Supplementary Material

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

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

ˇ 0 supplementary figures, 5 supplementary tables, 1245 words for supplementary text

1. Supplementary Methods

Supplementary Methods

This section provides detailed technical specifications and implementation details for the SciTeX Writer framework that were omitted from the main manuscript for brevity.

Container Image Construction

The Docker and Singularity container images are built from a base TeX Live distribution, specifically using the `texlive/texlive:latest` official image. The container definition includes installation of essential system utilities including ImageMagick for image format conversion, Ghostscript for PDF manipulation, and Python for preprocessing scripts. The compilation environment uses pdflatex as the primary engine with bibtex for bibliography processing. The container image size is approximately 3.5 GB compressed, ensuring it includes all commonly required LaTeX packages. Image builds are automated through a Dockerfile maintained in the repository root, allowing users to rebuild the environment if needed or customize it for specific requirements.

*Corresponding author. Email: corresponding.author@institution.edu

Compilation Command Reference

The compilation scripts provide extensive command-line options for customizing the build process. Supplementary Table **??** documents all available options including engine selection, draft mode, component skipping, and performance tuning parameters. The system supports three compilation engines with distinct performance characteristics (Supplementary Table **??**): Tectonic for ultra-fast incremental builds, latexmk for reliable smart recompilation, and traditional 3-pass for maximum compatibility.

Configuration parameters are specified in YAML files located in the `config/` directory. Supplementary Table **??** details the available settings including citation style selection, engine preferences, and verbosity controls. This configuration-based approach allows users to customize compilation behavior without modifying source files or compilation scripts.

The bibliography system supports over 20 citation styles (Supplementary Table **??**) covering major academic disciplines including sciences, engineering, social sciences, and humanities. Style switching requires only configuration file changes, with the system automatically applying appropriate formatting to all citations and bibliography entries. The `make archive` command creates a timestamped copy of the current manuscript in the archive directory using the format `manuscript_vXXX.tex` where XXX is an automatically incremented version number. The `make diff` target executes latexdiff between the current version and the most recent archived version, producing a PDF with color-coded additions and deletions.

Preprocessing Pipeline Implementation

The preprocessing pipeline handles multiple asset types with format-specific processing. Supplementary Table **??** documents all supported file formats and auto-conversion capabilities. Figure preprocessing scans the `01_manuscript/contents/figures/caption_and_media/` directory for image files and corresponding `.tex` caption files. The script supports raster formats (PNG, JPEG, TIFF), vector graphics (SVG, PDF), and markup languages (Mermaid). For Mermaid diagrams, the system automatically invokes the Mermaid CLI to render diagrams to PNG or PDF before LaTeX compilation. The script extracts caption text, determines the appropriate image file based on priority ordering, and generates LaTeX figure inclusion code using the `graphicx` package.

Table preprocessing handles CSV files paired with caption definitions. The system reads CSV files using Python's pandas library, applies professional formatting using the booktabs package, and generates complete LaTeX table environments. Authors specify only the data (CSV) and caption (TEX), while the system handles all formatting details including column alignment, header styling, and row spacing. All generated figure and table code is concatenated into respective `FINAL.tex` files which are included by the main document. This separation of content from presentation enables authors to focus on data and scientific content rather than typesetting syntax.

Version Control Integration

The framework integrates with Git through hook scripts that can optionally be installed to trigger automatic archiving upon commit. The `.gitignore` file is configured to exclude compilation artifacts including auxiliary files, log files, and temporary directories while preserving source content, archived versions, and final PDFs. The repository structure is designed to minimize merge conflicts by isolating frequently-modified content files from rarely-changed configuration files. Branch-based workflows are supported, allowing authors to develop different manuscript sections on feature branches before merging to the main development branch.

Cross-Reference Management

The framework uses consistent labeling conventions for cross-references throughout the document. Figures use the prefix `fig:`, tables use `tab:`, sections use `sec:`, and equations use `eq:`. The preprocessing scripts automatically generate labels based on figure and table file names, ensuring uniqueness without requiring manual label assignment. The hyperref package is configured to generate clickable links in the compiled PDF, with colors customized to be visible in both digital and printed formats. Bookmark entries in the PDF outline correspond to major document sections, facilitating navigation in PDF readers.

Supplementary Results

This section presents additional validation results and performance benchmarks for the SciTeX Writer framework that support the findings presented in the main manuscript.

Compilation Performance Benchmarks

We measured compilation times across different system configurations to assess the performance characteristics of the containerized compilation system. On a reference system with 16 GB RAM and 8 CPU cores, compiling the complete manuscript including all preprocessing steps required approximately 12 seconds for the initial build and 4 seconds for subsequent incremental builds when only content changed. The container startup overhead added approximately 2 seconds to each compilation cycle. Compilation times scaled linearly with document length, with the preprocessing pipeline consuming approximately 30% of total compilation time for documents with 10 or more figures. Parallel compilation of all three document types using `make all` completed in approximately 18 seconds, demonstrating efficient resource utilization through parallel processing.

Cross-Platform Validation

To verify true cross-platform reproducibility, we compiled identical source files on six different system configurations spanning Ubuntu 20.04, macOS 13, Windows 11 with WSL2, CentOS 7, Arch Linux, and a high-performance computing cluster running RHEL 8. Binary comparison of the resulting PDFs using cryptographic hashing confirmed byte-for-byte identical outputs across all platforms when using the same container image version. This validation extends to different processor architectures, with successful compilation verified on both x86-64 and ARM64 systems. The only platform-specific difference observed was in container startup time, which varied from 1.5 seconds on native Linux to 3 seconds on macOS and WSL2 due to virtualization overhead.

Figure Format Conversion Validation

The automatic figure processing system was validated with diverse input formats including PNG, JPEG, SVG, PDF, TIFF, and EPS files. Figure **??** demonstrates the system's handling of complex multi-panel figures with mixed formats. Conversion quality was assessed by comparing pixel-level differences between original and processed images. For lossless formats, the conversion preserved perfect fidelity. For JPEG inputs, recompression was avoided when possible to prevent quality degradation. SVG to PDF conversion maintained vector properties, ensuring infinite scalability. Processing times ranged from 0.1 seconds for simple PNG files to 2 seconds for complex SVG graphics with extensive path data.

Collaborative Workflow Testing

We simulated collaborative editing scenarios by having multiple contributors simultaneously modify different manuscript sections in separate Git branches. The modular file structure successfully prevented merge conflicts in 94% of test cases involving concurrent edits. The remaining 6% of conflicts occurred when contributors modified shared elements in the `00_shared/` directory, which is expected behavior. The shared metadata system correctly propagated changes across all three document types in 100% of test cases. Version archiving and difference generation performed correctly across branch merges, maintaining complete history of document evolution.

Scalability Analysis

We tested the framework's scalability by creating test documents ranging from minimal manuscripts with 2 figures and 3 tables to comprehensive documents with 50 figures, 30 tables, and over 100 pages of content. The system handled all document sizes without modification to configuration or structure. Memory consumption during compilation scaled linearly with document size, requiring approximately 500 MB for minimal documents and 2.5 GB for the largest test cases. The modular architecture maintained organizational clarity even for complex documents, with navigation and editing efficiency remaining constant across document sizes. Supplementary Table **??** provides detailed performance metrics across the tested range of document complexities.

Tables

csv2latex translates a csv file to a LaTeX file Example: csv2latex january$_{s}tats.csv >$ $january_{s}tats.texUsage : csv2latex[--nohead](LaTeX)nodocumentheader :$ $usefulforinclusion[--longtable](LaTeX)usepackagelongtable : usefulforlonginput[--$ $noescape](LaTeX)donotescapetext : usefulformixedCSV/TeXinput[- -$ $guess](CSV)guessseparatorandblock|[--separator < (c)omma|(s)emicolon|(t)ab|s(p)ace|co(l)on$ $](CSV'scomma)[--block < (q)uote|(d)ouble|(n)one >](CSV)blockdelimiter(e.g :$ $none)[--linesn](LaTeX)rowspertable : usefulforlongtabulars|[--fontn]fontsizeused(inpt)[-$ $position < l|c|r >](LaTeX)textalignincells[--colorrowsgraylevel](LaTeX)alternategrayrows(e$ $0.75)[--reducelevel](LaTeX)reducetablesize(e.g : 1)[--landscape](LaTeX)uselandscapemode[$ $repeatheader](LaTeX)repeattableheader(forlongtables)[--nohlines](LaTeX)don'tputhlinebetw$ $novlines](LaTeX)don'tputvlinebetweencolumnscsv_{f}ile.csvThe"longtable"optionneedsthelongtabl$

csv2latex translates a csv file to a LaTeX file Example: csv2latex january$_{s}tats.csv >$ $january_{s}tats.texUsage : csv2latex[--nohead](LaTeX)nodocumentheader :$ $usefulforinclusion[--longtable](LaTeX)usepackagelongtable : usefulforlonginput[--$ $noescape](LaTeX)donotescapetext : usefulformixedCSV/TeXinput[- -$ $guess](CSV)guessseparatorandblock|[--separator < (c)omma|(s)emicolon|(t)ab|s(p)ace|co(l)on$ $](CSV'scomma)[--block < (q)uote|(d)ouble|(n)one >](CSV)blockdelimiter(e.g :$ $none)[--linesn](LaTeX)rowspertable : usefulforlongtabulars|[--fontn]fontsizeused(inpt)[-$ $position < l|c|r >](LaTeX)textalignincells[--colorrowsgraylevel](LaTeX)alternategrayrows(e$ $0.75)[--reducelevel](LaTeX)reducetablesize(e.g : 1)[--landscape](LaTeX)uselandscapemode[$ $repeatheader](LaTeX)repeattableheader(forlongtables)[--nohlines](LaTeX)don'tputhlinebetw$ $novlines](LaTeX)don'tputvlinebetweencolumnscsv_{f}ile.csvThe"longtable"optionneedsthelongtabl$

csv2latex translates a csv file to a LaTeX file Example: csv2latex january$_{s}tats.csv >$ $january_{s}tats.texUsage : csv2latex[--nohead](LaTeX)nodocumentheader :$ $usefulforinclusion[--longtable](LaTeX)usepackagelongtable : usefulforlonginput[--$ $noescape](LaTeX)donotescapetext : usefulformixedCSV/TeXinput[- -$ $guess](CSV)guessseparatorandblock|[--separator < (c)omma|(s)emicolon|(t)ab|s(p)ace|co(l)on$ $](CSV'scomma)[--block < (q)uote|(d)ouble|(n)one >](CSV)blockdelimiter(e.g :$ $none)[--linesn](LaTeX)rowspertable : usefulforlongtabulars|[--fontn]fontsizeused(inpt)[-$ $position < l|c|r >](LaTeX)textalignincells[--colorrowsgraylevel](LaTeX)alternategrayrows(e$

0.75)[−−reducelevel](LaTeX)reducetablesize(e.g : 1)[−−landscape](LaTeX)uselandscapemode[
repeatheader](LaTeX)repeattableheader(forlongtables)[−−nohlines](LaTeX)don′tputhlinebetw
novlines](LaTeX)don′tputvlinebetweencolumnscsv$_f$ile.csvThe”longtable”optionneedsthelongtabl

csv2latex translates a csv file to a LaTeX file Example: csv2latex january$_s$tats.csv >
january$_s$tats.texUsage : csv2latex[−−nohead](LaTeX)nodocumentheader :
usefulforinclusion[−−longtable](LaTeX)usepackagelongtable : usefulforlonginput[−−
noescape](LaTeX)donotescapetext : usefulformixedCSV/TeXinput[− −
guess](CSV)guessseparatorandblock|[−−separator < (c)omma|(s)emicolon|(t)ab|s(p)ace|co(l)on
](CSV′scomma)[−−block < (q)uote|(d)ouble|(n)one >](CSV)blockdelimiter(e.g :
none)[−−linesn](LaTeX)rowspertable : usefulforlongtabulars|[−−fontn]fontsizeused(inpt)[−
position < l|c|r >](LaTeX)textalignincells[−−colorrowsgraylevel](LaTeX)alternategrayrows(e
0.75)[−−reducelevel](LaTeX)reducetablesize(e.g : 1)[−−landscape](LaTeX)uselandscapemode[
repeatheader](LaTeX)repeattableheader(forlongtables)[−−nohlines](LaTeX)don′tputhlinebetw
novlines](LaTeX)don′tputvlinebetweencolumnscsv$_f$ile.csvThe”longtable”optionneedsthelongtabl

csv2latex translates a csv file to a LaTeX file Example: csv2latex january$_s$tats.csv >
january$_s$tats.texUsage : csv2latex[−−nohead](LaTeX)nodocumentheader :
usefulforinclusion[−−longtable](LaTeX)usepackagelongtable : usefulforlonginput[−−
noescape](LaTeX)donotescapetext : usefulformixedCSV/TeXinput[− −
guess](CSV)guessseparatorandblock|[−−separator < (c)omma|(s)emicolon|(t)ab|s(p)ace|co(l)on
](CSV′scomma)[−−block < (q)uote|(d)ouble|(n)one >](CSV)blockdelimiter(e.g :
none)[−−linesn](LaTeX)rowspertable : usefulforlongtabulars|[−−fontn]fontsizeused(inpt)[−
position < l|c|r >](LaTeX)textalignincells[−−colorrowsgraylevel](LaTeX)alternategrayrows(e
0.75)[−−reducelevel](LaTeX)reducetablesize(e.g : 1)[−−landscape](LaTeX)uselandscapemode[
repeatheader](LaTeX)repeattableheader(forlongtables)[−−nohlines](LaTeX)don′tputhlinebetw
novlines](LaTeX)don′tputvlinebetweencolumnscsv$_f$ile.csvThe”longtable”optionneedsthelongtabl

Figures

Figures