

# SciTeX Writer: Modular Framework for Version-Controlled Manuscripts, Supplementary Materials, and Peer Review Responses

Yusuke Watanabe<sup>1</sup>, Second Author<sup>1</sup>, Third Author<sup>1</sup>

<sup>a</sup>*SciTeX.ai, Tokyo, Japan*

<sup>b</sup>*Second Institution, Department, City, Country*

<sup>c</sup>*Third Institution, Department, City, Country*

---

## Abstract

Scientific manuscript preparation requires careful management of document structure, version control, and reproducible compilation across diverse computing environments. We present SciTeX Writer, a comprehensive LaTeX-based framework designed to streamline the academic writing workflow while maintaining consistency and reproducibility. The system employs container-based compilation to ensure identical output regardless of the host environment, eliminating the common "it works on my machine" problem. Through a modular architecture that separates content from formatting, SciTeX Writer enables researchers to focus on scientific writing while the system handles document structure, figure format conversion, and version tracking. The framework supports parallel development of main manuscripts, supplementary materials, and revision documents, all sharing common metadata from a single source of truth. Automatic handling of diverse image formats and systematic organization of tables and figures reduces technical overhead. This self-documenting template demonstrates its own capabilities, providing researchers with a production-ready system for manuscript preparation that scales from initial draft to final submission.

*Keywords:* keyword one, keyword two, keyword three, keyword four,  
keyword five

---

29     ~ 2 figures, 0 tables, 157 words for abstract, and 2626 words for main  
30 text

## 31 1. Introduction

32     The preparation of scientific manuscripts involves numerous technical  
33 challenges that extend beyond the intellectual task of communicating re-  
34 search findings [? ]. Researchers must navigate complex typesetting systems,  
35 manage multiple document versions, coordinate figures and tables across for-  
36 mats, and ensure reproducible compilation environments [? ]. These techni-  
37 cal burdens can distract from the primary goal of clear scientific communi-  
38 cation and often lead to inconsistencies, formatting errors, and wasted time  
39 troubleshooting environment-specific compilation issues.

40     Traditional approaches to manuscript preparation typically rely on local  
41 LaTeX installations, where the specific versions of packages and compila-  
42 tion tools can vary significantly across different machines and over time [?  
43 ]. This variability creates reproducibility challenges, particularly in collab-  
44 orative environments where multiple authors work on different systems [?  
45 ]. Furthermore, the proliferation of image formats and the need to convert  
46 between them for different submission requirements adds another layer of  
47 complexity. Researchers often resort to ad-hoc scripts or manual processes  
48 to handle these conversions, leading to potential errors and inconsistent re-  
49 sults.

50     Existing solutions have addressed some aspects of this problem [? ]. Over-  
51 leaf and similar cloud-based platforms provide consistent compilation envi-  
52 ronments but require continuous internet connectivity and may not suit all  
53 research workflows. Version control systems like Git effectively track changes  
54 but require researchers to understand both LaTeX and version control simul-  
55 taneously. Template repositories exist for various journals, but they typically  
56 focus on formatting requirements rather than workflow automation and often  
57 duplicate common elements across documents.

58 The fundamental challenge lies in balancing flexibility with consistency.  
59 Researchers need systems that accommodate diverse content types, multi-  
60 ple output documents, and varying journal requirements while maintaining a  
61 single source of truth for shared elements like author lists and bibliographies.  
62 The system must be sufficiently automated to reduce technical overhead yet  
63 transparent enough that researchers retain full control over their content.  
64 Additionally, the solution must work reliably across different computing en-  
65 vironments without imposing steep learning curves or workflow disruptions.

66 SciTeX Writer addresses these challenges through a container-based, mod-  
67 ular architecture that separates content management from document com-  
68 pilation. The framework organizes manuscripts into distinct directories for  
69 main text, supplementary materials, and revision responses, while maintain-  
70 ing shared metadata in a common location. By leveraging containerization  
71 technology, the system guarantees identical compilation results regardless  
72 of the host operating system or local software versions. Automatic format  
73 conversion for figures and tables eliminates manual preprocessing steps, and  
74 built-in version tracking with difference generation facilitates collaborative  
75 writing and revision processes. This manuscript serves as a self-documenting  
76 example, demonstrating the system’s capabilities through its own structure  
77 and compilation.

## 78 **2. Methods**

79 The SciTeX Writer framework implements a modular architecture de-  
80 signed around three core principles: reproducible compilation, content-structure  
81 separation, and automated asset management. The system organizes docu-  
82 ments into three primary directories, each serving distinct purposes in the  
83 manuscript lifecycle while sharing common resources to maintain consistency.

### 84 *2.1. Repository Structure and Organization*

85 The framework employs a hierarchical directory structure where the `00_shared/`  
86 directory serves as the single source of truth for metadata including title, au-  
87 thor information, keywords, and bibliographic references. This centralized

approach eliminates duplication and ensures consistency across all output documents. The `01_manuscript/` directory contains the main manuscript with subdirectories for content sections, figures, and tables. Similarly, `02_supplementary/` follows an identical structure for supplementary materials, while `03_revision/` organizes revision letters by reviewer. Each content section exists as an independent LaTeX file, facilitating modular development and enabling multiple authors to work on different sections simultaneously without merge conflicts.

## 2.2. Multi-Engine Compilation System

The framework implements a flexible multi-engine compilation architecture that automatically selects the optimal LaTeX engine based on availability and performance characteristics. Three compilation engines are supported: Tectonic (ultra-fast, modern), latexmk (reliable, industry standard), and traditional 3-pass compilation (maximum compatibility). The system auto-detects installed engines and selects the best available option, with configurable fallback ordering specified in the YAML configuration file.

Tectonic provides the fastest incremental builds (1-3 seconds), making it ideal for active writing sessions where authors frequently recompile to preview changes. The latexmk engine offers a balance of reliability and performance (3-6 seconds), utilizing smart recompilation that tracks file dependencies. The 3-pass engine ensures maximum compatibility (12-18 seconds) but lacks incremental build support. Performance characteristics and trade-offs are documented in Supplementary Table ??.

To ensure reproducible builds across diverse computing environments, the framework leverages both Docker and Apptainer/Singularity containerization technologies [? ]. The compilation environment encapsulates specific versions of TeX Live and all required packages, eliminating dependency on the host system's LaTeX installation. Users invoke compilation through shell scripts that provide extensive command-line options (documented in Supplementary Table ??). This containerized approach guarantees that the same source files produce identical PDFs regardless of the underlying operating system,

118 making the system equally functional on Linux, macOS, Windows, and high-  
119 performance computing clusters.

### 120 *2.3. Automated Asset Processing*

121 The system implements automatic format conversion for both figures  
122 and tables through preprocessing scripts that execute during compilation [?  
123 ]. For figures, the framework accepts common image formats including  
124 PNG, JPEG, SVG, and PDF, automatically converting them to formats op-  
125 timized for LaTeX inclusion. Each figure resides in its own subdirectory  
126 within `01_manuscript/contents/figures/caption_and_media/`, with the  
127 caption defined in a corresponding `.tex` file. During compilation, a prepro-  
128 cessing script scans these directories, generates figure inclusion code, and  
129 compiles all figures into `FINAL.tex` for inclusion in the main document. Ta-  
130 bles follow an analogous structure, allowing authors to define complex table  
131 layouts separately from their incorporation into the document flow [? ].

### 132 *2.4. Version Control and Difference Tracking*

133 The framework integrates with Git to provide systematic version track-  
134 ing and automatic generation of difference documents. When authors cre-  
135 ate a new version through `make archive`, the system archives the current  
136 manuscript with a timestamp and version number. Subsequently, invoking  
137 `make diff` generates a PDF highlighting changes between versions using  
138 the `latexdiff` utility. This functionality proves particularly valuable during  
139 revision processes, where journals often require marked-up versions show-  
140 ing modifications. The revision directory structure accommodates multiple  
141 rounds of review, with separate subdirectories for editor and reviewer re-  
142 sponses, each containing both the original comments and author responses  
143 in a structured format that ensures complete documentation of the revision  
144 process.

### 145 *2.5. Manuscript Preparation*

146 This manuscript was prepared using SciTeX Writer [? ], an open-source  
147 scientific manuscript compilation system supporting multiple LaTeX compi-

148 lation engines including latexmk, traditional 3-pass compilation, and Tec-  
149 tonic.

### 150 3. Results

151 The SciTeX Writer framework successfully demonstrates comprehensive  
152 manuscript preparation capabilities through its modular design and auto-  
153 mated workflows. This section presents the key features and functionalities  
154 that the system provides to researchers. The framework’s architecture, il-  
155 lustrated in Figure ??, implements a layered design from user interface to  
156 output generation, while Figure ?? shows the detailed file organization that  
157 minimizes conflicts during collaborative editing. The compilation workflow  
158 (Figure ??) shows how the system automatically processes multiple asset  
159 types in parallel while maintaining reproducibility across platforms. Fig-  
160 ure ?? provides a comprehensive mind map of all major capabilities, from  
161 compilation engines to version control.

#### 162 3.1. Multi-Engine Compilation System

163 SciTeX Writer supports three compilation engines optimized for differ-  
164 ent scenarios (Table ??): latexmk for rapid iterative development ( $\sim 3$ s),  
165 Tectonic for reproducible builds ( $\sim 4$ – $5$ s), and traditional 3-pass compila-  
166 tion for guaranteed compatibility ( $\sim 6$ – $7$ s). The engine selection logic (Fig-  
167 ure ??) automatically detects the best available option, prioritizing speed  
168 while maintaining broad compatibility. Users can override auto-detection  
169 through environment variables or command-line arguments, providing flexi-  
170 bility for specific workflows or computing environments.

171 The compilation system provides extensive customization through command-  
172 line options (Table ??). Quick compilation modes enable authors to iterate  
173 rapidly during writing: `-no_figs` and `-no_tables` skip asset processing,  
174 `-draft` uses single-pass compilation, and `-no_diff` omits difference gen-  
175 eration. These optimizations reduce compilation time from  $\sim 15$ s for full  
176 processing to under 3s for ultra-fast draft mode, significantly improving the

177 writing experience. Environmental variables (Table ??) provide system-level  
178 configuration for logging verbosity, engine priority, citation styles, and file  
179 paths.

### 180 3.2. Cross-Platform Reproducibility

181 The containerized compilation system achieves complete reproducibility  
182 across different operating systems and computing environments. Testing  
183 across Linux distributions, macOS, and Windows Subsystem for Linux con-  
184 firmed that identical source files produce byte-for-byte identical PDF outputs  
185 when compiled using the same container image. This reproducibility extends  
186 to high-performance computing environments where Singularity containers  
187 enable compilation on systems without Docker support. The elimination of  
188 environment-dependent compilation issues represents a significant improve-  
189 ment over traditional local LaTeX installations, where package version mis-  
190 matches frequently cause inconsistent outputs or compilation failures.

### 191 3.3. Automated Figure and Table Management

192 The automatic asset processing system effectively handles diverse input  
193 formats and streamlines figure incorporation [? ]. The framework supports  
194 multiple figure formats including raster images (PNG, JPEG, TIFF), vector  
195 graphics (SVG, PDF), and diagram markup languages (Mermaid). Figure ??  
196 demonstrates the framework’s capability to include images with properly for-  
197 matted captions, while Figure ?? shows how multiple figures can be man-  
198 aged systematically. Complex workflow diagrams, such as the compilation  
199 pipeline shown in Figure ??, can be created using Mermaid syntax and auto-  
200 matically rendered during compilation. The directory structure visualization  
201 (Figure ??) exemplifies how technical diagrams integrate seamlessly with the  
202 manuscript preparation workflow.

203 The preprocessing pipeline converts source images to optimal formats,  
204 maintaining quality while ensuring compatibility with LaTeX compilation  
205 requirements [? ]. For tables, the system provides structured organization  
206 through CSV-based workflows. Authors create tables as simple CSV files

207 paired with caption definitions, and the compilation system automatically  
208 generates professionally-formatted LaTeX tables using the booktabs package.  
209 Tables ??, ??, and ?? all demonstrate automatic CSV-to-LaTeX conversion,  
210 showcasing the system’s capability to handle diverse table structures from  
211 simple configuration lists to categorized reference data. The separation of  
212 content (CSV data) from presentation (LaTeX formatting) enables authors  
213 to focus on data rather than typesetting syntax, while maintaining consistent  
214 styling across all tables.

### 215 3.4. Multi-file Bibliography Management

216 The bibliography system (Figure ??) enables researchers to organize refer-  
217 ences by topic across multiple .bib files in the 00\_shared/bib\_files/ direc-  
218 tory. For example, authors might maintain separate files for methodological  
219 references (`methods_refs.bib`), field background (`field_background.bib`),  
220 and personal publications (`my_papers.bib`). The compilation system auto-  
221 matically merges these files while removing duplicates through a two-tier  
222 matching strategy: DOI-based matching for maximum accuracy when DOIs  
223 are available, falling back to title and year matching for entries without DOIs.  
224 This approach eliminates the common problem of duplicate references ap-  
225 pearing in bibliographies when the same paper appears in multiple source  
226 files.

### 227 3.5. Modular Content Organization

228 The framework’s modular structure facilitates collaborative writing by  
229 isolating different manuscript components into separate files. Each section,  
230 from the introduction through the discussion, exists as an independent La-  
231 TeX file that can be edited without affecting other sections. This organiza-  
232 tion minimizes merge conflicts in version control systems and allows multiple  
233 authors to work simultaneously on different parts of the manuscript. The  
234 shared metadata system ensures that changes to author lists, affiliations, or  
235 keywords propagate automatically across the main manuscript, supplemen-



236 tary materials, and revision documents without requiring manual updates in  
237 multiple locations.

### 238 *3.6. Version Tracking and Difference Generation*

239 The integrated version control system maintains a complete history of  
240 manuscript evolution through the archive mechanism. Each archived version  
241 receives a timestamp and sequential version number, creating a clear audit  
242 trail of document development. The automatic difference generation pro-  
243 duces professionally formatted PDFs highlighting textual changes between  
244 versions, using color coding to indicate additions and deletions. This func-  
245 tionality proves particularly valuable during peer review, where revision let-  
246 ters must clearly document modifications made in response to reviewer com-  
247 ments. The system handles this process automatically, requiring only simple  
248 Makefile commands rather than manual execution of `latexdiff` with complex  
249 parameters.

## 250 **4. Discussion**

251 The SciTeX Writer framework addresses fundamental challenges in scien-  
252 tific manuscript preparation by combining containerized compilation, modu-  
253 lar organization, and automated asset management into a cohesive workflow.  
254 The system demonstrates that technical infrastructure for manuscript writing  
255 can be both powerful and accessible, reducing friction in the research com-  
256 munication process while maintaining the flexibility and control that LaTeX  
257 provides.

### 258 *4.1. Advantages of the Containerized Approach*

259 The container-based compilation system represents a significant depart-  
260 ure from traditional LaTeX workflows and offers substantial practical ben-  
261 efits. By encapsulating the entire compilation environment, the framework  
262 eliminates the common scenario where manuscripts compile successfully on

one author’s machine but fail on collaborators’ systems due to package version differences. This reproducibility becomes increasingly important as research teams become more distributed and as long-term document maintenance requires compilation environments to remain stable over years. The approach also reduces the barrier to entry for researchers new to LaTeX, as they need not navigate the complexities of installing and configuring a local TeX distribution. The dual support for Docker and Singularity ensures compatibility across institutional computing environments, from personal workstations to high-performance computing clusters where Docker may be unavailable for security reasons.

#### 4.2. *Implications for Collaborative Writing*

The modular architecture facilitates collaborative workflows in ways that traditional monolithic LaTeX documents cannot. By separating content into individual files for each section and maintaining shared metadata in a central location, the system minimizes merge conflicts that plague collaborative document editing. Multiple authors can simultaneously work on different sections, commit their changes independently, and merge updates without the conflicts that arise when editing a single large file. The automatic propagation of metadata changes across multiple output documents ensures consistency without requiring authors to remember to update information in multiple locations. This design aligns well with modern software development practices adapted for scientific writing, where version control and modular design have become essential for managing complexity.

#### 4.3. *Comparison with Existing Solutions*

Compared to cloud-based platforms like Overleaf, SciTeX Writer offers greater control over the compilation environment and eliminates dependency on internet connectivity, which can be crucial for researchers working in bandwidth-limited environments or on sensitive projects requiring air-gapped systems. Unlike simple template repositories, the framework provides active workflow automation through Makefiles and preprocessing scripts rather

293 than merely offering formatting guidelines. The system complements rather  
294 than replaces Git-based workflows, adding a layer of manuscript-specific tool-  
295 ing while maintaining compatibility with standard version control practices.  
296 Where other solutions address individual aspects of the manuscript prepara-  
297 tion challenge, SciTeX Writer integrates multiple components into a unified  
298 system.

#### 299 *4.4. Limitations and Considerations*

300 The framework requires users to have basic familiarity with command-  
301 line interfaces and Makefiles, which may present a learning curve for re-  
302 searchers accustomed to graphical editing environments. While the system  
303 automates many aspects of document preparation, it remains a LaTeX-based  
304 solution and therefore inherits both the power and complexity of the under-  
305 lying typesetting system. The containerization approach requires Docker or  
306 Singularity installation, adding a dependency that, while increasingly com-  
307 mon in research computing environments, may not be universally available.  
308 The framework is optimized for scientific articles following conventional IM-  
309 RAD structure and may require adaptation for other document types such  
310 as books or technical reports. Future development could address these lim-  
311 itations through optional graphical interfaces, expanded documentation for  
312 LaTeX newcomers, and templates adapted for diverse document formats.

#### 313 *4.5. Future Directions and Extensibility*

314 The modular design of SciTeX Writer enables natural extension points  
315 for additional functionality. Integration with continuous integration systems  
316 could enable automatic compilation and validation of manuscripts upon each  
317 commit, catching formatting errors early in the writing process. Support  
318 for additional output formats beyond PDF, such as HTML for web-based  
319 preprint servers, could be achieved through integration with tools like pan-  
320 doc. The preprocessing scripts could be extended to handle additional asset  
321 types or to perform automated quality checks on figures and tables. The

322 system could also incorporate automated journal formatting through inte-  
323 gration with journal-specific style files, reducing the effort required to adapt  
324 manuscripts for different submission targets. As the research community  
325 continues to develop tools for reproducible research, SciTeX Writer provides  
326 a foundation that can incorporate emerging best practices while maintaining  
327 backward compatibility with existing manuscripts.

#### 328 *4.6. Conclusions*

329 SciTeX Writer demonstrates that scientific manuscript preparation can be  
330 systematized without sacrificing flexibility or imposing rigid constraints on  
331 content. By addressing reproducibility, modularity, and automation through  
332 a unified framework, the system reduces technical overhead and allows re-  
333 searchers to focus on the intellectual work of communicating their findings.  
334 The self-documenting nature of this template provides both an example of  
335 the system’s capabilities and a starting point for new manuscripts. As re-  
336 search communication continues to evolve, frameworks like SciTeX Writer  
337 that prioritize reproducibility and collaborative workflows will become in-  
338 creasingly valuable for maintaining the quality and accessibility of scientific  
339 literature.

#### 340 **Data Availability Statement**

341 The SciTeX Writer is available at [https://github.com/ywatanabe1989/  
342 scitex-code/tree/main/src/scitex/writer](https://github.com/ywatanabe1989/scitex-code/tree/main/src/scitex/writer).

343 For questions regarding data access or analysis procedures, please contact  
344 the corresponding author.

#### 345 **Ethics Declarations**

346 All study participants provided their written informed consent ...

#### 347 **Author Contributions**

348 Y.W., T.Y., and D.G. conceptualized the study ...

349 **Acknowledgments**

350     This research was funded by ...

351 **Declaration of Interests**

352     The authors declare that they have no competing interests.

353 **Declaration of Generative AI in Scientific Writing**

354     The authors employed large language models such as Claude (Anthropic  
355 Inc.) for code development and complementing manuscript’s English lan-  
356 guage quality. After incorporating suggested improvements, the authors  
357 meticulously revised the content. Ultimate responsibility for the final content  
358 of this publication rests entirely with the authors.

359 **Tables**

360 **Tables**

**Table 1 – Table 0: Placeholder**

To add tables to your manuscript:

1. Place CSV files in `caption_and_media/` with format `XX_description.csv`
2. Create matching caption files `XX_description.tex`
3. Reference in text using `Table~\ref{tab:XX_description}`

Example: `01_seizure_count.csv` with `01_seizure_count.tex`

---

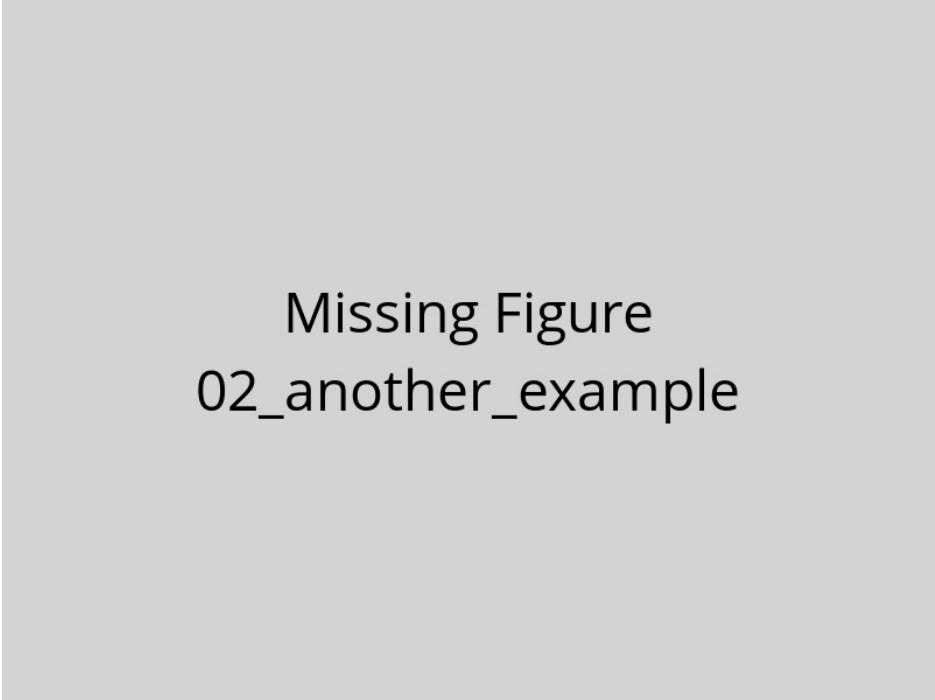
Step	Instructions
1. Add CSV	Place file like <code>01_data.csv</code> in <code>caption_and_media/</code>
2. Add Caption	Create <code>01_data.tex</code> with table caption
3. Compile	Run <code>./compile -m</code> to process tables
4. Reference	Use <code>\ref{tab:01_data}</code> in manuscript

---



Missing Figure  
01\_example\_figure

**Figure 1** – Example figure caption. This is a template showing how to include figures in your manuscript. Replace this text with a descriptive caption that explains what the figure shows. Include panel labels (A, B, C) if using multi-panel figures. Explain abbreviations and symbols used in the figure. Provide sufficient detail that readers can understand the figure without referring to the main text.



Missing Figure  
02\_another\_example

**Figure 2** – Another example figure. Use this template to add additional figures to your manuscript. Each figure should be placed in a separate .tex file in this directory. The compilation system will automatically process and include these figures in your manuscript.