance, with Claude Code consistently favored by both automated and human evaluators. However, persistent challenges related to factual accuracy and data integration highlight the continued need for human oversight. Overall, our findings suggest that current LLM-based systems can aid NEPA drafting workflows while leaving clear room for improvement through targeted adaptation and improved grounding.

## 8 Limitations

While DraftNEPABench provides a useful foundation for evaluating LLMs on NEPA-related drafting tasks, it has several limitations. In some cases, reference drafts contained missing citations or outdated regulatory information, which may have affected accuracy judgments. Because LLM agents had access to more recent information through web searches, their outputs sometimes diverged from the ground truth while remaining practically relevant, leading to stricter grading by LLM judges (Figure 8). SMEs were instructed to accept newer information where appropriate.

In addition, the evaluation criteria capture broad aspects of draft quality but do not fully reflect legal defensibility or project-specific nuance. Expanding the benchmark with more up-to-date references, additional section types, and alternative evaluation schemes would

enable a more detailed assessment of model behavior, particularly with respect to grounding, citation practices, and section-level variability.

Finally, since we started this work, newer and more capable models have been released from all three frontier models, and improvements have been made to the coding agents. Therefore, the results using these coding agents might be different with the new models and capabilities in place.

## 9 Ethical Considerations

All human-involved research in the project, including the SME involvement, was ruled as non-human subject research (non-HSR) by the Institutional Review Board (IRB). All SMEs were compensated for their time at their regular hourly rates.

## Acknowledgments

This work was supported by the Office of Policy, U.S. Department of Energy, and Pacific Northwest National Laboratory (PNNL), which is operated by Battelle Memorial Institute for the U.S. Department of Energy under Contract DE-AC05–76RLO1830. This paper has been cleared by PNNL for public release as PNNL-SA-219252.

## References

Shubham Agarwal, Gaurav Sahu, Abhay Puri, Issam Hadj Laradji, Krishnamurthy DJ Dvijotham, Jason Stanley, Laurent Charlin, and Christopher Pal. 2024. LitLLMs, LLMs for literature review: Are we there yet? *Transactions on Machine Learning Research*, 2025.

Anthropic. 2025a. Claude Code. `https://claude.com/product/claude-code`.

Anthropic. 2025b. Claude Sonnet 4.5. `https://www.anthropic.com/claude/sonnet`.

Christian Basile, Stefan D. Anker, and Gianluigi Savarese. 2025. Large language models to write scientific manuscripts: To be considered but not trusted. *Global Cardiology*, 3.

Hengxing Cai, Xiaochen Cai, Junhan Chang, Sihang Li, Lin Yao, Changxin Wang, Zhifeng Gao, Hongshuai Wang, Yongge Li, Mujie Lin, Shuwen Yang, Jiankun Wang, Yuqi Yin, Yaqi Li, Linfeng Zhang, and Guolin Ke. 2024. SciAssess: Benchmarking LLM proficiency in scientific literature analysis. *arXiv preprint, arXiv:2403.01976*.

Andrés Castellanos-Gómez. 2023. Good practices for scientific article writing with ChatGPT and other artificial intelligence language models. *Nanomanufacturing*, 3:135–138.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint, arXiv:2507.06261*.

Tu Anh Dinh, Carlos Mullov, Leonard Bärmann, Zhaolin Li, Danni Liu, Simon Reiß, Jueun Lee, Nathan Lerzer, Fabian Ternava, Jianfeng Gao, and 1 others. 2024. SciEx: Benchmarking large language models on scientific exams with human expert grading and automatic grading. *arXiv preprint, arXiv:2406.10421*.

Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Songyang Zhang, Kai Chen, Zongwen Shen, and Jidong Ge. 2023. LawBench: Benchmarking legal knowledge of large language models. *arXiv preprint, arXiv:2309.16289*.

Shubham Gandhi, Dhruv Shah, Manasi S. Patwardhan, Lovekesh Vig, and Gautam M. Shroff. 2025. ResearchCodeAgent: An LLM multi-agent system for automated codification of research methodologies. *arXiv preprint, arXiv:2504.20117*.

Fan Gao, Hang Jiang, Rui Yang, Qingcheng Zeng, Jinghui Lu, Moritz Blum, Dairui Liu, Tianwei She, Yuang Jiang, and Irene Li. 2023. Evaluating large language models on Wikipedia-style survey generation. In *Annual Meeting of the Association for Computational Linguistics*.

Krishna Garg, Firoz Shaik, Sambaran Bandyopadhyay, and Cornelia Caragea. 2025. Let's use ChatGPT to write our paper! Benchmarking LLMs to write the introduction of a research paper. *arXiv preprint, arXiv:2508.14273*.

Alireza Ghafarollahi and Markus J. Buehler. 2024. SciAgents: Automating scientific discovery through bioinspired multi-agent intelligent graph reasoning. *Advanced Materials*, 37:2413523.

Google. 2025. Gemini CLI. `https://github.com/google-gemini/gemini-cli`.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Yuanzhuo Wang, and Jian Guo. 2024. A survey on LLM-as-a-Judge. *arXiv preprint, arXiv:2411.15594*.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin M. K. Peters, Brandon Waldon, Daniel N. Rockmore, Diego A. Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, and 21 others. 2023. LegalBench: A collaboratively built benchmark for measuring legal reasoning in large language models. *arXiv preprint, arXiv:2308.11462*.

Siyuan Guo, Cheng Deng, Ying Wen, Hechang Chen, Yi Chang, and Jun Wang. 2024. DS-Agent: Automated data science by empowering large language models with case-based reasoning. *arXiv preprint, arXiv:2402.17453*.

Taicheng Guo, Kehan Guo, Bozhao Nan, Zhenwen Liang, Zhichun Guo, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2023. What indeed can GPT models do in chemistry? A comprehensive benchmark on eight tasks. *arXiv preprint, arXiv:2305.18365v1*.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021a. CUAD: An expert-annotated NLP dataset for legal contract review. *arXiv preprint, arXiv:2103.06268*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. *NeurIPS*.

Lekang Jiang and Stephan Goetz. 2024. Natural language processing in the patent domain: A survey. *Artificial Intelligence Review*, 58:214.

Abhinav Joshi, Shounak Paul, Akshat Sharma, Pawan Goyal, Saptarshi Ghosh, and Ashutosh Modi. 2024. IL-TUR: Benchmark for Indian legal text understanding and reasoning. *arXiv preprint, arXiv:2407.05399*.

Yuta Koreeda and Christopher D Manning. 2021. ContractNLI: A dataset for document-level natural language inference for contracts. *arXiv preprint, arXiv:2110.01799*.

Jinqi Lai, Wensheng Gan, Jiayang Wu, Zhenlian Qi, and Philip S. Yu. 2023. Large language models in law: A survey. *arXiv preprint, arXiv:2312.03718*.

Haitao Li, You Chen, Qingyao Ai, Yueyue Wu, Ruizhe Zhang, and Yiqun Liu. 2024. LexEval: A comprehensive Chinese legal benchmark for evaluating large language models. *Advances in Neural Information Processing Systems*, 37:25061–25094.

Haitao Li, Jiaying Ye, Yiran Hu, Jia Chen, Qingyao Ai, Yueyue Wu, Junjie Chen, Yifan Chen, Cheng Luo, Quan Zhou, and 1 others. 2025. CaseGen: A benchmark for multi-stage legal case documents generation. *arXiv preprint, arXiv:2502.17943*.

Junwei Liu, Kaixin Wang, Yixuan Chen, Xin Peng, Zhenpeng Chen, Lingming Zhang, and Yiling Lou. 2024a. Large language model-based agents for software engineering: A survey. *arXiv preprint, arXiv:2409.02977*.

Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. 2024b. HD-Eval: Aligning large language model evaluators through hierarchical criteria decomposition. *arXiv preprint, arXiv:2402.15754*.

Meredith Ringel Morris. 2023. Scientists' perspectives on the potential for generative AI in their fields. *arXiv preprint, arXiv:2304.01420*.

Savinay Narendra, Kaushal Shetty, and Adwait Ratnaparkhi. 2024. Enhancing contract negotiations with LLM-based legal document comparison. *Proceedings of the Natural Legal Language Processing Workshop 2024*.

OpenAI. 2025a. Codex CLI. https://developers.openai.com/codex/cli.

OpenAI. 2025b. GPT-5. https://openai.com/gpt-5.

OpenAI. 2025c. text-embedding-3-small. https://platform.openai.com/docs/models/text-embedding-3-small.

Naseela Pervez and Alexander J. Titus. 2024. Inclusivity in large language models: Personality traits and gender bias in scientific abstracts. *arXiv preprint, arXiv:2406.19497*.

Ramon Pires, Roseval Malaquias Junior, and Rodrigo Nogueira. 2025. Automatic legal writing evaluation of LLMs. *arXiv preprint, arXiv:2504.21202*.

Cheng Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2023. ChatDev: Communicative agents for software development. In *Annual Meeting of the Association for Computational Linguistics*.

Jaromir Savelka and Kevin D. Ashley. 2023. The unreasonable effectiveness of large language models in zero-shot semantic annotation of legal texts. *Frontiers in Artificial Intelligence*, 6.

Ryan Shea and Zhou Yu. 2025. AutoSpec: An agentic framework for automatically drafting patent specification. *arXiv preprint, arXiv:2509.19640*.

Wenqi Shi, Ran Xu, Yuchen Zhuang, Yue Yu, Jieyu Zhang, Hang Wu, Yuanda Zhu, Joyce C. Ho, Carl Yang, and M. D. Wang. 2024. EHRAgent: Code empowers large language models for few-shot complex tabular reasoning on electronic health records. *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, 2024:22315–22339.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, and 431 others. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint, arXiv:2206.04615*.

Weihang Su, Baoqing Yue, Qingyao Ai, Yiran Hu, Jiaqi Li, Changyue Wang, Kaiyuan Zhang, Yueyue Wu, and Yiqun Liu. 2025. JuDGE: Benchmarking judgment document generation for Chinese legal system. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3573–3583.

Liangtai Sun, Yang Han, Zihan Zhao, Da Ma, Zhennan Shen, Baocai Chen, Lu Chen, and Kai Yu. 2024. SciEval: A multi-level large language model evaluation

benchmark for scientific research. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19053–19061.

Amulya Suravarjhula, Rashi Chandrashekhar Agrawal, Sakshi Jayesh Patel, and Rahul Gupta. 2025. Retrieval-augmented multi-agent system for rapid statement of work generation. *arXiv preprint, arXiv:2508.07569*.

Arianna Trozze, Toby Davies, and Bennett Kleinberg. 2024. Large language models in cryptocurrency securities cases: Can a GPT model meaningfully assist lawyers? *Artificial Intelligence and Law*, 33:691–737.

Jian Wang, Yinpei Dai, Yichi Zhang, Ziqiao Ma, Wenjie Li, and Joyce Chai. 2025a. Training turn-by-turn verifiers for dialogue tutoring agents: The curious case of LLMs as your coding tutors. In *Annual Meeting of the Association for Computational Linguistics*.

Steven H. Wang, Maksim Zubkov, Kexin Fan, Sarah Harrell, Yuyang Sun, Wei Chen, Andreas Lindhardt Plesner, and Roger Wattenhofer. 2025b. ACORD: An expert-annotated retrieval dataset for legal contract drafting. *arXiv preprint, arXiv:2501.06582*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290.

Yue Wu, Yewen Fan, So Yeon Min, Shrimai Prabhumoye, Stephen McAleer, Yonatan Bisk, Ruslan Salakhutdinov, Yuanzhi Li, and Tom Mitchell. 2024. AgentKit: Structured LLM reasoning with dynamic graphs. *arXiv preprint, arXiv:2404.11483*.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Adriano Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024. SWE-agent: Agent-computer interfaces enable automated software engineering. *arXiv preprint, arXiv:2405.15793*.

Ziming You, Yumiao Zhang, Dexuan Xu, Yiwei Lou, Yandong Yan, Wei Wang, Huaming Zhang, and Yu Huang. 2025. DatawiseAgent: A notebook-centric LLM agent framework for automated data science. *arXiv preprint, arXiv:2503.07044*.

Yi Zhan, Qi Liu, Weibo Gao, Zheng Zhang, Tianfu Wang, Shuanghong Shen, Junyu Lu, and Zhenya Huang. 2025. CoderAgent: Simulating student behavior for personalized programming learning with large language models. In *International Joint Conference on Artificial Intelligence*.

Kechi Zhang, Jia Li, Ge Li, Xianjie Shi, and Zhi Jin. 2024a. CodeAgent: Enhancing code generation with tool-integrated agent systems for real-world repo-level coding challenges. In *Annual Meeting of the Association for Computational Linguistics*.

Yuntong Zhang, Haifeng Ruan, Zhiyu Fan, and Abhik Roychoudhury. 2024b. AutoCodeRover: Autonomous program improvement. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 1592–1604.

Jianing Zhao, Peng Gao, Jiannong Cao, Zhiyuan Wen, Chen Chen, Jianing Yin, Ruosong Yang, and Bo Yuan. 2025. CodeEdu: A multi-agent collaborative platform for personalized coding education. *arXiv preprint, arXiv:2507.13814*.

Yuxuan Zhou, Xien Liu, Chen Ning, and Ji Wu. 2024. MultifacetEval: Multifaceted evaluation to probe LLMs in mastering medical knowledge. *arXiv preprint, arXiv:2406.02919*.

# A Example Task and Rubric

## A.1 Example Task Instruction

### Task Instruction Example

**Description:** We need to draft an affected environment section describing land use and visual resources at Los Alamos National Laboratory, which will be included in the Surplus Plutonium Disposition Program final environmental impact statement . . .
**References:**

- BLM (U.S. Bureau of Land Management). 1986. Visual Resource Contrast Rating. BLM Manual Handbook 8431, Washington, D.C. January *https://example.gov/docs/example.pdf*
- ...

**Detailed Instructions:** When drafting this section, please make sure to

- use the provided references to access the necessary facts and values.
- structure the section into five paragraphs.
- in the second paragraph, describe the visual resources in general and what can be viewed at the facility, including geology, vegetation, and any resulting visual effects from environmental impacts, and describe what the public can view of the facility, including the extent of nighttime lights.
- . . .

**Success Criteria:** A good draft of the subsections is defined based on the following criteria:

- This section should provide a short definition of land use and visual resources.
- A reference map should be provided for the information presented.
- . . .

Figure 4: An example of the task instruction for a case.

## A.2 Agent Instructions

### Simplified Agent Instructions

**Outcomes:**

- A high-quality final report
- Working artifacts like Plan, Scratchpad, images

**Steps:**

- **Step 1:** Set up and read task instruction
- **Step 2:** Plan
- **Step 3:** Convert reference PDFs to text
- **Step 4:** Explore references
- **Step 5:** Extract figures
- **Step 6:** Draft report
- **Step 7:** Quality and completeness check
- **Step 8:** Finalize and organize

Figure 5: Steps followed by the coding agents to complete a DraftNEPABench task.

## A.3 Example Rubric

### Grading Rubric Example

**Scoring Scale:**

| Score | Description |
|---|---|
| 5 | Fully correct and complete |
| 4 | Mostly correct, minor omissions or unclear points |
| 3 | Partially correct, moderate omissions or clarity issues |
| 2 | Incomplete or contains multiple factual errors |
| 1 | Missing, seriously flawed, or irrelevant |

**Evaluation Criteria:**

1. **Section Structure and Organization:** Does the draft contain all five required paragraphs in the correct order?
   - Intro: LANL land use overview
   - Visual resources and viewshed
   - TA-55 land use and visual context
   - TA-52 land use and visual context
   - Visual contrast assessment using BLM methodology
2. **Correct Definitions and Context:** Are the definitions for "land use" and "visual resources" included and correct?
3. **LANL Geographic and Facility Data:** Does the draft include key details such as location, acreage, technical areas, development percentage, square footage, structures, and residential/agricultural use?
4. **Visual Resource and Viewshed Description:** Does the draft describe topography, vegetation, geologic colors, viewshed changes, visibility of development, public visibility, and nighttime lights?
5. **TA-55 Accuracy:** Are facts about TA-55 (acreage, development, access, land use, vegetation, facilities, visibility, nearest offsite receptor) correct and clear?
6. **TA-52 Accuracy:** Are facts about TA-52 (acreage, access, land use, vegetation, nearest offsite receptor) correct and clear?
7. **Visual Impact Assessment Using BLM Ratings:** Are BLM visual contrast ratings applied accurately to developed areas, north/south of Pajarito Road, and undeveloped areas?
8. **Reference and Source Attribution:** Are references included for key data points? Are citations consistent, and do they include figures/maps where applicable?
9. **Clarity, Style, and Tone:** Is the tone formal, objective, and nonspeculative? Is the language precise and free of advocacy?

Figure 6: Rubric and scoring sheet for evaluating the LANL land use and visual resources section.

# B Additional Details on Case Creation

## B.1 Details About Case Sections

The dataset comprises 102 selected cases spanning a range of National Environmental Policy Act (NEPA) case sections. Table 6 reports the number of cases associated with each section. The distribution shows coverage across core environmental resource areas (e.g., biological resources, water resources, and air quality), as well as socioeconomic, infrastructure, and planning-related sections, with some sections represented more frequently than others.

## B.2 Baseline Evaluation Query Creation

We used the following prompt to create the "contextualized queries" for the baseline evaluation from the queries extracted from the task instructions:

```
You are a Subject Matter Expert (SME) in drafting
Environmental Impact Statements (EIS).

You have been provided with a task instruction intended to
guide the drafting of an EIS section. However, the
```

Table 6: Selected case sections.

| Case Section | Count | Case Section | Count |
|---|---|---|---|
| Alternatives | 1 | Purpose and need | 3 |
| Air quality | 6 | Socioeconomic | 7 |
| Biological resources | 14 | Recreation and open space | 5 |
| Cultural and historic resources | 7 | Transportation and traffic | 6 |
| Geology and soils | 7 | Utilities and infrastructure | 3 |
| Land use and zoning | 7 | Waste management | 6 |
| Mitigation measures | 3 | Water resources | 9 |
| Noise and vibration | 7 | Visual resources | 5 |
| Public health and safety | 5 | Important species and habitats | 1 |

```
instruction is not currently well-suited for
retrieval-augmented generation (RAG), which relies on
precise and contextualized queries to retrieve
relevant information from a vector database.

Your objective is to rewrite or restructure the instruction
to make it more effective for information retrieval. You may:
- Rephrase the instruction for clarity and specificity.
- Break it down into smaller, more focused sub-instructions
if that improves retrieval accuracy.
- Ensure the reformulated instruction is self-contained and
  contextually rich, even though you only have access to the
  original instruction.

Here is the original task instruction:
{task_instruction}

Return only the improved instruction(s), separated by
newlines, optimized for retrieval. Do not include any
additional commentary or explanation. Limit the output
to a maximum of 20 instructions.
```

### B.3 Baseline Draft Generation Prompt

Once we have retrieved the passages, we used the following prompt to generate the draft from the baseline models:

```
You are a subject matter expert at drafting Environmental
Impact Statement for National Environmental Policy Act(NEPA).
Using given task instruction and context from references,
draft a section that meets all the success criteria.

Context from references: {context}

Here is the task instruction: {task_instruction}

If context from the refrences is not provided use the urls
provided in the reference section of task instruction.
Check if the success criteria is met but do not include
in the final draft. Do not forget to add references.

Strictly return only the generated draft suitable for saving
as markdown with all heading and sections.
```

## C Performance by Estimated Task Difficulty

To assess whether task difficulty correlates with model and agent performance, we analyze results across four difficulty levels assigned by subject matter experts (SMEs). Importantly, these difficulty ratings were designed to reflect *anticipated human drafting effort* rather than the intrinsic complexity for language models or agents. As shown below, this distinction is critical: higher human-rated difficulty does not correspond to lower agent performance.

### C.1 Difficulty Rating Framework

Each case section was assigned a difficulty rating ranging from 1 (simple) to 4 (complex) based on the SMEs' assessment of the anticipated effort required to draft that section. Ratings were determined by considering the section length, degree of technical detail, and environmental impact statement (EIS) section type (e.g., affected environment, environmental consequences, purpose and need, alternatives, and mitigating measures).

Table 7 summarizes the four-tier difficulty framework, and Table 8 shows the distribution of case sections across difficulty levels. The resulting dataset intentionally spans a broad range of drafting complexity to enable the analysis of performance trends across task difficulty.

### C.2 Dataset Characteristics by Difficulty

Table 9 reports the descriptive statistics for the drafts at each difficulty level, including the number of drafts, average draft length, and average number of references per draft.

**Draft length and reference statistics.** As expected, higher difficulty ratings correspond to longer drafts, with difficulty 4 sections being substantially longer on average than difficulty 1 sections. Difficulty 2 sections include the largest number of references, reflecting resource-intensive affected environment sections that require the synthesis of multiple data sources. These trends indicate that the difficulty ratings capture meaningful differences in the anticipated human effort and content complexity.

### C.3 Performance Across Difficulty Levels

Figure 7 and Table 10 present the average scores for the baseline models and coding agents across difficulty levels, aggregated by judge.

**Observed performance trends.** Contrary to expectations, agent performance does not decline as task difficulty increases. Coding agents—particularly Claude Code and Codex CLI—exhibit stable or improving performance at higher difficulty levels across all judges. In

Table 7: Difficulty rating criteria for environmental documentation.

| Difficulty | Criteria |
|---|---|
| 1 | Affected environment sections for straightforward resource areas, typically land use, air quality, waste management, noise, utilities and infrastructure, geology, socioeconomics, and cultural. |
| 2 | (1) Affected environment for resource areas that require substantial input—typically, biological resources and water resources.<br>(2) Environmental consequences for straightforward resource areas with simple proposed actions. |
| 3 | (1) Environmental consequences for (a) resource areas with greater complexity but simple proposed actions or (b) straightforward resource areas with more complex proposed actions.<br>(2) Affected environment + environmental consequences (simple complexity). |
| 4 | (1) Environmental consequences for resource areas with greater complexity and complex proposed actions.<br>(2) Affected environment + environmental consequences (greater complexity).<br>(3) Mitigating measures, purpose and need, and alternatives. |

several cases, the highest scores are achieved on difficulty 3 and 4 tasks. In contrast, baseline models show relatively flat or mildly declining performance as the difficulty increases, indicating a limited sensitivity to task difficulty.

**Implications.** These results reveal a misalignment between the human-perceived task difficulty and agent difficulty. Tasks that are more complex for human authors may provide richer structure, clearer constraints, or more explicit context that benefits agent-based systems. Consequently, difficulty ratings grounded in the anticipated human effort do not directly translate to an increased difficulty for language models or coding agents.

# D Extended SME Analysis

This appendix provides additional analysis supporting the SME validation results reported in the main paper. The discussion below summarizes the criterion-level scores and recurring observations noted during the SMEs' review of agent-generated drafts.

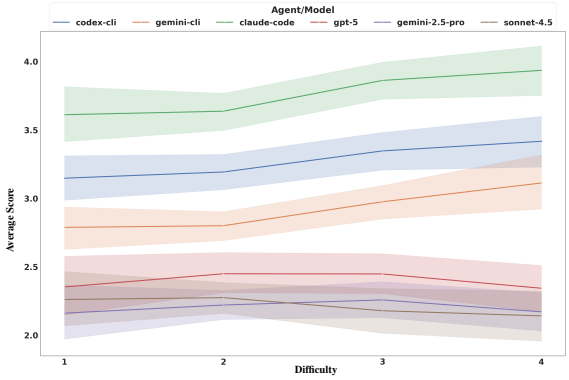


Figure 7: Overall performance of the baseline models and coding agents across different difficulty levels.

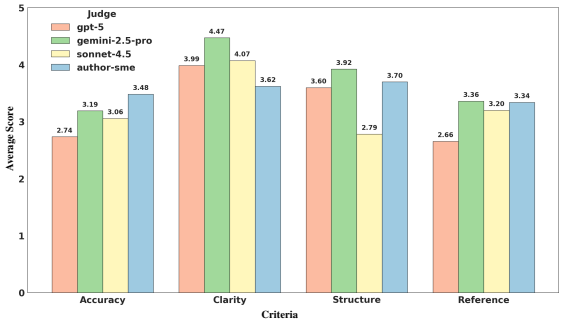


Figure 8: Average scores across evaluation criteria for the sampled cases by LLM judges & Author SMEs.

## D.1 Further Analysis of SME Versus LLM Scores

Figure 8 provides additional insight into the differences in evaluation emphasis between LLM judges and Author SMEs. This figure suggests that LLM judges tend to be stricter on accuracy and, in some cases, structure, whereas SMEs place greater emphasis on clarity and, to a lesser extent, structure, which are especially important in the context of NEPA drafting. Despite these differences, the overall consistency between the LLM and SME evaluations suggests that LLM judges can provide meaningful and reliable signals for automatic evaluation while still reflecting the systematic differences from expert human judgment.

## D.2 Criterion-Level SME Scores

Table 11 reports the breakdown of Author SME scores across evaluation criteria. While the main paper focuses on aggregate trends, this table provides additional detail on how agents perform with respect to accuracy, clarity, reference use, and structure. Across all criteria, Claude Code receives the highest average scores, consistent with the overall SME rankings discussed in the main paper.

## D.3 Observed Model-Specific Patterns During SME Review

In addition to numerical scores, we summarize the recurring patterns observed during the SMEs' review of agent-generated drafts. These observations are intended

Table 8: Difficulty rating criteria and counts for the environmental documentation.

| EIS Case Section | Low Complexity | Medium Complexity | High Complexity |
|---|---|---|---|
| Affected Environment | 23 | 26 | 4 |
| Environmental Consequences | 2 | 17 | 6 |
| Combined affected environment and environmental consequences | 2 | 6 | 7 |
| Purpose and need, alternatives and mitigating measures | - | 2 | 5 |

**Difficulty:** 1 2 3 4

Table 9: Data statistics based on estimated difficulty.

| Difficulty | # Drafts | Avg. Length | # References |
|---|---|---|---|
| 1 | 22 | 1,054 | 7 |
| 2 | 31 | 1,225 | 8 |
| 3 | 30 | 1,310 | 5 |
| 4 | 17 | 1,557 | 6 |

to contextualize the quantitative results and reflect the issues encountered during evaluation, rather than a separate qualitative annotation process.

**Gemini CLI.** During the SMEs' review, Gemini CLI outputs were generally observed to follow the prescribed outlines and maintain a consistent professional tone. However, reviewers frequently encountered issues related to imprecise data placement, geographic specificity, and in-text citation accuracy, as well as reliance on generic NEPA language.

**Codex CLI.** Codex CLI outputs were often concise and structured, but usability varied across sections. While some drafts were largely usable, others required substantial revision, particularly due to issues with data integration and citation consistency.

**Claude Code.** Claude Code drafts were generally observed to be more consistently structured, with clearer narrative flow and more effective reference use. In several cases, SMEs noted that these drafts required fewer revisions relative to other agents. Nonetheless, occasional data-related issues were identified, and outputs were often more verbose than those of other systems.

### D.4 Notes on Evaluation Alignment

As discussed in the main paper, the alignment between the LLM judges and SMEs is quantified using the mean absolute difference (MAD). The criterion-level results and observed patterns reported here are consistent with the aggregate agreement trends reported in Sections 5.3 and 5.4.

## E Extended Discussion

This appendix provides an extended qualitative analysis supporting the discussion in the main paper, including model-specific observations, section-level performance differences, and additional considerations related to evaluation alignment.

### E.1 Model-Specific Benefits and Issues

#### E.1.1 Gemini CLI

Based on SME reviews of content generated using Gemini CLI, we observe strengths in following prescribed outlines, maintaining logical progression, and using a consistent professional tone. However, Gemini CLI struggles to reliably place data, maintain geographic specificity, and produce accurate in-text citations. SMEs also note a tendency toward vague or generic NEPA language, rather than the use of section-specific or project-specific terminology.

#### E.1.2 Codex CLI

Codex CLI produces concise and, in some cases, accurate content. However, we find substantial variability in usability across sections. While the expected outlines are generally rendered correctly and some content is directly usable, other outputs require substantial revision, in some cases exceeding the effort required to draft the section without artificial intelligence (AI) assistance. Data integration and citation accuracy remain recurring issues.

#### E.1.3 Claude Code

Overall, we find that Claude Code produces the strongest content across models and section types. SME reviewers note appropriate document structure, logical subsection organization, clear transitions, and readable introductions. Data, locations, and references are generally used more effectively and correctly than in outputs from other systems, and in several cases, SMEs judged the generated content to be comparable to or better than the ground truth drafts. Nonetheless, SMEs identify occasional data issues, indicating the continued need for careful verification. Reviewers also note that Claude Code outputs are often more verbose than those of other models, which raises practical concerns given NEPA document length constraints.

Table 10: Overall performance grouped by judge and agent across difficulty levels.

| Judge | Agent/Model | Difficulty 1 | Difficulty 2 | Difficulty 3 | Difficulty 4 |
|---|---|---|---|---|---|
| GPT-5 | GPT-5 | $2.52 \pm 0.71$ | $2.56 \pm 0.60$ | $2.51 \pm 0.59$ | $2.47 \pm 0.54$ |
| | Gemini 2.5 Pro | $2.19 \pm 0.59$ | $2.19 \pm 0.40$ | $2.26 \pm 0.56$ | $2.18 \pm 0.43$ |
| | Claude Sonnet 4.5 | $2.37 \pm 0.63$ | $2.36 \pm 0.47$ | $2.33 \pm 0.68$ | $2.22 \pm 0.63$ |
| | Codex CLI | $2.91 \pm 0.56$ | $2.97 \pm 0.52$ | $3.16 \pm 0.51$ | $3.14 \pm 0.53$ |
| | Gemini CLI | $2.62 \pm 0.60$ | $2.63 \pm 0.48$ | $2.83 \pm 0.55$ | $2.98 \pm 0.65$ |
| | Claude Code | $\mathbf{3.31 \pm 0.72}$ | $\mathbf{3.36 \pm 0.55}$ | $\mathbf{3.57 \pm 0.53}$ | $\mathbf{3.56 \pm 0.57}$ |
| Gemini 2.5 Pro | GPT-5 | $2.37 \pm 1.07$ | $2.54 \pm 0.93$ | $2.44 \pm 0.86$ | $2.31 \pm 0.74$ |
| | Gemini 2.5 Pro | $2.21 \pm 1.00$ | $2.35 \pm 0.70$ | $2.27 \pm 0.74$ | $2.24 \pm 0.58$ |
| | Claude Sonnet 4.5 | $2.24 \pm 1.06$ | $2.34 \pm 0.72$ | $2.09 \pm 0.85$ | $2.12 \pm 0.77$ |
| | Codex CLI | $3.45 \pm 0.74$ | $3.55 \pm 0.75$ | $3.79 \pm 0.71$ | $3.90 \pm 0.68$ |
| | Gemini CLI | $3.02 \pm 0.78$ | $3.05 \pm 0.54$ | $3.25 \pm 0.65$ | $3.43 \pm 0.80$ |
| | Claude Code | $\mathbf{3.82 \pm 0.80}$ | $\mathbf{3.95 \pm 0.72}$ | $\mathbf{4.19 \pm 0.58}$ | $\mathbf{4.27 \pm 0.63}$ |
| Sonnet-4.5 | GPT-5 | $2.37 \pm 0.63$ | $2.36 \pm 0.47$ | $2.33 \pm 0.68$ | $2.22 \pm 0.63$ |
| | Gemini 2.5 Pro | $2.24 \pm 1.06$ | $2.34 \pm 0.72$ | $2.09 \pm 0.85$ | $2.12 \pm 0.77$ |
| | Claude Sonnet 4.5 | $2.18 \pm 0.82$ | $2.13 \pm 0.49$ | $2.12 \pm 0.75$ | $2.09 \pm 0.64$ |
| | Codex CLI | $3.09 \pm 0.67$ | $3.06 \pm 0.60$ | $3.10 \pm 0.58$ | $3.21 \pm 0.49$ |
| | Gemini CLI | $2.73 \pm 0.61$ | $2.73 \pm 0.49$ | $2.85 \pm 0.63$ | $2.93 \pm 0.55$ |
| | Claude Code | $\mathbf{3.71 \pm 0.98}$ | $\mathbf{3.61 \pm 0.72}$ | $\mathbf{3.83 \pm 0.75}$ | $\mathbf{3.97 \pm 0.72}$ |

Table 11: Overall performance based on Author SME scores across different criteria.

| Agent/Model | Accuracy | Clarity | Reference | Structure |
|---|---|---|---|---|
| Codex CLI | $3.06 \pm 0.83$ | $3.18 \pm 1.14$ | $3.22 \pm 0.94$ | $3.43 \pm 1.33$ |
| Gemini CLI | $3.30 \pm 0.84$ | $3.62 \pm 1.37$ | $2.76 \pm 1.41$ | $3.75 \pm 1.32$ |
| Claude Code | $\mathbf{4.12 \pm 0.76}$ | $\mathbf{4.09 \pm 0.97}$ | $\mathbf{4.08 \pm 0.83}$ | $\mathbf{3.94 \pm 1.19}$ |

### E.2 Section-Specific Performance and Task Sensitivity

Of the 25 SME-reviewed tasks, six tasks had identical resource areas (e.g., there were multiple visual resources sections). This enabled analysis across similar sections although the actions, lead agencies, and ground truth documents differed in scope, level of detail, and tone.

Across models, we observe occasional instances where generated content is generally usable with targeted edits. However, less analytically complex affected environment sections, such as Air Quality, are more consistently amenable to agent-assisted drafting, particularly when using Claude Code. In contrast, sections requiring tighter data integration, cross-referencing, or analytical judgment exhibit greater variability in quality and consistency across models.

### E.3 Evaluation Alignment and Bias Considerations

An additional analysis of the evaluation alignment indicates that while LLM judges tend to apply stricter criteria for accuracy, SMEs place greater emphasis on clarity, structure, and communicative effectiveness. We find that the alignment between the Author and External SMEs indicates that author involvement does not systematically inflate scores. These complementary differences suggest that automated and human evaluations capture overlapping but distinct aspects of draft quality, supporting the use of hybrid evaluation strategies for long-form generation tasks.