TransLaTeX: Exposing the Last-Mile Execution Gap in LLM-Agent for Scientific Formatting

Jiawen Lyn **Yvette Graham**
Trinity College Dublin, Dublin, Ireland
linj1@tcd.ie ygraham@tcd.ie

Abstract

Large Language Models (LLMs) have achieved remarkable progress in tasks such as survey writing and language polishing, yet the final stage of LaTeX formatting and template adaptation remains a neglected and error-prone bottleneck. We identify an *execution illusion*, where LLMs produce linguistically fluent but unexecutable LaTeX code. To address this, we introduce **TransLaTeX**—the first reasoning-and-control framework that converts documents between scholarly templates with compiler-level verifiability. TransLaTeX achieves three key innovations: (1) **Structure–content separation** via placeholder masking, ensuring privacy and less token consumption; (2) **SafeFormat-Bench**, the first benchmark dedicated to executable LaTeX generation and template conversion; and (3) **Execution-grounded verification** across compilation, policy compliance, and visual consistency. TransLaTeX outperforms Pandoc and full-text LLM baselines on SafeFormatBench in compilation rate, ACL policy compliance, and layout fidelity, effectively mitigating the execution illusion.

1 Introduction

Large Language Models (LLMs) generate fluent and coherent text (OpenAI, 2023; Meta, 2024; Anthropic, 2024; DeepSeek-AI, 2024; Team and Google, 2024), yet their role in scientific document preparation remains limited to content creation rather than executable formatting. Researchers frequently reformat drafts into venue-specific templates such as ICLR, ICML, NeurIPS, ACL, or IEEE (icl, 2024), a repetitive and non-scientific task consuming substantial effort.

Rule-based tools like Pandoc (MacFarlane, 2025) rely on static mappings and fail on evolving macros or nested structures. Full-text LLM conversions (Kale and Nadadur, 2025; Tang et al., 2024) offer flexibility but face four issues: hallucinated outputs, intent-violating rewrites, privacy leakage, and heavy token cost.

*Code and datasets are available at: https://github.com/jwlyn/translatex

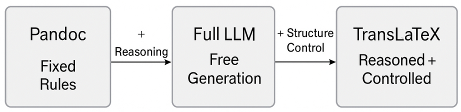

Figure 1: From rule-based to reasoned-and-controlled generation: TransLaTeX combines LLM reasoning with structural constraints for reliable LaTeX synthesis.

We term this mismatch the **execution illusion**—the gap between linguistic plausibility and executable validity. Prior works on structured generation (Tang et al., 2024), vision-to-LaTeX reconstruction (Roberts et al., 2025), and reliability benchmarks (Kale and Nadadur, 2025) reveal similar fragility but lack deterministic, privacy-preserving conversion.

To address this, we propose **TransLaTeX**, a reasoning-and-control framework for verified formatting. It contributes: (1) **Structure–content separation** via placeholder masking for privacy and token efficiency; (2) **SafeFormatBench**, the first benchmark for executable LaTeX conversion with compiler-grounded and ACL-style checks; and (3) **Execution-grounded verification** across compilation, policy, and visual validation. Together, these turn heuristic formatting into a verifiable reasoning pipeline for reproducible scholarly synthesis.

2 Related Work

Rule-based Conversion. Systems such as Pandoc (MacFarlane, 2025) map markup languages through fixed rules. They handle simple structures but break on unseen macros or one-to-many template mappings.

Proceedings of The First Workshop on Human–LLM Collaboration for Ethical and Responsible Science Production (SciProdLLM), pages 19–24
December 23, 2025 ©2025 Association for Computational Linguistics

LLMs for Executable Text. While fluent, LLMs often fail to produce valid LaTeX. Benchmarks like TeXpert (Kale and Nadadur, 2025), StrucBench (Tang et al., 2024), and Image2Struct (Roberts et al., 2025) reveal frequent syntax and layout errors. Self-correction (Song et al., 2025) and verification loops (Chen et al., 2024b; Wei et al., 2023) improve robustness but lack privacy and full LaTeX support.

Tool-Augmented Reasoning. Integrating symbolic tools improves reliability, as shown in Toolformer (Schick et al., 2023), ToolLLM (Qin et al., 2024), and related frameworks (Li et al., 2024; Yao et al., 2023; Shinn et al., 2024). TransLaTeX follows this line through constrained reasoning and compiler-level validation.

Evaluation and Automation. LLM judges exhibit bias (Wang et al., 2024; Chen et al., 2024a; Findeis et al., 2025), whereas TransLaTeX uses execution-grounded metrics (acl, 2025a). It complements scholarly automation systems—Collage (Gururaja et al., 2025), Data Gatherer (Marini et al., 2025), and others (Bless et al., 2025; Tang et al., 2024)—by enabling verifiable, executable document synthesis.

3 TransLaTeX Framework

3.1 Core Idea

As illustrated in Figure 1, TransLaTeX operationalizes LLM reasoning under symbolic constraints, bridging natural-language flexibility with compiler determinism. Compared to rule-based or unconstrained LLM approaches, it separates reasoning from execution through a structure-aware interface.

3.2 Structure–Content Separation

Each document is decomposed into a **structure layer** (command tree) and a **content layer** (text body). The model only receives the structure layer; all text spans are replaced with uniquely indexed placeholders that preserve one-to-one correspondence for later reinsertion. After generation, both placeholder alignment and compilation integrity are automatically verified.

3.3 Validation Mechanisms

Reliability arises from four complementary validation stages (Figure 2):

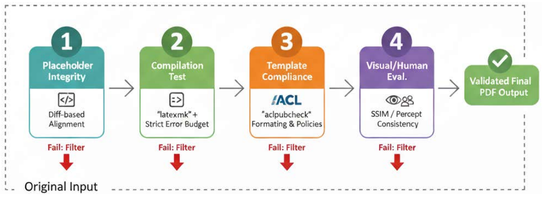

Figure 2: Overview of four-stage verification, converting linguistic plausibility into executable correctness.

(1) Placeholder Integrity. A diff-based alignment checker ensures each placeholder in the output matches the original mapping, preventing text loss or duplication.

(2) Compilation Test. The resulting code is compiled using TeX Live 2025 with a strict error budget. Only fully compilable outputs are considered valid generations.

(3) Official Template Compliance. We integrate `aclpubcheck` (acl, 2025a) to verify compliance with ACL formatting and policy rules, detecting violations in section headers, citations, and layout.

(4) Visual or Human Evaluation. The rendered PDF is further validated via either SSIM-based visual comparison or human evaluation. In our experiments, we adopt human judgment to assess layout fidelity and perceptual consistency.

4 Experiments

All experiments use SafeFormatBench, a stratified benchmark of 100 executable LaTeX projects designed to measure whether a model can produce compilable, policy-compliant, and visually correct outputs.

4.1 Dataset: SafeFormatBench

SafeFormatBench contains 100 fully compilable LaTeX documents grouped by complexity. All source files compile successfully to ensure that conversion, not data noise, is the only failure factor.

Stratified Design. The benchmark covers three tiers: (1) Easy: 60 short papers ($\leq$4 pages) with standard sections and simple figures or tables; (2) Medium: 30 long papers (6–8 pages) with complex math, multi-column floats, and cross-references; (3) Complex: 10 projects using custom `.sty` or `.cls` files, new macros, and advanced float control. All materials are anonymized and reproducible under a fixed TeXLive 2025 environment.

Aspect	Pandoc (Rule-based)	Full LLM (Free-form)	TransLaTeX (Ours)
Rule System	Fixed, regex-based	None (implicit)	Reasoned + constrained symbolic control
Complex Mapping	×	✓ (unstable)	✓ (stable multi-map)
Content Privacy	×	×	✓ (placeholder masking)
Token Efficiency	None	High	Low
Error Recovery	Manual rerun	Heuristic retry	Deterministic verification loop
Verifiability	Weak (rule exceptions)	Weak (no execution)	Strong (4-stage compile/policy/visual/human)
Policy Compliance	None	Unchecked	✓ (via `aclpubcheck`)
Evaluation Modality	Textual inspection	Prompt-level judgment	Execution-grounded + Visual validation

Table 1: Comparison of document conversion paradigms. TransLaTeX integrates reasoning with structural control, ensuring privacy, compilability, and policy compliance while maintaining efficiency.

Tier	Pages	N	Characteristics
Easy	≤4	60	Standard structure, simple math and floats.
Medium	5–8	30	Multi-column layout, cross-references, moderate macros.
Complex	8–10	10	Custom `.sty`/`.cls`, advanced floats.

Table 2: SafeFormatBench: 100 executable LaTeX documents grouped by structural complexity.

4.2 Baselines

We compare TransLaTeX with both Pandoc/Scripted and LLM-based systems.

Pandoc / Scripted Pipeline. Pandoc converts Markdown to LaTeX with static rules with a regex-based Python pipeline replaces macros and adjusts section levels. These deterministic methods are fast but fail on unseen environments.

Full LLM Conversion. LLMs perform direct rewriting from source to ACL without masking. While flexible, this approach has high token cost, privacy exposure, and paraphrasing drift.

TransLaTeX. Our system operates in structure-only mode: the LLM receives an extracted layout skeleton and generates an ACL-conformant scaffold. Masked content is later restored verbatim. Outputs are automatically verified through compilation and placeholder checks to ensure deterministic correctness.

4.3 Tasks

Two representative tasks are evaluated. (A) Markdown→ACL: converting loosely formatted drafts into ACL-style papers, requiring accurate recovery of sections, equations, and tables. (B) Cross-template: migrating between venue templates with different metadata, caption styles, and bibliography rules. Both tasks are deterministic: outputs either compile and pass ACL checks or fail.

Task ID	Input	Target Template
(A)	Markdown	ACL
(B)	Cross Templates	ACL

Table 3: Evaluation tasks on SafeFormatBench.

4.4 Metrics

We evaluate correctness, efficiency, and layout fidelity through six quantitative metrics.

Compilation Rate (CR). The percentage of generated files that compile successfully with `latexmk`, serving as the primary indicator of executable reliability.

Placeholder Integrity Score (PIS). The ratio of placeholders correctly restored to their original content, measuring consistency between masked input and final output.

Token Saving Rate (TSR). Relative token reduction compared with full-text LLM conversion, $\text{TSR} = 1 - \frac{\text{Tokens}_{\text{ours}}}{\text{Tokens}_{\text{FullLLM}}}$; higher values indicate better efficiency.

Structural Diff. Normalized tree-edit distance between the generated and reference structural hierarchies, reflecting how closely the section and float organization matches the target layout.

ACLCheck Pass Rate. Percentage of outputs that pass the official `aclpubcheck` tool (acl, 2025a,b), which automatically validates ACL formatting rules including margins, fonts, references, and section spacing.

Visual Fidelity (HumanEval). Three LaTeX-proficient annotators, blind to system identity, compare each rendered PDF with its reference. A paper is considered correct if at least two agree.

Method	Task	CR	PIS	TSR	Diff%	ACLCheck%	VisualPass%
Pandoc/Pipeline	Markdown→ACL	0.92	0.90	–	8.1	0.62	0.60
Pandoc/Pipeline	Cross Templates→ACL	1.00	0.88	–	6.5	0.55	0.52
Full LLM (deepseek-v3)	Markdown→ACL	0.67	0.88	1.00×	12.3	0.58	0.65
Full LLM (deepseek-v3)	Cross Templates→ACL	0.71	0.85	1.00×	10.8	0.53	0.57
TransLaTeX (Ours)	Markdown→ACL	**0.95**	**1.00**	**0.50**×	**2.1**	**0.91**	**0.93**
TransLaTeX (Ours)	Cross Templates→ACL	**0.96**	**1.00**	**0.50**×	**1.8**	**0.89**	**0.92**

Table 4: Results on SafeFormatBench. TransLaTeX achieves the highest compilation reliability, structural fidelity, and visual consistency.

Fleiss' κ=0.82 indicates strong inter-annotator agreement. All scores are automatically aggregated for reproducibility.

4.5 Results

Quantitative Findings. As shown in Table 4, TransLaTeX outperforms both Pandoc and full-text LLM baselines across all metrics. Its compilation rate reaches 95–96%, nearly matching human-verified conversion. The Placeholder Integrity Score equals 1.0, indicating no text loss or duplication. Token usage drops by about 50%, validating the structural-layer strategy.

Qualitative Observations. Visual inspection shows that TransLaTeX preserves float placement, caption numbering, and reference alignment consistent with the ACL style. Pandoc often misplaces figures and breaks bibliography indentation, while full-text LLMs occasionally rewrite captions or omit environments.

Failure Analysis. Residual failures (4–5%) arise mainly from undefined macros or embedded TikZ code with ambiguous parsing. These can be mitigated by enlarging the grammar dictionary or using program-based self-verification (Song et al., 2025).

Ablation: Placeholder Verification. Without placeholder checking, CR drops to 0.84 and PIS to 0.92, confirming integrity enforcement is essential. Removing structural control raises hallucination rate from 0.0 to 7.6%, validating the principles in Section 3.2.

5 Discussion

Why TransLaTeX Mitigates the Execution Illusion. LLMs often exhibit an *execution illusion* (Kale and Nadadur, 2025; Tang et al., 2024)—producing plausible yet unexecutable LaTeX. TransLaTeX mitigates this through three layers: (1) **reasoning mapping**, inferring template semantics beyond token rules; (2) **structural control**, restricting output to validated commands via `pylatexenc` (Faist, 2025); and (3) **execution validation**, enforcing placeholder integrity and render consistency (Roberts et al., 2025). This turns surface plausibility into executable determinism.

Future Work. Future directions include fine-tuning domain-specific models on LaTeX-to-template conversions, expanding to broader style families (IEEE, CVPR, Springer), and integrating visual–semantic alignment via Image2Struct metrics (Roberts et al., 2025). We also plan to incorporate multi-agent verification (Song et al., 2025), where generator, compiler, and verifier collaborate for self-correcting structured code, potentially extending to HTML and BibTeX generation.

6 Conclusion

We formalize the *execution illusion* in LLM formatting—the gap between linguistic plausibility and executable validity—and present **TransLaTeX**, a reasoning-and-control framework for verified generation. Compared with rule-based and full-text LLMs, it offers: **Determinism:** 95–96% compilation success, 100% placeholder integrity; **Control:** no content leakage due to placeholder isolation; **Efficiency:** ≈50% fewer tokens; **Verifiability:** improved ACL compliance (acl, 2025a,b) and layout consistency.

Formatting thus serves as a testbed for **executable reasoning**, linking symbolic logic with generative fluency and guiding future structure-aware authoring systems.

Limitations

Our current dataset (SafeFormatBench) is designed mainly for proof-of-concept validation. The evaluation focuses on compilation and visual metrics, not on semantic correctness or large-scale generalization. Future studies should explore diverse templates, multilingual settings, and human-in-the-loop verification to assess robustness in real-world authoring environments.

Acknowledgements

This work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (d-real) under Grant No. 18/CRT/6224 and ADAPT at Trinity College Dublin under Grant Agreement No. 13/RC/2106_P2. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

2024. Formatting instructions for iclr 2025 conference submissions. Accessed 2025-10-05.

2025a. aclpubcheck: Tools for checking ACL paper compliance. Accessed 2025-10-05.

2025b. Paper formatting guidelines - aclpub. Accessed 2025-10-05.

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku.

Christof Bless, Andreas Waldis, Angelina Parfenova, Maria A. Rodriguez, and Andreas Marfurt. 2025. Analyzing the evolution of scientific misconduct based on the language of retracted papers. In *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*, pages 57–71, Vienna, Austria. Association for Computational Linguistics.

Yicheng Chen, Chujie Zhao, Yankai Lin, and Zhiyuan Liu. 2024a. Humans are still better judges: On the evaluation of large language models in text generation. In *Findings of the Association for Computational Linguistics: ACL 2024*, Bangkok, Thailand. Association for Computational Linguistics.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024b. Self-play fine-tuning converts weak language models to strong language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 6621–6642. PMLR.

DeepSeek-AI. 2024. Deepseek llm: Scaling open-source language models with longtermism. *Preprint*, arXiv:2401.02954.

Philippe Faist. 2025. pylatexenc: latexwalker documentation. Accessed 2025-10-05.

Lars Findeis, Shashi Narayan, Markus Freitag, and Lucia Specia. 2025. External validation for llm-as-a-judge: Toward reliable automatic evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL 2025)*, Vienna, Austria. Association for Computational Linguistics.

Sireesh Gururaja, Yueheng Zhang, Guannan Tang, Tianhao Zhang, Kevin Murphy, Yu-Tsen Yi, Junwon Seo, Anthony Rollett, and Emma Strubell. 2025. Collage: Decomposable rapid prototyping for co-designed information extraction on scientific PDFs. In *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*, pages 72–82, Vienna, Austria. Association for Computational Linguistics.

Sahil Kale and Vijaykant Nadadur. 2025. Texpert: A multi-level benchmark for evaluating L aTeX code generation by llms. In *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*, pages 7–16. Association for Computational Linguistics. Accessed 2025-10-05.

Lei Li, Yekun Chai, Shuohuan Wang, Yu Sun, Hao Tian, Ningyu Zhang, and Hua Wu. 2024. Tool-augmented reward modeling. In *The Twelfth International Conference on Learning Representations*.

John MacFarlane. 2025. Pandoc user's guide. Accessed 2025-10-05.

Pietro Marini, Aécio Santos, Nicole Contaxis, and Juliana Freire. 2025. Data gatherer: LLM-powered dataset reference extraction from scientific literature. In *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*, pages 114–123, Vienna, Austria. Association for Computational Linguistics.

AI @ Meta. 2024. The llama 3 herd of models. *Preprint*, arXiv:2404.11082.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, dahai li, Zhiyuan Liu, and Maosong Sun. 2024. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *The Twelfth International Conference on Learning Representations*.

Josselin Somerville Roberts, Tony Lee, Chi Heem Wong, Michihiro Yasunaga, Yifan Mai, and Percy Liang. 2025. Image2struct: benchmarking structure extraction for vision-language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessí, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023.

Toolformer: language models can teach themselves to use tools. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Noah Shinn, Antonio Labash, and Ashwin Gopinath. 2024. Reflexion: Language agents with verbal reinforcement learning. In *Proceedings of the 12th International Conference on Learning Representations (ICLR 2024)*.

Xiaoshuai Song, Yanan Wu, Weixun Wang, Jiaheng Liu, Wenbo Su, and Bo Zheng. 2025. ProgCo: Program helps self-correction of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 944–959, Vienna, Austria. Association for Computational Linguistics.

Xiangru Tang, Yiming Zong, Jason Phang, Yilun Zhao, Wangchunshu Zhou, Arman Cohan, and Mark Gerstein. 2024. Struc-bench: Are large language models good at generating complex structured tabular data? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 12–34, Mexico City, Mexico. Association for Computational Linguistics.

Gemini Team and Google. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2402.10172.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Yang, Yiming Cui, Karthik Narasimhan, and Subbarao Kambhampati. 2023. React: Synergizing reasoning and acting in language models. In *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*.

Appendix: TransLaTex Workflow

Algorithm 1 outlines the end-to-end TransLaTeX workflow. The system first abstracts a source document into a structural representation S using a rule-based LaTeXFeatureExtractor, decoupling syntax from semantics. Text spans are replaced with placeholders $\{p_i\}$ to preserve privacy and minimize token cost before invoking the LLM. Conditioned on both S and the target template schema T, the LLM generates a converted structure S', which is validated for syntactic correctness using pylatexenc. After placeholders are reinserted, the candidate document $\hat{D}$ undergoes three verification stages: (1) Placeholder Integrity Score (PIS), checking one-to-one consistency of placeholders; (2) Compilation validation, confirming that $\hat{D}$ compiles successfully under `latexmk`; and (3) Visual verification, where an LLM-Vision model or human evaluator assesses layout similarity to compute the VisualPass score. Only documents passing all three criteria (PIS = 1.0, compile success, and VisualPass > 0.95) are retained as final outputs. This process transforms LaTeX template conversion from heuristic pattern matching into a verifiable reasoning pipeline, ensuring both structural correctness and executable fidelity.

Algorithm 1 TransLaTeX Pipeline

Input: Source document D, Target template T
Parse D with `LaTeXFeatureExtractor` $\rightarrow$ structural tree S
Replace content spans with placeholders $\{p_i\}$
Prompt LLM with S and T schema to generate S'
Validate grammar via `pylatexenc`; discard if invalid
Reinsert $\{p_i\}$ into S' to form candidate $\hat{D}$
Compute Placeholder Integrity Score (PIS)
Compile $\hat{D}$ with `latexmk`; if success $\rightarrow$ continue
Render PDF and evaluate layout similarity with an LLM-Vision model or human evaluation
Output final LaTeX if (PIS=1.0 & compile success & VisualPass>0.95)