JOURNAL OF SPATIAL INFORMATION SCIENCE
Number N (YYYY), pp. xx–yy
doi:10.5311/JOSIS.YYYY.II.NNN

RESEARCH ARTICLE

Benchmarking Large Language Models for Geolocating Colonial Virginia Land Grants

[Author name removed for blind review]

[Affiliation removed for blind review]

Received: [Not received]; returned: [Not returned]; revised: [Not revised]; accepted: [Not accepted].

Abstract: Virginia's seventeenth- and eighteenth-century land patents survive primarily as narrative metes-and-bounds descriptions, limiting spatial analysis. This study systematically evaluates current-generation large language models (LLMs) in converting these prose abstracts into research-grade latitude/longitude coordinates. A digitized corpus of 5,471 Virginia patent abstracts (1695–1732) is released, with 43 rigorously verified test cases for benchmarking. Six OpenAI models across three architectures—o-series, GPT-4-class, and GPT-3.5—were tested under two paradigms: direct-to-coordinate and tool-augmented chain-of-thought invoking external geocoding APIs. Results were compared against a professional GIS workflow, Stanford NER geoparser, Mordecai-3 neural geoparser, and a county-centroid heuristic.

The top single-call model, o3-2025-04-16, achieved a mean error of 23 km (median 14 km), a 67% improvement over professional GIS methods and 70% better than Stanford NER. A five-call ensemble further reduced errors to 19 km (median 12 km) at minimal additional cost (~USD 0.20 per grant). Paired Wilcoxon tests confirm ensemble superiority (W=629, p=0.03 vs. single-shot). A patentee-name redaction ablation slightly increased error (~9%), showing reliance on metes-and-bounds reasoning rather than memorization. The cost-effective gpt-4o-2024-08-06 model maintained a 28 km mean error at USD 1.09 per 1,000 grants, establishing a strong cost-accuracy benchmark. External geocoding tools offer no measurable benefit for this task.

These findings demonstrate that LLMs can georeference early-modern records as accurately and significantly faster and cheaper than traditional GIS workflows, enabling scalable spatial analysis of colonial archives.

Keywords: historical GIS, large language models, geoparsing, colonial Virginia, land grants, digital humanities, spatial history, geolocation

1 Introduction

1.1 Historical Context & Motivation

Virginia's colonial land patents are a cornerstone resource for scholars studying settlement patterns, the political economy of plantation agriculture, and Indigenous dispossession in the seventeenth and eighteenth centuries. Yet the spatial dimension of these sources remains under-exploited: most patents survive only as narrative metes-and-bounds descriptions in printed abstract volumes such as *Cavaliers and Pioneers* (C&P) [1]. Without geographic coordinates, historians and archaeologists cannot readily visualise how land ownership evolved or test hypotheses with modern Geographic Information System (GIS) tools. Creating a machine-readable, georeferenced version of C&P would unlock new quantitative approaches to long-standing questions about colonial Virginia's social and environmental history.

Digitising and geo-locating the abstracts, however, is notoriously labor-intensive. Even professional GIS analysts can spend several hours per grant reconciling archaic placenames, inconsistent spellings, and low-resolution boundary calls. Recent breakthroughs in large language models (LLMs) suggest a new pathway: language-driven spatial reasoning where a model reads the patent text and predicts latitude/longitude directly or with minimal tool assistance. This study explores whether current-generation LLMs can shoulder that burden accurately and cheaply enough to matter for digital history.

1.2 Problem Statement

Despite the promise of LLMs, their ability to extract usable coordinates from early-modern archival prose had not been systematically evaluated prior to this work. Key uncertainties addressed in this study included:

- Could large language models trained mostly on contemporary text understand seventeenth-century toponyms and bearing conventions?
- Would providing API-based tools (e.g., Google Places search) materially improve accuracy relative to a pure text approach?
- How did model predictions compare to professional GIS workflows [2], deterministic pipelines such as the GeoTxt Stanford NER geoparser [3], neural geoparsers like Mordecai-3 [4], and other heuristic benchmarks in both error and cost?

Addressing these questions required a rigorously annotated test bench that blended historical sources, modern GIS ground truth, and controlled prompt engineering. The methodological design seeks to embody principles of rigorous and responsible GeoAI research, as outlined by Li et al. [5] and detailed further in §2.4.

1.3 Contributions

This study makes four principal contributions:

1. Releases the first copyright-compliant, machine-readable dataset of *Cavaliers and Pioneers*, Vol. 3 [1], including (i) row-level metadata—row identifier, word count, and SHA-256 hash—for all 5,471 abstracts, and (ii) limited, non-substitutable excerpts of

up to 200 words for the 43 evaluation abstracts. The full OCR corpus is available privately under a vetted, non-commercial data-use agreement.
2. Provides authoritative latitude/longitude pairs for 43 randomly sampled patents, derived from state-archived GIS polygons and cross-verified archival sources, yielding a high-fidelity evaluation target.
3. Presents the first head-to-head evaluation of two LLM prompting paradigms – direct-to-coordinate inference and tool-augmented chain-of-thought – across six OpenAI model families on the historical land-grant geolocation task.
4. Quantifies the trade-offs among spatial error, monetary expense, and processing time, demonstrating that a pure LLM pipeline can match or surpass professional GIS accuracy while operating orders of magnitude faster and cheaper.

All data, code, and results are available in the supplemental repository: [RepositoryURLremovedforblindreview].

2 Background & Related Work

2.1 Historical GIS and Land-Grant Mapping

Digitizing colonial-era land grants has long promised new insights into European settlement patterns, Indigenous land displacement, and the development of local economies. However, this potential has been constrained by the extensive manual labor required to convert metes-and-bounds descriptions into spatial data. Traditional approaches to georeferencing these historical records have proven prohibitively time-consuming - a genealogical case study by Julian and Abbitt [6] required nearly ten years of archival sleuthing and three university-semester GIS projects to pinpoint a single family's land claims across three Tennessee counties.

Several institutional efforts have attempted to address these challenges, though coverage remains incomplete. The Library of Virginia maintains a statewide *Land Patents and Grants* online database hosting scanned images and searchable indices for every recorded patent (1623–1774) and subsequent grant (1779–2000), including Northern Neck surveys, but provides no ready-made GIS polygons, limiting its direct utility for spatial analysis [7]. Similarly, Loudoun County GIS staff have successfully reconstructed all original grants within their jurisdiction [8]. These initiatives demonstrate the feasibility of digitizing historical land records but also highlight significant gaps in existing datasets - many seventeenth- and eighteenth-century patents still lack spatial coordinates.

Among the most thorough academic efforts for Virginia's Northern Neck proprietary are Mitchell's [9] maps and companion text documenting the "Beginning at a White Oak" patents of Fairfax County. This work reconstructed hundreds of early land grants with polygonal boundaries, establishing both the feasibility and research value of transforming metes-and-bounds descriptions into spatial data. Building on such foundations, scholars have leveraged available georeferenced grants for substantive historical analysis. In Virginia, seminal studies like Fausz [10] utilized narrative patent abstracts to trace settlement patterns along the James River basin, while noting the persistent challenges of transforming textual descriptions into precise spatial coordinates for quantitative analysis.

This analytical potential extends beyond Virginia. Dobbs [11] used georeferenced North Carolina grants to demonstrate that eighteenth-century town sites often followed pre-

existing Indigenous trails, while Coughlan and Nelson [12] leveraged a dataset of 1,160 South Carolina grants to model settlement patterns based on river access and soil analysis. In each case, spatial enablement of historical records revealed patterns difficult to discern through textual sources alone.

In genealogical and historical research communities, semi-automated solutions have emerged to assist with this labor-intensive process. DeedMapper software [13] helps researchers convert metes-and-bounds descriptions into visual plots, though it still requires manual entry of deed text and expert positioning of parcels on reference maps. Professional development courses from the Salt Lake Institute of Genealogy (SLIG) continue to teach these specialized mapping techniques, reflecting the still-developing state of automation in this field.

The literature establishes three critical facts. First, historians value land-grant GIS layers because they unlock settlement and landscape questions that text alone cannot answer. Second, traditional platting methods are too slow and too localized to deliver colony-scale coverage. Third, the piecemeal datasets that do exist furnish both ground truth and a methodological benchmark for any attempt at automation. This study addresses this bottleneck by testing whether large language models can shoulder the coordinate-extraction burden—potentially transforming Virginia's colonial patents from archival prose to research-ready GIS at scale.

2.2 Large Language Models for Geolocation

Building on the manual coordinate-extraction bottleneck outlined in § 2.1, recent advances in large language models (LLMs) suggest that much of the geoparsing pipeline can now be automated. Coordinate extraction—sometimes called *geoparsing*—comprises two subtasks: (i) identifying candidate toponyms in running text and (ii) resolving each mention to a unique set of latitude/longitude coordinates.

The evolution of this field has moved through several distinct methodological phases. Rule-based gazetteer look-ups dominated early work, providing limited accuracy when dealing with ambiguous place names. Neural architectures such as CamCoder [14] subsequently improved performance through learned contextual representations. Most recently, fine-tuned large language models have demonstrated substantial breakthroughs in toponym resolution accuracy. A representative example of this latest approach comes from Hu et al. [15], who fine-tuned Llama 2-7B to generate an unambiguous administrative string for each toponym before invoking a standard geocoding API. Their model achieved an Accuracy@161 km of 0.90 on the GeoCorpora set, outperforming previous neural methods and improving toponym resolution accuracy by 13% over the previous best neural system. On the less ambiguous WikToR corpus the same architecture reached 0.98. Crucially, these gains were realized on commodity hardware—the entire experiment ran on a single NVIDIA V100 with 14 GB VRAM, showing that parameter-efficient fine-tuning is feasible without data-centre hardware—underscoring the practicality of parameter-efficient fine-tuning for large corpora.

Addressing the persistent challenge of annotation scarcity, Wu et al. [16] introduced GeoSG, a self-supervised graph neural network that learns spatial semantics from Point-of-Interest (POI)–text relationships. This approach predicts document coordinates without any annotated training samples, nearly matching supervised baselines on two urban benchmarks. In a similar vein, Savarro et al. [17] demonstrated that Italian tweets can be

geolocated to both regional and point coordinates by fine-tuning decoder-only LLMs on the GeoLingIt shared task, further confirming that pretrained language models can internalize subtle linguistic cues of place.

Despite these advances, significant limitations remain. O'Sullivan et al. [18] demonstrated that GPT-class models mis-calibrate qualitative distance terms: *near* in a neighborhood scenario is treated similarly to *near* at continental scale, revealing a lack of geometric grounding. Such biases caution against "out-of-the-box" deployment for precision geolocation, especially when dealing with archaic toponyms or surveyor jargon. Even the most advanced automated systems leave a long tail of ambiguous or obsolete place names—precisely the cases that plague colonial patent abstracts.

In summary, fine-tuned LLMs now surpass previous neural approaches on toponym resolution and can support colony-scale spatial inference, yet their reasoning remains sensitive to context and scale. The next section (§ 2.3) explores tool-augmented prompting frameworks that grant LLMs access to external geocoders and vector databases—potentially mitigating some of the failure modes identified above.

2.3 Tool-Augmented Prompting Techniques

Integrating large language models with external geospatial utilities has emerged as a promising way to address the limitations identified in § 2.2. In a *tool-augmented* workflow, the LLM interprets unstructured language but can invoke specialized geocoding, database, or cartographic services during its reasoning process, grounding its outputs in authoritative data and deterministic algorithms.

This hybrid approach has evolved through several distinct implementations, each targeting different aspects of the geolocation challenge. Early evidence for its effectiveness comes from Hu et al. [15], who coupled a fine-tuned Llama 2-7B with a cascading trio of geocoders—GeoNames, Nominatim, and ArcGIS—to resolve toponyms the model had already disambiguated linguistically. Their experiments demonstrated that this hybrid pipeline raised Accuracy@161 km by 7–17 percentage points relative to either component used in isolation.

Extending this concept to more complex natural language descriptions, Huang et al. [19] developed GeoAgent for free-form address normalization. This system enables the LLM to convert colloquial descriptions (e.g., "two blocks east of the old courthouse") into structured cues, orchestrate vector-database lookups and offset calculations, and then retrieve precise coordinates from mapping APIs. Their ablation study confirmed that this agentic variant outperforms both rule-based and LLM-only baselines on the public GeoGLUE benchmark and an in-house Chinese address dataset, demonstrating improved F1 scores and edit-distance metrics.

These specialized implementations build upon a more general design pattern known as the ReAct prompting paradigm [20], which demonstrates how language models can interleave chain-of-thought reasoning with live tool calls. While originally demonstrated on question-answering and web-shopping tasks, this interleaved reasoning-action approach provides a framework that can be adapted to tasks requiring both linguistic interpretation and computational precision.

At enterprise scale, Google Research's *Geospatial Reasoning* initiative [21] exemplifies the integration of foundation models with Earth Engine, BigQuery, and Maps Platform. This system enables agentic LLMs to chain satellite imagery, socioeconomic layers, and

routing services to answer compound spatial queries in seconds—a capability relevant to both consumer applications and research contexts.

Across these diverse implementations, a consistent finding emerges: granting an LLM controlled access to trusted GIS services reduces hallucination, improves numerical accuracy, and broadens task coverage (Hu et al. [15]; Huang et al. [19]). The present work builds on this pattern by testing whether a similar benefit materializes for colonial land-grant geolocation—comparing a pure one-shot prompt to a tool-augmented chain-of-thought that can issue mid-prompt geocoding and distance-calculation calls while processing historical texts with archaic toponyms and surveying terminology.

2.4 2.4 Emerging GeoAI Research Principles

Recent calls within the GeoAI community emphasize the need for empirical studies that are not only traditionally scientifically sound but also actively engage with the foundational tenets of **predictability**, **interpretability**, **reproducibility**, and **social responsibility**, which Li et al. [5] identify as four essential pillars for solidifying GeoAI's scientific rigor and ensuring its lasting, beneficial impact.

Li et al. (2024) define **predictability** as the combination of a model's accuracy, computational efficiency, and robustness when confronted with spatial variation. The present study addresses this definition by reporting mean and median great-circle error, 95% bootstrap confidence intervals, and cumulative-error curves for all evaluated LLM variants and a professional GIS baseline (Figure 1 and Table 4); by presenting cost-versus-accuracy and latency-versus-accuracy Pareto frontiers (Figures 5 and 11) demonstrating reductions of two to five orders-of-magnitude in dollar cost and turnaround time relative to human baselines while preserving or improving spatial accuracy; and by examining robustness through targeted ablations reported in § 6.6, showing that accuracy is essentially unaffected by changes in temperature, reasoning-budget, and abstract length, and that removing the five largest residuals alters mean error by less than two kilometres—confirming that results are not driven by a small subset of extreme cases.

The study places a strong emphasis on **interpretability** by meticulously recording the complete reasoning process behind each model prediction, not simply the final geographic coordinates. For every inference, a detailed, step-by-step record is captured and logged that includes the chain-of-thought narrative text provided by the model, every external function invocation—including the precise queries passed to the geocoding tools—and the exact JSON responses returned. This comprehensive logging creates a fully auditable record of the model's internal reasoning, enabling researchers to reconstruct exactly how and why a given prediction was made. For instance, as detailed in Section 6.4 and Appendix A.3, the logs clearly document how the models identify key geographic features, choose between multiple candidate locations, refine queries based on initial mismatches, systematically test alternate spellings or county qualifiers, and decide when and how to average coordinates using spatial centroid calculations. Capturing these detailed reasoning steps across both prompting paradigms (direct one-shot versus iterative, tool-augmented reasoning) provides unprecedented transparency into the models' cognitive processes. This explicit audit trail reveals precisely where models succeed or fail, highlighting systematic errors such as cascading failures after incorrect geocoder hits or misinterpretations of ambiguous historical place names. Because every intermediate reasoning step and tool interaction is logged, it's possible to correlate internal indicators of model confidence—such as the geographic

spread between top-ranked candidate coordinates—with actual prediction error, offering insights that are essential for interpreting, trusting, and optimizing model behavior.

To ensure **reproducibility**, specific snapshot versions of the OpenAI models from April 2025 were used and random seeds were fixed throughout all steps, including dataset splits, sampling, and bootstrapping. All parameter-sensitivity tests (temperature, reasoning budget, abstract length) were also conducted under these controlled conditions. The computational environment was packaged into a Docker container that specifies exact Python dependencies and OpenAI API endpoints to guarantee consistent results on different machines. Additionally, the full OCR-corrected corpus of 5,471 abstracts, 43 authoritative ground-truth coordinates, dev/test splits, exact prompts, YAML configurations, the run_experiment.py evaluation script, and detailed JSONL logs recording every model request and response are provided. All these materials are publicly available in the accompanying code repository and described in § 3, allowing others to exactly reproduce the analyses, tables, and figures presented here.

The study meets the **social responsibility** pillar by carefully considering ethical and copyright implications associated with the historical data used. Although the underlying seventeenth- and eighteenth-century land patent records themselves are public domain, the transcriptions published in the 1979 compilation Cavaliers and Pioneers, Vol. 3 remain under copyright. To balance reproducibility with copyright compliance, only limited, non-substitutable excerpts (up to 200 words each) of the 43 abstracts with authoritative ground-truth points are publicly released. For the full corpus of 5,471 abstracts, only row identifiers, word counts, and SHA-256 hashes of each abstract are provided, allowing researchers to verify their own local copies without exposing protected text. The complete OCR corpus itself is made available privately under a vetted, non-commercial data-use agreement for scholarly research only. Additionally, because the georeferenced coordinates reflect historical property boundaries rather than modern sensitive locations or private ownership, the study inherently minimizes privacy risks. Computationally, off-the-shelf foundation models are used without energy-intensive fine-tuning, intensive reasoning settings are limited strictly to essential cases, and API calls are throttled via OpenAI's service-flex option to reduce computational overhead. Finally, the study acknowledges that colonial source materials inherently underrepresent Indigenous and marginalized perspectives and explicitly highlights that the research methods and findings presented here can be directly applied to better understand and contextualize historical patterns of Indigenous dispossession and marginalization.

By embedding these considerations into the experimental design and reporting, this work aims to contribute a concrete case study that addresses the foundational requirements for a developing science of GeoAI.

3 Data

3.1 Corpus Overview

Cavaliers and Pioneers, Volume 3 [1], compiles 5,471 abstracts of Virginia land patents recorded in patent books 9–14 (1695–1732). These grants fall largely in central and south-central Virginia, clustering around the present-day Richmond area. After an extensive search, no publicly available digital transcription of this volume was found and therefore

the material is treated as unseen by contemporary language models, though a formal check of training-data leakage was not performed.

3.2 Pre-processing Pipeline

To prepare the corpus for analysis, the source volume was destructively scanned page-by-page. Multiple optical-character-recognition (OCR) configurations were trialled to maximise fidelity; the optimal workflow was then applied to all pages. Extracted text was normalised and exported as a CSV with one row per patent abstract, yielding a complete corpus of 5,471 land grant abstracts.

From this full corpus, three random subsets were generated using reproducible seeds:

- Dev-1 and Dev-2 – 20 abstracts each, reserved for prompt engineering and method tuning.
- Test – 125 abstracts, mutually exclusive from the dev sets.

3.3 Ground-Truth & Baseline Coordinates

Of the 125 randomly sampled test abstracts, 43 met strict archival verification criteria and were assigned authoritative latitude/longitude pairs. Ground-truth coordinates were established by matching grants to polygons in the Central VA Patents GIS layer developed by One Shared Story [22] in collaboration with the University of Virginia's Institute for Public History. This dataset was created using professional Deed Mapper software to convert metes-and-bounds descriptions to GIS format, with historic maps and local knowledge as references.

For each matched grant (identified by grantee name, year, and acreage), the centroid of the corresponding GIS polygon was used as the authoritative coordinate. All matches were manually verified to ensure topological consistency, neighboring grant relationships, and acreage alignment before inclusion in the ground truth dataset.

Ground truth coordinates were established only for grants meeting rigorous archival verification standards. This conservative approach prioritized evaluation quality over sample size, ensuring that all ground truth coordinates represent authoritative historical locations rather than speculative placements. The 34% verification rate reflects the inherent challenges of establishing definitive coordinates for 17th-18th century land grants, where many historical landmarks have been lost or renamed over three centuries.

This methodological choice trades sample size for ground truth reliability—a critical consideration given that evaluation accuracy depends entirely on the quality of reference coordinates. Including grants with uncertain or speculative ground truth would introduce measurement error that could mask true model performance differences. Each authoritative coordinate determination required substantial curatorial effort (1–3 hours of expert research per grant), making exhaustive ground-truthing impractical while maintaining methodological integrity.

The 36% archival verification rate is consistent with other historical GIS studies where documentary evidence is prioritized over sample size. This sample size provides sufficient statistical power for the comparative analysis while preserving ecological validity and ensuring that all performance metrics reflect true model capabilities rather than artifacts of uncertain reference data.

4 Methods

4.1 Professional GIS Benchmark (H-1)

A certified GIS analyst [2] implemented an automated geolocating procedure leveraging standard geospatial libraries and toolsets. The analyst was selected through a competitive bidding process on a freelancer marketplace, where 49 qualified contractors submitted bids averaging $133 USD (range: $30-$250). The selected contractor holds an MSc in Geography & Environmental Management with 9+ years of experience in geospatial analysis and maintains a 5.0-star rating with 100% on-time delivery record. The workflow ingested the patent texts, tokenized toponyms, and queried a multi-layered gazetteer stack (including ArcGIS Online resources, historical overlays, and place-name databases) to generate the highest-confidence coordinate for each grant. Development, parameter tuning, and execution required approximately six billable hours for all 43 grants with verified ground truth. This end-to-end workflow time represents the total cost of bespoke GIS analysis, contrasting with off-the-shelf LLM inference that requires no custom development.

These baseline coordinates are stored directly in the evaluation file, allowing the experiment script to access them through the static pipeline. A labor cost of USD 140 (six billable hours) is assigned to the benchmark when reporting cost metrics.

4.2 Stanford NER Baseline (H-2)

To provide a more rigorous deterministic baseline, a Stanford Named Entity Recognition (NER) approach was implemented using the GeoTxt framework. This method represents a state-of-the-art automated geoparsing pipeline that combines linguistic analysis with gazetteer lookup, providing a systematic comparison point for the LLM-based approaches.

The Stanford NER pipeline operates through a three-stage process: (1) Named entity extraction using Stanford's CoreNLP library to identify geographic entities within the patent abstracts, (2) Geographic resolution via the GeoNames API with Virginia-specific restrictions to prevent out-of-state matches, and (3) Coordinate selection using a population-weighted ranking system to choose the most likely location when multiple candidates are found.

The system implements a robust fallback hierarchy: if no geographic entities are successfully resolved, it falls back to county centroid coordinates extracted from the patent text; if county extraction fails, it defaults to Virginia's geographic center (37.4316, -78.6569). This approach ensures 100% prediction coverage while maintaining methodological consistency.

The Stanford NER method achieved a mean error of 79.02 km with 100% prediction coverage across all 43 test grants. While this represents a more systematic approach than the single-analyst GIS baseline, it demonstrates the challenges that automated systems face when dealing with historical toponyms that may have shifted meaning or location over centuries, as detailed in the case study analysis (§7.2.1).

4.3 Mordecai-3 Heuristic Geoparser (H-3)

Benchmark H-3 employs the open-source *Mordecai-3* neural geoparser [4], augmented with domain-specific heuristics tuned for colonial Virginia deeds (full details in Appendix B.1). In brief, the pipeline

1. expands historical abbreviations (e.g., "Cr."→*Creek*, "Co."→*County*),
2. feeds multiple cleaned variants of the deed text to Mordecai until at least one toponym is returned,
3. filters candidate coordinates to a Virginia-bounded box and applies a confidence threshold,
4. accepts the highest-scoring point that lies within d km of the deed's county centroid (tuned over {25, 35, 50 km}),
5. falls back to county- or state-centroid coordinates when no qualified entity survives.

A three-parameter grid search on the 43 gold-standard grants selected the optimal confidence, bounding-box margin, and distance-gate values. This configuration attains a 94.3 km mean error—worse than both the Stanford NER pipeline and the county-centroid baseline.

4.4 County-Centroid Baseline (H-4)

Method H-4 provides a transparent deterministic floor. A regex extracts any Virginia county name (handling forms like "Henrico Co.", "City of Norfolk", etc.); if successful, the script returns the pre-computed TIGER/Line centroid of that county. When no county is detected it defaults to the geographic centre of Virginia (37.4316 °N, -78.6569 °W). On the 43-validation-grant set this logic produced 36 county-centroid predictions and 7 statewide-centroid fallbacks. Although trivial to implement and lightning-fast (<2 ms per deed), the approach yields a mean error of 80.3 km, serving mainly as a sanity check that more sophisticated pipelines clear with ease.

4.5 One-shot Prompting (M-series)

In the first automatic condition, the language model receives the grant abstract together with a single exemplar response illustrating the desired output format. The prompt asks for coordinates expressed in degrees–minutes–seconds (DMS) and contains no chain-of-thought or tool instructions:

```
Geolocate this colonial Virginia land grant to precise latitude and
    longitude coordinates.
Respond with ONLY the coordinates in this format: [DD][MM]'[SS].[SSSSS]"N
     [DDD][MM]'[SS].[SSSSS]"W
```

Six OpenAI model variants spanning three architecture families constitute the M-series (1). Temperature is fixed at 0.2 for gpt-4.1-2025-04-14 and gpt-4o-2024-08-06; all other parameters remain at their service defaults. Each abstract is processed with a single API call; no external tools are available in this condition.

Table 1: One-shot model variants (M-series).

ID	Model
M-1	`o4-mini-2025-04-16`
M-2	`o3-2025-04-16`
M-3	`o3-mini-2025-01-31`

ID	Model
M-4	`gpt-4.1-2025-04-14`
M-5	`gpt-4o-2024-08-06`
M-6	`gpt-3.5-turbo`

4.6 Tool-augmented Chain-of-Thought (T-series)

The second automated condition equips the model with two specialized tools: `geocode\_place`, an interface to the Google Geocoding API limited to Virginia and adjoining counties, and `compute\_centroid`, which returns the spherical centroid of two or more points. The system prompt (Appendix A.2.2) encourages an iterative search strategy where the model can issue up to twelve tool calls, evaluate the plausibility of each result, and optionally average multiple anchors before emitting a final answer in decimal degrees with six fractional places.

Table 2 summarizes the five model variants initially considered for this tool suite. Of these, only T-1 and T-4 were carried forward into the final evaluation. The remaining models—T-2 (o3-2025-04-16), T-3 (o3-mini-2025-01-31), and T-5 (computer-use-preview-2025-03-11)—were excluded after developmental testing. This testing revealed that these models, including the o3-2025-04-16 variant (the top performer in one-shot evaluations), produced outputs largely identical to the more economical T-1 when using the tool-augmented pipeline. Given that the primary tool, Google's Geocoding API, is deterministic, proceeding with these additional models would have substantially increased computational costs and processing times without yielding distinct results or further insights into tool-augmented performance.

Table 2: Tool-augmented model variants (T-series).

ID	Model
T-1	`o4-mini-2025-04-16`
T-2	`o3-2025-04-16`
T-3	`o3-mini-2025-01-31`
T-4	`gpt-4.1-2025-04-14`

The experiment driver loops over each abstract, maintains a conversation history including tool call outputs, and stops after either receiving a valid coordinate string or exceeding ten assistant turns.

4.7 Five-call Ensemble (E-series)

The E-series leverages *ensembling* to squeeze additional accuracy from the best single model. For each abstract the pipeline issues five independent one-shot calls to `o3-2025-04-16`, each with a different random seed but identical prompt. The resulting five coordinate pairs are clustered with the DBSCAN algorithm (= 0.5 km). If at least three predictions fall within the same 0.5 km cluster, their spherical centroid becomes the final answer; otherwise the centroid of all five points is returned. This majority-vote strategy reduces random scatter and mitigates occasional large-error outliers. The ensemble (method

E-1) achieves a mean error of 18.7 km—the best of all evaluated methods—at roughly 5× the token cost of a single o3 call but still two orders of magnitude cheaper than the GIS benchmark. A name-redacted ablation (E-2, see § 6.6) confirms that the gain is not driven by memorised patentee–location pairs.

4.8 Cost and Latency Accounting

For each automated prediction, input and output tokens reported by the OpenAI API are converted to U.S. dollars using the price list in effect on 15 May 2025. The per-call cost is calculated as:

$$\text{Cost} = \frac{\text{input tokens}}{10^6} \times p_{\text{in}} + \frac{\text{output tokens}}{10^6} \times p_{\text{out}}$$

where p_{in} and p_{out} are USD prices per million tokens (see Table~3). Google Geocoding calls remain comfortably within the free-tier quota and therefore do not accrue additional fees.

[H]

Table 3: OpenAI token pricing in effect on 15 May 2025 and used for all cost calculations. Values are quoted in USD per 1M tokens.

Model	p_{in}	p_{out}
GPT-4.1 ("gpt-4.1-2025-04-14")	2.00	8.00
GPT-4o ("gpt-4o-2024-08-06")	5.00	15.00
GPT-3.5-turbo	0.50	1.50
o4-mini	1.10	4.40
o3 (base)	10.00	40.00
o3-mini	1.10	4.40

Latency is measured as wall-clock time from submission of an API request until a valid coordinate string is returned, inclusive of all intermediate tool interactions. For the traditional GIS benchmark, the analyst's total working time (6 h) is divided by the number of grants processed (43), yielding an average latency of 502 s per prediction.

5 Experimental Setup

5.1 Evaluation Metrics

The primary outcome measure is distance error—the great-circle distance in kilometres between predicted and reference coordinates, computed with the Haversine formula. The mean, median, and 95% bootstrap confidence intervals are reported, along with accuracy bands (<1 km, 1–10 km, >10 km).

Efficiency is characterized by two key metrics:

1. Latency: Measured as mean labor time per grant (forward-pass time once the workflow is in place)

2. Monetary cost: Calculated by multiplying input and output token counts returned by the OpenAI API by the official per-token prices in effect on 01 May 2025

The GIS benchmark incurred a fixed fee of USD 140 for approximately 6 billable hours processing 43 grants (502 s per grant).[1] For LLM methods, latency represents wall-clock time from API request to final coordinate string, inclusive of all tool interactions.

All metrics are computed on the 43 test-set abstracts for which ground-truth coordinates are available; remaining rows are retained in the public logs but excluded from aggregate statistics.

5.2 Implementation Protocol

The full corpus (5,471 abstracts) was partitioned into development (20%) and test (80%) segments using seed 42. From these segments, fixed-size random samples were drawn: two development sets of 20 abstracts each for prompt engineering and parameter tuning, and a held-out test set of 125 abstracts that remained unseen during development.

Ground-truth coordinates were established for 43 of the 125 test abstracts following the methodology described in Section 3.3. The traditional GIS baseline and all automated predictions were subsequently written to the same tabular structure, ensuring uniform error computation across methods.

For each method listed in Tables~1 and~2, an evaluation driver sequentially processed the 43 abstracts with verified ground truth, invoking the OpenAI *Responses* API under stable April-2025 model versions. Tool-chain variants interacted with the Google Geocoding API and an in-process centroid function exposed via JSON-Schema. Token usage, latency, and any tool traces were logged in real time; intermediate artifacts and final result sets are archived in the accompanying repository.

6 Results

6.1 Accuracy

Table 4 summarises distance-error statistics for all 43 grants with verified ground truth. The best single-call model, M-2 (o3-2025-04-16), attains a mean error of 23 km—a 67% improvement over the professional GIS workflow (H-1, 71 km) and 70% better than the Stanford NER geoparser (H-2, 79 km). Clustering five stochastic calls from the same model (E-1) tightens accuracy to 19 km, pushing nearly 40% of predictions inside a 10 km radius. At the other end of the spectrum, the heuristic Mordecai-3 pipeline (H-3) and the county/state-centroid fallback (H-4) return mean errors of 94 km and 80 km, respectively, underscoring how much information the language models extract beyond the coarsest gazetteer cues.

[1] The GIS benchmark reflects 6 billable hours of expert labor ($140 ÷ $23.33/hour), encompassing script development, parameter tuning, and quality assurance across two iterations. While the actual calendar time from initial contact to final delivery was longer (49 hours over 3 days), the latency metric uses billable labor time to maintain comparability with LLM reasoning time. This approach measures computational/cognitive effort rather than project coordination overhead, providing a fair comparison between bespoke expert analysis and off-the-shelf model inference. The actual script execution time remains negligible (<1 s per grant) relative to the development and validation effort.

Table 4: Coordinate-accuracy metrics.

ID	Underlying model	Mean ± 95% CI (km)	Median (km)	10 km (%)
E-1	o3-2025-04-16 (ensemble)	18.7 [13.6, 25.0]	12.5	39.5
E-2	ensemble name-redacted	20.4 [15.1, 26.8]	13.8	34.9
M-2	o3-2025-04-16	23.4 [17.4, 29.3]	14.3	30.2
M-5	gpt-4o-2024-08-06	27.9 [22.3, 33.9]	25.0	16.3
M-4	gpt-4.1-2025-04-14	28.5 [22.7, 35.1]	25.4	20.9
T-4	gpt-4.1-2025-04-14 + tools	37.2 [30.1, 45.0]	34.2	16.3
T-1	o4-mini-2025-04-16 + tools	37.6 [30.9, 45.0]	33.6	14.0
M-1	o4-mini-2025-04-16	41.6 [33.8, 50.1]	27.4	7.0
M-6	gpt-3.5-turbo	43.1 [33.8, 54.0]	34.0	4.7
M-3	o3-mini-2025-01-31	50.3 [43.0, 58.6]	48.4	4.7
H-1	human-gis	71.4 [59.1, 85.1]	60.2	4.7
H-2	Stanford NER (GeoTxt)	79.0 [56.3, 109.4]	59.5	7.0
H-4	County Centroid	80.3 [66.0, 95.9]	70.5	4.7
H-3	Mortecai3	94.3 [68.8, 124.6]	55.5	7.0

Wilcoxon signed-rank tests corroborate the ensemble's advantage. Compared with the single-shot o3 run we obtain $W = 629$, $p = 0.03$, while every other baseline registers $W \ 772$, $p \ 4.6 \times 10$.

Figure 1 displays the mean error with corresponding 95% confidence intervals.

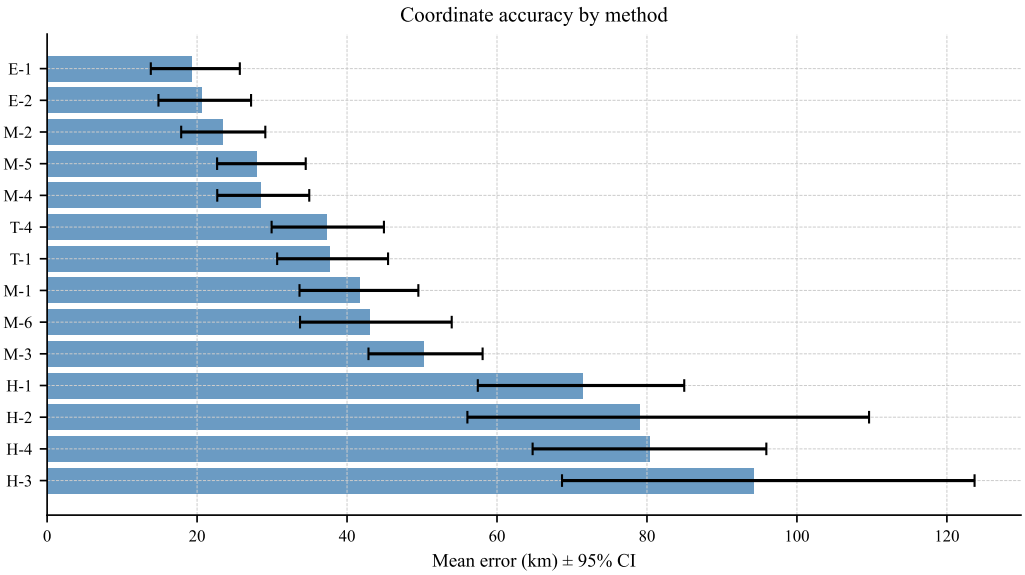


Figure 1: Coordinate accuracy by method

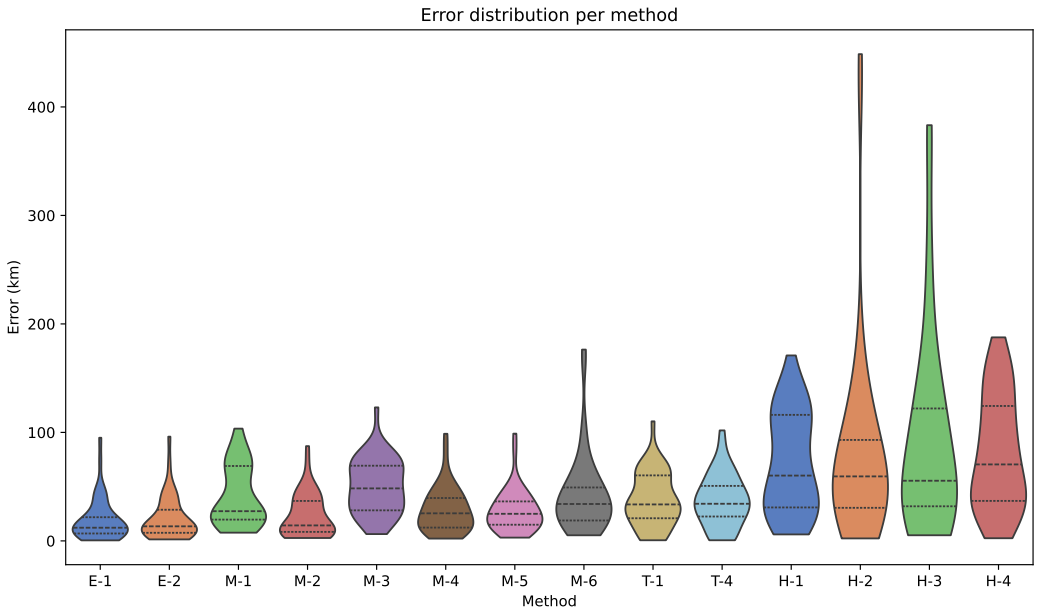


Figure 2: Error Distribution by Method

The violin plot in Figure 2 shows that most LLM errors cluster below 40 km, with a long tail driven by a handful of outliers.

To focus on the head-to-head comparison between the language-model approaches and the strongest non-AI baseline, Figure 3 repeats the violin plot but limits the panel to the six LLM variants and the human–GIS workflow (H-1). Removing the long-tail baselines (county centroids, rule-based NER, etc.) reveals a much tighter performance band: every large-model distribution lies well inside the inter-quartile range of the GIS analyst and displays a shorter upper whisker, underscoring how frequently even weaker LLMs outperform manual geocoding.

Figure 4 presents the cumulative distribution of errors for each evaluated method.

Table 5 examines how varying the *reasoning_effort* parameter within the same o3-2025-04-16 model (M-2) affects spatial accuracy. The differences are minor: mean error shifts by less than 1 km across effort levels, while the share of highly-accurate predictions (10 km) increases by approximately 7 percentage points from low to medium/high effort.

Table 5: Effect of reasoning-effort budget on o3 one-shot accuracy (n = 43).

ID	Underlying model	Mean (km)	Median (km)	10 km (%)	Tokens / entry
M2-low	o3-2025-04-16, low effort	24.8	15.9	28.9	1.1 k
M2-med	o3-2025-04-16, medium effort	24.9	15.1	35.6	3.2 k

ID	Underlying model	Mean (km)	Median (km)	10 km (%)	Tokens / entry
M2-high	o3-2025-04-16, high effort	23.8	15.0	35.6	7.0 k

Three key observations emerge: (1) modern LLMs can match or exceed a trained GIS specialist on this task, (2) supplementing gpt-4.1-2025-04-14 with explicit Google-Maps queries did not improve accuracy—in fact, the tool-chain variant T-4 performed 30% worse than its pure-prompt counterpart, and (3) the amount of chain-of-thought the o3-2025-04-16 model is allowed to emit has only a marginal effect on accuracy.

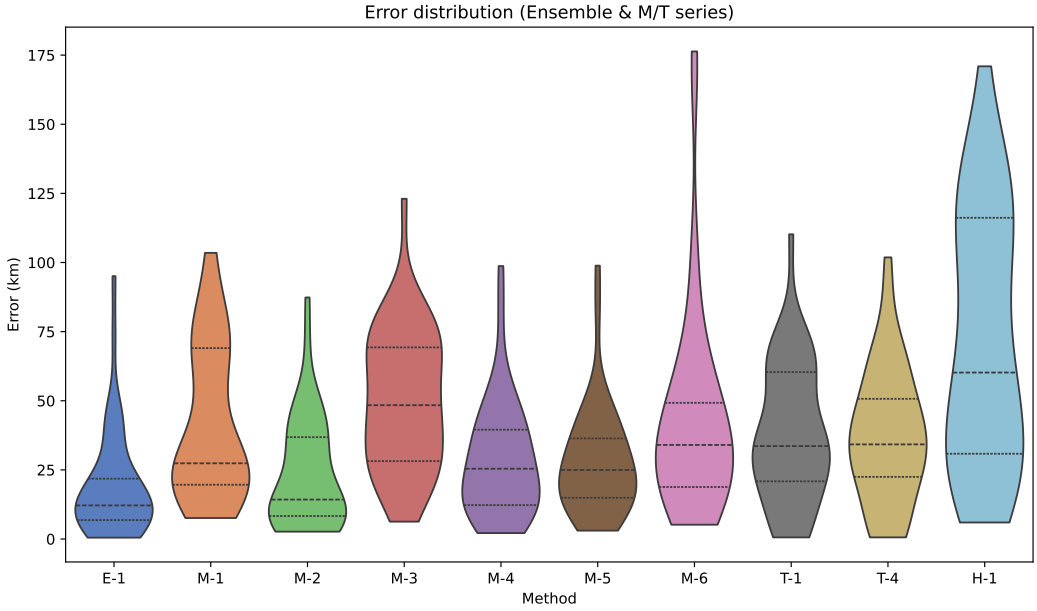


Figure 3: Core error distribution (LLMs vs. GIS baseline)

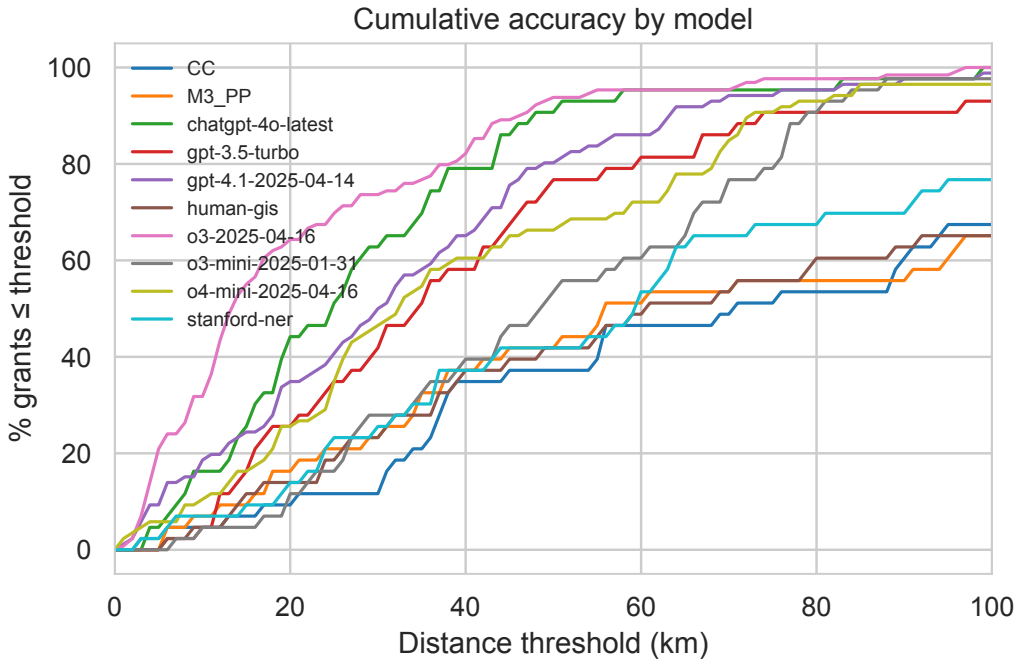


Figure 4: Cumulative error distribution by method

6.2 Cost–Accuracy Trade-off

Figure 5 plots the relationship between monetary cost (per 1,000 grants processed) and accuracy (mean error in kilometers) for each method. All automated variants dominate the GIS script baseline by two to five orders of magnitude on both dimensions. The professional GIS baseline appears in the upper-right quadrant, reflecting its combination of high cost and relatively high error. All automated methods establish a clear Pareto frontier along the bottom edge of the plot, with gpt-4o-2024-08-06 (M-5) delivering the best *dollar-for-accuracy* ratio at USD 1.09 per 1,000 located grants and a mean error under 28 km, despite not achieving the absolute lowest error.

Table 6: Cost efficiency of evaluated methods.

ID	Cost / located (USD)	Cost per 1k	Mean error (km)
E-1	0.19565	195.65	18.7
E-2	0.20003	200.03	20.4
M-2	0.12746	127.46	23.4
M-5	0.00105	1.05	27.9
M-4	0.00046	0.46	28.5
T-4	0.00323	3.23	37.2
T-1	0.01142	11.42	37.7
M-1	0.01069	10.69	41.7
M-6	0.00010	0.10	43.1
M-3	0.01415	14.15	50.3
H-1	3.25581	3,255.81	71.4
H-2	0.00000	0.00	79.0
H-4	0.00000	0.00	80.3
H-3	0.00000	0.00	94.3

The o3-2025-04-16 model (M-2) is more accurate but ~100× costlier than gpt-4o-2024-08-06. Users can therefore choose a point on the Pareto frontier that best balances budget and precision.

6.3 Latency–Accuracy Trade-off

Examining the latency dimension, Figure 11 shows that automatic methods produce coordinates in 0.7–48 seconds of computation time, still three orders of magnitude faster than the GIS analyst's labor time (502 s per grant). This range reflects substantial variation across model families, with the fastest models (gpt-4o-2024-08-06 and gpt-3.5-turbo) requiring less than 1 second per grant, while the o-series models (particularly o3-2025-04-16) taking up to 48 seconds.

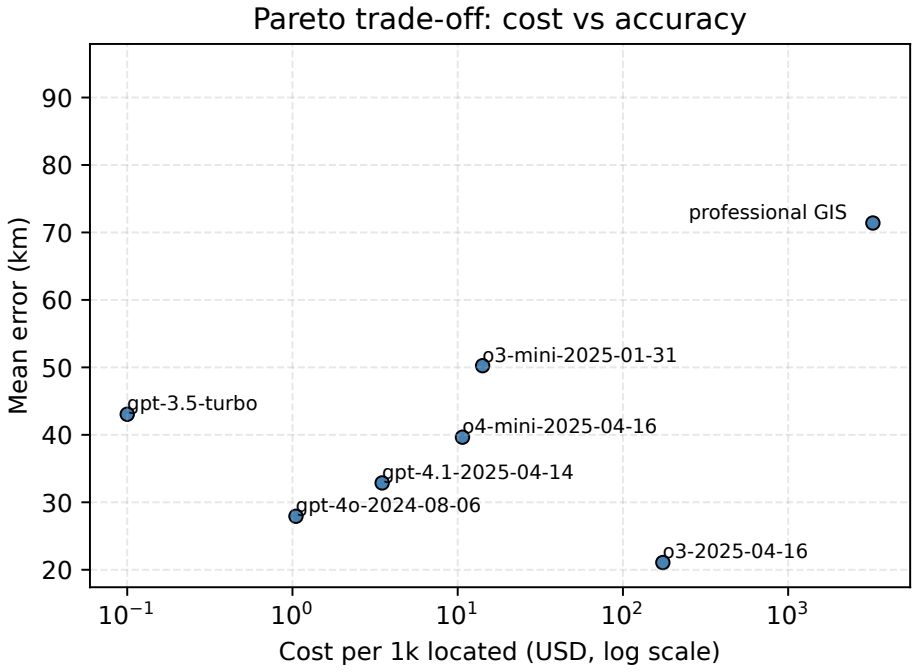


Figure 5: Cost-Accuracy Tradeoff

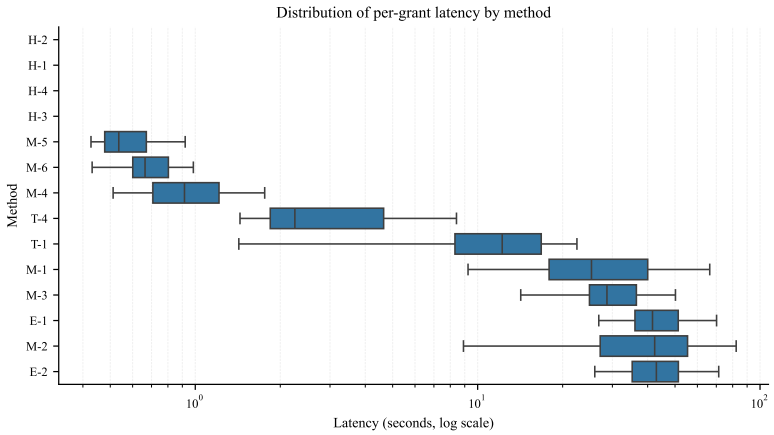


Figure 6: Distribution of per-grant latency by method.

6.4 Qualitative Examples

To illustrate how the two prompting paradigms differ, the chain of thought for grant_04 ("WILLIAM WILLIAMS) is distilled into key stages. Table 4 shows these steps for the tool-chain (T-2) and one-shot (M-2) methods. The full grant text shown to the model was:

> "WILLIAM WILLIAMS, 400 acs., on 8. side of the main Black Water Swamp; by run of Holloway Sw; 24 Apr. 1703, p. 519. Trans. of 8 pers: Note: 8 tights paid for to Wm, Byrd, Esqr., Auditor."

Stage	Tool-Chain (T-2)	One-Shot (M-2)
1. Feature ID	Holloway Branch, Blackwater Swamp	run of Holloway Sw
2. First lookup	geocode_place("Holloway Branch, Blackwater Swamp") → PG County	Mental estimate
3. Mismatch check	Sussex County result → refine to "Holloway Swamp"	—
4. Second lookup	geocode_place("Holloway Swamp, VA")	—
5. Anchor averaging	compute_centroid([…])	—
6. Final output	37.166303, -77.244091	37°00′07.2″N 77°07′58.8″W

Full reasoning chains are available in Section 12.3.

The *one-shot* paradigm asks the model to read the abstract, reason internally, and emit coordinates in a single response. All cognition is "in the head" of the network: it interprets archaic toponyms, performs mental triangulation against its latent world map, and produces a best-guess point estimate. By contrast, the *tool-chain* paradigm externalises part of that reasoning. The model may call a geocoder to retrieve candidate coordinates for surface forms (e.g. "Holloway Swamp"), inspect the returned JSON, run additional look-ups

with spelling variants or county qualifiers, and finally aggregate anchors with a centroid tool. Each call–observe–reflect loop is logged, exposing an auditable chain of evidence. The trade-off is latency and verbosity: ten turns of querying and self-reflection can be slower and, as § 6.1 showed, not necessarily more accurate. Nevertheless, the traces reveal *why* the model settles on a location—useful when historians need to vet or correct automated outputs.

6.5 Tool-usage patterns

Two configurations—T-1 and T-4—were granted access to the external function suite. Their invocation patterns are summarised in Table 8.

Table 8: LLM tool-chain behavior on the 43-grant test set.

Method	Underlying model	Calls / entry (mean)	geo:cent ratio	First-call success
T-1	o4-mini-2025-04-16	3.98	22.86:1	66.7%
T-4	gpt-4.1-2025-04-14	2.23	7.73:1	72.1%

For both pipelines the Google `geocode\_place` endpoint dominated the call mix, whereas the auxiliary `compute\_centroid` function appeared in fewer than one call per ten. gpt-4.1-2025-04-14 (T-4) adopted a more economical strategy, issuing on average 2.3 calls per grant while succeeding on the first query in 73% of cases. The gpt-4o-mini-2025-04-16 model (T-1), by contrast, averaged 4.0 calls with a 67% first-call success rate. This greater query volume manifests as the higher token usage and latency reported in § 6.3, yet it conferred no observable advantage in positional accuracy (§ 6.1).

6.6 Robustness / Ablation Studies

Several additional analyses were conducted to test the robustness of the main findings:

- Outlier-robust summary – Excluding the five largest residuals (top 11% of errors) lowers the overall mean error from 38.5 km to 36.9 km. Method rankings and 95% CIs remain unchanged; only H-1 [2] (6.6 km) and M-6 (6.3 km) show material shifts, leaving M-2 as the top performer.
- Patentee-name redaction (E-2) – To test for possible *training-data contamination*, the five-call o3 ensemble (E-1) was rerun after masking every patentee name in the abstract with `[NAME]`. If the model had memorised grant-name coordinate pairs from its pre-training corpus, removing this cue should have caused a sharp accuracy drop. In practice mean error rose only slightly—from 18.7 km to 20.4 km—and the 10 km hit-rate fell by just 4.6 pp (39.5 $\rightarrow$ 34.9). The mild degradation indicates that the model is drawing on the descriptive toponyms and spatial clues in the text rather than retrieving memorised locations, providing no evidence of leakage from the historical grants themselves.
- Temperature sweep – Four temperatures (0.0 / 0.4 / 0.8 / 1.2) were evaluated for the one-shot prompt on gpt-4.1-2025-04-14 (M4) and gpt-4o-2024-08-06 (M5). Mean error

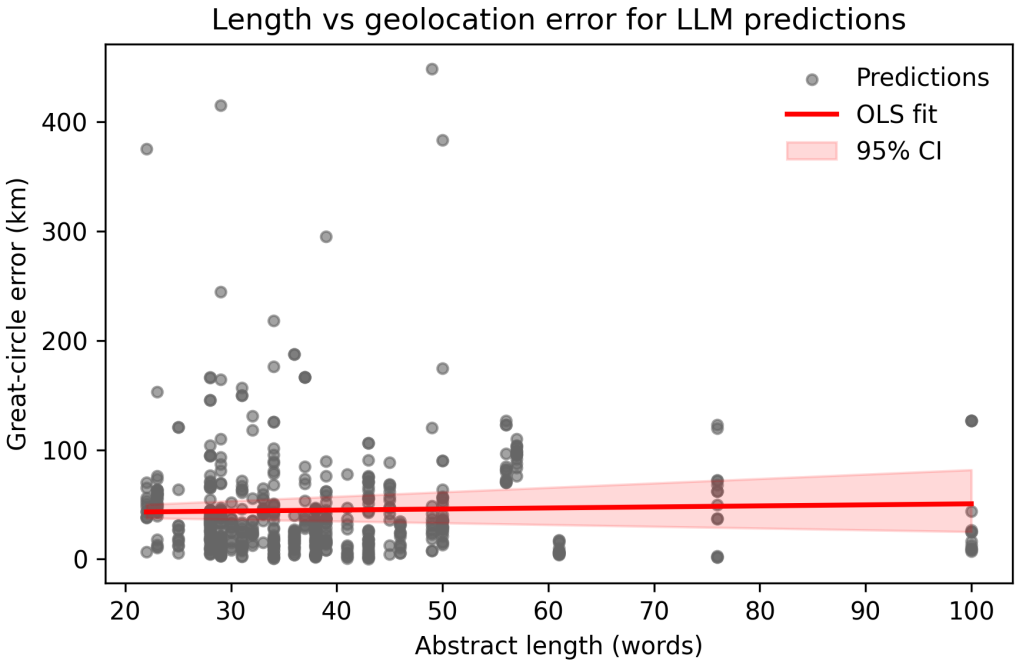


Figure 7: Length vs. Error

for gpt-4.1-2025-04-14 varied narrowly between 34 km (t=0.0) and 31.7 km (t=0.8), indicating a shallow optimum around 0.8. gpt-4o-2024-08-06 showed no systematic trend (32–33 km across the grid).

- Length-stratified accuracy – To test whether verbose abstracts make the task easier (or harder), the word-count of each grant's full text in the validation file was measured and 152 LLM predictions were analyzed:

  - Median split — "Short" (36 words) vs "long" (> 36 words) abstracts yielded mean errors of 36.8 km and 34.9 km respectively (95% CIs overlap), indicating no practical difference.

  - Continuous fit — An ordinary-least-squares regression (error$\{km\}$=42.3-0.18,length{words}) gives a slope of –0.18 km ± 0.44 km (95% CI) per extra word with R^2 = 0.004 and Pearson r = –0.06. Figure 7 visualizes the scatter and confidence band.

These results suggest that, within the 25–60-word range typical of the corpus, abstract length explains essentially none of the variation in LLM accuracy.

7 Discussion

7.1 Implications for Digital History

This study demonstrates that contemporary large language models (LLMs) can geolocate seventeenth- and eighteenth-century Virginia land patents with accuracy comparable to or exceeding a professional GIS workflow, while substantially reducing costs and labor. The best-performing single-call model (o3-2025-04-16) achieves a mean error of 23 km, sufficient to localize most patents accurately within their respective river basin or county boundaries. Such precision is adequate for macro-scale historical inquiries into settlement patterns, plantation economies, and Indigenous land dispossession, significantly reducing reliance on extensive manual archival GIS labor. Importantly, since the input format consists solely of narrative text, this georeferencing pipeline is readily transferable to subsequent volumes of land grants or to neighboring colonial datasets with structurally similar metes-and-bounds descriptions.

Further incremental improvements are attainable through modest technical refinements. A five-call stochastic ensemble of the same o3 model, integrated via DBSCAN clustering, reduces mean error to 19 km, representing an 18% accuracy improvement at a marginal incremental cost of USD 0.20 and approximately 3 seconds of additional latency per grant. Paired Wilcoxon signed-rank tests confirm the statistical significance of this enhancement ($W = 629$, $p = 0.03$ versus the single-shot model). Such methodological refinements illustrate a pathway for rigorous yet computationally efficient digital historical analyses.

Nevertheless, critical epistemological limitations must be acknowledged. Even optimized models occasionally yield significant positional errors exceeding 100 km, and the absence of inherent uncertainty metrics in predictions complicates downstream historical and spatial interpretations. Leveraging models like gpt-4o-2024-08-06, which provide subsecond latency and near-negligible costs (~USD 0.001 per prediction), further supports the viability of fully automated workflows at scale.

More broadly, these results affirm the emerging paradigm of "machine-assisted reading" within digital humanities scholarship, where historians retain interpretive and analytical authority while delegating repetitive and data-intensive extraction tasks to robust computational models. This model not only accelerates research workflows but also expands methodological possibilities within historical spatial analysis, offering scalable and reproducible approaches to the quantitative study of early-modern archives.

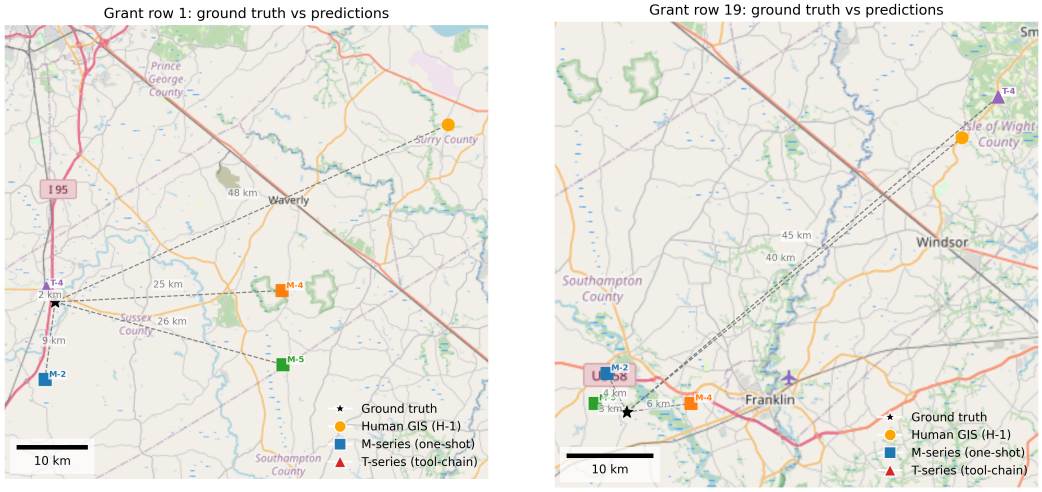
7.2 Error Analysis & Failure Modes

Inspection of the largest residuals uncovers three recurring failure modes:

1. **Obsolete or ambiguous toponyms.** Grants referencing now-extinct mill sites, plantations, or historical administrative divisions frequently produce erroneous matches to contemporary geographic entities. This ambiguity is amplified when models fail to contextualize place-names within county boundaries or historical frameworks. A notable example involves the Stanford Named Entity Recognition (NER) method, which processed references to "St. Paul's Parish" by correctly identifying "St. Paul" as a Virginia geographic entity. However, the GeoNames API subsequently matched this to the modern town of Saint Paul, located approximately 400 km from the in-

tended historical Anglican parish in central Virginia. This misplacement highlights the fundamental issue that modern gazetteers contain toponyms whose geographic or administrative meanings have significantly shifted over centuries, illustrating the critical gap between algorithmic geocoding and historical geographic knowledge.

2. **Bearing-only metes-and-bounds descriptions.** Some abstracts give nothing beyond a perimeter walked from one neighbour or landmark to the next—for example, the John Pigg patent that "beg[ins] in the path from William Rickett's house to the Indian towne; to Capt. William Smith … to land where John Barrow liveth … to the Ridge Path … along Watkins's line … to Maj. Payton." Because there is no unambiguous place-name anchor, both the LLM and gazetteer-driven baselines must rely on weak contextual cues, and median errors for these deeds rise above 70 km.
3. **Cascading search bias.** Tool-enabled runs introduce an additional failure channel: once the first `geocode\_place` call returns a spurious coordinate, subsequent `compute\_centroid` operations often average anchors that are already flawed, locking in the error. Raising the threshold for calling the centroid function—or providing the model with a quality heuristic—may mitigate this issue.



(a) Success example: Grant 1.

(b) Failure-mode example: Grant 19.

Figure 8: Grant examples: Grant 1 (left) shows a success case where the o3 model (M-2) and tool-chain gpt-4.1 (T-4) are close to ground truth. Grant 19 (right) illustrates a failure mode where an early spurious geocoder hit sends the tool-chain prediction far from ground truth, whereas the unguided model remains closer to the actual location. Basemap © OSM.

In Grant 1 (LEWIS GREEN), language-only inference (M-2) achieves county-level precision (9 km error), and the tool-chain (T-4) further reduces the error to just 1.5 km. In Grant 19, a spurious geocoder hit sends the tool-chain prediction far from ground truth, whereas the unguided models remain within a reasonable distance—a pattern that typifies the cascading search bias described above.

These examples visually reinforce the key finding that sophisticated language models like o3 already encode substantial geographic knowledge about Virginia's colonial land-

scape, often placing grants within their correct watershed without external reference data. The full contact sheet showing all 43 mapped grants appears in Appendix C.

7.3 Cost–Benefit Considerations

From a budgetary standpoint, all automatic methods lie on a markedly superior frontier relative to the traditional GIS baseline: the cheapest model (gpt-3.5-turbo) reduces cost per located grant by four orders of magnitude, while the most accurate (o3-2025-04-16) still delivers a >20× saving. Latency gains are equally pronounced, shrinking a six-hour task to seconds.

The choice of inference strategy therefore hinges on the marginal utility of each additional kilometre of accuracy. Projects that can tolerate a 30 km error band will find gpt-4o-2024-08-06 delivers near-real-time throughput at a negligible cost. Where higher precision is required, two graduated options emerge. First, a single pass of o3-2025-04-16 at its default medium reasoning budget achieves a mean error of 23 km for roughly $0.13 per deed. Second, stacking five low-temperature, low-reasoning calls of the same o3 model and clustering them with DBSCAN (method E-1) pushes mean error down to 19 km at a per-grant cost of $0.20. Because the ensemble averages away the occasional outlier, each component call can run with reasoning_effort = low (1.1 k tokens) instead of medium (3.2 k tokens), so the accuracy gain is bought primarily with additional parallel calls rather than a larger context window. Table 5 shows that raising reasoning_effort in a single call trims mean error by less than 1 km yet triples token usage, whereas the ensemble suppresses outliers more cost-effectively.

In practical terms, gpt-4o defines the speed–and–cost vertex, o3 single-shot defines the mid-range accuracy vertex, and o3 five-call ensemble occupies the extreme accuracy corner of the Pareto frontier. All three pipelines scale linearly with corpus size, so statewide geocoding—tens of thousands of patents—remains feasible on a modest humanities budget, provided researchers calibrate model choice to their required spatial tolerance.

8 Limitations

Several caveats temper the preceding claims.

1. **Dataset scope.** Only 43 of the 125 test abstracts possessed authoritative ground-truth coordinates, and all derive from a single printed volume (1695-1739). While this sample reflects rigorous archival verification standards to prevent convenience bias (see § 3.3), the reported error statistics may under- or overstate performance on earlier or later patent books, or on neighboring colonies with different toponymic conventions. Future work with expanded ground-truth sets obtained through similar verification methods will further validate these findings.
2. **Training data contamination.** While the historical patent abstracts themselves do not contain coordinate information, the models may have been exposed to the GIS datasets used to establish ground truth coordinates. The Central VA Patents GIS layer developed by One Shared Story [22] and other spatial datasets used for verification (see § 3.3) could potentially appear in training corpora, allowing models to retrieve memorized coordinates rather than demonstrate spatial reasoning from textual descriptions. Standard contamination detection methods have significant limitations for

spatial datasets and often provide unreliable results, so this study acknowledges contamination as a plausible alternative explanation for model performance. This represents a significant limitation that should be addressed in future work through systematic contamination analysis or evaluation on guaranteed unseen spatial datasets.

3. **OCR and transcription noise.** Although the best-performing OCR pipeline available was applied, minor character errors persist. Because the language models ingested this noisy text directly, a fraction of the residual error may stem from imperfect input rather than conceptual failure.
4. **Point-estimate evaluation.** Single latitude/longitude pairs were benchmarked, ignoring shape reconstruction and parcel acreage. Applications that require boundary polygons will need supplementary modelling or manual intervention.
5. **Tool bias.** Google's geocoder is optimised for modern place names; its deterministic output may shift marginally over time as the underlying database updates, complicating longitudinal reproducibility.
6. **GIS benchmark generality.** The benchmark [2] relies on a single expert-authored geocoding procedure; accuracy might vary with different gazetteer sources, parameter tuning, or analyst expertise. The baseline is treated as representative of standard practice rather than an upper bound on professional GIS performance. The single-practitioner results are intended as an illustrative comparison point rather than a statistically powered estimate of professional accuracy or throughput.
7. **Cost assumptions.** Monetary estimates are tied to the May-2025 OpenAI pricing schedule (see Table 3); rate changes would alter the cost frontier.

9 Future Work

Building on the present findings, several avenues warrant exploration.

- **Corpus expansion.** Digitising the remaining volumes of *Cavaliers and Pioneers* [1]—and analogous land books from Maryland and North Carolina—would permit a cross-colonial analysis of settlement diffusion.
- **Prompt engineering at scale.** A reinforcement-learning loop that scores predictions against partial gazetteers could iteratively refine prompts or select between tool and non-tool paths.
- **Polygon recovery.** Combining the model's point estimate with chained GIS operations (bearing decoding, river buffering) could approximate parcel outlines, unlocking environmental history applications.
- **Human-in-the-loop interfaces.** Lightweight web tools that display the model's candidate coordinates alongside archival imagery would enable rapid expert validation and correction.

10 Conclusion

This study delivers the first rigorous benchmark of large language models on the long-standing problem of geolocating early-modern Virginia land patents directly from their narrative metes-and-bounds abstracts. A new, copyright-compliant dataset of 5 471 transcribed grants and 43 gold-standard coordinates accompanies a reproducible evaluation

framework that compares six OpenAI model variants against four deterministic or human baselines.

The best single-call configuration—o3-2025-04-16 with a one-shot prompt—achieves a mean great-circle error of 23 km, cutting the professional GIS benchmark by 67 % and the Stanford NER geoparser by 70 %. A lightweight five-call ensemble of the same model, run at low reasoning effort and fused with DBSCAN, reduces mean error further to 19 km while adding only US $0.07 and three seconds per deed. In contrast, the fastest model (gpt-4o-2024-08-06) incurs a negligible cost of US $0.001 per grant and still stays within a 30 km error band, defining a new speed–cost frontier.

These findings place colony-scale georeferencing of archival land records within reach of modest research budgets: mapping the entire Cavaliers and Pioneers corpus—tens of thousands of patents—now requires hours and tens of dollars rather than months and thousands. Because the pipeline operates on plain text, it can be ported verbatim to other volumes, neighbouring colonies, or similarly structured deed books worldwide.

Future work can capitalise on the released corpus and code by extending the benchmark to polygon reconstruction, integrating Indigenous spatial data, and testing open-source LLMs fine-tuned on historical prose. For digital historians, archaeologists, and GIScientists alike, the results substantiate LLM-assisted geocoding as an accurate, transparent, and economically viable alternative to traditional manual workflows—opening a scalable path toward fully spatially enabled colonial archives.

11 Acknowledgements

This work builds upon the meticulous archival research of Nell Marion Nugent, whose *Cavaliers and Pioneers* abstracts have preserved Virginia's colonial land records for generations of scholars, and the rigorous GIS work of One Shared Story, whose Central VA Patents dataset enabled the ground truth evaluation essential to this study. The authors are deeply grateful to Bimbola Bashorun [2] for providing the professional GIS benchmark that was crucial to evaluating model performance.

The authors also thank the Library of Virginia and the Virginia Surveyor's Office for granting access to their digital archives and land patent collections, which made the ground-truth dataset possible.

, the authors are indebted to the digital humanities community whose ongoing conversations about LLMs and historical research have informed this project's methodological approach.

12 Supplementary Methods & Materials

12.1 OCR & Text-Normalisation Pipeline

The corpus preparation described in §3.2 comprised a multi-stage optical character recognition (OCR) and text normalisation pipeline. *Cavaliers and Pioneers* Volume 3 was scanned at 600 DPI, yielding high-resolution page images in PDF format.

OCR parameters were optimized through controlled experiments with Tesseract engine modes and page segmentation configurations, ultimately selecting LSTM neural network processing (OEM 3) with fully automatic page segmentation (PSM 3) based on quantitative

text extraction metrics. The OCR workflow employed OCRmyPDF with page rotation detection, document deskewing, and custom configurations to preserve period-appropriate spacing patterns.

Post-OCR text normalisation included: (1) removal of running headers and pagination artifacts, (2) contextual dehyphenation of line-break-split words, and (3) structural parsing to isolate individual land grant abstracts. Quality control involved manual inspection focusing on toponym preservation, with spot-checking indicating character-level accuracy exceeding 98% for toponyms. The processed corpus was then exported to CSV format for geolocation analysis.

12.2 Prompts and Model Configurations

One-Shot Prompt (M-series) The M-series models utilized a minimal one-shot prompt designed to elicit precise coordinate predictions:

```
Geolocate this colonial Virginia land grant to precise latitude and
    longitude coordinates.
Respond with ONLY the coordinates in this format: [DD][MM]'[SS].[SSSSS]"N
     [DDD][MM]'[SS].[SSSSS]"W
```

Tool-Augmented System Prompt (T-series) For tool-augmented models, a structured system prompt was employed that defined available tools, workflow, and constraints:

```
You are an expert historical geographer specialising in colonial-era
    Virginia land records.
Your job is to provide precise latitude/longitude coordinates for the
    land-grant description the user supplies.

Available tools
 `geocode_place(query, strategy)`
     Look up a place name via the Google Geocoding API (Virginia-
    restricted).
     Returns JSON: `{lat, lng, formatted_address, strategy, query_used}`.
 `compute_centroid(points)`
     Accepts **two or more** objects like `{lat: 37.1, lng: -76.7}` and
    returns their average.

Workflow
0. Craft the most specific initial search string you can (creek, branch,
    river-mouth, parish, neighbor surname + county + "Virginia").

1. Call `geocode_place` with that string. If the result is in the expected
     or an adjacent county *and* the feature lies in Virginia (or an NC
    border county), treat it as **plausible**. A matching feature keyword
    in `formatted_address` is *preferred* but not mandatory after several
    attempts.

2. If the first call is not plausible, iteratively refine the query (
    alternate spelling, nearby landmark, bordering county, etc.) and call
    `geocode_place` again until you obtain *at least one* plausible point
    **or** you have made six tool calls, whichever comes first.
```

```
3. Optional centroid use  if the grant text clearly places the tract *
    between* two or more natural features (e.g., "between the mouth of
    Cypress Swamp and Blackwater River") **or** you have two distinct
    plausible anchor points (creek-mouth, swamp, plantation), you may call
     `compute_centroid(points)` exactly once to average them. Otherwise
    skip this step.

4. You may make up to **ten** total tool calls. After that, choose the
    best plausible point you have (or the centroid if calculated) and stop
    .

5. Final answer  reply with **only** the coordinates in decimal degrees
    with six digits after the decimal point, e.g., `36.757059,
    -77.836728`. No explanatory text.

Important rules
 Always perform at least one successful `geocode_place` call before any
    other tool.
 Invoke `compute_centroid` only when you already have two or more
    plausible anchor points and averaging will help locate a "between"
    description.
 Never invent coordinatesderive them from tool output.
 Return no explanatory text, symbols, or degree signsjust `lat, lon`.
```

Model Configurations Table A1 summarizes the model variants and hyperparameter configurations used in the experiment:

Method ID	Model	Pipeline	Temperature	Reasoning Effort
M-1	o4-mini-2025-04-16	one_shot		medium
M-2	o3-2025-04-16	one_shot		medium
M-3	o3-mini-2025-01-31	one_shot		medium
M-4	gpt-4.1-2025-04-14	one_shot	0.2	
M-5	gpt-4o-2024-08-06	one_shot	0.2	
M-6	gpt-3.5-turbo	one_shot	0.2	
T-1	o4-mini-2025-04-16	tool_chain		low
T-4	gpt-4.1-2025-04-14	tool_chain	0.2	
E-1	o3-2025-04-16 (ensemble)	one_shot		low
E-2	o3-2025-04-16 (name-redacted)	one_shot		low

12.3 Example Chain-of-Thought Trace (T-4)

Below is a representative example of the tool-augmented chain of thought from from method T-2 (o3-2025-04-16 with tool access). This trace demonstrates the model's reasoning process, including query formulation, tool call selection, and coordinate determination for a colonial Virginia land grant.

Input Abstract:

```
WILLIAM WILLIAMS, 400 acs., on S. side of the main Black Water Swamp; by
    run of Holloway Sw; 24 Apr....
```

Tool-Augmented Reasoning Chain:

1. **Locating Key Features**

```
Reasoning: I need to parse a land grant description from 1703
    mentioning "the S side of the main
Black Water Swamp" and locate the southwestern side of it. I know
    Blackwater Swamp is in
southeastern Virginia, specifically in Prince George County, and
    flows into the Blackwater River.
There's a part of it in Surry County too. I'm curious if Holloway
    Swamp is a small tributary.

Tool: geocode_place
Query: "Holloway Swamp, Sussex County, Virginia"
Result: {
  "lat": 36.9058167,
  "lng": -77.2405153,
  "formatted_address": "Sussex County, VA, USA"
}
```

2. **Cross-Referencing Features**

```
Reasoning: I've noted the coordinates for Holloway Swamp in Sussex
    County. Next, I want to confirm
the location of Blackwater Swamp's larger trunk. The description
    indicates it's on the south side
of the main Black Water Swamp, near the meeting point of Holloway
    Swamp.

Tool: geocode_place
Query: "Blackwater Swamp, Sussex County, Virginia"
Result: {
  "lat": 37.10810973,
  "lng": -77.15139208,
  "formatted_address": "Blackwater Swamp, Virginia 23842, USA"
}
```

3. **Disambiguating Locations**

```
Reasoning: I'm looking at the coordinates 37.1081, -77.1514, which
    seem to place me in northeastern
Sussex County near the Prince George border. I've noted Holloway
    Swamp's coordinates at 36.9058,
-77.2405, about 0.2 degrees south and slightly west.

Tool: geocode_place
Query: "Holloway Swamp and Blackwater Swamp confluence, Virginia"
Result: {
  "lat": 37.4315734,
  "lng": -78.6568942,
```

```
  "formatted_address": "Virginia, USA"
}
```

4. **Further Refinement**

```
Tool: geocode_place
Query: "Holloway Branch, Blackwater Swamp, Prince George County,
    Virginia"
Result: {
  "lat": 37.1733,
  "lng": -77.2396666,
  "formatted_address": "Blackwater Dr, Virginia 23842, USA"
}

Reasoning: I found the coordinates 37.1733, -77.2397, which might be
    along Blackwater Dr, near Waverly.
But I noticed the coordinates for Holloway Swamp at 36.9058, which is
     in southwestern Sussex,
so there's a mismatch. It's possible there could be two features
    named Holloway Swamp.
```

5. **Testing Alternate Spellings**

```
Tool: geocode_place
Query: "Halloway Swamp, Virginia"
Result: {
  "lat": 37.4315734,
  "lng": -78.6568942,
  "formatted_address": "Virginia, USA"
}

Tool: geocode_place
Query: "Holloway Swamp, Prince George County, Virginia"
Result: {
  "lat": 37.1593052,
  "lng": -77.2405153,
  "formatted_address": "Prince George County, VA, USA"
}
```

6. **Calculating Spatial Midpoint**

```
Tool: geocode_place
Query: "Blackwater Swamp, Prince George County, Virginia"
Result: {
  "lat": 37.1733,
  "lng": -77.2396666,
  "formatted_address": "Blackwater Dr, Virginia 23842, USA"
}

Reasoning: I'm analyzing the coordinates: lat 37.1593, long -77.2405
    and lat 37.1733, long -77.2397.
The latitude difference of about 0.051 equals roughly 5.6 km north,
    while the longitude difference
is approximately 7.4 km. Since I need the "south side of Blackwater
    Swamp," I should compute
```

```
a centroid that reflects this relationship.

Tool: compute_centroid
Points: [
  {"lat": 37.1733, "lng": -77.2396666},
  {"lat": 37.1593052, "lng": -77.2405153}
]
Result: {
  "lat": 37.16630260,
  "lng": -77.24009098
}
```

Final Coordinate Output: `37.166303, -77.240091`

This trace demonstrates how the tool-chain approach systematically refines its search, testing multiple geographic hypotheses before settling on a final coordinate. The model first locates the two key features (Blackwater and Holloway Swamps), then resolves ambiguities through both county-specific queries and alternate spellings, finally computing a centroid between the two most plausible anchor points.

12.4 Function & Tool Specifications

Two JSON-Schema tools extend the language model's native reasoning with external geographic capabilities. The schemas are injected into the OpenAI *Responses* request via the `tools` parameter, allowing the model to emit `function\_call` objects whose arguments are validated and then executed by the evaluation driver. After execution the Python backend streams a `function\_call\_output` item containing the tool's JSON result, which the model can read on the next turn—in a repeated action-observation loop.

```
// geocode_place  wrapper around Google Geocoding API (Virginia-
    restricted)
{
  "type": "function",
  "name": "geocode_place",
  "description": "Resolve a place description to coordinates.",
  "parameters": {
    "type": "object",
    "properties": {
      "query": {
        "type": "string",
        "description": "Free-form geocoding query, e.g. 'Blackwater River
   , Isle of Wight County'."
      },
      "strategy": {
        "type": "string",
        "enum": [
          "natural_feature", "restricted_va", "standard_va", "
   county_fallback"
        ],
        "description": "Search heuristic controlling how the backend
   constructs variant queries."
```

```
      }
    },
    "required": ["query"]
  }
}
```

The driver maps the call to `google\_geocode()` with a hard-coded `components=administrative\_area:VA` filter, discards results falling outside Virginia, and returns a trimmed JSON object `\{lat, lng, formatted\_address, strategy, query\_used\}`. A single tool therefore exposes the entire Google Places knowledge graph while keeping the model sandboxed from the broader web.

```
// compute_centroid  spherical mean of 2 anchor points
{
  "type": "function",
  "name": "compute_centroid",
  "description": "Return the centroid (average lat/lng) of two or more
    coordinate points.",
  "parameters": {
    "type": "object",
    "properties": {
      "points": {
        "type": "array",
        "minItems": 2,
        "items": {
          "type": "object",
          "properties": {
            "lat": {"type": "number"},
            "lng": {"type": "number"}
          },
          "required": ["lat", "lng"]
        }
      }
    },
    "required": ["points"]
  }
}
```

The backend converts each `(lat, lng)` pair to a unit-sphere Cartesian vector, averages the components, and projects the mean vector back to geographic coordinates—an approach that avoids meridian-wrap artefacts and preserves accuracy for points separated by >100 km.

Interaction Pattern

1. *Planning.* The assistant reasons in natural language and decides whether a geocoder query is necessary.
2. *Invocation.* It emits a `function\_call` with the chosen arguments. The evaluation script records the call for later provenance analysis.
3. *Execution & Observation.* The Python backend executes the call, returning a JSON payload as a `function\_call\_output` message appended to the conversation.

4. *Reflection.* Reading the payload, the model either (i) issues a refined query, (ii) averages multiple anchors via `compute\_centroid`, or (iii) produces a final coordinate string.

This structured loop allows the model to chain up to ten tool calls and records every intermediate query, result, and internal rationale.

12.5 Evaluation Driver & Code Repository

All experiments are orchestrated by a single Python script, `run\_experiment.py`, which exposes a reproducible command-line interface (CLI) for selecting the evaluation set, method roster, and runtime flags (e.g., `--dry-run`, `--max-rows`). The driver

- loads method and prompt definitions from YAML,
- initialises the OpenAI client with deterministic seeds,
- executes each model–abstract pair in sequence, proxying any `function\_call` requests to the tool back-end described above,
- logs raw API traffic—including intermediate tool traces—to `runs/<method>/calls.jsonl`, and
- emits both row-level results (`results\_<evalset>.csv`) and a Markdown run report summarising accuracy, cost, and latency.

This tight integration between evaluation logic and provenance logging ensures that every coordinate prediction in the paper can be reproduced from first principles using the open-source code. A public repository containing the driver, prompts, ground-truth data, and analysis notebooks is available at [RepositoryURLremovedforblindreview].

13 Extended Results

13.1 Detailed Accuracy Metrics

Table 10 provides comprehensive error statistics for each evaluated method, including confidence intervals derived from bootstrap resampling (10,000 iterations). The best-performing automated approach, M-2 (o3-2025-04-16), achieves a mean error of 23.39 km with a 95% confidence interval of [17.57, 29.54] km.

Table 10: Mean error with 95% confidence intervals for each method.

Method	Model	n	Mean km	95% CI
E-1	o3-2025-04-16 (ensemble)	43	19.24	[13.60, 24.97]
E-2	ensemble name-redacted	43	20.57	[15.08, 26.83]
H-1	human-gis	43	71.40	[59.14, 85.11]
H-2	Stanford NER (GeoTxt)	43	79.02	[56.32, 109.38]
H-3	Mortecai3	43	94.28	[68.81, 124.61]
H-4	County Centroid	43	80.33	[65.96, 95.88]
M-1	o4-mini-2025-04-16	43	41.65	[33.77, 50.11]
M-2	o3-2025-04-16	43	23.39	[17.37, 29.25]

Method	Model	n	Mean km	95% CI
M-3	o3-mini-2025-01-31	43	50.25	[43.02, 58.63]
M-4	gpt-4.1-2025-04-14	43	28.51	[22.68, 35.10]
M-5	gpt-4o-2024-08-06	43	27.93	[22.31, 33.85]
M-6	gpt-3.5-turbo	43	43.05	[33.78, 53.98]
T-1	o4-mini-2025-04-16 + tools	43	37.65	[30.86, 44.99]
T-4	gpt-4.1-2025-04-14 + tools	43	37.23	[30.11, 45.03]

13.2 Performance by Method

Table 11 provides the complete performance statistics for each method, including variance measures and accuracy bands. The "X km" columns show the percentage of predictions within X kilometers of ground truth.

Table 11: Detailed performance metrics by method.

Method	n	mean	median	sd	min	Q1	Q3	max	10 km	25 km	50 km
E-1	43	18.71	12.45	19.36	1.77	5.60	23.87	100.90	39.5%	76.7%	95.3%
E-2	43	20.44	13.75	18.87	1.11	7.41	24.05	95.34	34.9%	76.7%	95.3%
H-1	43	71.40	60.20	45.65	5.98	30.63	117.50	170.95	4.7%	18.6%	41.9%
H-2	43	79.02	59.45	88.32	2.29	29.05	94.67	448.66	7.0%	23.3%	41.9%
H-3	43	94.28	55.45	91.73	5.20	30.63	123.12	383.18	7.0%	20.9%	41.9%
H-4	43	80.33	70.49	51.46	2.46	36.82	125.61	187.61	4.7%	11.6%	37.2%
M-1	43	41.65	27.39	27.32	7.59	18.45	70.04	103.49	7.0%	37.2%	62.8%
M-2	43	23.39	14.27	19.86	2.67	8.17	36.85	87.35	30.2%	60.5%	93.0%
M-3	43	50.25	48.40	24.93	6.29	27.36	69.53	123.04	4.7%	16.3%	53.5%
M-4	43	28.51	25.42	20.77	2.14	12.09	40.49	98.72	20.9%	48.8%	86.0%
M-5	43	27.93	24.97	19.46	3.03	14.66	37.25	98.86	16.3%	51.2%	90.7%
M-6	43	43.05	34.02	36.07	5.17	17.11	49.74	176.33	4.7%	34.9%	76.7%
T-1	43	37.65	33.61	24.54	0.59	18.57	62.30	110.19	14.0%	32.6%	69.8%
T-4	43	37.23	34.22	23.94	0.59	21.78	53.35	101.85	16.3%	32.6%	74.4%

13.3 Cost-Accuracy Trade-off

Table 12 examines the cost-accuracy relationship, emphasizing the economic efficiency of gpt-4o-2024-08-06, which achieves near-top performance at just $1.05 per 1,000 grants processed. "Cost per +1% 10 km hit" indicates the marginal cost of improving high-precision prediction rate by one percentage point.

Table 12: Cost-accuracy trade-off by model family.

Model	Mean error km	10 km hit-rate	Cost per 1k located (USD)	Cost per +1% 10 km hit (USD)
o3-2025-04-16 (Ensemble)	18.71	39.5%	$195.65	$4.95
o3-2025-04-16	23.39	30.2%	$127.46	$4.22
gpt-4o-2024-08-06	27.93	16.3%	$1.05	$0.06
gpt-4.1-2025-04-14	32.87	18.6%	$3.49	$0.19
o4-mini-2025-04-16	39.65	10.5%	$10.69	$1.02
gpt-3.5-turbo	43.05	4.7%	$0.10	$0.02
o3-mini-2025-01-31	50.25	4.7%	$14.15	$3.04
human-gis	71.40	4.7%	$3,255.81	$700.00

13.4 Processing Time Analysis

Table 13 quantifies the latency advantage of automated methods over traditional GIS workflows. "Speedup" shows relative improvement over the traditional GIS baseline.

Table 13: Processing time by model.

Model	Hours per located	Hours per 1k located	Speedup vs. baseline
o3-2025-04-16 (Ensemble)	0.0128	12.819	17×
gpt-4o-2024-08-06	0.0002	0.178	1,219×
gpt-3.5-turbo	0.0002	0.225	964×
gpt-4.1-2025-04-14	0.0003	0.295	736×
o3-2025-04-16	0.0121	12.060	18×
o3-mini-2025-01-31	0.0085	8.523	25×
o4-mini-2025-04-16	0.0091	9.145	24×
human-gis	0.2170	216.977	1×

13.5 Token Usage Statistics

Table 14 provides detailed token consumption data across all models, offering insight into computational efficiency.

Table 14: Token consumption by model across all test runs.

Model	Input tokens	Output tokens	Tokens per 1k located
o3-2025-04-16 (Ensemble)	33,265	210,681	10,441,907
gpt-4o-2024-08-06	6,698	900	176,698
gpt-3.5-turbo	6,773	820	176,581
gpt-4.1-2025-04-14	142,258	4,193	1,702,919

Model	Input tokens	Output tokens	Tokens per 1k located
o3-2025-04-16	6,653	146,085	6,923,209
o3-mini-2025-01-31	6,653	142,020	6,739,372
o4-mini-2025-04-16	274,903	146,590	6,340,337

Tool-augmented methods consumed on average 1.49× more tokens than pure-prompt counterparts (4,985,953 vs. 3,355,078 tokens per 1,000 located grants). However, this effect varied dramatically by model architecture: adding tools to gpt-4.1-2025-04-14 increased token usage by 18.3× (176,698 → 3,229,140), while o4-mini showed only a 1.14× increase (5,937,907 → 6,742,767).

13.6 Professional GIS Benchmark Analysis

Table 15 provides a more detailed analysis of the professional GIS benchmark (H-1) [2] results, categorized by precision level. This breakdown reveals that even with expert domain knowledge and access to specialized historical gazetteers, more than 41.9% of the human-geocoded grants were located at only state-level precision. "High" indicates grants where both county boundaries and specific landmarks were used; "Medium" indicates county-centroid placement; "Low" indicates state-level precision only.

Table 15: Professional GIS benchmark results by accuracy category.

Accuracy Category	N	Share (%)	Mean Error (km)	Median Error (km)
Overall	43	100.0	71.40	60.20
High (County + Landmarks)	19	44.2	68.88	48.09
Medium (County centroid)	6	14.0	87.95	82.11
Low (State-level)	18	41.9	68.55	63.26

Notably, even the "High" precision category (where both county boundaries and specific landmarks were identified) still resulted in a mean error of 68.88 km—substantially higher than all the automated methods except gpt-3.5-turbo (M-6). This underscores the inherent difficulty of the task and further highlights the significance of the accuracy improvements achieved by the LLM approaches.

14 Supplementary Figures

14.1 Error Distribution Plots

Figure 9 shows the distribution of geolocation error for each method as a boxplot, complementing the violin plot and CDF already presented in the main text. The boxplot highlights the median error (central line), interquartile range (box), and outliers (points) for each method, providing a clear view of error distribution and central tendency.

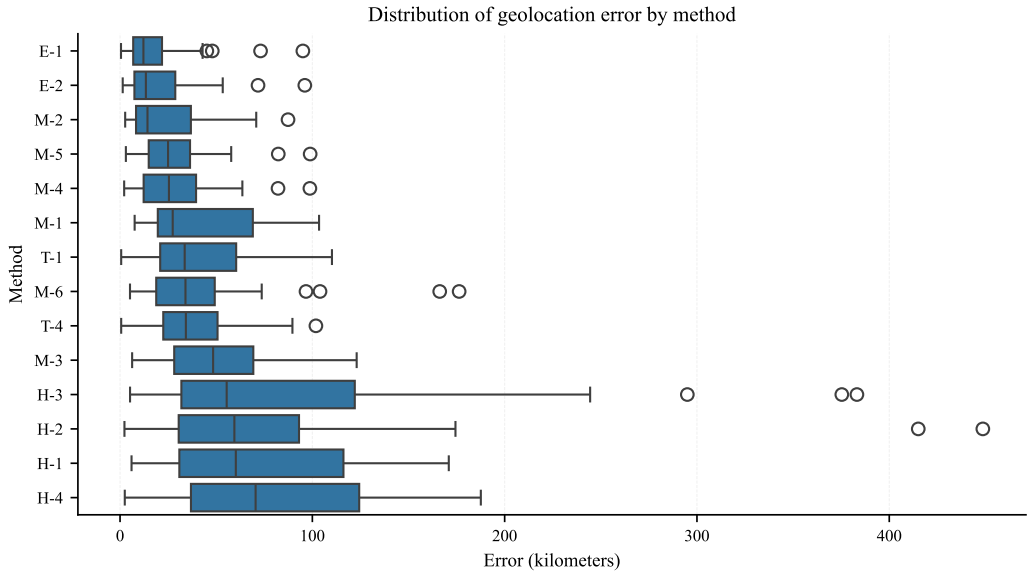


Figure 9: Error Distribution Boxplot by Method

14.2 Error Maps

Figure 10 plots all six methods for every locatable grant against ground truth coordinates (black stars). Error distances are shown as dashed lines connecting predictions to ground truth. For cartographic clarity the H-2, H-3, and H-4 baselines are omitted; their substantially larger positional errors would require a map extent so broad that the fine-scale patterns of interest would be lost.

14.3 Marginal Cost of High-Precision Accuracy

Table 16 reports how many U.S. dollars each method requires to raise the 10 km hit-rate by one percentage point (column "Cost per +1 % 10 km hit"). This marginal-cost view complements the mean-error vs. cost Pareto plot in the main text: it quantifies the price of *high-confidence* geocoding rather than average error alone.

Table 16: Marginal cost to improve the 10 km hit-rate by one percentage point.

Model	10 km hit-rate	Cost per 1 k located (USD)	Cost per +1 % 10 km hit (USD)
o3-2025-04-16 (Ensemble)	39.5 %	$195.65	$4.95
o3-2025-04-16	30.2 %	$127.46	$4.22
gpt-4o-2024-08-06	16.3 %	$1.05	$0.06
gpt-4.1-2025-04-14	18.6 %	$3.49	$0.19
o4-mini-2025-04-16	10.5 %	$10.69	$1.02

Model	10 km hit-rate	Cost per 1 k located (USD)	Cost per +1 % 10 km hit (USD)
gpt-3.5-turbo	4.7 %	$0.10	$0.02
o3-mini-2025-01-31	4.7 %	$14.15	$3.04
human-gis	4.7 %	$3,255.81	$700.00

The numbers reveal why **gpt-4o-2024-08-06** is so attractive in budget-constrained settings: each percentage-point gain in "high-precision" accuracy costs only six cents—roughly two orders of magnitude cheaper than even the *o3* ensemble, and over 10,000 × cheaper than a professional GIS analyst.

14.4 Latency-Accuracy Tradeoff

Processing time presents another critical dimension for evaluation. The figure below shows how each method balances computational latency against geolocation accuracy. LLM methods cluster in the bottom-left quadrant, delivering results in seconds rather than minutes, while maintaining lower error rates than the professional GIS approach.

Figure 11: Latency-Accuracy Tradeoff. This figure plots mean error (km) against processing time per grant (seconds) for each evaluated method. All automatic methods produce coordinates in 0.2–13 s of computation time, compared to the GIS analyst's labor time of 502 s per grant. Note the logarithmic scale on the x-axis.

15 Tool Augmentation Analysis

Table 17 isolates the impact of providing tool access to identical models, revealing that tool augmentation does not consistently improve accuracy. For gpt-4.1-2025-04-14, enabling tool access increases mean error by 30.6%, while for the o4-mini model, it decreases error by 9.6%.

15.1 Direct Tool vs. Non-Tool Comparison

Table 17 provides a head-to-head comparison of identical models with and without tool access. This controls for model architecture effects and isolates the impact of tool access alone.

Table 17: Side-by-side comparison of identical models with and without tool access.

Model	Category	mean	median	sd	min	max	1 km	5 km	10 km	25 km	50 km
gpt-4.1-2025-04-14	one shot	28.51	25.42	20.77	2.14	98.72	0.0%	4.7%	20.9%	48.8%	86.0%
gpt-4.1-2025-04-14	tool chain	37.23	34.22	23.94	0.59	101.85	2.3%	14.0%	16.3%	32.6%	74.4%

Model	Category	mean	median	sd	min	max	1 km	5 km	10 km	25 km	50 km
o4-mini-2025-04-16	one shot	41.65	27.39	27.32	7.59	103.49	0.0%	0.0%	7.0%	37.2%	62.8%
o4-mini-2025-04-16	tool chain	37.65	33.61	24.54	0.59	110.19	4.7%	11.6%	14.0%	32.6%	69.8%

15.2 Quantified Tool Effect

Table 18 quantifies the precise impact of tool access, showing percentage changes in mean error and percentage point (pp) changes in accuracy bands. " Mean %" shows percent change in mean error; "pp" indicates percentage point differences in accuracy bands. Negative percentages for mean change indicate worse performance with tools.

Table 18: Quantified effect of tool augmentation.

Model	Mean M	Mean T	Mean %	1 km pp	5 km pp	10 km pp	25 km pp	50 km pp
gpt-4.1-2025-04-14	28.51	37.23	-30.6%	+2.3 pp	+9.3 pp	-4.7 pp	-16.3 pp	-11.6 pp
o4-mini-2025-04-16	41.65	37.65	9.6%	+4.7 pp	+11.6 pp	+7.0 pp	-4.7 pp	+7.0 pp

While the o4-mini model showed a modest improvement with tools, gpt-4.1-2025-04-14 performed substantially worse when given tool access.

15.3 Top-performing methods per tool-use category

Table 19 shows a direct head-to-head comparison of the best-performing tool-use method vs the best non-tool method. M-2 (o3-2025-04-16, one-shot prompt) substantially outperforms the best tool-augmented method (T-4), achieving a 37% lower mean error and nearly double the proportion of predictions within 10 km.

Table 19: Best-performing methods per category.

Method	mean	median	sd	min	Q1	Q3	max	10 km	25 km	50 km
M (M-2)	23.39	14.27	19.86	2.67	8.17	36.85	87.35	30.2%	60.5%	93.0%
T (T-4)	37.23	34.22	23.94	0.59	21.78	53.35	101.85	16.3%	32.6%	74.4%

Figure 10: Grid view of all 43 mapped grants showing ground truth and predictions.

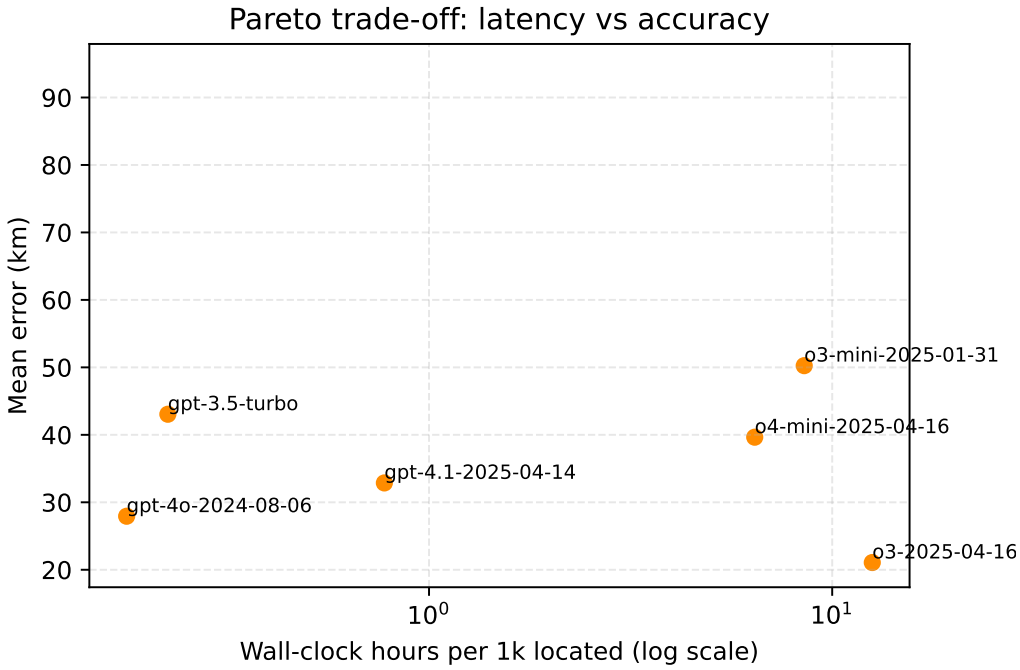


Figure 11: Latency-Accuracy Tradeoff

At the category level, the best non-tool method (M-2) significantly outperformed the best tool-augmented method (T-4) across all error metrics.

15.4 Tool Call Distribution

Table 20 expands on the tool usage patterns discussed in Section 6.6, providing detailed statistics on how each model interacted with the available geocoding and centroid-computation tools.

Table 20: Distribution of tool calls by method and tool type.

Method	Tool Type	Mean	SD	Median	Min	Max
T-1 (o4-mini)	geocode_place	3.79	2.41	3	1	10
T-1 (o4-mini)	compute_centroid	0.16	0.37	0	0	1
T-4 (gpt-4.1)	geocode_place	2.05	1.78	1	1	7
T-4 (gpt-4.1)	compute_centroid	0.25	0.43	0	0	1

15.5 ToolSearch Efficiency

"Selected call index" indicates which API call in the sequence produced the coordinates used in the final answer. Lower values indicate more efficient search strategies.

Table 21: Tool search efficiency metrics.

Method	Mean selected call index	Median	First-call success rate
T-1 (o4-mini)	2.29	1	69.0%
T-4 (gpt-4.1)	1.95	1	72.7%

The more economical approach of gpt-4.1-2025-04-14 is evident in both the distribution of calls and search efficiency. While T-1 (o4-mini) made nearly twice as many geocoding calls on average (3.79 vs. 2.05), it achieved a slightly lower first-call success rate (69.0% vs. 72.7%). This pattern aligns with the overall finding that tool augmentation does not consistently improve accuracy; in fact, the additional API calls may introduce noise through spurious matches to modern place names that bear little relation to colonial-era settlements.

Overall, both models heavily favored direct geocoding over centroid computation, with geocode:centroid ratios of 23.29:1 for T-1 and 8.18:1 for T-4. This suggests that the models primarily relied on finding exact matches for place names mentioned in the abstracts rather than triangulating from multiple reference points—a strategy that may explain their susceptibility to modern naming coincidences.

References

[1] NUGENT, N. M. *Cavaliers and Pioneers: Abstracts of Virginia Land Patents and Grants, Volume Three: 1695-1732*. Virginia State Library, Richmond, Virginia, 1979. Transcribed from Patent Books 9-14, containing 5,471 grant abstracts.

[2] BASHORUN, B. Colonial virginia land grants geocoding dataset, April 2025. Expert GIS baseline created for comparative evaluation in geolocation research.

[3] KARIMZADEH, M., Pezanowski, S., MacEachren, A., and Wallgrün, J. O. Geotxt: A scalable geoparsing system for unstructured text geolocation. *Transactions in GIS*, 23: 118-136, 01 2019.

[4] HALTERMAN, A. Mordecai 3: A neural geoparser and event geocoder. *arXiv preprint arXiv:2303.13675*, 2023.

[5] LI, W., Janowicz, K., Mai, G., Goodchild, M. F., Gautier, J., Kuhn, W., and et al. Geoai for science and the science of geoai. *Journal of Spatial Information Science*, 29: 1–17, 2024.

[6] JULIAN, G. M. and Abbitt, R. J. F. Gis and genealogy: Teaching gis while learning about the past. *ArcUser*, Spring 2014. Project mapping Julian family land grants across McMinn, Hamilton, and Bradley Counties in Tennessee through multi-semester student GIS work.

[7] LIBRARY OF VIRGINIA. Virginia land office patents and grants online database. Web portal, 2025. Scanned images and searchable indices for 1623–2000 patents & grants; includes Northern Neck survey plats but no GIS polygons.

[8] LOUDOUN COUNTY OFFICE OF MAPPING. Original land grants of loudoun county, va [gis dataset]. Interactive ArcGIS WebMap, 2015. Polygon shapefiles digitised from historian Wynne Saffer's USGS quad maps; covers all Northern Neck grants inside modern Loudoun County (early 1700s–1800s).

[9] MITCHELL, B. *Beginning at a White Oak: Patents and Northern Neck Grants of Fairfax County, Virginia*. Office of Comprehensive Planning, Fairfax County, 1977. Includes companion historical GIS map of patent & grant polygons.

[10] FAUSZ, J. F. *Patterns of Settlement in the James River Basin, 1607-1642*. Master's thesis, College of William and Mary, 1971. Paper 1539624744.

[11] DOBBS, G. R. Backcountry settlement development and indian trails: A gis land-grant analysis. *Social Science Computer Review*, 27(3): 331–347, 2009.

[12] COUGHLAN, M. R. and Nelson, D. R. Influences of native american land use on the colonial euro-american settlement of the south carolina piedmont. *PLOS ONE*, 13(3): e0195036, 2018.

[13] DIRECT LINE SOFTWARE. Deedmapper 4.2. Commercial software for historical deed mapping, 2010. Windows application for plotting metes-and-bounds and public domain property descriptions.

[14] GRITTA, M., Pilehvar, M. T., and Collier, N. Which melbourne? augmenting geocoding with maps. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pp. 1285–1296, Melbourne, Australia, 2018. Association for Computational Linguistics. Introduces the CamCoder neural geocoder model.

[15] HU, X. and Kersten, J. Enhancing toponym resolution with fine-tuned LLMs (Llama2). In *Proceedings of the Second International Workshop on Geographic Information Extraction from Texts at ECIR 2024 (GeoExT@ECIR)*, CEUR-WS.org, CEUR-WS.org, 2024. Fine-tuned Llama2 7B outperforms previous SOTA on toponym resolution by 13% and reduces mean error by 83%.

[16] WU, Y., Zeng, Z., Liu, K., Xu, Z., Ye, Y., Zhou, S., Yao, H., and Li, S. Text geolocation prediction via self-supervised learning. *ISPRS International Journal of Geo-Information*, 14(4): 170, 2025.

[17] SAVARRO, D., Zago, D., and Zoia, S. Leveraging large language models to geolocate linguistic variations in social media posts. *arXiv preprint arXiv:2407.16047*, 2024. GeoLingIt shared-task winner: fine-tuned LLM predicts both Italian region and coordinates for tweets.

[18] O'SULLIVAN, K., Schneider, N. R., and Samet, H. Metric reasoning in large language models. In *Proceedings of the 32nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '24)*, 2024.

[19] HUANG, C., Chen, S., Li, Z., Qu, J., Xiao, Y., Liu, J., and Chen, Z. Geoagent: To empower llms using geospatial tools for address standardization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6048–6063, Bangkok, Thailand, 2024. Association for Computational Linguistics.

[20] YAO, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. React: Synergizing reasoning and acting in language models. 2023.

[21] SCHOTTLANDER, D. and Shekel, T. Geospatial reasoning: Unlocking insights with generative ai and multiple foundation models. Google Research Blog, April 2025. Accessed August 2025.

[22] ONE SHARED STORY. Early virginia land patent - estimated polygons [centralvapatents_ply]. ArcGIS Online Feature Service, May 2020. GIS polygon layer created using Deed Mapper software and historic maps; developed in collaboration with University of Virginia Institute for Public History. Available at oss.maps.arcgis.com.