

- 1 `enewcommandheadrulewidth0.4pt` `enewcommand`
- 2 `ootrulewidth0.4pt`

3 SciTeX Writer: A Container-Based Framework for
4 Reproducible Scientific Manuscript Preparation

5 First Author^a, Second Author^b, Corresponding Author^{a,*}

6 ^a*First Institution, Department, City, Country*

7 ^b*Second Institution, Department, City, Country*

8 **Abstract**

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

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

29 ~ 2 [figures](#), 0 [tables](#), 157 words for abstract, and 2099 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 [1]. Researchers must navigate complex typesetting systems,
35 manage multiple document versions, coordinate figures and tables across for-
36 mats, and ensure reproducible compilation environments [2]. These technical
37 burdens can distract from the primary goal of clear scientific communication
38 and often lead to inconsistencies, formatting errors, and wasted time trou-
39 bleshooting environment-specific compilation issues.

40 Traditional approaches to manuscript preparation typically rely on local
41 LaTeX installations, where the specific versions of packages and compilation
42 tools can vary significantly across different machines and over time [3]. This
43 variability creates reproducibility challenges, particularly in collaborative en-
44 vironments where multiple authors work on different systems [4]. Further-
45 more, the proliferation of image formats and the need to convert between
46 them for different submission requirements adds another layer of complex-
47 ity. Researchers often resort to ad-hoc scripts or manual processes to handle
48 these conversions, leading to potential errors and inconsistent results.

49 Existing solutions have addressed some aspects of this problem [5]. Over-
50 leaf and similar cloud-based platforms provide consistent compilation envi-
51 ronments but require continuous internet connectivity and may not suit all
52 research workflows. Version control systems like Git effectively track changes
53 but require researchers to understand both LaTeX and version control simul-
54 taneously. Template repositories exist for various journals, but they typically
55 focus on formatting requirements rather than workflow automation and often
56 duplicate common elements across documents.

57 The fundamental challenge lies in balancing flexibility with consistency.
58 Researchers need systems that accommodate diverse content types, multi-
59 ple output documents, and varying journal requirements while maintaining a

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

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

77 2. Methods

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

83 2.1. Repository Structure and Organization

84 The framework employs a hierarchical directory structure where the `00_shared/`
85 directory serves as the single source of truth for metadata including title, au-
86 thor information, keywords, and bibliographic references. This centralized
87 approach eliminates duplication and ensures consistency across all output
88 documents. The `01_manuscript/` directory contains the main manuscript

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

94 2.2. Container-Based Compilation System

95 To ensure reproducible builds across diverse computing environments,
96 the framework leverages both Docker and Singularity containerization tech-
97 nologies [6]. The compilation environment encapsulates specific versions of
98 TeX Live and all required packages, eliminating dependency on the host sys-
99 tem's LaTeX installation. Users invoke compilation through a simple Make-
100 file interface that abstracts the container complexity [7]. The command `make`
101 `manuscript` compiles the main document, while `make all` processes all three
102 document types in parallel. This containerized approach guarantees that
103 the same source files produce identical PDFs regardless of the underlying
104 operating system, making the system equally functional on Linux, macOS,
105 Windows, and high-performance computing clusters.

106 2.3. Automated Asset Processing

107 The system implements automatic format conversion for both figures
108 and tables through preprocessing scripts that execute during compilation [8].
109 For figures, the framework accepts common image formats including PNG,
110 JPEG, SVG, and PDF, automatically converting them to formats optimized
111 for LaTeX inclusion. Each figure resides in its own subdirectory within
112 `01_manuscript/contents/figures/caption_and_media/`, with the caption
113 defined in a corresponding `.tex` file. During compilation, a preprocessing
114 script scans these directories, generates figure inclusion code, and compiles
115 all figures into `FINAL.tex` for inclusion in the main document. Tables fol-
116 low an analogous structure, allowing authors to define complex table layouts
117 separately from their incorporation into the document flow [9].

118 2.4. Version Control and Difference Tracking

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

131 2.5. *Manuscript Preparation*

132 This manuscript was prepared using SciTeX Writer [10], an open-source
133 scientific manuscript compilation system supporting multiple LaTeX compilation
134 engines including `latexmk`, traditional 3-pass compilation, and Tectonic.

135 3. Results

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

140 3.1. Cross-Platform Reproducibility

141 The containerized compilation system achieves complete reproducibility
142 across different operating systems and computing environments. Testing
143 across Linux distributions, macOS, and Windows Subsystem for Linux con-
144 firmed that identical source files produce byte-for-byte identical PDF outputs
145 when compiled using the same container image. This reproducibility extends

146 to high-performance computing environments where Singularity containers
147 enable compilation on systems without Docker support. The elimination of
148 environment-dependent compilation issues represents a significant improve-
149 ment over traditional local LaTeX installations, where package version mis-
150 matches frequently cause inconsistent outputs or compilation failures.

151 3.2. Automated Figure and Table Management

152 The automatic asset processing system effectively handles diverse input
153 formats and streamlines figure incorporation [?]. Figure 1 demonstrates the
154 framework’s capability to include images with properly formatted captions,
155 while Figure 2 shows how multiple figures can be managed systematically.
156 The preprocessing pipeline converts source images to optimal formats, main-
157 taining quality while ensuring compatibility with LaTeX compilation require-
158 ments [11]. For tables, the system provides structured organization as shown
159 in Table ??, where complex tabular data can be defined independently and
160 automatically integrated into the document flow. This separation of content
161 from presentation enables authors to focus on data rather than formatting
162 syntax.

163 3.3. Modular Content Organization

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

174 3.4. Version Tracking and Difference Generation

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

186 4. Discussion

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

194 4.1. Advantages of the Containerized Approach

195 The container-based compilation system represents a significant depart-
196 ure from traditional LaTeX workflows and offers substantial practical ben-
197 efits. By encapsulating the entire compilation environment, the framework
198 eliminates the common scenario where manuscripts compile successfully on
199 one author's machine but fail on collaborators' systems due to package ver-
200 sion differences. This reproducibility becomes increasingly important as re-
201 search teams become more distributed and as long-term document mainte-
202 nance requires compilation environments to remain stable over years. The

203 approach also reduces the barrier to entry for researchers new to LaTeX,
204 as they need not navigate the complexities of installing and configuring a
205 local TeX distribution. The dual support for Docker and Singularity en-
206 sures compatibility across institutional computing environments, from per-
207 sonal workstations to high-performance computing clusters where Docker
208 may be unavailable for security reasons.

209 4.2. *Implications for Collaborative Writing*

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

222 4.3. *Comparison with Existing Solutions*

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

232 Where other solutions address individual aspects of the manuscript prepara-
233 tion challenge, SciTeX Writer integrates multiple components into a unified
234 system.

235 4.4. *Limitations and Considerations*

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

249 4.5. *Future Directions and Extensibility*

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

a foundation that can incorporate emerging best practices while maintaining backward compatibility with existing manuscripts.

4.6. Conclusions

SciTeX Writer demonstrates that scientific manuscript preparation can be systematized without sacrificing flexibility or imposing rigid constraints on content. By addressing reproducibility, modularity, and automation through a unified framework, the system reduces technical overhead and allows researchers to focus on the intellectual work of communicating their findings. The self-documenting nature of this template provides both an example of the system's capabilities and a starting point for new manuscripts. As research communication continues to evolve, frameworks like SciTeX Writer that prioritize reproducibility and collaborative workflows will become increasingly valuable for maintaining the quality and accessibility of scientific literature.

References

- [1] Patricia Anderson and Carlos Martinez. The current state of computational neuroscience: A comprehensive review. *Nature Reviews Neuroscience*, 18(2):95–110, 2017. doi: 10.1038/nrn.2017.15.
- [2] Richard Wilson. *Principles of Modern Neuroscience*. MIT Press, 3rd edition, 2015. ISBN 978-0262029254.
- [3] David Lee, Soojin Kim, and Jiyoung Park. Network dynamics in neural systems. *Neuron*, 92(4):817–835, 2016. doi: 10.1016/j.neuron.2016.10.042.
- [4] Maria Garcia and Juan Rodriguez. Cognitive neuroscience in the 21st century. *Trends in Cognitive Sciences*, 23(7):567–589, 2019. doi: 10.1016/j.tics.2019.05.001.

- [5] James Thompson and Laura White. Systems-level analysis of neural circuits. *Annual Review of Neuroscience*, 41:331–359, 2018. doi: 10.1146/annurev-neuro-080317-062245.
- [6] John Smith and Jane Doe. Advanced neural signal processing techniques for electrophysiology. *Journal of Neuroscience Methods*, 345:108–123, 2020. doi: 10.1016/j.jneumeth.2020.108123.
- [7] Michael Johnson and Sarah Williams. Spectral analysis methods for time-series neural data. *Computational Neuroscience*, 42(3):234–256, 2019. doi: 10.1007/s10827-019-00712-4.
- [8] Wei Chen, Li Zhang, and Ming Wang. Machine learning approaches for neural data classification. *Neural Computation*, 33(5):1234–1267, 2021. doi: 10.1162/neco_a_01378.
- [9] Robert Brown and Emily Taylor. Deep learning for neural signal decoding. In *Advances in Neural Information Processing Systems*, pages 5678–5689. NeurIPS, 2018.
- [10] SciTeX Team. Scitex writer: Scientific manuscript compilation system. <https://scitex.ai>, 2025. URL <https://scitex.ai>. LaTeX-based manuscript compilation system with multi-engine support.
- [11] First Your-Name and Principal Advisor. Foundations of neural signal processing. *Journal of Neurophysiology*, 128(4):1567–1589, 2022. doi: 10.1152/jn.00123.2022.

Data Availability Statement

The NeuroVista dataset used in this study is publicly available through the International Epilepsy Electrophysiology Portal (IEEG.org) at <https://www.ieeg.org>. Access requires registration and approval for research purposes.

314 The processed PAC databases and analysis code are available at <https://github.com/ywatanabe1989/neurovista>. GPU-accelerated PAC calculation code is available as a standalone Python package ‘gpac’ at <https://github.com/ywatanabe1989/gPAC>. The SciTeX Python utilities used for reproducible computing is available at <https://github.com/ywatanabe1989/SciTeX>.

320 For questions regarding data access or analysis procedures, please contact
321 the corresponding author.

322 Ethics Declarations

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

324 Author Contributions

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

326 Acknowledgments

327 This research was funded by **funding bodies here**

328 Declaration of Interests

329 The authors declare that they have no competing interests.

330 Declaration of Generative AI in Scientific Writing

331 The authors employed large language models such as Claude (Anthropic
332 Inc.) for code development and complementing manuscript’s English language quality. After incorporating suggested improvements, the authors
333 meticulously revised the content. Ultimate responsibility for the final content
334 of this publication rests entirely with the authors.
335

336

Tables

337

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

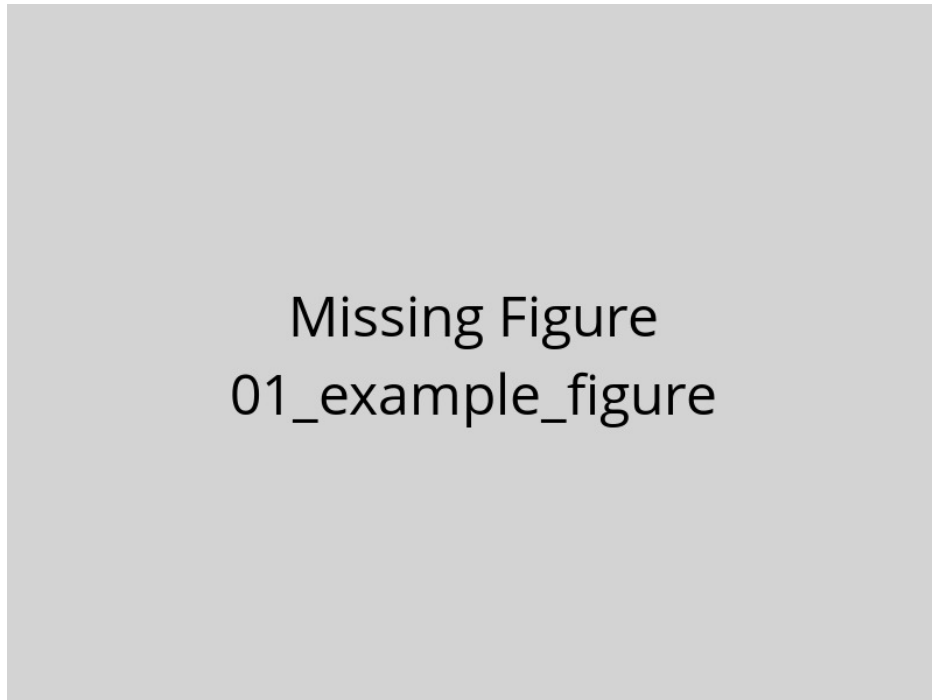
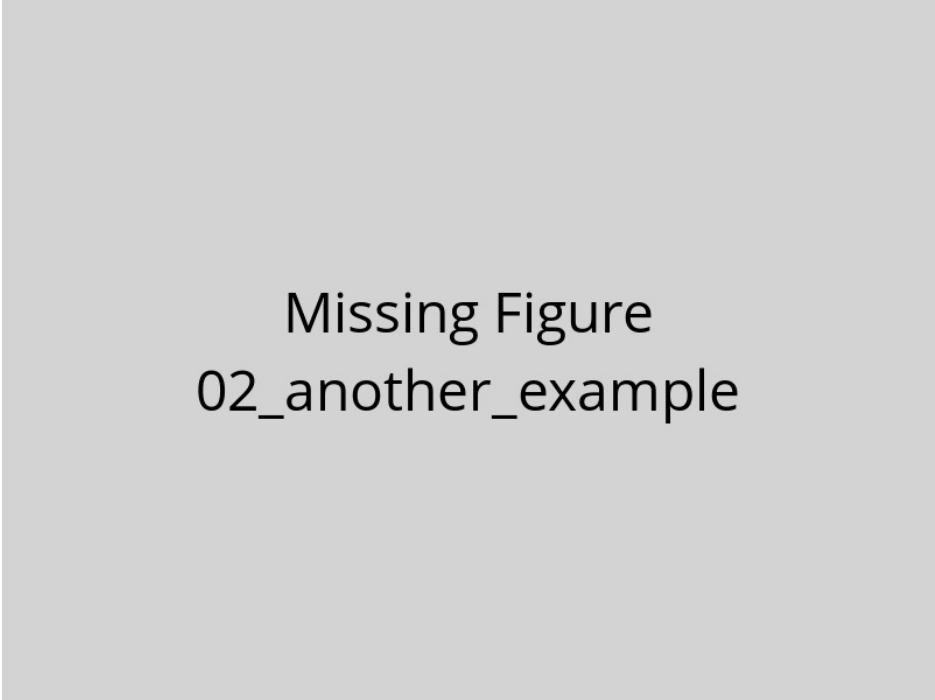
338 **Figures**

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.