1    enewcommandheadrulewidth0.4pt enewcommand

2    ootrulewidth0.4pt

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

First Author[a], Second Author[b], Corresponding Author[a,*]

[a]*First Institution, Department, City, Country*
[b]*Second 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

˘ 2 figures, 0 tables, 157 words for abstract, and 2099 words for main

30 text

## 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

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

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

## 2. Methods

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

### 2.1. Repository Structure and Organization

The framework employs a hierarchical directory structure where the `00_shared/` directory serves as the single source of truth for metadata including title, author 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

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].

*2.4. Version Control and Difference Tracking*

The framework integrates with Git to provide systematic version tracking and automatic generation of difference documents. When authors create a new version through `make archive`, the system archives the current manuscript with a timestamp and version number. Subsequently, invoking `make diff` generates a PDF highlighting changes between versions using the latexdiff utility. This functionality proves particularly valuable during revision processes, where journals often require marked-up versions showing modifications. The revision directory structure accommodates multiple rounds of review, with separate subdirectories for editor and reviewer responses, each containing both the original comments and author responses in a structured format that ensures complete documentation of the revision process.

*2.5. Manuscript Preparation*

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

## 3. Results

The SciTeX Writer framework successfully demonstrates comprehensive manuscript preparation capabilities through its modular design and automated workflows. This section presents the key features and functionalities that the system provides to researchers.

*3.1. Cross-Platform Reproducibility*

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

to high-performance computing environments where Singularity containers enable compilation on systems without Docker support. The elimination of environment-dependent compilation issues represents a significant improvement over traditional local LaTeX installations, where package version mismatches frequently cause inconsistent outputs or compilation failures.

## 3.2. Automated Figure and Table Management

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

## 3.3. Modular Content Organization

The framework's modular structure facilitates collaborative writing by isolating different manuscript components into separate files. Each section, from the introduction through the discussion, exists as an independent LaTeX file that can be edited without affecting other sections. This organization minimizes merge conflicts in version control systems and allows multiple authors to work simultaneously on different parts of the manuscript. The shared metadata system ensures that changes to author lists, affiliations, or keywords propagate automatically across the main manuscript, supplementary materials, and revision documents without requiring manual updates in 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 depar-
196 ture 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

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 than merely offering formatting guidelines. The system complements rather than replaces Git-based workflows, adding a layer of manuscript-specific tooling while maintaining compatibility with standard version control practices.

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

<sup>235</sup> *4.4. Limitations and Considerations*

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

<sup>249</sup> *4.5. Future Directions and Extensibility*

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

<sub>314</sub>      The processed PAC databases and analysis code are available at `https:`
<sub>315</sub> `//github.com/ywatanabe1989/neurovista`. GPU-accerelated PAC calcu-
<sub>316</sub> lation code is available as a standalone Python package 'gpac' at `https://`
<sub>317</sub> `github.com/ywatanabe1989/gPAC`. The SciTeX Python utilities used for re-
<sub>318</sub> producible computing is available at `https://github.com/ywatanabe1989/`
<sub>319</sub> `SciTeX`.
<sub>320</sub>      For questions regarding data access or analysis procedures, please contact
<sub>321</sub> the corresponding author.

## Ethics Declarations

<sub>323</sub>      All study participants provided their written informed consent ...

## Author Contributions

<sub>325</sub>      Y.W., T.Y., and D.G. conceptualized the study ...

## Acknowledgments

## Declaration of Interests

<sub>329</sub>      The authors declare that they have no competing interests.

## Declaration of Generative AI in Scientific Writing

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

**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`

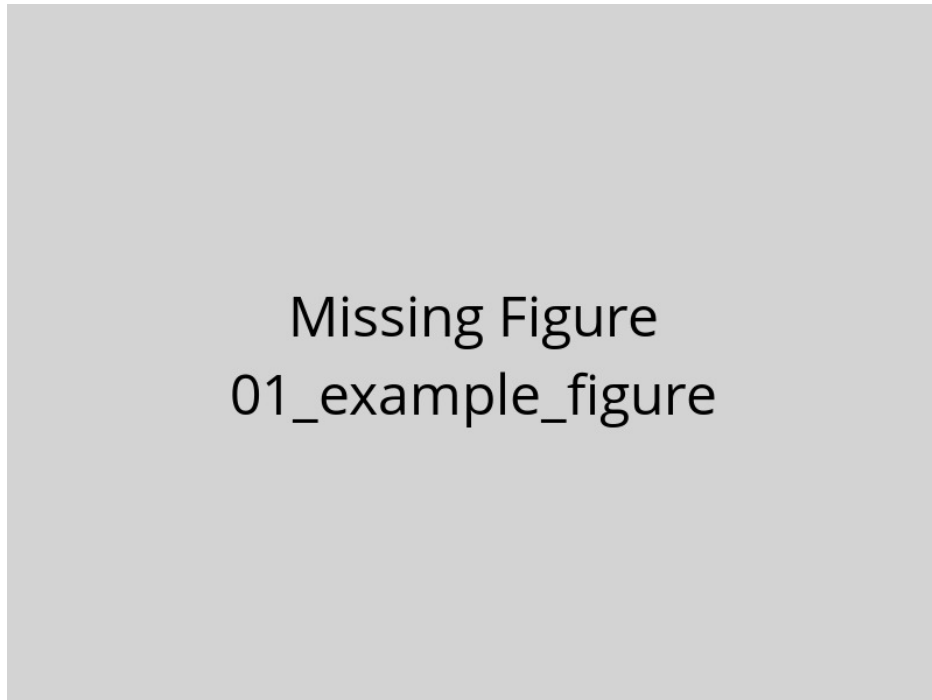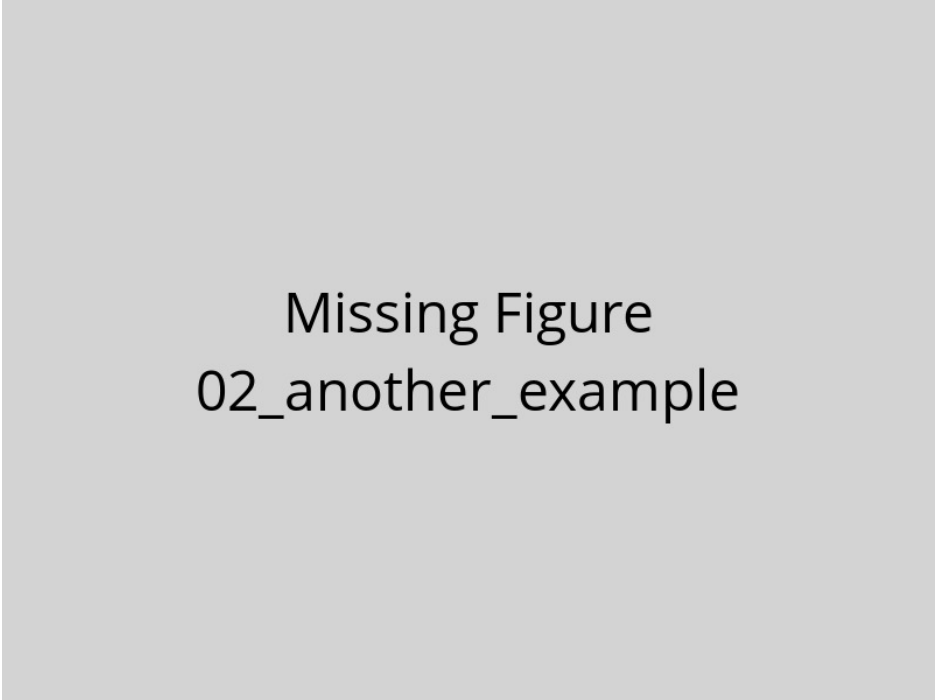| Step | Instructions |
|---|---|
| 1. Add CSV | Place file like `01_data.csv` in `caption_and_media/` |
| 2. Add Caption | Create `01_data.tex` with table caption |
| 3. Compile | Run `./compile -m` to process tables |
| 4. Reference | Use `\ref{tab:01_data}` in manuscript |

## 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.

**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.