UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

DÉPARTEMENT DES SCIENCES ADMINISTRATIVES

WORKING PAPER NO. 45

JoF-250: Learning Scientific Writing Conventions from Pre-LLM Financial Research

A measured-convention writing engine for Claude Code, built from 276 Journal of Finance articles, with a preregistered held-out evaluation

Simon-Pierre Boucher

Département des sciences administratives
Université du Québec en Outaouais

simon-pierre.boucher@uqo.ca

Gatineau – Pavillon Alexandre-Taché
283, boulevard Alexandre-Taché
Gatineau, Québec, Canada J9A 1L8

November 2026

Version 1.0

Abstract

Keywords: scientific writing, rhetorical moves, large language models, retrieval-augmented generation, research integrity, Journal of Finance, Claude Code

JEL Classification: A14, C88, G00, O33

1. Related literature

This paper sits at the intersection of four literatures: the linguistic study of scientific genres, the empirical study of writing conventions and readability in economics and finance, retrieval-augmented and agentic uses of large language models (LLMs) in scholarly writing and review, and the evaluation of LLM-based research tools under contamination and reproducibility constraints. We review each in turn and then state what JoF-250 adds.

1.1. Genre analysis and rhetorical moves in scientific writing

The notion that research articles follow a learnable rhetorical structure originates in genre analysis. Swales (1990) describes the research article as a genre whose introductions "create a research space" (CARS) through a small number of recurring moves: establishing a territory, establishing a niche, and occupying the niche; Swales (2004) later revisits the model and extends it to other research genres. The CARS framework has since been operationalised as *move analysis*, in which trained coders segment a text into functional units and record their frequency, order and linguistic realisation. Samraj (2002) shows that move structure varies even between neighbouring subfields and that the original model requires elaboration to capture, among other things, how centrality claims and niche establishment are intertwined. Kanoksilapatham (2005) extends the method beyond introductions by proposing a fifteen-move description of entire biochemistry articles, and Lin and Evans (2012), in a cross-disciplinary sample of empirical articles, document that the canonical introduction–methods–results–discussion template is only one of several macro-structures actually in use. Two lessons follow for our purposes. First, rhetorical conventions are discipline- and even subfield-specific, which argues against importing a generic template into finance. Second, conventions are measurable: once an ontology of moves is fixed, their distribution can be estimated on a corpus rather than asserted.

A parallel strand studies how authors modulate commitment and guide readers. Hyland (1996, 1998) analyses hedging in science research articles as a pragmatic resource for calibrating claims and anticipating objection, and Hyland and Tse (2004) reappraise metadiscourse—the text about the text through which writers organise arguments and engage readers. Hyland and Jiang (2017) trace changes in informality markers across four disciplines over fifty years and find that the trends differ by field. These studies motivate the hedging, transition and claim–evidence dimensions of our annotation schema. Computational linguistics has long tried to automate such analyses. Teufel and Moens (2002) introduced argumentative zoning, assigning a rhetorical status (background, own contribution, contrast, and so on) to sentences of scientific articles, and Cohan et al. (2019) show that pretrained language models classify the rhetorical role of consecutive sentences in abstracts with high accuracy. Large open corpora of structured full text (Lo et al., 2020) and dedicated extraction

tools such as GROBID (Lopez, 2009) made document-level analysis practical, although finance articles—heavy in tables, footnotes and equations—remain a demanding parsing target.

Within economics, writing guidance has mostly taken the form of essays by senior scholars rather than corpus studies. McCloskey (1985) offers rules of economical writing in *Economic Inquiry*, and Thomson (1999) gives a detailed guide to writing economic theory in the *Journal of Economic Literature*. Such guidance is valuable but prescriptive: it reflects experience and taste, not measured practice in a specific journal. Our profiles are intended to complement it with descriptive estimates, while remaining explicit that observed frequency is not evidence of editorial superiority.

1.2. Writing conventions and readability in economics and finance

Finance has an unusually developed empirical literature on *writing*, albeit mostly about corporate disclosures rather than research articles. Li (2008) measures the readability of annual reports with the Fog index and document length and relates it to current earnings and earnings persistence. Loughran and McDonald (2011) show that general-purpose sentiment dictionaries misclassify a large share of financial vocabulary and build finance-specific word lists; Loughran and McDonald (2014) argue that the Fog index is poorly specified for 10-Ks and propose file size as a simpler readability proxy. Loughran and McDonald (2016) survey textual analysis in accounting and finance, and Gentzkow et al. (2019) review text-as-data methods across economics. These papers are relevant to JoF-250 in two ways: they establish that domain-specific lexicons and measurement choices matter for financial text, and they furnish deterministic measures (lengths, lexical features, formula-based readability) that can be computed on research articles as well as on filings.

A smaller literature treats the economics research article itself as an object of measurement. Card and DellaVigna (2013) document, among nine trends in the top-five economics journals since 1970, that published papers have become roughly three times longer and have more coauthors, and Card and DellaVigna (2014) study how two journals' page limits changed submission length and composition. Hengel (2022) applies readability scores to abstracts in top economics journals and finds that female-authored papers are better written and that the gap widens during peer review, evidence that writing standards are enforced in review. Dowling et al. (2018) ask whether more readable finance articles attract more citations. Outside economics, Plavén-Sigray et al. (2017) find a steady decline in the readability of scientific abstracts across more than 700,000 abstracts since 1881, and Hartley (2014) summarises research on structured abstracts, which tend to be more informative than unstructured ones. Taken together, this work shows that section lengths, abstract organisation and readability are measurable properties of research articles that vary across journals and over time, but no study we are aware of estimates the distribution of *rhetorical* conventions—contribution placement, identification framing, robustness presentation—for a finance journal. More recently, Korinek (2023) catalogues use cases of generative AI across the research

workflow, including writing and editing, and anticipates that economists will increasingly rely on such tools; this motivates asking what conventions those tools should be calibrated to.

1.3. Retrieval-augmented generation and LLM-assisted writing and review

Retrieval-augmented generation (RAG) combines a parametric language model with a non-parametric memory accessed through a retriever. Lewis et al. (2020) introduce the canonical formulation for knowledge-intensive tasks, Guu et al. (2020) integrate retrieval into pre-training, and Karpukhin et al. (2020) show that dense passage retrieval learned from question–passage pairs outperforms sparse baselines in open-domain question answering. Gao et al. (2023b) survey the subsequent proliferation of "naive", "advanced" and "modular" RAG designs. Virtually all of this work retrieves *content*—passages whose facts are to be reproduced. The JoF-250 engine retrieves something different: structural annotations and summaries that describe *how* an argument is organised, with excerpts limited to what is necessary and every result linked to provenance. Condition C in our experiment is a conventional content-RAG baseline precisely so that this distinction can be tested rather than assumed.

LLMs are already used for scholarly writing and review at scale. Liang et al. (2024a) estimate, with a population-level statistical framework applied to almost a million papers, a sharp rise in LLM-modified text after late 2022, and Kobak et al. (2025) use excess-vocabulary analysis of PubMed abstracts to infer that a substantial fraction of 2024 biomedical abstracts were processed with LLMs. Liang et al. (2024c) apply the same framework to peer reviews at AI conferences, and Latona et al. (2024) report, in a quasi-experimental study of ICLR 2024, that AI-assisted reviews are associated with higher scores. Else (2023) and Gao et al. (2023a) show that blinded reviewers identify only part of the ChatGPT-generated abstracts shown to them, and Van Noorden and Perkel (2023) report mixed expectations among 1,600 surveyed researchers. On the usefulness side, Liang et al. (2024b) find in a large-scale analysis that GPT-4 feedback on manuscripts overlaps with human reviewer comments and is judged helpful by many authors, whereas Du et al. (2024) show that LLM-generated reviews of NLP papers are often deficient in ways experts readily spot. Thelwall (2024) finds that single ChatGPT research-quality scores correlate only weakly with an expert's own ratings, with averaging over repeated runs improving agreement, and Hosseini and Horbach (2023) discuss the integrity risks of LLM-assisted peer review. Si et al. (2024) run a controlled comparison of LLM and expert research ideas, a reminder that evaluating LLM research assistance requires blinding and controls. Shumailov et al. (2024) show that training recursively on model-generated text degrades models, one argument for anchoring writing tools in human-written, pre-LLM text.

Because our primary outcome is scored by an LLM judge, the literature on LLM-as-a-judge bears directly on our design. Zheng et al. (2023) document position, verbosity and self-enhancement biases and show that strong judges can nevertheless agree with human preferences at human-level

rates on their benchmarks. Wang et al. (2024) show that the ordering of candidates can flip pairwise judgments, and Panickssery et al. (2024) link self-preference to a model's ability to recognise its own output. We therefore randomise presentation, blind the judge to condition, and treat judge scores as a supplement to deterministic fidelity metrics and to human ratings where available.

1.4. Evaluation, contamination and reproducibility of LLM-based research tools

Any experiment in which a foundation model is evaluated on published articles must confront the possibility that those articles were in its training data. Magar and Schwartz (2022) distinguish memorisation from exploitation of contaminated data, Sainz et al. (2023) argue that contamination must be measured benchmark by benchmark, and Golchin and Surdeanu (2024) and Deng et al. (2024) propose completion-based probes for detecting it. Carlini et al. (2023) quantify how memorisation grows with model size, duplication and prompt length, and Lee et al. (2022) show that deduplicating training data reduces verbatim regurgitation. These findings motivate our contamination statement, the withholding of the published introduction from research briefs, and the overlap audit between generated text and the corpus. More broadly, Kapoor and Narayanan (2023) document leakage as a pervasive source of irreproducibility in machine-learning-based science and propose model information sheets; Bowman and Dahl (2021) discuss what credible benchmarks require; Dodge et al. (2019) call for reporting budgets and variance rather than best runs; and Pineau et al. (2021) report on the NeurIPS reproducibility programme and checklist. Nosek et al. (2018) make the general case for preregistration: specifying hypotheses, outcomes and analyses before data are collected. Our protocol follows these recommendations by freezing document-level splits, preregistering hypotheses and primary outcomes, logging tokens, cost, latency and model versions, and reporting failed runs.

1.5. Positioning of this paper

JoF-250 differs from the work above in four respects. First, its writing conventions are *corpus-derived and measured*: rather than prescriptive advice (McCloskey, 1985; Thomson, 1999), we estimate medians, interquartile ranges and sample sizes for rhetorical moves and structural features on 276 pre-ChatGPT *Journal of Finance* articles, using a move ontology in the tradition of Swales (1990) and Kanoksilapatham (2005). Second, profile construction is separated from evaluation by frozen, document-level held-out splits, and contamination is acknowledged and probed rather than ignored (Sainz et al., 2023; Kapoor and Narayanan, 2023). Third, the engine's retrieval targets rhetorical structure rather than content, and its outputs are subjected to deterministic fidelity audits—numerical preservation, citation verification, corpus overlap—so that improvements in judged quality can be checked against unchanged facts, addressing the known weaknesses of LLM

judges (Zheng et al., 2023; Wang et al., 2024). Fourth, the engine is implemented as a tested Claude Code extension with a preregistered benchmark, so that the paper studies a system that is independently usable for future manuscripts.

2. Corpus, inclusion rules, rights, and descriptive statistics

2.1. Provenance and scope

The reference corpus consists of the research content of *The Journal of Finance* (Wiley, on behalf of the American Finance Association), volumes 72 to 75, published between 2017 and 2020. The PDF files were obtained on 3 November 2026 through the library proxy of the Université du Québec en Outaouais, under the institution's subscription and for the research use of a subscribing faculty member. The download produced one file per table-of-contents entry together with a bibliographic index (`index.csv`) listing, for each file, the title, authors, page range, publication date, DOI, table-of-contents section, year, volume and issue as displayed by the publisher. The original files are treated as read-only: no PDF is modified, moved, or redistributed, and the derived full-text structures described in Section 3 are never committed to version control.

Two features of the scope deserve an explicit statement. First, the project is labelled "JoF-250" because the mandate anticipated roughly 250 articles from 2018–2020; the folder as delivered also contained volume 72 (2017), and we retained it because the sole substantive criterion of the design is that every document predates the public release of ChatGPT on 30 November 2022 (amendment A-002 in `docs/PROGRESS.md`). The label is therefore a name, not a count; the verified count is 276 included documents. Second, "pre-ChatGPT" does not mean "free of algorithmic writing assistance": spelling and grammar tools, and in some cases earlier language models, were available to authors throughout the period. The corpus is best described as a sample of peer-reviewed finance writing produced before large language models became a routine drafting tool, not as a sample of unassisted human prose.

2.2. Inventory and verification procedure

The inventory (`jof250 corpus audit`; module `inventory.py`) scans the corpus folder recursively, skipping the download tooling directory, and records for every file its relative path, size, and SHA-256 digest. Each PDF is then probed with PyMuPDF: a file is considered valid when it opens, has at least one page, and yields more than 50 characters of text over its first three pages. The three-page window is deliberate: presidential addresses open with a full-page portrait of the speaker, and a one-page test would have flagged them as broken. The probe also extracts the document's embedded metadata and searches the first three pages for a DOI of the form `10.1111/jofi.`*nnnnn*.

An early version of this search failed on a subset of files because the publisher's typesetting encodes the ligature "fi" as the single Unicode code point U+FB01, so that "jofi" did not match a plain ASCII pattern; all PDF text is now normalised with Unicode NFKC before any pattern matching, both in the inventory and in the parser.

The scanned records are joined to `index.csv` on the relative path, which attaches the publisher's bibliographic fields to every file. Every DOI is then verified against the Crossref public API (polite pool, cached locally so that the verification is reproducible offline). A record is marked `crossref_ok` when the request returns status 200, the container title contains "Journal of Finance", and the fuzzy similarity between the index title and the Crossref title is at least 85 on a 0–100 scale. Missing metadata are never inferred: a file without a bibliographic record, or whose in-PDF DOI disagrees with the index DOI, is routed to a `review` decision rather than silently included. The final manifest (`corpus_manifest.parquet`), the human-readable audit (`corpus_audit.md`), and a frozen, hashed list of included documents (`corpus_v1_frozen.json`) are written together with an append-only inventory ledger.

2.3. Inclusion and exclusion

The decision rule is applied in a fixed order. A file is excluded if it is not a PDF, is smaller than 20,000 bytes, fails the validity probe, duplicates the content of another file (identical SHA-256), falls outside the years in scope, or belongs to a table-of-contents section other than *Articles*, *Replications and Corrigenda*, *Corrigendum*, or the retraction notice section. A final title-based rule excludes items that the publisher's table of contents lists under *Articles* but that are not research articles. One such case occurred: the participant schedule of the 2021 AFA annual meeting (volume 75, issue 6), which was mislabelled as an article by the publisher and is excluded with the reason "not a research article". The included documents are typed as regular articles, presidential addresses (identified by their title prefix), replications or corrigenda (by section), and retracted articles (by section). One retracted article is present; it is retained and flagged rather than removed, since the object of study is the writing conventions of the published record, not the validity of its findings.

Table 1 reports the outcome. Of 278 files scanned (277 PDFs), 276 are included and 2 excluded (the index file and the AFA schedule); no exact duplicates and no duplicate DOIs were found, and no record required manual review. All 276 included records pass the Crossref check. The included corpus spans 11,735 pages (median 43 pages per document) and 231 MB.

2.4. Document-level split

Before any profile was computed or any prompt was tuned, the included documents were split at the document level into a training set and a held-out set (`data/splits/heldout_v1.json`; script

Table 1. Corpus inventory: included documents by publication year and by article type. Source: `data/manifests/corpus_audit.md`, generated by `jof250 corpus audit`.

Year	Included	Article type	Included
2017 (vol. 72)	63	Research article	263
2018 (vol. 73)	65	Presidential address	4
2019 (vol. 74)	74	Replication or corrigendum	8
2020 (vol. 75)	74	Retracted article (flagged)	1
Total	276	Total	276

Notes: files scanned 278 (277 PDFs); excluded 2 (`index.csv` and the mislabelled AFA 2021 participant schedule); exact duplicates 0; Crossref-verified included records 276; pages 11,735 (median 43); size 231 MB.

`03_make_splits.py`, seed 45). Within each (year, article type) stratum, 20% of the documents were drawn at random into the held-out set, giving 221 training and 55 held-out documents. Profiles (Section 4.2) are built from the training set only, and the retrieval index records the split of every unit so that the writing engine can be restricted to training material. Author overlap between the two sets is measured and reported rather than eliminated: 20 of the 55 held-out documents share at least one author surname with a training document. In a single-journal corpus with a dense co-authorship network, removing every such document would have left too few held-out articles to be useful; we therefore treat author overlap as a documented limitation of the split rather than a solved problem, and the preregistration (Section 5) states this explicitly.

2.5. Rights policy and data exposure

The policy is recorded in `config/rights.yaml` and `docs/RIGHTS_AND_ETHICS.md`. Three classes of material are distinguished. *Private* material comprises the PDFs and the structured full texts produced by the parser (`data/derived/parsed/`); it stays on the author's machine, is excluded from version control, and is never redistributed or used for fine-tuning. *Material sent to the API* for annotation is bounded: for one document and one call, the payload contains the abstract, the paragraphs of the introduction and of the conclusion, the heading and the first paragraph of every section, table titles with notes truncated to 350 characters, figure captions truncated to 300 characters, and section-level counts, with a hard cap of 60,000 characters per call enforced by trimming paragraphs from the middle of the introduction and conclusion. No full article is sent in a single call, no fine-tuning is performed, and every call is written to a ledger. *Committed* material comprises metadata, counts, structural annotations, paraphrased summaries, aggregate profiles, and the retrieval index, whose textual units are capped at 400 characters so that the index cannot reconstruct a protected text. The engine is also instructed never to reuse sentences from the corpus, and every output is screened for eight-word overlap with the corpus (Section 4.5). None of this amounts to an endorsement by the journal or its publisher; the corpus is used as evidence about

conventions, not as a style authority.

2.6. Descriptive structural statistics

Table 3 and Figure 1 summarise the structural statistics extracted by the parser of Section 3. Two sets of numbers are reported. The first column block gives medians over all 276 included documents, as recorded in the parse summary; the second gives medians with interquartile ranges over the 221 training documents, as recorded in the `corpus_all` profile, which is the quantity the writing engine actually consults. The two agree closely, as expected from a stratified random split.

A typical included document has a body of 11009 words (interquartile range 10035–12407), an introduction of 1804 words in 7 paragraphs, that is a 0.16 share of the body, 8 tables, 4 figures, 10 numbered equations, 52 reference entries, and 37 footnotes; the abstract has 101 words, which reflects the journal's abstract length limit, and the main text is divided into 5 roman-numbered sections. The conclusion is short, a 0.03 share of the body. Minimum values of the distributions are driven by the corrigenda and the one-page replication notes, which the profiles retain for the `corpus_all` aggregate but which are excluded by genre from the rhetoric profiles. We stress, here and wherever a profile is quoted, that these distributions describe what published articles did in 2017–2020; they are not evidence that any of these choices is editorially superior or required.

Composition by genre and subfield. The annotation stage of Section 3.5 assigns each included document a genre and a subfield; Table 2 reports the resulting composition over all 276 validly annotated documents (training and held-out). These labels are produced by a language model under the controlled vocabularies of Table 4 and should be read with the caveats of Section 3.8; they are used for stratifying profiles and filtering retrieval, not as a bibliometric classification of the journal. Theory papers are a sizeable minority (40 purely theoretical and 75 mixed), which is why the ontology and the engine are explicitly designed not to impose an empirical-paper template.

3. Extraction pipeline and annotation ontology

3.1. Parser choice

The mandate asked that installed scientific PDF parsers be evaluated before one was chosen. On the host used for this study, GROBID (Lopez, 2009) was not installed and the Docker daemon required to run it was not available, so a server-based parser was ruled out; the core of the system must in any case run on a single machine without network services. PyMuPDF (version 1.27) and poppler's `pdftotext` were available. We built the structural parser on PyMuPDF, which exposes the font name, font size, bounding box, and writing direction of every text span, and retained `pdftotext`

Table 2. Composition of the annotated corpus by genre and by primary subfield (model annotation, ontology version 1.0, all 276 validly annotated included documents). Source: `data/annotations/annotations_table.parquet`.

Genre	*n*	Subfield	*n*
Empirical	142	Asset pricing	93
Mixed theory and empirical	75	Corporate finance	48
Theoretical	40	Financial intermediation and banking	34
Replication or correction	9	Household finance	27
Methodological or econometric	7	Market microstructure	24
Survey or address	3	Behavioural finance	15
		Macro-finance	12
		Entrepreneurial finance and private equity	11
		International finance	8
		Financial econometrics	4

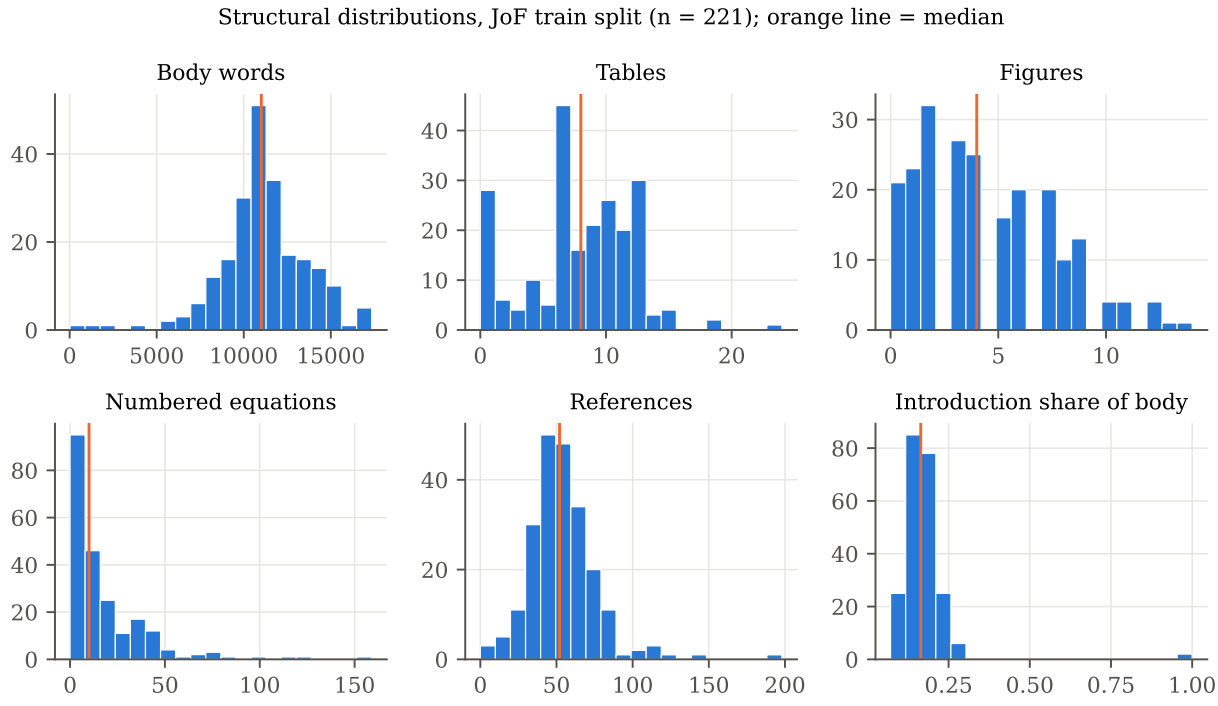


Figure 1. Structural distributions of the training split (221 documents): body words, tables, figures, numbered equations, reference entries, and introduction share of the body. The vertical line marks the median. Counts come from the heuristic PyMuPDF parser (Section 3); numbered-equation counts in particular are approximate. Held-out documents are not shown.

Table 3. Descriptive structural statistics of the corpus. Left block: medians over all 276 included documents (`parse_summary.parquet`). Right block: median and interquartile range over the 221 training documents (`profiles/corpus_all.yaml`, structure block). Bootstrap 95% intervals for the medians (500 resamples, seed 45) are stored in the profile file. Cells printed as TODO await a macro in `numbers.tex`.

	All included	Training split	
Statistic	Median	Median	IQR
Pages	43	43	38–47
Body words	10965	11009	10035–12407
Abstract words	101	101	96–104
Numbered sections	5	5	5–6
Introduction words	—	1804	1568–2132
Introduction paragraphs	—	7	6–9
Introduction share of body	—	0.16	0.14–0.19
Conclusion share of body	—	0.03	0.02–0.05
Tables	8	8	6–11
Figures	4	4	2–7
Numbered equations	10	10	4–23
Reference entries	52	52	41–64
Footnotes	37	37	29–48
Share with a robustness section	—	26	
Share with a separate literature section	—	17	
Share with an in-text appendix	—	50	
Share referring to an Internet Appendix	—	89	

as an independent second extraction for quality control (Section 3.4). The parser is tuned to the journal's LaTeX template; it is not a general-purpose scientific parser, and its heuristics would need to be revisited for another journal.

3.2. Layout heuristics

The parser (`parsing.py`, version `pymupdf-jof-heuristics-1.3`) proceeds in two passes over the text lines of a document. Every line is first normalised (Unicode NFKC, removal of non-printable characters, unification of the minus sign) and tagged with its dominant span's font, size, bold and italic attributes, and whether the font belongs to a mathematical family. Lines in the running head, the publisher's download watermark (a very small Times font in the margin), bare page numbers near the foot of the page, and narrow vertical fragments are discarded. The *body font size* is estimated as the modal size of lines set in the template's roman text font; all subsequent thresholds are expressed relative to it.

The front-matter pass identifies the title (bold lines at least 2.5 points larger than the body, on the first three pages), the author block (smaller non-bold lines between the title and the "ABSTRACT" marker), and the abstract (the small-type lines that follow the marker until body-size text resumes). The structural pass then walks the remaining lines and applies, in order, the following rules. A bold body-size line reading "REFERENCES" opens the reference list; a bold line beginning with "Supporting Information" records the page of the Internet Appendix pointer. Section headings are bold body-size lines that either begin with a roman numeral followed by a period ("`III. Data`") or are centred on the page; headings beginning with "Appendix" switch the parser to appendix mode so that appendix words are counted separately. Subsection headings are italic body-size lines matching the template's "`A. Title`" pattern. Table captions are bold small-type lines matching "`Table` *`roman`*", followed by a bold title and small-type notes; landscape tables, whose lines are rotated by ninety degrees, are collected in a separate pass over rotated lines so that their captions and notes are not lost, and the "Continued" marker suppresses duplicate captions. Figure captions are bold-initial small-type lines matching "`Figure` *`n`*."; small-type lines located above a figure caption on the same page are treated as text inside the figure and ignored. Numbered equations are detected from their tag, a parenthesised number set flush with the right margin on a line in a mathematical or italic font or containing an operator; the equation text is reconstructed from the lines in the same vertical band. Footnotes are small-type lines in the lower part of the page, grouped into one footnote per leading number and per page. Reference entries are split using the hanging indent of the template: a line starting at the page's left margin begins a new entry, and each entry is parsed into an author string, a year, the remainder, and a DOI when present. Finally, body lines are grouped into paragraphs using the first-line indent, hyphenation across lines is repaired, and in-text author–year citations are collected with a regular expression.

The output of the parser for one document is a JSON record holding the metadata, the abstract, the sections with their paragraphs and page numbers, subsections, equations, tables, figures, footnotes, references, in-text citations, page markers (references, appendix, supporting information), and summary counts; it is kept private because it contains the full text.

3.3. Confidence scores

Each parsed document carries a confidence score per component and an overall score. Title and authors receive 1 when found and 0 otherwise; the abstract receives 1 when its length lies between 400 and 2,500 characters, 0.5 when a shorter or longer abstract was captured, and 0 when none was; sections receive 1 when at least three roman-numbered headings were found in increasing order, 0.6 when two were found, and 0.2 otherwise; references receive 1 for at least ten entries and 0.5 for a non-empty list; equations and tables receive 0.9 when at least one was found and 0.7 otherwise. The overall score is the mean of the component scores. These scores are coarse by design: they are meant to route documents, not to measure accuracy. Over the included corpus the mean overall confidence is 0.951, and 3 documents fall below 0.8; these are the corrigenda and one-page notes, whose layout does not follow the article template and which are excluded by genre from the rhetoric profiles.

3.4. Parser quality assurance

The mandate called for a manually checked gold-standard sample. No human annotator other than the author was available in the time frame of this study, and we did not want to present the author's own inspection as an independent gold standard. We therefore report an *automated cross-parser agreement* check (`scripts/02_parser_qa.py`, results in `data/derived/parser_qa.md`, reproduced per document in Appendix C): the structural counts produced by the PyMuPDF parser are compared with counts obtained by regular expressions on the layout-preserving output of `pdftotext`, on a stratified sample of 26 documents (five randomly chosen articles per year, seed 45, plus every document of a non-article type). This is a comparison between two automated extractions, each with its own failure modes; agreement between them is evidence that the counts are not artefacts of one tool, but it is not an accuracy figure against human judgement, and we do not describe it as such.

On this sample, the number of roman-numbered sections agrees exactly for 85% of documents, the number of tables agrees exactly for 77% and within one for 96%, the number of figures agrees exactly for 92%, and the number of reference entries agrees within 10% for 92%. The abstract is found by both tools in 100% of documents. Numbered-equation counts are the weak point: exact agreement is reached for only 8% of documents and agreement within 10% for 35%, because the

two tools disagree on what constitutes an equation tag (the regular expression on the layout text also matches enumerated items and footnote markers in parentheses at the end of a line, while the PyMuPDF parser requires a right-margin position and a mathematical font). Equation counts should therefore be read as approximate throughout the paper. As a corpus-wide check of the front matter, the title extracted by the parser matches the Crossref title with a fuzzy similarity of at least 90 for 99% of the 276 included documents. The per-document table in Appendix C shows that the worst disagreements are concentrated in corrigenda, whose reference lists and headings do not follow the article template.

3.5. Annotation ontology

The mandate specified twelve families of properties to annotate, from the structure of the abstract to the boundedness of the conclusion, plus rhetorical moves, claim–evidence relations, hedging, and contribution types. We implemented them as a single document-level JSON schema (ontology version 1.0, `rhetoric.py`; full listing in Appendix A) whose enumerated fields use the controlled vocabularies of Table 4. Table 5 maps each family of the mandate to the fields of the schema. Two families are deliberately annotated at a coarser level than the mandate's wording suggests. The *literature* family is captured by the location of the literature positioning (inside the introduction, in a separate section, both, or none) and by the `related_literature_positioning` move, rather than by a graph of contrasts and lineage, because the bounded payload (Section 2.5) does not contain the full literature discussion. The *data* and *empirical design* families are captured through section roles, the identification block (estimand, strategy, key assumptions, threats addressed, a 1–5 causal-language score, and a flag for explicit causal claims), and the table conventions block (units stated in notes, description of standard errors), rather than through variable lists. Theory papers are explicitly allowed not to follow the empirical template: the annotator is instructed to assign section roles honestly and to use the "none", "unclear", or "not applicable" values for empirical-only fields.

The rhetorical-move inventory for introductions (motivation, tension or puzzle, literature gap, preview of approach, preview of results, contribution statement, mechanism or theory sketch, identification defence, robustness mention, roadmap, related-literature positioning, policy or broader implication, other) is an adaptation of move analysis in the tradition of genre studies (Swales, 1990, 2004) to the conventions of empirical finance, where the preview of results with magnitudes and the defence of identification are moves in their own right. Hedging is recorded as a three-level intensity with free-text devices, following the applied-linguistics literature on hedging in academic prose (Hyland, 1996, 1998), rather than as a word count, since the payload is a sample of the text.

Table 4. Controlled vocabularies of the annotation ontology (version 1.0), as defined in `src/jof250/rhetoric.py`. Free-text and numeric fields are listed in Appendix A.

Field	Allowed values
`genre`	`empirical`, `theoretical`, `mixed_theory_empirical`, `methodological_econometric`, `survey_or_address`, `replication_or_correction`
`subfield`	`asset_pricing`, `corporate_finance`, `financial_intermediation_banking`, `household_finance`, `market_microstructure`, `international_finance`, `behavioral_finance`, `macro_finance`, `financial_econometrics`, `entrepreneurial_finance_private_equity`, `other`
`primary_method`	`reduced_form_regression`, `difference_in_differences`, `instrumental_variables`, `regression_discontinuity`, `natural_experiment_event_study`, `structural_estimation`, `calibration_simulation`, `theoretical_model_analytical`, `machine_learning_textual`, `survey_or_experiment`, `other`, `none`
`introduction.moves[].move`	`motivation`, `tension_or_puzzle`, `literature_gap`, `preview_of_approach`, `preview_of_results`, `contribution_statement`, `mechanism_or_theory_sketch`, `identification_defense`, `robustness_mention`, `roadmap`, `related_literature_positioning`, `policy_or_broader_implication`, `other`
`introduction.contribution_types[]`	`new_evidence`, `new_data_or_measure`, `new_identification`, `new_theory_or_model`, `new_method`, `resolution_of_puzzle`, `replication_or_correction`, `synthesis`
`section_roles[].role`	`introduction`, `literature`, `theory`, `data`, `empirical_design`, `results`, `robustness`, `conclusion`, `appendix`, `other`
`robustness.types[]`	`alternative_specification`, `placebo`, `subsample`, `alternative_measure`, `alternative_sample_period`, `falsification`, `sensitivity_to_parameters`, `other`
`introduction.literature_positioning_location`	`inside_introduction`, `separate_section`, `both`, `none`
`results_presentation.statistical_significance_language`	`stars_and_tstats`, `confidence_intervals`, `p_values`, `mixed`, `unclear`
`robustness.location`	`main_text_section`, `within_results`, `appendix`, `internet_appendix`, `mixed`, `none_seen`
`tables.standard_errors_description`	`clustered`, `robust`, `bootstrap`, `newey_west`, `mixed`, `unclear`, `none`
`figures.uncertainty_shown`	`yes`, `no`, `mixed`, `unclear`, `no_figures`
`hedging.level`	`low`, `moderate`, `high`
`claim_evidence[].evidence_type`	`table`, `figure`, `equation_or_proof`, `prior_literature`, `descriptive_statistic`, `none_stated`

Table 5. Mapping of the twelve annotation families of the mandate to the fields of the ontology (version 1.0).

Mandate family	Schema fields
1. Abstract	`abstract_moves.{question, gap, method, results, magnitude, contribution}`, each with `present` and `sentence_index`
2. Introduction	`introduction.moves[]` (paragraph index, move), `contribution_paragraph_index`, `contribution_types[]`, `roadmap_present`
3. Literature	`introduction.literature_positioning_location`; move `related_literature_positioning`
4. Theory	`genre`; section role `theory`; move `mechanism_or_theory_sketch`; evidence type `equation_or_proof`
5. Data	section role `data`; `primary_method`; `subfield`
6. Identification	`identification.{estimand, strategy, key_assumptions[], threats_addressed[], causal_language_strength, uses_causal_claim}`
7. Empirical design	section role `empirical_design`; `primary_method`; `tables.standard_errors_description`
8. Results	`results_presentation.{economic_significance_discussed, statistical_significance_language, magnitude_statements_in_intro}`
9. Robustness	`robustness.{location, types[]}`; move `robustness_mention`
10. Tables	`tables.{convention_notes, units_stated_in_notes, standard_errors_description}`
11. Figures	`figures.uncertainty_shown`
12. Conclusion	`conclusion.{bounded_claims, policy_implications, open_questions, restates_contribution}`
Moves, claims, hedging	`introduction.moves[]`, `claim_evidence[]` (claim, evidence type, evidence reference), `hedging.{level, typical_devices[]}`, `structural_summary`, `writing_profile_keywords[]`, `confidence`

3.6. Annotation procedure

Each document is annotated by one call to the Claude API (`claude-opus-5-5`, the annotation model recorded in `config/models.yaml`, with reasoning effort `medium` and a 4,000-token output limit) using prompt `document_annotation` version 1.0 (Appendix B). The system prompt consists of the prompt body, the list of allowed values for every enumerated field, and the JSON schema itself; it is marked for prompt caching so that the per-document cost is dominated by the payload. The user message is the bounded payload of Section 2.5, serialised as JSON. The API's constrained-decoding option for structured outputs was tried first and rejected the full schema as too large to compile into a grammar; the model is therefore asked in the prompt to return a single JSON object, and the response is parsed and validated locally against the strict schema (JSON Schema draft 2020-12, enumerations enforced, no additional properties, every listed property required). A response that fails validation is stored with its validation errors and marked invalid; invalid documents are re-run with a fresh call rather than repaired by hand, and documents that remain invalid are excluded from the rhetoric profiles and from the annotation units of the index. The record written for each document holds the annotation, the prompt and ontology versions, the model identifier returned by the API, the effort, the timestamp, the source document hash, the parser version, the payload size, token usage, cost, the API request identifier, and the code version. In the full run, all 276 included documents were annotated and 276 validate against the strict schema after re-runs; roughly one first-pass response in ten failed strict validation (typically an enumeration value outside the vocabulary or a missing property) and was re-run, at a separately logged retry cost of 0.96 USD. The measured cost is 0.075 USD per valid document (20.62 USD in total). Every count quoted from the annotations in this paper is printed from macros regenerated from the recorded artefacts.

3.7. Deterministic checks and model–rule agreement

Independently of the model, a set of rule-based signals is computed from the same payload for every document (`deterministic_checks` in `rhetoric.py`): the presence of a roadmap paragraph (regular expression on "the remainder of the paper", "Section *n* presents", and similar phrasings), of contribution language in the introduction ("contribute", "this paper is the first", "this paper adds"), and of a magnitude in the abstract (a number followed by a percentage, basis-point, or standard-deviation unit); the number of sentences in the abstract and of paragraphs and words in the introduction; counts of causal and hedging words in the abstract; whether table notes mention significance stars or clustered standard errors; whether the introduction or the headings mention an Internet Appendix; and a rule-based role for every section derived from its heading (`segmentation.py`). These signals serve two purposes. They enter the profiles alongside the model's judgements, so that a reader can compare, for instance, the share of introductions with a roadmap according to

the model and according to the regular expression. And they provide a check on the model: for each annotated document we compute the agreement between the model's section roles and the rule-based roles over sections where the rule assigns a role, the agreement between the model's roadmap flag and the regular expression, and the agreement between the model's identification of a contribution paragraph and the contribution regular expression.

Over the 276 valid annotations, the median per-document agreement on section roles is 1.00, the roadmap flags agree for 97% of documents, and the contribution flags agree for 62% of documents. The lower contribution agreement is expected: the regular expression requires explicit contribution vocabulary, whereas the model also recognises contribution paragraphs phrased as "we show" or "our results", and the disagreements are informative about the limits of lexical rules rather than necessarily about the model.

3.8. What the agreement figures do and do not mean

We state the limits of this evidence plainly. The annotations are produced by one language model under one prompt; the agreement figures above compare that model with lexical rules, and the self-consistency of the profiles compares the model with itself. Neither is inter-rater reliability in the sense used in content analysis, which requires at least two independent human coders and a chance-corrected statistic such as Krippendorff's α (Krippendorff, 2019). A human audit protocol (`experiments/protocols/`) has been prepared for the day qualified coders are available, and the rhetoric profiles should be read, until then, as a model's structured reading of the corpus whose aggregate patterns are checked against simple rules, not as a validated coding of it. The same caution applies to the downstream uses of the annotations: structured retrieval units and profile summaries inherit whatever systematic bias the annotator has, and the experiment of Section 5 evaluates the usefulness of the resulting engine, not the correctness of the annotations.

4. JoF-250 architecture and Claude Code implementation

4.1. Overview

JoF-250 is a local-first pipeline implemented as a Python package (`jof250`, Python 3.12) with a command-line interface (Typer) and a thin Claude Code layer on top. Table 6 lists its stages, their inputs and outputs, and the artefacts each stage leaves behind; every artefact carries a timestamp, the git code version (with a `-dirty` suffix when the working tree had uncommitted changes), and the host description. The design choices follow from the constraints of the mandate: the corpus is copyrighted and must stay on one machine; the system must be usable on a single Mac without network services; every number in this paper must be traceable to a recorded artefact; and the engine

must be useful for manuscripts other than this one. Only two components call an external service: the annotation and generation stages call the Claude API, and the citation audit calls the Crossref API. Everything else, including embeddings, runs locally.

Table 6. Pipeline stages of JoF-250. Modules are in `src/jof250/`; artefacts marked *private* are excluded from version control.

Stage (module)	Input	Output / artefacts
Inventory (`inventory`)	PDF folder, `index.csv`, Crossref	`corpus_manifest.parquet`, `corpus_audit.md`, frozen manifest, inventory ledger
Extraction (`parsing`)	included PDFs	one structured JSON per document (*private*), `parse_summary.parquet`, parse ledger
Segmentation (`segmentation`)	parsed JSON	rule-based section roles; bounded annotation payload (rights cap)
Annotation (`rhetoric`, `llm`)	payload, prompt v1.0, Claude API	schema-validated annotation per document, `annotations_table.parquet`, API ledger, response cache
Splits (`scripts/03`)	manifest	`heldout_v1.json` (train / held-out document identifiers)
Profiles (`features`)	parsed JSON, annotations, train split	`profiles/*.yaml`, `features.parquet`, figures
Index (`embeddings`)	parsed JSON, annotations	`units.parquet`, `embeddings.npy`, `index_meta.json`, 8-gram shingles
Retrieval (`retrieval`)	query, filters	ranked units with provenance
Engine (`generation`)	manuscript, profile, retrieved units, Claude API	proposal file, report (change log, flags, audit, diff), engine-run ledger
Audits (`auditing`)	manuscript, bibliography, Crossref	deterministic audit report; evidence lookup
Evaluation (`evaluation`)	fixtures, protocol, Claude API	`runs.jsonl`, `judgments.jsonl`, `summary.json`
Integration (`integrations.claude_code`)	command, agent and skill definitions	`.claude/{commands,agents,skills}` (project scope) or `~/.claude/` (user scope, on request)

4.2. Measured writing profiles

A profile is a YAML file that records, for a named subset of the training split, the empirical distribution of structural and rhetorical features. The structural features are computed from the parsed documents (`features.py`): page count; words in the body and in the appendix; abstract words; numbers of sections, subsections, paragraphs, equations, tables, figures, footnotes, reference entries, and in-text citations; words and paragraphs of the introduction and its share of the body; words and share of the conclusion; the body shares of sections whose rule-based role is theory, data,

results, or robustness; words per table; references and equations per thousand words; and indicator variables for the presence of a robustness section, a separate literature section, an in-text appendix, and an Internet Appendix. For each numeric feature the profile stores the sample size, median, quartiles, mean, minimum, maximum, and a 95% bootstrap interval for the median (500 resamples, seed 45). When the subset contains annotated documents, a `rhetoric` block adds the distribution of genres, subfields, and methods; the share of abstracts in which each of the six abstract moves is present, and the median sentence position of each; the number of abstract sentences; the distribution of the first introduction move and of the most common move sequences (consecutive duplicates collapsed); the position of the contribution paragraph, absolute and relative to the length of the introduction; the share of introductions with a roadmap according to the model and according to the regular expression; the location of literature positioning; contribution types; the causal-language score; results-presentation conventions; robustness location and types; standard-error and unit conventions in table notes; uncertainty in figures; the conclusion flags; hedging; the share of claims with a stated evidence type; and, as diagnostics, the model–rule role agreement and the model's own confidence.

Profiles are stratified by selection rules on genre and subfield (Appendix D): an all-document profile (`corpus_all`), an all-empirical profile, and subfield profiles for empirical asset pricing, corporate finance, financial intermediation and banking, household and behavioural finance, financial econometrics, and theoretical finance. Because the subfield of a document is an annotation output, the subfield profiles depend on the annotation run; with the full run, the training split yields 221 documents in `corpus_all`, 173 in `empirical_all`, 58 in `empirical_asset_pricing`, 28 in `corporate_finance`, 21 in `financial_intermediation_banking`, 32 in `household_behavioral_finance`, 32 in `theoretical_finance`, and 8 in `financial_econometrics` (Appendix D). The engine falls back from a subfield profile with fewer than eight documents to the all-empirical profile and then to the all-document profile. To give one illustration of what a profile summary conveys, in the `empirical_all` profile the annotator finds a results move in 100% of abstracts but an explicit magnitude in only 29%, the first move of the introduction is motivation in 90% of articles, the contribution paragraph sits at a median relative position of 0.62 (interquartile range 0.20–0.80) in the introduction, a roadmap paragraph is present in 65% of articles according to the model and 64% according to the regular expression, the median causal-language score is 3.0 on the 1–5 scale, and economic significance is discussed in 92% of articles. Figures 2 to 4 show the corresponding distributions for the training split. Every profile file carries the same caveat in its header, which the engine repeats to the model in every prompt that uses a profile: observed frequencies describe what published articles did in 2017–2020; they are not evidence that these choices are editorially superior or required. The point is not rhetorical modesty. A convention can be frequent because it is useful, because it is enforced, or because it is habitual, and a corpus of

accepted articles cannot distinguish these cases; the profiles are offered as a measured description that an author can consult and deviate from, and the engine's prompts ask the model to say when it deviates.

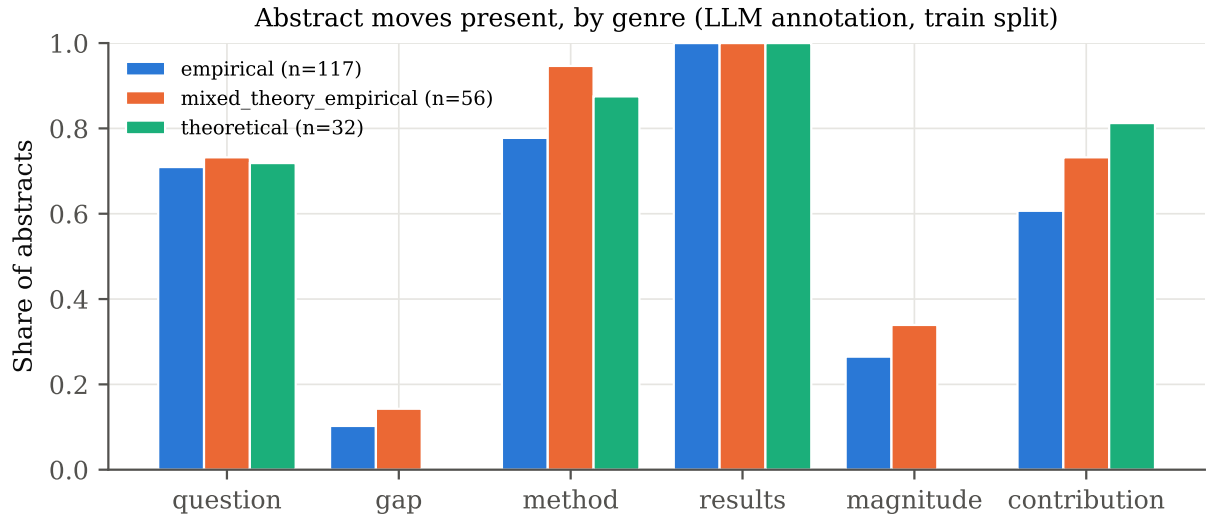


Figure 2. Share of abstracts in which each of the six abstract moves (question, gap, method, results, magnitude, contribution) is present, by annotated genre, training split only; genres with fewer than five annotated documents are omitted. Presence is judged by the model annotator (prompt `document_annotation` v1.0) from the abstract text and has not been validated by human coders; the deterministic magnitude check of Section 3.7 is reported in the profile files alongside the model's judgement.

4.3. Private hybrid retrieval

The retrieval index (`embeddings.py`, `retrieval.py`) is built from *units* rather than from passages of full text. Seven unit types are indexed. Four are derived from the annotations and contain no quoted text: a document summary (title, genre, subfield, method, the annotator's structural summary, and keywords), an identification summary (estimand, strategy, assumptions, threats, causal-language score), one claim–evidence unit per annotated claim, and one introduction-move unit per introduction paragraph with at least one annotated move. Three contain short excerpts capped at 400 characters, which the rights policy permits: the opening of each section with its role, length, and subsection headings; the title and notes of each table; and figure captions (at most eight per document). Each unit records the document identifier, the SHA-256 digest of the source PDF, DOI, year, title, article type, page, section, role, genre, subfield, method, the annotator's confidence, and the split of the document. The current index holds 9,642 units from 276 documents (276 with an annotation): 2,165 section openings, 2,105 table-note units, 1,150 captions, 1,908 introduction-move units, 1,762 claim–evidence units, 276 document summaries, and 276 identification summaries.

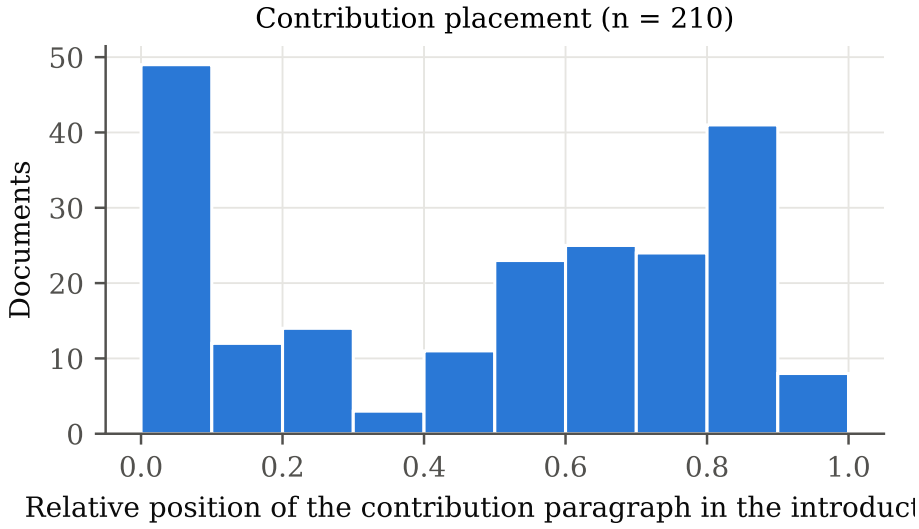


Figure 3. Relative position of the contribution paragraph within the introduction (paragraph index divided by the number of introduction paragraphs), training split, validly annotated documents in which the annotator identified a contribution paragraph. The paragraph index is the model annotator's; paragraph boundaries come from the heuristic parser.

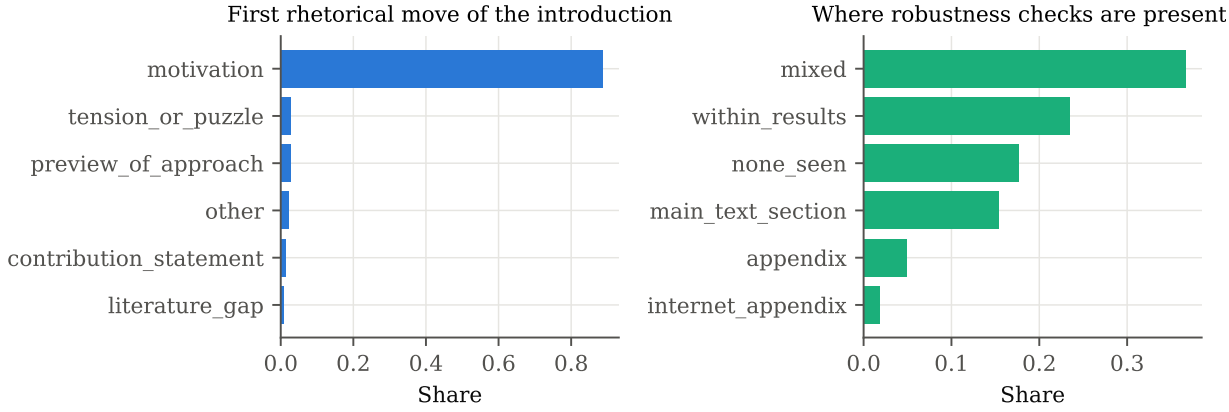


Figure 4. Left: the first rhetorical move of the introduction (six most frequent values). Right: where robustness checks are presented (six most frequent values). Training split, validly annotated documents; both classifications are the model annotator's under the controlled vocabularies of Table 4, and "mixed" denotes robustness material spread across several locations.

Queries are scored by a hybrid of BM25 (Robertson and Zaragoza, 2009) over a simple alphanumeric tokenisation and cosine similarity between normalised embeddings computed locally with a MiniLM sentence encoder (Reimers and Gurevych, 2019; Wang et al., 2020) of dimension 384; the two scores are min–max normalised within the filtered candidate set and combined with equal weight (the weight, and pure BM25 or pure dense modes, are parameters). Metadata filters restrict the candidates by split (training only by default), year, genre, subfield, section role or heading, and unit type; the `annotation` unit filter selects the four structural unit types and the `raw` filter the three excerpt types, which is how the conventional-RAG and structured-retrieval conditions of Section 5 are distinguished. To avoid a single article dominating the context, at most two units per document are returned. Every hit carries its provenance record, which the engine writes into its reports so that a user can trace any retrieved pattern to a document, page, and section.

4.4. The writing engine

The engine (`generation.py`) operates on a manuscript directory or file. The loader (`auditing.load_manuscript`) accepts LaTeX projects, locating `paper/main.tex` or `main.tex`, stripping comments, resolving `\input` and `\include` recursively, reading the UQO template's `\WP...` metadata macros, splitting the text into sections and subsections, and extracting tables, figures, equations, labels, cross-references, and citation commands; it also scans the project's BibTeX files for keys and DOIs. Markdown manuscripts are accepted with the same section logic. Each section receives a rule-based role (introduction, literature, theory, data, empirical design, results, robustness, conclusion, appendix) so that a request for "the results section" can be resolved against arbitrary headings. A cheap lexical heuristic then detects the manuscript's genre (theoretical versus empirical) and subfield from the title, abstract, and headings, without an API call, and the profile is selected by the rules of Section 4.2; the user may override the profile. For manuscripts outside the corpus's core genre (theory, methods, computer science, real estate), the engine injects a genre note instructing the model to apply the corpus conventions only where the argument structure is comparable and to say in the change log which conventions it did not apply.

The material handed to the model is bounded to 60,000 characters: title, abstract, metadata, an optional verified research brief, the list of sections with roles and lengths, the current text of the target section (up to 12,000 characters, or a note that it is absent), the other sections truncated proportionally, the captions and number counts of up to twenty tables, up to twenty equations, and the list of available BibTeX keys. The task is one of three modes: *outline* (a paragraph-by-paragraph plan with the rhetorical move and the evidence of each paragraph), *write*, or *upgrade* (rewrite an existing section preserving every number, citation, claim, and result). Conditions A to E of the experimental design are implemented as prompt families (Appendix B) that share one system prompt with hard constraints (never invent results, numbers, or citations; use only listed BibTeX

keys or a visible `\todo` placeholder; preserve verified facts; never copy sentences or imitate an author; return the section text followed by a change log and a list of flags). Condition A adds a one-line instruction; B a strong academic-writing instruction; C adds six raw excerpts retrieved by similarity from the training split, in the manner of conventional retrieval-augmented generation (Lewis et al., 2020); D adds the profile summary (sample size, caveat, length medians, abstract and introduction move patterns, contribution placement, identification, results, robustness, table and conclusion conventions) together with six structured units with provenance, and a four-step procedure (identify what the evidence supports, choose the sequence of moves and state it in the change log, write, list unsupported statements under flags). Condition E, the default engine, runs D and then two further stages: a deterministic audit of the draft (numeric preservation against the material, citation keys against the bibliography, unsupported-claim sentences, and eight-word overlap with the corpus), and a specialist-referee pass in which the model receives the material, the draft, and the audit and must return a revised draft that fixes every audit item while keeping every verified fact, strengthening the methodological framing, and replacing unsupported statements with placeholders. The audit is re-run on the revised text. Ablations of E switch off retrieval, the profile, the referee pass, or the fact-check individually.

The engine never edits the manuscript. Its output is a new file whose header records the mode, section, condition, model, timestamp, and manuscript hash and states that the file is a proposal, together with a sibling report containing the change log, the flags, the retrieval provenance, the deterministic audit, and a unified diff against the current section. Each invocation is appended to an engine-run ledger with its cost, latency, token usage, number of API calls, prompt versions, and detected genre.

4.5. Reviews, audits, and evidence lookup

Three further commands reuse the loader. `jof250 review` produces role-based reviews from five prompts (referee, writing editor, table-and-figure editor, citation and integrity auditor, replication auditor; Appendix B); each receives the manuscript material, the deterministic audit facts, and, by default, the measured conventions of the selected profile with the genre note, and is instructed to tie every comment to a section, table, or equation and to leave every number unchanged. `jof250 audit` makes no API call: it checks that tables have captions and notes and are referenced, that labelled equations and figures are referenced and that every cross-reference resolves, that every citation key exists in the bibliography, and that the DOIs in the bibliography resolve through Crossref (cached, at most sixty per run); it lists sentences with strong claim verbs that carry no citation, cross-reference, number, or hedge; and it reports the maximum share of the manuscript's eight-word shingles found in any single corpus document, computed from locally stored hashes. The limits of these checks are stated in `docs/LIMITATIONS.md`: numeric preservation compares

sets of numbers, not their meaning; unsupported-claim detection is a lexical heuristic; and citation verification establishes that a key and a DOI exist, not that the cited work supports the claim. `jof250 evidence` locates, for a claim given as text, the tables whose bodies contain the claim's numbers or keywords and the sentences that mention them, and returns a status of supported by a table, mentioned in text, or no evidence found.

4.6. Claude Code integration

The host ran Claude Code version 2.1.317, which supports project-scoped slash commands as Markdown files with YAML front matter under `.claude/commands/`, sub-agents under `.claude/agents/`, and skills under `.claude/skills/`. The integration module generates these files from a single source of truth (`integrations/claude_code.py`) so that the project copy and an optional user-scope copy cannot drift. Ten commands are installed (Table 7): status, corpus audit, profile explanation, outline, write, review, upgrade, audit, evidence, and benchmark. The installed version passes the raw text typed after a command to the command file as `$ARGUMENTS`; rather than relying on a parser that the version does not provide, each command file instructs the assistant how to parse its arguments (for example, `<paper-path> [-section <name>] [-condition X]`, with stated defaults), which CLI invocation to run, which output files to read, and what to show the user. The front matter restricts the tools each command may use to the project's CLI binary, a few read-only shell utilities, and file reading and searching, and the command bodies state that the manuscript is never modified and that proposals go to a `jof250_proposals/` or `jof250_reviews/` folder inside the user's paper directory. Seven specialist agents (Table 8) wrap the review and audit commands with role-specific instructions and declared models; a skill file describes the engine, its rules for the assistant (always pass an explicit output path, relay every flag and audit item, state which conventions do not apply to non-finance manuscripts, warn that API calls cost money), and how to repair a missing environment. The project-scope installation is the default; because the author asked that the commands be available in every session, the CLI also offers a user-scope installation that copies only the files prefixed `jof-` or named `jof250` into the user's Claude Code directory and touches nothing else there, with documented removal. The commands were exercised from a separate sample research repository, as the mandate requires; the results of those exercises are reported with the engineering tests, not as experimental findings.

4.7. Provenance, ledger, and budget

Every API call goes through one wrapper (`llm.py`) that appends a record to `api_ledger.jsonl` with the purpose, prompt identifier and version, model identifier as returned by the API, effort, input, output and cache token counts, cost computed from the list prices recorded in `config/models.`

Table 7. Slash commands installed under `.claude/commands/` and the CLI call each one wraps.

Command	Arguments	Purpose
`/jof-status`	`--`	Show JoF-250 pipeline status (corpus, parsing, annotations, profiles, index, spend)
`/jof-corpus-audit`	`--`	Re-run the JoF corpus inventory/verification and summarize the audit
`/jof-profile`	`[profile-name]`	Explain a measured JoF writing profile (empirical_asset_pricing, corporate_finance, financial_econometrics, theoretical_finance, ...)
`/jof-outline`	`<paper-path> -section <name>`	Propose a section outline for a manuscript using JoF-250 (profiles + structured retrieval)
`/jof-write`	`<paper-path> -section <name> [-condition A|B|C|D|E]`	Draft a manuscript section with the full JoF-250 engine (profiles, structured retrieval, verification, specialist referee)
`/jof-review`	`<paper-path> [-role referee|editor|table_editor|integrity|replication] [-full]`	Produce a referee-style review of a manuscript (roles: referee, editor, table_editor, integrity, replication)
`/jof-upgrade`	`<paper-path> -section <name> [-preserve-results]`	Rewrite/upgrade an existing section while preserving every result, number and citation
`/jof-audit`	`<paper-path> [-tables] [-equations]`	Deterministic manuscript audit: tables, equations, citations/DOIs, unsupported claims, corpus overlap
`/jof-evidence`	`<paper-path> -claim ''...''`	Locate the evidence for a claim inside a manuscript (tables, numbers, sentences)
`/jof-benchmark`	`[-protocol heldout_v1] [-dry-run]`	Run (or dry-run) the preregistered JoF-250 evaluation protocol

Table 8. Specialist agents installed under `.claude/agents/` (frontmatter `model` field as declared).

Agent	Model	Role
`jof-corpus-auditor`	haiku	verifies the JoF-250 corpus inventory, manifest integrity, duplicates and bibliographic records
`jof-rhetorical-analyst`	sonnet	explains the rhetorical structure of a section/manuscript against the measured JoF profiles
`jof-writing-editor`	sonnet	proposes rewrites that keep every fact, number and citation unchanged
`jof-econometric-referee`	opus	identification, inference, robustness critique
`jof-table-figure-editor`	sonnet	captions, notes, units, uncertainty
`jof-integrity-auditor`	sonnet	citations, DOIs, unsupported claims, numeric consistency, overlap
`jof-replication-auditor`	sonnet	what a replicator needs and what is missing

`yaml`, latency, stop reason, request identifier, cache key, and code version. Responses are cached by the hash of model, prompts, schema, effort, and output limit, so that re-running the annotation stage does not re-spend; generation and judging calls bypass the cache because nondeterminism is part of what the experiment measures. The wrapper refuses to place a call once cumulative spend reaches the budget cap fixed by amendment A-001 (`budget_usd_cap`), retries transient API errors with back-off, and records refusals. At the time of writing the ledger holds 423 calls for 38.12 USD, of which 20.58 USD for annotation, 0.96 USD for annotation retries, 11.52 USD for engine writing calls, 4.88 USD for referee passes, and 0.19 USD for review calls; benchmark spending is reported with the results. Engine invocations and benchmark runs are written to their own ledgers, and failed or refused runs are kept, not dropped. Together with the frozen manifest, the per-document hashes, the parser version stamped in every parsed file, and the seeds recorded in the split, profile, and fixture scripts, these ledgers allow every table and figure of this paper to be regenerated from recorded data by the commands listed in Appendix F.

5. Experimental design, hypotheses, and preregistration

This section transcribes the preregistered protocol `heldout_v1` (`experiments/preregistration.md`), which was frozen on 3 November 2026, before any benchmark run, at the code version recorded in the repository history, following the logic of preregistration as a commitment device against analytic flexibility (Nosek et al., 2018). Any departure from it is logged as an amendment in `docs/PROGRESS.md`. No result is reported here; Sections **??** to **??** report only what was actually run.

5.1. Research question and hypotheses

The question is whether abstract, domain-specific writing conventions extracted from a pre-ChatGPT finance corpus improve LLM-assisted scholarly writing and review beyond standard prompting and conventional retrieval, while maintaining originality, factual accuracy, and reproducibility. The five hypotheses are directional and are tested as paired contrasts within task × document (and repetition), with the conditions defined in Table 9.

H1 D (structured profiles and rhetorical retrieval) exceeds B (strong academic-writing prompt) on judge-rated *argument clarity* and *contribution positioning*. The mandate states H1 in terms of blind human ratings; because human raters were not available in this study (Section 5.8), H1 can only be tested here with the model judge and is reported as such.

H2 D exceeds C (conventional retrieval over raw excerpts) on the finance writing tasks, on the same two outcomes and on *usefulness*.

H3 E (D plus verification and a specialist referee pass) exceeds D on *methodological rigour* and *fidelity*, with no loss in numeric preservation, that is, the preserved rate of E is at least that of D.

H4 E has fewer *unverifiable citations* (author–year strings absent from the source reference list) and fewer *unsupported-claim sentences* than A, B, and C.

H5 Any advantage of D or E persists on held-out documents and on the out-of-domain manuscript, and is not explained by overlap: the maximum eight-word overlap share with any corpus document is below 0.02 for every output.

5.2. Conditions and ablations

The five conditions form a ladder from a minimal prompt to the full engine, so that each step isolates one ingredient (Table 9). Conditions B, C, and D share the same strong-prompt instruction; C and D differ only in what is retrieved and from which units (raw excerpts versus structured annotation units with a profile summary); E adds the two post-draft stages. Four ablations of E remove one ingredient at a time. All prompts are listed verbatim in Appendix B.

5.3. Tasks and materials

The experimental unit is a task × source document × condition × repetition. Source documents are drawn from the frozen held-out split (55 documents never used for profiles or prompt tuning): two empirical articles of different subfields, selected by the fixture script (seed 45) among held-out empirical articles with at least six tables, plus one out-of-domain manuscript, UQO Working Paper 29, a methods-oriented, computer-science-flavoured LaTeX paper by the author with verified results. The fixture manifest records, for each selected article, its identifier, DOI, source hash, and table and reference counts, together with the code version at which the fixtures were built. For each

Table 9. Experimental conditions and ablations (`config/evaluation.yaml`; prompts in Appendix B). Retrieval is restricted to the training split.

Condition	Definition
A `baseline`	Shared system prompt; one-line instruction ("Write the *section* of this paper").
B `strong_prompt`	A plus an experienced-economist writing instruction (question early, explicit contribution, closest prior work without a laundry list, preview of findings with magnitudes, proportionate claims, plain English, one idea per paragraph, manuscript's own notation and numbers).
C `conventional_rag`	B plus six raw excerpts (at most 400 characters each: section openings, table notes, captions) retrieved by hybrid similarity, presented as examples of tone and organisation only.
D `profiles_retrieval`	B plus the measured profile summary (sample size, caveat, length and move distributions, contribution placement, identification, results, robustness, table and conclusion conventions) and six structured rhetorical units with provenance; a four-step procedure (evidence, move sequence stated in the change log, draft, flags); genre note when applicable.
E `full_engine`	D plus a deterministic fact-check of the draft (numeric preservation, citation keys, unsupported claims, corpus overlap) and a specialist-referee revision pass that must fix every audit item while keeping every verified fact; the audit is re-run on the revised text.
E `no_retrieval`	E without the six structured units.
E `no_profile`	E without the profile summary.
E `no_referee`	E without the referee pass (draft plus audit only).
E `no_factcheck`	E without the deterministic audit (the referee pass receives no audit).

held-out article the fixture contains a Markdown manuscript built from the parsed structure in which the published introduction is withheld, a copy with the introduction for the judge's reference material where the task permits, the published introduction as a separate file, and metadata; no fabricated finance findings are used anywhere. The eight tasks follow the mandate (Table 10); each is implemented as a section role, an engine mode, and a research brief that states what material may be used and what must not be invented. The degraded literature passage used by the rewrite task is built deterministically from the paper's own literature paragraphs, so that the task has a known ground truth of admissible citations.

Table 10. Evaluation tasks (`evaluation.py`, `TASKS`). "Mode" is the engine mode; the brief is paraphrased. The published introduction is withheld from every finance fixture.

Task	Section role	Mode	Brief (summary)
`draft_introduction`	introduction	write	Draft the introduction from the verified material (abstract, section structure, tables, figures, references); no numbers beyond the material.
`rewrite_literature`	literature	upgrade	Rewrite a deterministically degraded "laundry-list" positioning passage into paragraphs that relate, differentiate, and build on the cited work, citing only works in the reference list.
`explain_specification`	empirical design	upgrade	Improve the explanation of the research design: estimand, identifying assumptions, fixed effects, inference, as stated; add no specification.
`interpret_results`	results	write	Interpret the main results from the tables' captions and notes and the stated facts; economic and statistical significance only as far as supported; never invent coefficients.
`improve_table`	results	upgrade	Propose an improved caption and notes for the first main regression table (sample, unit of observation, period, definitions, standard errors, significance conventions) without inventing values.
`flag_causal_claims`	results	write	Identify every causal claim; classify as supported by the design, over-stated, or associational; name the design element that would support it; propose a bounded rewording.
`referee_report`	results	write	Referee report: summary, contribution, major comments (identification, inference, measurement, robustness, presentation), minor comments, recommendation; every comment tied to a section or table.
`improve_abstract`	abstract	upgrade	Improve the abstract without altering any finding, number, or claim; 100–150 words.

5.4. Controls

All conditions use the same generator model (`claude-opus-5-5`), the same reasoning effort (`medium`) and output limit, and the same manuscript material and brief. Temperature is not a configurable parameter of this model, which uses adaptive reasoning; nondeterminism is therefore handled by repetition rather than by fixing a seed. Repetition 1 covers the full grid of conditions and ablations; repetition 2 covers the main conditions A to E on four tasks (introduction, abstract, referee report, interpretation of results). The out-of-domain manuscript is run on three tasks (introduction, abstract, referee report) under the five main conditions only. Cost, tokens, latency, retries, prompt versions, and model identifiers are logged per call. Token budgets are not equalised: condition E makes two generation calls and uses more tokens than A to D; cost and latency are therefore reported as outcomes rather than controlled, and this asymmetry is stated as a limitation.

5.5. Outcomes and judging

The six primary outcomes are judge ratings on a 1–7 scale for clarity, argument logic, contribution positioning, methodological rigour, fidelity, and usefulness, with calibrated anchors (Appendix B, judge rubric version 1.0). The judge is `claude-sonnet-5-5`, a different member of the generator's model family, chosen to limit self-preference while keeping the rubric interpretable to the same vendor's models; the judge is blind to condition labels, receives all condition outputs of one (task, document, repetition) set in a random order under random letters together with the task and the source material, is told not to reward length, and returns scores, a count of invented facts, and a one-line justification per output. Judge consistency is measured by re-judging a random 25% of the sets in a second random order and reporting the mean absolute score difference and the rank correlation between passes; judge biases that this design does not remove (verbosity, style preference, same-vendor affinity) are acknowledged (Zheng et al., 2023; Wang et al., 2024; Panickssery et al., 2024). Deterministic secondary outcomes are computed for every output without a model: the preserved rate and new rate of numbers relative to the source material, the number of unverifiable author–year citations (surname and year not co-occurring in the source reference list or material), the number of unknown BibTeX keys, the number of unsupported-claim sentences, the maximum eight-word overlap share with the corpus, the number of `\todo` placeholders, latency, cost, and tokens.

5.6. Analysis plan

For each hypothesis contrast and outcome, the primary statistic is the mean of the paired differences within (task, document, repetition), with a 95% cluster bootstrap interval (clusters are task $\times$ document; 2,000 resamples; seed 45) and a paired Wilcoxon signed-rank test. The Holm procedure

(Holm, 1979) is applied across the six primary outcomes within each hypothesis. Effect sizes are the mean paired difference on the 1–7 scale and Cliff's δ (Cliff, 1993). Secondary outcomes use the same paired bootstrap, with counts also reported as rates. Each ablation is compared with full E by the same procedure. The out-of-domain manuscript is reported separately and descriptively, since it is a single document. Significance is declared only when the Holm-adjusted p-value is below 0.05 *and* the bootstrap interval excludes zero. The sample, two finance documents and eight tasks, is small by construction: the study is a pilot of the protocol, powered to detect only large effects, and this is stated wherever results are reported. No result is written before the runs complete, and failed or refused runs are reported rather than silently dropped.

5.7. Contamination and leakage statement

The generator and the judge are foundation models whose pretraining data may include the held-out articles, which are published and widely cited; nothing in this design can exclude that possibility, and the literature on benchmark contamination and memorisation shows that such exposure can inflate apparent performance (Sainz et al., 2023; Golchin and Surdeanu, 2024; Carlini et al., 2023). Three mitigations reduce its consequences without removing it: the published introduction is withheld from the material of every finance fixture, so that the drafting task cannot be solved by recall of the visible text; every output is screened for eight-word overlap with every corpus document, and H5 sets a preregistered ceiling; and author–year citations in outputs are verified against the source reference list, so that recalled but unlisted references count as unverifiable. Leakage between splits is addressed at the document level (Section 2.4): profiles and the retrieval index used by the engine are restricted to training documents, and the author-overlap count between splits is reported. Results are to be interpreted as conditional on these caveats.

5.8. Human evaluation

Blind evaluation by qualified human reviewers was not available in this study. A rating protocol and rating sheets were prepared (`experiments/protocols/human_rating_sheet.md`): at least two PhD-level finance researchers, blind to condition, each rating all outputs of a (task, document) set in random order under random letters on the six primary dimensions plus a count of invented facts, with a joint calibration on two anchor sets, reliability measured by Krippendorff's α (Krippendorff, 2019), and disagreements above two points resolved by discussion while the original scores are kept for analysis. Until such ratings exist, every quality outcome of this paper is a model-judge rating and is labelled as such; the mandate's H1, which is phrased in terms of human ratings, remains untested in its original form.

References

Bowman, Samuel R. and George Dahl, "What Will It Take to Fix Benchmarking in Natural Language Understanding?," in "Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies" Association for Computational Linguistics 2021, pp. 4843–4855. verified Crossref 2026-11-03.

Card, David and Stefano DellaVigna, "Nine Facts about Top Journals in Economics," *Journal of Economic Literature*, 2013, *51* (1), 144–161. verified Crossref 2026-11-03.

_ and _ , "Page Limits on Economics Articles: Evidence from Two Journals," *Journal of Economic Perspectives*, 2014, *28* (3), 149–168. verified Crossref 2026-11-03.

Carlini, Nicholas, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang, "Quantifying Memorization Across Neural Language Models," arXiv preprint arXiv:2202.07646 2023. verified arXiv 2026-11-03.

Cliff, Norman, "Dominance Statistics: Ordinal Analyses to Answer Ordinal Questions," *Psychological Bulletin*, 1993, *114* (3), 494–509. verified Crossref 2026-11-03.

Cohan, Arman, Iz Beltagy, Daniel King, Bhavana Dalvi, and Dan Weld, "Pretrained Language Models for Sequential Sentence Classification," in "Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)" Association for Computational Linguistics 2019, pp. 3691–3697. verified Crossref 2026-11-03.

Deng, Chunyuan, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan, "Investigating Data Contamination in Modern Benchmarks for Large Language Models," in "Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)" Association for Computational Linguistics 2024, pp. 8706–8719. verified Crossref 2026-11-03.

Dodge, Jesse, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith, "Show Your Work: Improved Reporting of Experimental Results," in "Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)" Association for Computational Linguistics 2019, pp. 2185–2194. verified Crossref 2026-11-03.

Dowling, Michael, Helmi Hammami, and Ousayna Zreik, "Easy to Read, Easy to Cite?," *Economics Letters*, 2018, *173*, 100–103. verified Crossref 2026-11-03.

Du, Jiangshu, Yibo Wang, Wenting Zhao, Zhongfen Deng, Shuaiqi Liu, Renze Lou, Henry Peng Zou, Pranav Narayanan Venkit, Nan Zhang, Mukund Srinath, Haoran Ranran Zhang, Vipul Gupta, Yinghui Li, Tao Li, Fei Wang, Qin Liu, Tianlin Liu, Pengzhi Gao, Congying Xia, Chen Xing, Jiayang Cheng, Zhaowei Wang, Ying Su, Raj Sanjay Shah, Ruohao Guo, Jing Gu, Haoran Li, Kangda Wei, Zihao Wang, Lu Cheng, Surangika Ranathunga, Meng Fang, Jie Fu, Fei Liu, Ruihong Huang, Eduardo Blanco, Yixin Cao, Rui Zhang, Philip S. Yu, and Wenpeng Yin, "LLMs Assist NLP Researchers: Critique Paper (Meta-)Reviewing," arXiv preprint arXiv:2406.16253 2024. verified arXiv 2026-11-03.

Else, Holly, "Abstracts Written by ChatGPT Fool Scientists," *Nature*, 2023, *613* (7944), 423. verified Crossref 2026-11-03.

Gao, Catherine A., Frederick M. Howard, Nikolay S. Markov, Emma C. Dyer, Siddhi Ramesh, Yuan Luo, and Alexander T. Pearson, "Comparing Scientific Abstracts Generated by ChatGPT to Real Abstracts with Detectors and Blinded Human Reviewers," *npj Digital Medicine*, 2023, *6* (1). verified Crossref 2026-11-03.

Gao, Yunfan, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang, "Retrieval-Augmented Generation for Large Language Models: A Survey," arXiv preprint arXiv:2312.10997 2023. verified arXiv 2026-11-03.

Gentzkow, Matthew, Bryan Kelly, and Matt Taddy, "Text as Data," *Journal of Economic Literature*, 2019, *57* (3), 535–574. verified Crossref 2026-11-03.

Golchin, Shahriar and Mihai Surdeanu, "Time Travel in LLMs: Tracing Data Contamination in Large Language Models," arXiv preprint arXiv:2308.08493 2024. verified arXiv 2026-11-03.

Guu, Kelvin, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang, "REALM: Retrieval-Augmented Language Model Pre-Training," arXiv preprint arXiv:2002.08909 2020. verified arXiv 2026-11-03.

Hartley, James, "Current Findings from Research on Structured Abstracts: An Update," *Journal of the Medical Library Association*, 2014, *102* (3), 146–148. verified Crossref 2026-11-03.

Hengel, Erin, "Publishing While Female: Are Women Held to Higher Standards? Evidence from Peer Review," *The Economic Journal*, 2022, *132* (648), 2951–2991. verified Crossref 2026-11-03.

Holm, Sture, "A Simple Sequentially Rejective Multiple Test Procedure," *Scandinavian Journal of Statistics*, 1979, *6* (2), 65–70. verified JSTOR stable URL 2026-11-03 (no DOI registered).

Hosseini, Mohammad and Serge P. J. M. Horbach, "Fighting Reviewer Fatigue or Amplifying Bias? Considerations and Recommendations for Use of ChatGPT and Other Large Language Models in Scholarly Peer Review," *Research Integrity and Peer Review*, 2023, *8* (1). verified Crossref 2026-11-03.

Hyland, Ken, "Writing Without Conviction? Hedging in Science Research Articles," *Applied Linguistics*, 1996, *17* (4), 433–454. verified Crossref 2026-11-03.

_ , *Hedging in Scientific Research Articles* Pragmatics & Beyond New Series, Amsterdam: John Benjamins, 1998. verified Crossref 2026-11-03.

_ **and Feng (Kevin) Jiang**, "Is Academic Writing Becoming More Informal?," *English for Specific Purposes*, 2017, *45*, 40–51. verified Crossref 2026-11-03.

_ **and Polly Tse**, "Metadiscourse in Academic Writing: A Reappraisal," *Applied Linguistics*, 2004, *25* (2), 156–177. verified Crossref 2026-11-03.

Kanoksilapatham, Budsaba, "Rhetorical Structure of Biochemistry Research Articles," *English for Specific Purposes*, 2005, *24* (3), 269–292. verified Crossref 2026-11-03.

Kapoor, Sayash and Arvind Narayanan, "Leakage and the Reproducibility Crisis in Machine-Learning-Based Science," *Patterns*, 2023, *4* (9), 100804. verified Crossref 2026-11-03.

Karpukhin, Vladimir, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih, "Dense Passage Retrieval for Open-Domain Question Answering," in "Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)" Association for Computational Linguistics 2020, pp. 6769–6781. verified Crossref 2026-11-03.

Kobak, Dmitry, Rita González-Márquez, Emőke-Ágnes Horvát, and Jan Lause, "Delving into LLM-Assisted Writing in Biomedical Publications through Excess Vocabulary," *Science Advances*, 2025, *11* (27), eadt3813. verified Crossref 2026-11-03.

Korinek, Anton, "Generative AI for Economic Research: Use Cases and Implications for Economists," *Journal of Economic Literature*, 2023, *61* (4), 1281–1317. verified Crossref 2026-11-03.

Krippendorff, Klaus, *Content Analysis: An Introduction to Its Methodology*, 4 ed., Thousand Oaks, CA: SAGE Publications, 2019. verified Crossref 2026-11-03.

Latona, Giuseppe Russo, Manoel Horta Ribeiro, Tim R. Davidson, Veniamin Veselovsky, and Robert West, "The AI Review Lottery: Widespread AI-Assisted Peer Reviews Boost Paper Scores and Acceptance Rates," arXiv preprint arXiv:2405.02150 2024. verified arXiv 2026-11-03.

Lee, Katherine, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini, "Deduplicating Training Data Makes Language Models Better," in "Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)" Association for Computational Linguistics 2022, pp. 8424–8445. verified Crossref 2026-11-03.

Lewis, Patrick, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," arXiv preprint arXiv:2005.11401 2020. verified arXiv 2026-11-03.

Li, Feng, "Annual Report Readability, Current Earnings, and Earnings Persistence," *Journal of Accounting and Economics*, 2008, *45* (2–3), 221–247. verified Crossref 2026-11-03.

Liang, Weixin, Yaohui Zhang, Zhengxuan Wu, Haley Lepp, Wenlong Ji, Xuandong Zhao, Hancheng Cao, Sheng Liu, Siyu He, Zhi Huang, Diyi Yang, Christopher Potts, Christopher D. Manning, and James Y. Zou, "Mapping the Increasing Use of LLMs in Scientific Papers," arXiv preprint arXiv:2404.01268 2024. verified arXiv 2026-11-03.

_ , Yuhui Zhang, Hancheng Cao, Binglu Wang, Daisy Yi Ding, Xinyu Yang, Kailas Vodrahalli, Siyu He, Daniel Scott Smith, Yian Yin, Daniel A. McFarland, and James Zou, "Can Large Language Models Provide Useful Feedback on Research Papers? A Large-Scale Empirical Analysis," *NEJM AI*, 2024, *1* (8). verified Crossref 2026-11-03.

_ , Zachary Izzo, Yaohui Zhang, Haley Lepp, Hancheng Cao, Xuandong Zhao, Lingjiao Chen, Haotian Ye, Sheng Liu, Zhi Huang, Daniel A. McFarland, and James Y. Zou, "Monitoring AI-Modified Content at Scale: A Case Study on the Impact of ChatGPT on AI Conference Peer Reviews," arXiv preprint arXiv:2403.07183 2024. verified arXiv 2026-11-03.

Lin, Ling and Stephen Evans, "Structural Patterns in Empirical Research Articles: A Cross-Disciplinary Study," *English for Specific Purposes*, 2012, *31* (3), 150–160. verified Crossref 2026-11-03.

Lo, Kyle, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld, "S2ORC: The Semantic Scholar Open Research Corpus," in "Proceedings of the 58th Annual Meeting of the

Association for Computational Linguistics" Association for Computational Linguistics 2020, pp. 4969–4983. verified Crossref 2026-11-03.

Lopez, Patrice, "GROBID: Combining Automatic Bibliographic Data Recognition and Term Extraction for Scholarship Publications," in "Research and Advanced Technology for Digital Libraries (ECDL 2009)" Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2009, pp. 473–474. verified Crossref 2026-11-03.

Loughran, Tim and Bill McDonald, "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks," *The Journal of Finance*, 2011, *66* (1), 35–65. verified Crossref 2026-11-03.

_ **and** _ , "Measuring Readability in Financial Disclosures," *The Journal of Finance*, 2014, *69* (4), 1643–1671. verified Crossref 2026-11-03.

_ **and** _ , "Textual Analysis in Accounting and Finance: A Survey," *Journal of Accounting Research*, 2016, *54* (4), 1187–1230. verified Crossref 2026-11-03.

Magar, Inbal and Roy Schwartz, "Data Contamination: From Memorization to Exploitation," in "Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)" Association for Computational Linguistics 2022, pp. 157–165. verified Crossref 2026-11-03.

McCloskey, Donald N., "Economical Writing," *Economic Inquiry*, 1985, *23* (2), 187–222. verified Crossref 2026-11-03.

Noorden, Richard Van and Jeffrey M. Perkel, "AI and Science: What 1,600 Researchers Think," *Nature*, 2023, *621* (7980), 672–675. verified Crossref 2026-11-03.

Nosek, Brian A., Charles R. Ebersole, Alexander C. DeHaven, and David T. Mellor, "The Preregistration Revolution," *Proceedings of the National Academy of Sciences*, 2018, *115* (11), 2600–2606. verified Crossref 2026-11-03.

Panickssery, Arjun, Samuel R. Bowman, and Shi Feng, "LLM Evaluators Recognize and Favor Their Own Generations," arXiv preprint arXiv:2404.13076 2024. verified arXiv 2026-11-03.

Pineau, Joelle, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle, "Improving Reproducibility in Machine Learning Research (A Report from the NeurIPS 2019 Reproducibility Program)," arXiv preprint arXiv:2003.12206 2021. verified arXiv 2026-11-03.

Plavén-Sigray, Pontus, Granville James Matheson, Björn Christian Schiffler, and William Hedley Thompson, "The Readability of Scientific Texts Is Decreasing over Time," *eLife*, 2017, *6*, e27725. verified Crossref 2026-11-03.

Reimers, Nils and Iryna Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," 2019. verified arXiv 2026-11-03.

Robertson, Stephen and Hugo Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," *Foundations and Trends in Information Retrieval*, 2009, *3* (4), 333–389. verified Crossref 2026-11-03.

Sainz, Oscar, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier López de Lacalle, and Eneko Agirre, "NLP Evaluation in Trouble: On the Need to Measure LLM Data Contamination for Each Benchmark," in "Findings of the Association for Computational Linguistics: EMNLP 2023" Association for Computational Linguistics 2023, pp. 10776–10787. verified Crossref 2026-11-03.

Samraj, Betty, "Introductions in Research Articles: Variations Across Disciplines," *English for Specific Purposes*, 2002, *21* (1), 1–17. verified Crossref 2026-11-03.

Shumailov, Ilia, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal, "AI Models Collapse When Trained on Recursively Generated Data," *Nature*, 2024, *631* (8022), 755–759. verified Crossref 2026-11-03.

Si, Chenglei, Diyi Yang, and Tatsunori Hashimoto, "Can LLMs Generate Novel Research Ideas? A Large-Scale Human Study with 100+ NLP Researchers," arXiv preprint arXiv:2409.04109 2024. verified arXiv 2026-11-03.

Swales, John M., *Genre Analysis: English in Academic and Research Settings*, Cambridge: Cambridge University Press, 1990. verified Open Library (ISBN 0521338131) and Crossref-indexed reviews (10.2307/329786; 10.1017/s0272263100011773) 2026-11-03.

_ , *Research Genres: Explorations and Applications*, Cambridge: Cambridge University Press, 2004. verified Crossref 2026-11-03.

Teufel, Simone and Marc Moens, "Summarizing Scientific Articles: Experiments with Relevance and Rhetorical Status," *Computational Linguistics*, 2002, *28* (4), 409–445. verified Crossref 2026-11-03.

Thelwall, Mike, "Can ChatGPT Evaluate Research Quality?," *Journal of Data and Information Science*, 2024, *9* (2), 1–21. verified Crossref 2026-11-03.

Thomson, William, "The Young Person's Guide to Writing Economic Theory," *Journal of Economic Literature*, 1999, *37* (1), 157–183. verified Crossref 2026-11-03.

Wang, Peiyi, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui, "Large Language Models Are Not Fair Evaluators," in "Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)" Association for Computational Linguistics 2024, pp. 9440–9450. verified Crossref 2026-11-03.

Wang, Wenhui, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou, "MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers," 2020. verified arXiv 2026-11-03.

Zheng, Lianmin, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica, "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena," arXiv preprint arXiv:2306.05685 2023. verified arXiv 2026-11-03.

A. Ontology schema

The annotation schema (ontology version 1.0) is defined once, in `src/jof250/rhetoric.py`, as a JSON Schema (draft 2020-12) with `additionalProperties` set to false and every listed property required. The controlled vocabularies are those of Table 4; the complete list of fields follows. Paths use dot notation; `[]` denotes an array. Indices are zero-based (`sentence_index` within the abstract, `paragraph_index` within the introduction, `idx` over the sections of the payload); the value -1 means "not seen". The `causal_language_strength` scale runs from 1 (purely associational wording) through 3 (cautious causal wording with explicit caveats) to 5 (unqualified causal claims); `confidence` is the annotator's overall confidence in $[0, 1]$.

- `genre` — string: `empirical`, `theoretical`, `mixed_theory_empirical`, `methodological_econometric`, `survey_or_address`, `replication_or_correction`
- `subfield` — string: `asset_pricing`, `corporate_finance`, `financial_intermediation_banking`, `household_finance`, `market_microstructure`, `international_finance`, `behavioral_finance`, `macro_finance`, `financial_econometrics`, `entrepreneurial_finance_private_equity`, `other`
- `subfield_secondary` — string
- `primary_method` — string: `reduced_form_regression`, `difference_in_differences`, `instrumental_variables`, `regression_discontinuity`, `natural_experiment_event_study`, `structural_estimation`, `calibration_simulation`, `theoretical_model_analytical`, `machine_learning_textual`, `survey_or_experiment`, `other`, `none`
- `identification.estimand` — string
- `identification.strategy` — string
- `identification.key_assumptions[]` — array of string
- `identification.threats_addressed[]` — array of string
- `identification.causal_language_strength` — integer
- `identification.uses_causal_claim` — boolean
- `abstract_moves.question.present` — boolean
- `abstract_moves.question.sentence_index` — integer
- `abstract_moves.gap.present` — boolean
- `abstract_moves.gap.sentence_index` — integer
- `abstract_moves.method.present` — boolean
- `abstract_moves.method.sentence_index` — integer
- `abstract_moves.results.present` — boolean
- `abstract_moves.results.sentence_index` — integer
- `abstract_moves.magnitude.present` — boolean
- `abstract_moves.magnitude.sentence_index` — integer
- `abstract_moves.contribution.present` — boolean
- `abstract_moves.contribution.sentence_index` — integer

- introduction.moves[].paragraph_index — integer
- introduction.moves[].move — string: motivation, tension_or_puzzle, literature_gap, preview_of_approach, preview_of_results, contribution_statement, mechanism_or_theory_sketch, identification_defense, robustness_mention, roadmap, related_literature_positioning, policy_or_broader_implication, other
- introduction.contribution_paragraph_index — integer
- introduction.contribution_types[] — array of string: new_evidence, new_data_or_measure, new_identification, new_theory_or_model, new_method, resolution_of_puzzle, replication_or_correction, synthesis
- introduction.roadmap_present — boolean
- introduction.literature_positioning_location — string: inside_introduction, separate_section, both, none
- section_roles[].idx — integer
- section_roles[].role — string: introduction, literature, theory, data, empirical_design, results, robustness, conclusion, appendix, other
- results_presentation.economic_significance_discussed — boolean
- results_presentation.statistical_significance_language — string: stars_and_tstats, confidence_intervals, p_values, mixed, unclear
- results_presentation.magnitude_statements_in_intro — boolean
- robustness.location — string: main_text_section, within_results, appendix, internet_appendix, mixed, none_seen
- robustness.types[] — array of string: alternative_specification, placebo, subsample, alternative_measure, alternative_sample_period, falsification, sensitivity_to_parameters, other
- tables.convention_notes — string
- tables.units_stated_in_notes — boolean
- tables.standard_errors_description — string: clustered, robust, bootstrap, newey_west, mixed, unclear, none
- figures.uncertainty_shown — string: yes, no, mixed, unclear, no_figures
- conclusion.bounded_claims — boolean
- conclusion.policy_implications — boolean
- conclusion.open_questions — boolean
- conclusion.restates_contribution — boolean
- hedging.level — string: low, moderate, high
- hedging.typical_devices[] — array of string
- claim_evidence[].claim — string
- claim_evidence[].evidence_type — string: table, figure, equation_or_proof, prior_literature, descriptive_statistic, none_stated
- claim_evidence[].evidence_ref — string

- `structural_summary` — string
- `writing_profile_keywords[]` — array of string
- `confidence` — number
- `notes` — string

B. Prompts

The prompt files are reproduced verbatim from `prompts/` (YAML front matter omitted; prompt identifier and version shown). Placeholders in braces (`{section}`, `{retrieved}`, `{profile_summary}`, `{genre_note}`, `{conventions}`, ...) are substituted by the engine at run time. For the annotation prompt, the engine appends the list of allowed values (Table 4) and the JSON schema of Appendix A to the body shown here. In the writing engine, the system prompt is shared by every condition; the condition prompts are placed after the manuscript material in the user message, and condition E uses the condition-D prompt for its first call and the referee prompt for its second.

B.1. Annotation prompt

`annotation/document_annotation_v1.md` **(prompt id** `document_annotation`**, version 1.0).**

> You are a careful annotator of scientific writing in financial economics. You receive bounded excerpts of ONE published article (abstract, introduction paragraphs, conclusion paragraphs, section headings with the first paragraph of each section, table titles/notes, figure captions, counts). Your job is to describe HOW the argument is built — its genre, rhetorical moves, claim–evidence links, hedging and presentation conventions — using the fixed schema.
>
> Rules:
>
> 1. Annotate only what is visible in the excerpts; when something is not visible, say so with the provided "unclear"/"none_seen" values or -1 indices. Never guess a strategy or result that is not in the text.
>
> 2. Indices: ‘sentence_index‘ refers to the 0-based sentence position in the abstract; ‘paragraph_index‘ refers to the 0-based position in ‘introduction_paragraphs‘; section ‘idx‘ refers to the ‘idx‘ field of ‘sections‘.
>
> 3. Do NOT copy sentences. ‘structural_summary‘, ‘claim‘ fields and ‘convention_notes‘ must be your own abstract-level description (paraphrase at the level of moves, not wording). Keep each ‘claim‘ under 200 characters.
>
> 4. Theory papers do not have to follow the empirical template: assign roles honestly (theory, proofs in appendix) and set empirical-only fields to the "none"/"unclear"/"not applicable" values.

5. Identification: distinguish the estimand (what quantity is targeted), the strategy (how), assumptions (what must hold), and the strength of causal language (1 = purely associational wording, 3 = cautious causal wording with explicit caveats, 5 = unqualified causal claims).

6. ‘contribution_types‘: list every type that is explicitly claimed.

7. ‘confidence‘ is your overall confidence (0–1) that the annotation is faithful to the excerpts.

Return only the JSON object.

B.2. Writing prompts (system prompt and conditions A–E)

`writing/system_base_v1.md` **(prompt id** `writing_system_base`**, version 1.0).**

You are a scholarly writing engine for economics and finance manuscripts. You produce LaTeX-ready prose for a specified section of the user's manuscript.

Hard constraints (apply in every condition):

- Never invent results, data, numbers, coefficients, citations, or references. Every number you write must appear in the manuscript material you are given (abstract, existing sections, tables, research brief). Every citation must use an existing BibTeX key from the list provided (\citep{key}/\citet{key}); if you need a reference that is not in the list, write a visible placeholder ‘\todo{CITATION NEEDED: <what>}‘ instead of inventing one.

- Preserve the manuscript's verified facts exactly (numbers, signs, units, sample sizes, periods, variable names).

- Do not copy sentences from any other paper, do not imitate a specific author, and do not claim endorsement by any journal.

- Output format: first the LaTeX text of the section (no \section command unless asked, no preamble), then a line ‘=== CHANGE LOG ===‘ followed by a bullet list explaining the structural choices and any assumptions, then a line ‘=== FLAGS ===‘ followed by bullet items for anything that needs the author's verification (unsupported statements, missing evidence, placeholders). Keep the change log and flags concise.

`writing/condition_A_v1.md` **(prompt id** `condition_A_minimal`**, version 1.0).**

Write the {section} of this paper.

`writing/condition_B_v1.md` **(prompt id** `condition_B_strong_prompt`**, version 1.0).**

Write the {section} of this paper as an experienced academic economist would for a top general-interest finance or economics journal.

Guidance: state the research question early; motivate why it matters; make the contribution explicit and specific; relate to the closest prior work without a laundry list; preview the approach

and the main findings with their magnitudes; keep claims proportionate to the evidence; use precise, plain academic English; avoid hype, clichés and vague adjectives; prefer short paragraphs with one idea each; use the manuscript's own notation and numbers; write in the first person plural if the manuscript does, otherwise follow its voice.

`writing/condition_C_v1.md` **(prompt id** `condition_C_conventional_rag`**, version 1.0).**

Write the {section} of this paper as an experienced academic economist would for a top general-interest finance or economics journal.

Guidance: state the research question early; motivate why it matters; make the contribution explicit and specific; relate to the closest prior work without a laundry list; preview the approach and the main findings with their magnitudes; keep claims proportionate to the evidence; use precise, plain academic English; avoid hype, clichés and vague adjectives; prefer short paragraphs with one idea each; use the manuscript's own notation and numbers; write in the first person plural if the manuscript does, otherwise follow its voice.

Below are short excerpts retrieved by similarity from a private corpus of published Journal of Finance articles (2017–2020). Use them only as examples of tone and organization; do not reuse their sentences, numbers or claims, and do not cite them unless the manuscript's bibliography already contains them.

{retrieved}

`writing/condition_D_v1.md` **(prompt id** `condition_D_profiles_structured`**, version 1.0).**

Write the {section} of this paper as an experienced academic economist would for a top general-interest finance or economics journal.

Manuscript genre/subfield (detected): {genre} / {subfield}. Applicable measured writing profile: '{profile_name}' (built from {profile_n} published Journal of Finance articles, 2017–2020; measured frequencies describe practice, they are not rules — deviate when the manuscript's logic requires it and say so in the change log):

{profile_summary}

Structural patterns retrieved from the annotated corpus (rhetorical moves, claim–evidence links, identification framing; each with provenance). Use them to plan the section's moves — not as text to reuse:

{retrieved}

Procedure: (1) identify what this manuscript's actual evidence supports; (2) choose the sequence of rhetorical moves for the section and state it briefly in the change log; (3) write the section; (4) list every statement that the manuscript material does not support under FLAGS.

{genre_note}

`writing/condition_E_referee_v1.md` **(prompt id** `condition_E_specialist_referee`**, version 1.0).**

You are a specialist referee for a top finance journal (econometric referee + writing editor). You receive (a) the manuscript material, (b) a draft of its {section}, (c) a deterministic audit of the draft (numbers not present in the manuscript, citation keys not in the bibliography, possibly unsupported claims, overlap with a corpus). Produce a revised draft that fixes every audit item, keeps every verified fact unchanged, strengthens the methodological framing (identification, what the estimates can and cannot support), keeps claims bounded, and removes any statement the material does not support (replace with '\todo{...}' if essential).

Output exactly: the revised LaTeX text, then '=== CHANGE LOG ===' with the referee's edits (what, why), then '=== FLAGS ===' with remaining items for the author.

{retrieved}

B.3. Review prompts

`review/referee_v1.md` **(prompt id** `review_referee`**, version 1.0).**

You are an anonymous referee for a top general-interest finance journal. Write a referee report on the manuscript material provided (title, abstract, sections, tables, figures, equations, audit facts). Structure: (1) Summary of the paper in your own words (one paragraph); (2) Assessment of the contribution and positioning; (3) Major comments, numbered, each tied to a specific section/table/equation and stating what evidence or change would resolve it (identification and inference first, then measurement, then robustness, then presentation); (4) Minor comments (writing, tables, figures, notation, references); (5) Recommendation with a one-sentence justification. Be specific, fair, and proportionate; do not invent results that are not in the material; when the material is incomplete, say what is missing instead of guessing. {conventions}

`review/editor_v1.md` **(prompt id** `review_editor`**, version 1.0).**

You are a finance writing editor. For the manuscript material provided, assess the writing section by section against the measured conventions below (treat them as descriptive benchmarks, not rules): clarity of the research question, placement and specificity of the contribution, preview of findings with magnitudes, proportionality of claims, paragraph structure, transitions, notation consistency, hedging. For each section give: strengths, concrete rewrite suggestions (quote the problematic sentence, propose a replacement that keeps every fact), and a priority (high/medium/low). Then list global issues. Do not change any number or claim. {conventions}

`review/table_editor_v1.md` **(prompt id** `review_table_editor`**, version 1.0).**

> You are a table-and-figure editor for a finance journal. For every table and figure in the material: check that the caption states what is shown, the sample, the unit of observation and the period; that notes define variables, units, standard errors and significance conventions; that labels are referenced in the text; that figures state uncertainty where estimates are shown. Propose a corrected caption and notes for each (keeping all numbers unchanged) and rank the fixes. {conventions}

`review/integrity_v1.md` **(prompt id** `review_integrity`**, version 1.0).**

> You are a citation and integrity auditor. Using the manuscript material and the deterministic audit facts provided (unknown citation keys, DOI resolution results, numbers in text not found in tables, possibly unsupported claims, corpus overlap), produce an integrity report: (1) citations that cannot be verified or are used inconsistently; (2) claims in the text that lack a table/figure/equation/citation anchor; (3) numbers that appear in prose but not in any table; (4) statements whose causal strength exceeds the design; (5) reproducibility statements missing. For each item give location, severity and the minimal fix. Do not speculate beyond the material. {conventions}

`review/replication_v1.md` **(prompt id** `review_replication`**, version 1.0).**

> You are a replication auditor. From the manuscript material, extract what a replicator would need: data sources and access, sample construction steps and filters, variable definitions, estimation specifications, software, seeds, and where each is stated. List every item as present/absent with the location, then list the questions the authors must answer for the paper to be replicable. {conventions}

B.4. Judge rubric

`evaluation/judge_rubric_v1.md` **(prompt id** `judge_rubric`**, version 1.0).**

> You are a blind, calibrated judge of scholarly writing in economics and finance. You receive: the TASK, the SOURCE MATERIAL (verified facts the outputs had to respect), and several OUTPUTS labelled with random letters. The outputs were produced under different, undisclosed conditions; do not try to guess them, and do not reward length.
>
> Score every output on each dimension from 1 to 7 using these anchors:
>
> - clarity — 1: hard to follow, vague; 4: readable but uneven; 7: every paragraph has one clear idea, precise wording.
>
> - argument_logic — 1: claims without reasons or in a confused order; 4: mostly logical with gaps; 7: question → approach → evidence → bounded conclusion, each step motivated.

- contribution_positioning — 1: contribution absent or generic; 4: stated but not sharply differentiated from prior work; 7: specific, differentiated, placed early, proportionate.

- methodological_rigor — 1: ignores or misstates design/identification/inference; 4: mentions but does not engage; 7: engages estimand, assumptions, threats, inference, limits.

- fidelity — 1: invents or alters numbers/findings/citations; 4: faithful but omits important facts; 7: every number, sign, unit and claim matches the source and nothing is invented (unverifiable citations and numbers absent from the source are serious faults).

- usefulness — 1: the author could not use it; 4: partially usable; 7: could be adopted with light editing.

Also report for each output: n_invented_facts (numbers, findings or references not in the source), and a one-sentence justification.

Return only JSON: {"scores": {"<letter>": {"clarity": int, "argument_logic": int, "contribution_positioning": int, "methodological_rigor": int, "fidelity": int, "usefulness": int, "n_invented_facts": int, "justification": str}, ...}, "best": "<letter>", "notes": str}

C. Parser QA

Table 11 reproduces the per-document rows of `data/derived/parser_qa.md`. The sample comprises 26 documents: five randomly chosen research articles per year (seed 45) and every document of a non-article type. For each component the table gives the count produced by the PyMuPDF structural parser (P) and the count produced by regular expressions on `pdftotext -layout` output (R). Aggregate agreement rates are reported in Section 3.4: sections exact 85%, tables exact 77% (within one 96%), figures exact 92%, equations exact 8% (within 10% 35%), references within 10% 92%, abstract found by both 100%; title agreement with Crossref (similarity at least 90) over all 276 included documents 99%. The check compares two automated extractions; it is not a human gold standard. The last row (a corrigendum) illustrates the failure mode of both tools on a one-page note without the article template: the reference regular expression counts 169 lines where the structural parser finds 5 entries and neither finds sections, tables, or an abstract.

D. Profiles

Table 12 lists every profile file and its selection rule; Tables 13 and 14 give the structure medians and interquartile ranges of each, and Table 15 the main entries of their rhetoric blocks. All profiles are computed on the training split only (221 documents), with 95% bootstrap intervals for the medians (500 resamples, seed 45) stored in the YAML files. The engine falls back from a profile with fewer than eight documents to `empirical_all` and then to `corpus_all`. Rhetoric blocks

Table 11. Parser QA: per-document comparison of the PyMuPDF structural parser (P) against regex counts on `pdftotext -layout` output (R) for the stratified sample. Columns give P/R counts of numbered sections, tables, figures, numbered equations, references and abstracts. This is a cross-parser agreement check, not a human gold standard.

Document	Type	Sections	Tables	Figures	Equations	References	Abstract
jof_12484	article	5/5	6/5	13/12	22/20	65/64	1/1
jof_12499	article	4/4	13/13	4/4	17/8	35/35	1/1
jof_12495	article	5/5	12/12	2/2	5/15	64/64	1/1
jof_12488	article	5/6	6/6	8/8	41/37	57/57	1/1
jof_12465	article	5/5	13/11	4/4	32/45	64/63	1/1
jof_12684	article	3/3	4/4	5/5	24/30	73/73	1/1
jof_12691	article	7/7	14/14	2/2	6/16	35/35	1/1
jof_12584	article	7/7	0/0	9/9	12/13	56/56	1/1
jof_12590	article	8/9	1/1	3/3	90/52	18/18	1/1
jof_12726	article	6/6	10/10	2/2	11/33	47/47	1/1
jof_12732	article	4/4	6/6	10/10	19/21	43/40	1/1
jof_12749	article	6/6	8/7	0/0	2/11	75/75	1/1
jof_12773	article	6/6	3/3	12/11	45/33	59/60	1/1
jof_12734	article	5/5	8/8	3/3	15/21	44/43	1/1
jof_12765	article	7/7	9/9	6/6	19/27	64/64	1/1
jof_12842	article	5/5	6/6	9/9	27/13	89/89	1/1
jof_12856	article	7/7	1/0	0/0	19/19	44/43	1/1
jof_12843	article	6/7	11/10	5/5	10/27	60/60	1/1
jof_12890	article	6/6	11/10	5/5	16/24	34/34	1/1
jof_12862	article	7/8	4/4	1/1	33/35	58/0	1/1
jof_12530	presidential_address	10/10	4/4	7/7	0/1	69/66	1/1
jof_12548	corrigendum_or_replication	2/2	2/2	0/0	38/6	4/4	1/1
jof_12710	presidential_address	5/5	6/6	14/14	22/27	72/71	1/1
jof_12762	corrigendum_or_replication	4/4	0/0	0/0	19/20	12/10	1/1
jof_12782	presidential_address	3/3	0/0	9/9	38/40	45/44	1/1
jof_12786	corrigendum_or_replication	0/0	0/0	0/0	0/0	169/5	0/0

(abstract and introduction moves, contribution placement, identification, results, robustness, table, figure, conclusion, and hedging conventions) are present in a profile file only when the subset contains schema-valid annotations, which is the case for every profile after the full annotation run; the YAML files hold the complete distributions, of which Table 15 shows a selection. Theory papers are included in `corpus_all` and `theoretical_finance` but excluded by genre from the empirical profiles, so that empirical-paper conventions are never averaged with theory-paper conventions.

Table 12. Profile files and their selection rules (`PROFILE_DEFS` in `src/jof250/features.py`).

Profile	Selection rule (train split only)
`corpus_all`	all included train documents
`empirical_all`	genre in {empirical, mixed_theory_empirical}
`empirical_asset_pricing`	annotation.subfield == asset_pricing and genre in {empirical, mixed_theory_empirical}
`corporate_finance`	annotation.subfield == corporate_finance and genre in {empirical, mixed_theory_empirical}
`theoretical_finance`	genre == theoretical
`financial_econometrics`	subfield == financial_econometrics or genre == methodological_econometric
`financial_intermediation_banking`	annotation.subfield == financial_intermediation_banking (empirical/mixed)
`household_behavioral_finance`	subfield in {household_finance, behavioral_finance} (empirical/mixed)

Table 13. Structure medians (interquartile range in parentheses) of the profile files in `profiles/`, part A: all = `corpus_all`; empirical = `empirical_all`; asset pricing = `empirical_asset_pricing`; corporate finance = `corporate_finance`. Training split only. A cell prints TODO if its macro is absent from `numbers.tex`. Observed frequencies describe what published articles did; they are not editorial rules.

Statistic	all	empirical	asset pricing	corporate finance
n documents	221	173	58	28
Body words	11009 (10035–12407)	11009 (10052–12194)	10920 (9901–12250)	11150 (10415–11550)
Introduction words	1804 (1568–2132)	1804 (1588–2116)	1729 (1526–2160)	1927 (1738–2119)
Introduction paragraphs	7 (6–9)	7 (6–9)	8 (6–10)	8 (6–11)
Introduction share	0.16 (0.14–0.19)	0.17 (0.14–0.19)	0.16 (0.14–0.18)	0.17 (0.15–0.20)
Tables	8 (6–11)	9 (7–11)	10 (8–11)	10 (8–12)
Numbered equations	10 (4–23)	7 (2–15)	14 (9–25)	5 (2–8)
Reference entries	52 (41–64)	53 (43–64)	53 (44–68)	50 (36–59)
Conclusion share	0.03 (0.02–0.05)	0.03 (0.02–0.05)	0.03 (0.02–0.04)	0.04 (0.03–0.06)

E. Splits and fixtures

Split protocol. `heldout_v1` (`data/splits/heldout_v1.json`, `scripts/03_make_splits.py`, seed 45) is a document-level split of the 276 included documents: within each (publication year,

Table 14. Structure medians (interquartile range in parentheses) of the profile files in `profiles/`, part B: theoretical = `theoretical_finance`; econometrics = `financial_econometrics`; banking = `financial_intermediation_banking`; household = `household_behavioral_finance`. Training split only. A cell prints TODO if its macro is absent from `numbers.tex`. Observed frequencies describe what published articles did; they are not editorial rules.

Statistic	theoretical	econometrics	banking	household
n documents	32	8	21	32
Body words	11064 (10315–13576)	12412 (9017–14125)	11218 (10601–11678)	10576 (9716–12794)
Introduction words	1896 (1673–2359)	1512 (1001–2292)	1728 (1653–2069)	1830 (1598–2063)
Introduction paragraphs	8 (5–9)	5 (4–9)	6 (5–8)	7 (7–10)
Introduction share	0.16 (0.13–0.22)	0.13 (0.10–0.17)	0.16 (0.14–0.18)	0.17 (0.16–0.19)
Tables	0 (0–2)	6 (4–6)	10 (7–12)	8 (6–11)
Numbered equations	36 (22–63)	22 (8–37)	5 (2–7)	2 (0–7)
Reference entries	45 (38–64)	52 (39–67)	49 (45–59)	56 (45–68)
Conclusion share	0.03 (0.01–0.04)	0.04 (0.03–0.04)	0.04 (0.03–0.05)	0.04 (0.03–0.05)

Table 15. Rhetoric block of each profile (model annotation, training split only). Abstract columns: share (%) of abstracts in which the move is present. Roadmap: share (%) of introductions with a roadmap paragraph according to the model (LLM) and to the regular expression (regex). Contribution position: median relative position of the contribution paragraph in the introduction (interquartile range). Causal: median causal-language score (1–5). Econ. sig.: share (%) discussing economic significance. Bounded / open: share (%) of conclusions with bounded claims / open questions. A cell prints TODO if its macro is absent. Descriptive, not prescriptive.

		Abstract moves present (%)						Roadmap (%)					Conclusion (%)	
Profile	*n*	Quest.	Gap	Method	Results	Magn.	Contrib.	LLM	Regex	Contrib. pos.	Causal	Econ. sig.	Bounded	Open
all	221	70	14	83	99	24	69	63	60	0.58 (0.13–0.78)	3.0	79	82	37
empirical	173	72	12	83	100	29	65	65	64	0.62 (0.20–0.80)	3.0	92	83	39
asset pricing	58	66	12	84	100	21	69	67	64	0.60 (0.18–0.80)	2.0	93	79	40
corporate finance	28	57	11	79	100	29	71	57	57	0.78 (0.64–0.86)	3.0	93	89	29
theoretical	32	72	0	88	100	0	81	53	50	0.40 (0.00–0.66)	1.0	25	81	28
econometrics	8	62	62	88	100	12	100	62	50	0.45 (0.15–0.54)	1.0	50	75	25
banking	21	76	10	81	100	38	62	71	67	0.67 (0.46–0.75)	3.0	100	86	38
household	32	84	6	91	100	38	47	72	72	0.46 (0.02–0.69)	3.0	94	81	50

article type) stratum, 20% of the document identifiers are sampled without replacement into the held-out set, giving 221 training and 55 held-out documents. The file records the protocol name, seed, rule, counts, the two identifier lists, the author-overlap count (20 held-out documents share at least one author surname with a training document; overlap is reported, not eliminated), and the usual provenance stamp. Profiles, the training-only retrieval filter, and prompt tuning use the training list only; the held-out list is used solely to build evaluation fixtures. Document identifiers are of the form `jof_`*nnnnn*, the numeric suffix of the article's DOI.

Fixtures. `scripts/04_build_fixtures.py` (seed 45) selects, among held-out empirical articles with at least six tables, two articles of different subfields and writes for each a directory containing `manuscript.md` (the parsed structure rendered as Markdown with the published introduction withheld), `manuscript_with_intro.md`, `published_introduction.txt`, `meta.json`, and the deterministically degraded literature passage used by the rewrite task; the out-of-domain fixture points to the LaTeX source of UQO Working Paper 29. Fixture directories contain excerpts of copyrighted text and are therefore private (excluded from version control), as is the raw output of every engine run on them. The held-out identifiers are listed in the split file; they are not reproduced here.

F. Reproducibility

Table 16 lists the command that regenerates each artefact (`docs/REPRODUCIBILITY.md`). Every JSON artefact carries a `timestamp`, a `code_version` (git short hash, suffixed `-dirty` when uncommitted changes were present), and a `host` description; the environment is pinned in `pyproject.toml` (Python 3.12.13) and model identifiers in `config/models.yaml`. Stages that call a language model are not deterministic; their inputs, outputs, and costs are stored so that any analysis can be re-run on the recorded outputs.

G. User manual

This appendix condenses `docs/USER_GUIDE.md` and `docs/INSTALL.md`.

Installation. Create the environment (`uv venv -python 3.12 .venv` and `uv pip install -e ".[dev]"`), copy `.env.example` to `.env` and point `JOF250_ANTHROPIC_KEY_FILE` to a file holding the API key (never committed), and run `.venv/bin/jof250 -help`. The corpus root is set in `config/corpus.yaml`; derived artefacts are regenerated with the commands of Table 16. Opening Claude Code inside the repository makes the project-scoped commands available; `jof250`

Table 16. Regeneration commands (run from the repository root; `make` targets wrap `.venv/bin/jof250`).

Artefact	Command	Deterministic?
Corpus manifest and audit	`make audit`	yes, given the Crossref cache
Parsed documents	`make parse`	yes (parser version in each file)
Parser QA	`.venv/bin/python -I scripts/02_parser_qa.py`	yes (seed 45)
Splits	`.venv/bin/python -I scripts/03_make_splits.py`	yes (seed 45)
Annotations	`make annotate`	no (LLM; responses cached by prompt hash)
Profiles and figures	`make profiles`	yes (bootstrap seed 45)
Retrieval index	`make index`	yes (same embedding model)
Fixtures	`.venv/bin/python -I scripts/04_build_fixtures.py`	yes (seed 45)
Benchmark	`jof250 benchmark -config config/evaluation.yaml`	no (LLM; all raw runs stored)
Analysis, paper tables and figures	`.venv/bin/python -I scripts/05_analyze_benchmark.py`	yes (seed 45)
Number macros	`.venv/bin/python -I scripts/07_make_numbers.py`	yes
Paper	`cd paper && make`	yes
Tests	`make test`	yes (synthetic fixtures; no API calls)

`install-claude-code -scope user` copies only the `jof-` files into the user's Claude Code directory (removal: delete those files). Commands call the absolute path of the repository's `.venv/bin/jof250`; after moving the repository, re-run the install command.

Concepts. A *profile* is a measured distribution from the training split, to be read as what published articles did, not as a rule (`/jof-profile <name>`). *Structured retrieval* is hybrid BM25 and local-embedding search over annotation units and short excerpts, every hit carrying provenance (`jof250 retrieve -topic "..." [-unit annotation|raw] [-section results] [-year 2019]`). The *engine* (`jof250 write`) takes `-mode outline|write|upgrade`, `-section` (introduction, literature, data, empirical_design, results, robustness, conclusion, abstract), and `-condition` (A to E; default E).

Output contract. The engine always writes a *new* file at the requested `-output` path, headed by a comment stating that it is a proposal and that the manuscript was not modified, and a sibling `.report.md` holding the change log, the flags (statements the author must verify), the retrieval provenance, the deterministic audit (numeric preservation, unknown citation keys, unsupported-claim sentences, corpus overlap), and a unified diff against the current section. Reviews (`jof250 review -role referee|editor|table_editor|integrity|replication`) write a Markdown report with the deterministic audit appended; audits (`jof250 audit`) and evidence lookups (`jof250 evidence -claim "..."`) make no API call. The manuscript itself is never modified; patches are applied only on the user's explicit request.

Slash commands and agents. The ten commands and their arguments are listed in Table 7; Claude Code passes the text typed after the command as `$ARGUMENTS`, and each command file instructs the assistant how to parse it. The seven specialist agents (Table 8) wrap the review and audit commands and propose patches or reports without editing the manuscript. The skill file (`.claude/skills/jof250/SKILL.md`) states the rules for the assistant: always pass an explicit output path under the paper's `jof250_proposals/` or `jof250_reviews/` folder or under `/tmp`; relay every flag and audit item and never present a generated number or citation as verified; for theory, methods, or non-finance manuscripts, say which conventions do not apply; warn that API calls cost money (`jof250 status` shows spend) and dry-run the benchmark first; and repair a missing environment with `make install` and `make index`.

Manuscripts supported. LaTeX projects (a directory with `paper/main.tex` or `main.tex`; `\input` and `\include` resolved; UQO `\WP...` macros read; `*.bib` files scanned for keys and DOIs) and Markdown files. Theory, methods, and non-finance manuscripts are accepted; the engine

prints a genre note explaining which corpus conventions do not apply.

H. Claude Code command files

The ten command files under `.claude/commands/` are generated from `integrations/claude_code.py`. One is reproduced verbatim below (`jof-write.md`); the others follow the same pattern (front matter with a description, an argument hint, and an allowed-tools list restricted to the project CLI and read-only utilities; a body that tells the assistant how to parse `$ARGUMENTS`, which CLI command to run, which files to read afterwards, and what to show the user).

—

description: Draft a manuscript section with the full JoF-250 engine (profiles, structured retrieval, verification, specialist referee)

argument-hint: <paper-path> –section <name> [–condition A|B|C|D|E]

allowed-tools: Bash(/Users/simon-pierreboucher/Desktop/UQO/Recherche/Recherche-UQO/methodes/wp45-jof250/.venv/bin/jof250:*), Bash(ls:*), Bash(cat:*), Bash(mkdir:*), Read, Glob, Grep

—

Parse ‘$ARGUMENTS‘ as ‘<paper-path> [–section <name>] [–condition X]‘ (defaults: introduction, E_full_engine). Run ‘/Users/simon-pierreboucher/Desktop/UQO/Recherche/Recherche-UQO/methodes/wp45-jof250/.venv/bin/jof250 write –paper <path> –section <name> –mode write –condition <X> –output <paper-path>/jof250_proposals/<name>_<date>.tex‘ (create the folder; this is a NEW file, the manuscript is never modified). Then read the ‘.report.md‘ next to it and show: the proposed text location, the change log, every FLAG (facts the author must verify), the deterministic audit summary (numbers not in the manuscript, unknown citation keys, unsupported claims, corpus overlap), and retrieval provenance.