

FastWrite: A Local-First LaTeX Writing Assistant with AI-Powered Editing

Anonymous Author(s)

Abstract

Writing high-quality academic papers in LaTeX is a challenging task that demands both technical proficiency and strict adherence to stylistic guidelines. While Large Language Models (LLMs) have shown promise in text generation, their seamless integration into the LaTeX writing workflow remains an open challenge. We introduce **FastWrite**, a novel local-first writing assistant tailored for computer systems researchers. FastWrite combines a local backend server with a modern React-based frontend to offer real-time, context-aware writing assistance. Key features include *Diagnose*, *Refine*, and *QuickFix* modes, which enable users to iteratively improve their drafts with granular control. Our evaluation demonstrates that FastWrite significantly reduces the cognitive load of LaTeX editing and enhances the overall quality of submissions.

1 Introduction

The rise of generative AI has transformed academic writing, but for computer systems and security researchers, this transition remains challenging.

1.1 The Challenge

Top-tier conferences such as S&P, OSDI, and CCS require an authoritative tone, precise terminology, and rigorous logical flow. These elements are essential for effectively communicating complex ideas and meeting the high standards of these venues. While Overleaf excels in providing a collaborative environment, it lacks deep integration with advanced AI assistance that can handle the nuances of academic LaTeX. This deficiency hinders both the efficiency and quality of the writing process, making it difficult to meet the stringent requirements of these conferences.

To address this, integrating more sophisticated AI tools is essential. These tools should enhance the collaborative features of platforms like Overleaf and provide real-time feedback on structure, clarity, and technical accuracy. This integration would help authors produce papers that are technically sound, well-structured, and clearly articulated, thereby increasing their chances of acceptance at these highly competitive venues.

Generic large language model (LLM) interfaces often generate verbose or hallucinatory text, necessitating extensive manual correction. Additionally, pasting sensitive drafts into cloud-based chatbots raises significant privacy concerns. These challenges underscore the need for more sophisticated and secure AI tools that can support the unique requirements of academic writing in computer systems and security.

1.2 Our Solution: FastWrite

To address these challenges, we introduce **FastWrite**, a privacy-focused, local-first LaTeX editor. FastWrite bridges the gap between traditional editors and modern AI capabilities. Unlike plugins that merely overlay text, FastWrite parses the LaTeX structure to offer context-aware suggestions. Our contributions are as follows:

- **Deep LaTeX Integration:** FastWrite understands sections, figures, and citations, ensuring that AI suggestions are syntactically correct and contextually relevant.
- **Granular AI Modes:** We propose three distinct interaction modes—Diagnose, Refine, and QuickFix—tailored to different stages of the writing process.
- **Local-First Architecture:** By running the backend locally, FastWrite ensures that the full paper draft never leaves the user’s machine unless explicitly sent to an LLM provider for processing specific segments.

2 Related Work

In this section, we compare FastWrite with existing academic writing tools, summarizing the key differences in Table 1. While Overleaf excels at col-

Feature	Overleaf	Copilot	Grammarly	FastWrite
Real-time Collaboration	✓	-	-	-
Local-First Privacy	-	-	-	✓
LaTeX Structure Awareness	✓	-	-	✓
Context-Aware Refinements	-	✓	✓	✓
Academic Style Tuning	-	-	-	✓

Table 1: Comparison of FastWrite with other writing assistants.

laboration, it lacks the intelligent assistance provided by FastWrite. Copilot and Grammarly are general-purpose tools that often fail to grasp the specific constraints of LaTeX sizing and formatting. {sections/3-method input

Metric	Baseline	FastWrite
Avg. Editing Time (min)	45.2	28.5
Acceptance Rate (%)	N/A	78%
User Satisfaction (1-5)	3.4	4.7

Table 2: User study results showing improved efficiency and satisfaction with FastWrite.

3 Evaluation

To validate the effectiveness of FastWrite, we conducted a user study with 12 Ph.D. students from top universities. Participants were asked to refine a draft abstract using both FastWrite and a baseline editor.

3.1 Quantitative Results

We measured the time taken to reach a satisfactory version and the number of accepted AI suggestions.

3.2 Qualitative Feedback

Users praised the *Refine* mode for its ability to sound like a native speaker while preserving technical accuracy. One participant noted: "It's like having a senior professor review my work, but without the pressure."

4 Implementation Details

We implemented FastWrite as a desktop-class web application. The tech stack was chosen for performance and developer experience.

5 Discussion

While FastWrite shows promise, there are several open challenges and limitations worth noting.

5.1 Limitations

Context Awareness: The current AI model processes text in small segments, such as paragraphs or sections, which can overlook cross-sectional inconsistencies. For instance, a variable defined in Section 3 might be used differently in Section 2, and the AI may not detect this discrepancy. Expanding the context window to address this issue is a priority, but it increases latency and computational costs.

PDF Compilation: FastWrite focuses on the writing experience. For final PDF compilation, users rely on external tools like ‘`ltxmk`’ or Overleaf. We integrate a local PDF previewer using the native TeX engines installed by user.

5.2 Future Work

- **Local LLM Support:** To enhance privacy and offline capability, we are experimenting with quantized models, such as Llama-3-8B, running locally via Ollama.
- **Collaboration:** We will implement CRDT-based real-time collaboration, enabling multiple authors to edit the same local file through P2P connections.
- **Git Integration:** We will integrate directly with GitHub and GitLab to manage version history seamlessly within the UI.

6 Conclusion

We have presented FastWrite, a specialized writing assistant for computer systems researchers. By combining a privacy-first local architecture with high-quality, domain-specific AI prompts, FastWrite empowers authors to produce clearer, more authoritative prose. Our evaluation demonstrates that FastWrite reduces editing time by 37% while significantly improving user satisfaction compared to standard tools. As we continue to refine the system, we envision FastWrite becoming an indispensable tool in the academic writing toolkit.

Acknowledgment

We thank the anonymous reviewers for their insightful comments. This work was supported in part by the VibeDev Open Source Grant.