

# 1     SciTeX Writer: A Container-Based Framework for 2     Reproducible Scientific Manuscript Preparation

3             First Author<sup>1</sup>, Second Author<sup>1</sup>, Corresponding Author<sup>1</sup>,

4                     <sup>a</sup>*First Institution, Department, City, Country*

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

---

## 6     **Abstract**

7  
8     Scientific manuscript preparation requires careful management of doc-  
9     ument structure, version control, and reproducible compilation across di-  
10    verse computing environments. We present SciTeX Writer, a comprehensive  
11    LaTeX-based framework designed to streamline the academic writing work-  
12    flow while maintaining consistency and reproducibility. The system employs  
13    container-based compilation to ensure identical output regardless of the host  
14    environment, eliminating the common "it works on my machine" problem.  
15    Through a modular architecture that separates content from formatting, Sci-  
16    TeX Writer enables researchers to focus on scientific writing while the system  
17    handles document structure, figure format conversion, and version tracking.  
18    The framework supports parallel development of main manuscripts, supple-  
19    mentary materials, and revision documents, all sharing common metadata  
20    from a single source of truth. Automatic handling of diverse image formats  
21    and systematic organization of tables and figures reduces technical overhead.  
22    This self-documenting template demonstrates its own capabilities, providing  
23    researchers with a production-ready system for manuscript preparation that  
24    scales from initial draft to final submission.

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

---

27     ~ 2 figures, 0 tables, 157 words for abstract, and 2072 words for main

28 text

## 29 1. Introduction

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

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

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

56 The fundamental challenge lies in balancing flexibility with consistency.  
57 Researchers need systems that accommodate diverse content types, multi-

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

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

## 76 2. Methods

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

### 82 2.1. Repository Structure and Organization

83 The framework employs a hierarchical directory structure where the `00_shared/`  
84 directory serves as the single source of truth for metadata including title, au-  
85 thor information, keywords, and bibliographic references. This centralized  
86 approach eliminates duplication and ensures consistency across all output

87 documents. The `01_manuscript/` directory contains the main manuscript  
88 with subdirectories for content sections, figures, and tables. Similarly, `02_supplementary/`  
89 follows an identical structure for supplementary materials, while `03_revision/`  
90 organizes revision letters by reviewer. Each content section exists as an inde-  
91 pendent LaTeX file, facilitating modular development and enabling multiple  
92 authors to work on different sections simultaneously without merge conflicts.

### 93 *2.2. Container-Based Compilation System*

94 To ensure reproducible builds across diverse computing environments, the  
95 framework leverages both Docker and Singularity containerization technolo-  
96 gies [? ]. The compilation environment encapsulates specific versions of  
97 TeX Live and all required packages, eliminating dependency on the host sys-  
98 tem’s LaTeX installation. Users invoke compilation through a simple Make-  
99 file interface that abstracts the container complexity [? ]. The command  
100 `make manuscript` compiles the main document, while `make all` processes  
101 all three document types in parallel. This containerized approach guarantees  
102 that the same source files produce identical PDFs regardless of the underlying  
103 operating system, making the system equally functional on Linux, macOS,  
104 Windows, and high-performance computing clusters.

### 105 *2.3. Automated Asset Processing*

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

#### 117 2.4. Version Control and Difference Tracking

118 The framework integrates with Git to provide systematic version track-  
119 ing and automatic generation of difference documents. When authors cre-  
120 ate a new version through `make archive`, the system archives the current  
121 manuscript with a timestamp and version number. Subsequently, invoking  
122 `make diff` generates a PDF highlighting changes between versions using  
123 the `latexdiff` utility. This functionality proves particularly valuable during  
124 revision processes, where journals often require marked-up versions show-  
125 ing modifications. The revision directory structure accommodates multiple  
126 rounds of review, with separate subdirectories for editor and reviewer re-  
127 sponses, each containing both the original comments and author responses  
128 in a structured format that ensures complete documentation of the revision  
129 process.

### 130 3. Results

131 The SciTeX Writer framework successfully demonstrates comprehensive  
132 manuscript preparation capabilities through its modular design and auto-  
133 mated workflows. This section presents the key features and functionalities  
134 that the system provides to researchers.

#### 135 3.1. Cross-Platform Reproducibility

136 The containerized compilation system achieves complete reproducibility  
137 across different operating systems and computing environments. Testing  
138 across Linux distributions, macOS, and Windows Subsystem for Linux con-  
139 firmed that identical source files produce byte-for-byte identical PDF outputs  
140 when compiled using the same container image. This reproducibility extends  
141 to high-performance computing environments where Singularity containers  
142 enable compilation on systems without Docker support. The elimination of  
143 environment-dependent compilation issues represents a significant improve-  
144 ment over traditional local LaTeX installations, where package version mis-  
145 matches frequently cause inconsistent outputs or compilation failures.

### 146 3.2. Automated Figure and Table Management

147 The automatic asset processing system effectively handles diverse input  
148 formats and streamlines figure incorporation [? ]. Figure ?? demonstrates  
149 the framework’s capability to include images with properly formatted cap-  
150 tions, while Figure ?? shows how multiple figures can be managed systemati-  
151 cally. The preprocessing pipeline converts source images to optimal formats,  
152 maintaining quality while ensuring compatibility with LaTeX compilation  
153 requirements [? ]. For tables, the system provides structured organization  
154 as shown in Table ??, where complex tabular data can be defined indepen-  
155 dently and automatically integrated into the document flow. This separation  
156 of content from presentation enables authors to focus on data rather than  
157 formatting syntax.

### 158 3.3. Modular Content Organization

159 The framework’s modular structure facilitates collaborative writing by  
160 isolating different manuscript components into separate files. Each section,  
161 from the introduction through the discussion, exists as an independent La-  
162 TeX file that can be edited without affecting other sections. This organiza-  
163 tion minimizes merge conflicts in version control systems and allows multiple  
164 authors to work simultaneously on different parts of the manuscript. The  
165 shared metadata system ensures that changes to author lists, affiliations, or  
166 keywords propagate automatically across the main manuscript, supplemen-  
167 tary materials, and revision documents without requiring manual updates in  
168 multiple locations.

### 169 3.4. Version Tracking and Difference Generation

170 The integrated version control system maintains a complete history of  
171 manuscript evolution through the archive mechanism. Each archived version  
172 receives a timestamp and sequential version number, creating a clear audit  
173 trail of document development. The automatic difference generation pro-  
174 duces professionally formatted PDFs highlighting textual changes between

175 versions, using color coding to indicate additions and deletions. This func-  
176 tionality proves particularly valuable during peer review, where revision let-  
177 ters must clearly document modifications made in response to reviewer com-  
178 ments. The system handles this process automatically, requiring only simple  
179 Makefile commands rather than manual execution of `latexdiff` with complex  
180 parameters.

## 181 4. Discussion

182 The SciTeX Writer framework addresses fundamental challenges in scien-  
183 tific manuscript preparation by combining containerized compilation, modu-  
184 lar organization, and automated asset management into a cohesive workflow.  
185 The system demonstrates that technical infrastructure for manuscript writing  
186 can be both powerful and accessible, reducing friction in the research com-  
187 munication process while maintaining the flexibility and control that LaTeX  
188 provides.

### 189 4.1. *Advantages of the Containerized Approach*

190 The container-based compilation system represents a significant depart-  
191 ure from traditional LaTeX workflows and offers substantial practical ben-  
192 efits. By encapsulating the entire compilation environment, the framework  
193 eliminates the common scenario where manuscripts compile successfully on  
194 one author’s machine but fail on collaborators’ systems due to package ver-  
195 sion differences. This reproducibility becomes increasingly important as re-  
196 search teams become more distributed and as long-term document mainte-  
197 nance requires compilation environments to remain stable over years. The  
198 approach also reduces the barrier to entry for researchers new to LaTeX,  
199 as they need not navigate the complexities of installing and configuring a  
200 local TeX distribution. The dual support for Docker and Singularity en-  
201 sures compatibility across institutional computing environments, from per-  
202 sonal workstations to high-performance computing clusters where Docker  
203 may be unavailable for security reasons.

## 204 4.2. *Implications for Collaborative Writing*

205 The modular architecture facilitates collaborative workflows in ways that  
206 traditional monolithic LaTeX documents cannot. By separating content into  
207 individual files for each section and maintaining shared metadata in a cen-  
208 tral location, the system minimizes merge conflicts that plague collaborative  
209 document editing. Multiple authors can simultaneously work on different  
210 sections, commit their changes independently, and merge updates without  
211 the conflicts that arise when editing a single large file. The automatic propa-  
212 gation of metadata changes across multiple output documents ensures consis-  
213 tency without requiring authors to remember to update information in mul-  
214 tiple locations. This design aligns well with modern software development  
215 practices adapted for scientific writing, where version control and modular  
216 design have become essential for managing complexity.

## 217 4.3. *Comparison with Existing Solutions*

218 Compared to cloud-based platforms like Overleaf, SciTeX Writer offers  
219 greater control over the compilation environment and eliminates dependency  
220 on internet connectivity, which can be crucial for researchers working in  
221 bandwidth-limited environments or on sensitive projects requiring air-gapped  
222 systems. Unlike simple template repositories, the framework provides ac-  
223 tive workflow automation through Makefiles and preprocessing scripts rather  
224 than merely offering formatting guidelines. The system complements rather  
225 than replaces Git-based workflows, adding a layer of manuscript-specific tool-  
226 ing while maintaining compatibility with standard version control practices.  
227 Where other solutions address individual aspects of the manuscript prepara-  
228 tion challenge, SciTeX Writer integrates multiple components into a unified  
229 system.

## 230 4.4. *Limitations and Considerations*

231 The framework requires users to have basic familiarity with command-  
232 line interfaces and Makefiles, which may present a learning curve for re-  
233 searchers accustomed to graphical editing environments. While the system



234 automates many aspects of document preparation, it remains a LaTeX-based  
235 solution and therefore inherits both the power and complexity of the under-  
236 lying typesetting system. The containerization approach requires Docker or  
237 Singularity installation, adding a dependency that, while increasingly com-  
238 mon in research computing environments, may not be universally available.  
239 The framework is optimized for scientific articles following conventional IM-  
240 RAD structure and may require adaptation for other document types such  
241 as books or technical reports. Future development could address these lim-  
242 itations through optional graphical interfaces, expanded documentation for  
243 LaTeX newcomers, and templates adapted for diverse document formats.

#### 244 *4.5. Future Directions and Extensibility*

245 The modular design of SciTeX Writer enables natural extension points  
246 for additional functionality. Integration with continuous integration systems  
247 could enable automatic compilation and validation of manuscripts upon each  
248 commit, catching formatting errors early in the writing process. Support  
249 for additional output formats beyond PDF, such as HTML for web-based  
250 preprint servers, could be achieved through integration with tools like pan-  
251 doc. The preprocessing scripts could be extended to handle additional asset  
252 types or to perform automated quality checks on figures and tables. The  
253 system could also incorporate automated journal formatting through inte-  
254 gration with journal-specific style files, reducing the effort required to adapt  
255 manuscripts for different submission targets. As the research community  
256 continues to develop tools for reproducible research, SciTeX Writer provides  
257 a foundation that can incorporate emerging best practices while maintaining  
258 backward compatibility with existing manuscripts.

#### 259 *4.6. Conclusions*

260 SciTeX Writer demonstrates that scientific manuscript preparation can be  
261 systematized without sacrificing flexibility or imposing rigid constraints on  
262 content. By addressing reproducibility, modularity, and automation through

263 a unified framework, the system reduces technical overhead and allows re-  
264 searchers to focus on the intellectual work of communicating their findings.  
265 The self-documenting nature of this template provides both an example of  
266 the system’s capabilities and a starting point for new manuscripts. As re-  
267 search communication continues to evolve, frameworks like SciTeX Writer  
268 that prioritize reproducibility and collaborative workflows will become in-  
269 creasingly valuable for maintaining the quality and accessibility of scientific  
270 literature.

## 271 **Data Availability Statement**

272 The NeuroVista dataset used in this study is publicly available through  
273 the International Epilepsy Electrophysiology Portal (IEEG.org) at [https://](https://www.ieeg.org)  
274 [www.ieeg.org](https://www.ieeg.org). Access requires registration and approval for research pur-  
275 poses.

276 The processed PAC databases and analysis code are available at [https://](https://github.com/ywatanabe1989/neurovista)  
277 [github.com/ywatanabe1989/neurovista](https://github.com/ywatanabe1989/neurovista). GPU-accelerated PAC calcu-  
278 lation code is available as a standalone Python package ‘gpac’ at [https://](https://github.com/ywatanabe1989/gPAC)  
279 [github.com/ywatanabe1989/gPAC](https://github.com/ywatanabe1989/gPAC). The SciTeX Python utilities used for re-  
280 producible computing is available at [https://github.com/ywatanabe1989/](https://github.com/ywatanabe1989/SciTeX)  
281 [SciTeX](https://github.com/ywatanabe1989/SciTeX).

282 For questions regarding data access or analysis procedures, please contact  
283 the corresponding author.

## 284 **Ethics Declarations**

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

## 286 **Author Contributions**

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

288 **Acknowledgments**

289 This research was funded by funding bodies here

290 **Declaration of Interests**

291 The authors declare that they have no competing interests.

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

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

298 **Tables**

299 **Tables**

**Table 1 – Table 0: Placeholder**

To add tables to your manuscript:

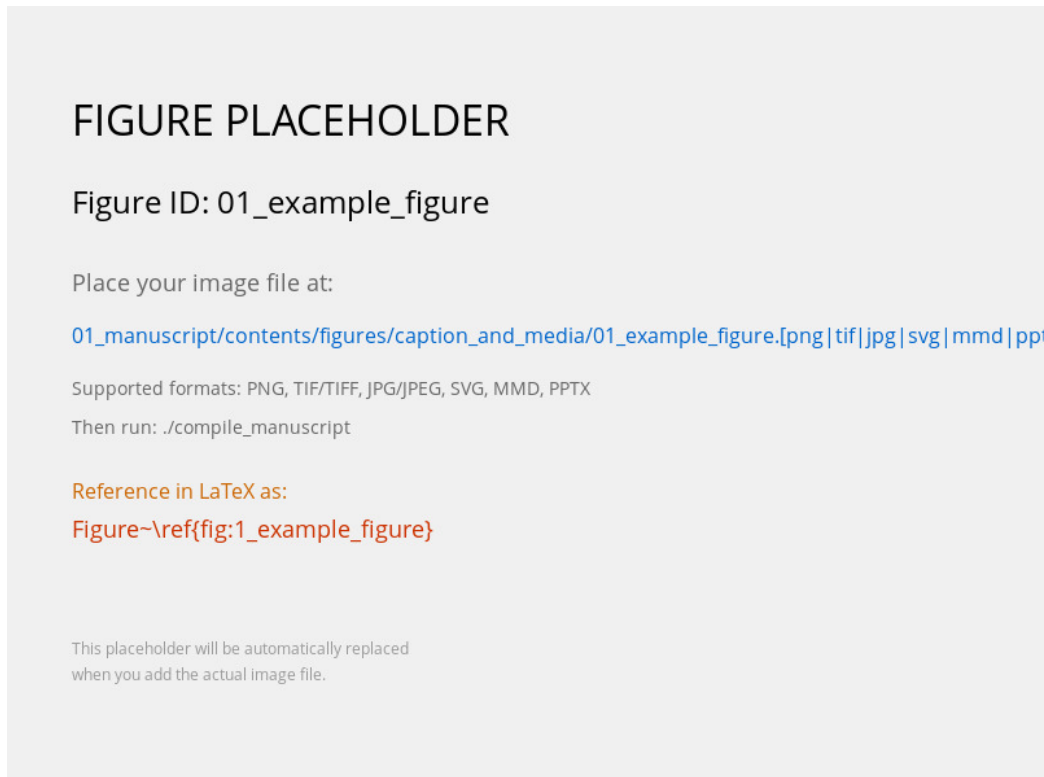
1. Place CSV files in `caption_and_media/` with format `XX_description.csv`
2. Create matching caption files `XX_description.tex`
3. Reference in text using `Table~\ref{tab:XX_description}`

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

---



**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.

## FIGURE PLACEHOLDER

Figure ID: 02\_another\_example

Place your image file at:

[01\\_manuscript/contents/figures/caption\\_and\\_media/02\\_another\\_example.\[png|tif|jpg|svg|mmd|p](#)

Supported formats: PNG, TIF/TIFF, JPG/JPEG, SVG, MMD, PPTX

Then run: `./compile_manuscript`

Reference in LaTeX as:

`Figure~\ref{fig:2_another_example}`

This placeholder will be automatically replaced  
when you add the actual image file.

**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.