PAPER2WEB: Let's Make Your Paper Alive!

Yuhang Chen[1*], **Tianpeng Lv**[1*], **Yao Wan**[1†], **Philip S. Yu**[2], **Dongping Chen**[3‡]
[1]Huazhong University of Science and Technology,
[2]University of Illinois Chicago, [3]University of Maryland
{u202315752, wanyao}@hust.edu.cn, dongping@umd.edu
[*] Equal Contribution. [†] Corresponding Author. [‡]Project Leader.

Abstract

Academic project websites can more effectively disseminate research when they clearly present core content and enable intuitive navigation and interaction. However, current approaches such as direct *Large Language Model* (LLM) generation, templates, or direct HTML conversion struggle to produce layout-aware, interactive sites, and a comprehensive evaluation suite for this task has been lacking. In this paper, we introduce PAPER2WEB, a benchmark dataset and multi-dimensional evaluation framework for assessing academic webpage generation. It incorporates rule-based metrics and human-verified LLM-as-a-Judge, and PaperQuiz, which measures paper-level knowledge retention. We further present PWAGENT, an autonomous pipeline that converts scientific papers into interactive and multimedia-rich academic homepages. The agent iteratively refines both content and layout through tool use. Our experiments show that PWAGENT consistently outperforms end-to-end baselines like template-based webpages and arXiv/alphaXiv versions by a large margin while maintaining low cost. The code is available[1] and the demonstration video is provided[2].

1 Introduction

Research papers are predominantly distributed in PDF format, conveying information solely through static text and images (Tkaczyk et al., 2015; Li et al., 2020; Clark and Divvala, 2016; Lo et al., 2020). However, PDFs offer limited support for interactivity and multimedia content (W3C Web Accessibility Initiative, 2018; Government Digital Service and Central Digital and Data Office, 2024; NHS Digital, 2025; Kumar and Wang, 2024), resulting in substantial information loss during dissemination (Tkaczyk et al., 2015; Li et al., 2020).

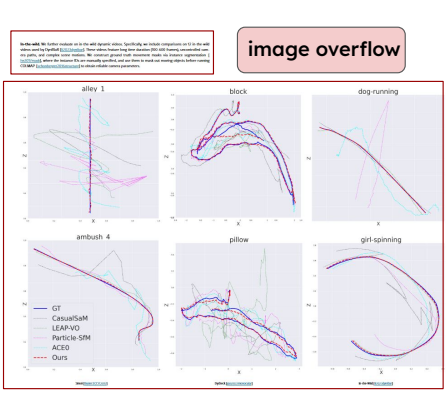
Recently, growing efforts have sought richer ways to transform scholarly articles—such as converting papers into concise posters with Paper2Poster (Pang et al., 2025), presentation slides with PresentAgent (Shi et al., 2025), videos with Paper2Video (Zhu et al., 2025), public-facing content with AutoPR (Chen et al., 2025). However, these approaches either discard the fine-grained details present in the original text or retain only the main ideas while overlooking the communicative advantages of multimedia content such as videos and animated graphics. This creates a gap for formats that preserve core knowledge while integrating multimedia to enhance scientific communication across communities.

Compared with prior methods, online webpages integrate text with multimedia in a coordinated and navigable manner. Recent efforts convert full academic papers into web pages to broaden accessibility. The arXiv HTML initiative (Frankston et al., 2024) is one example, yet such approaches often produce disordered layouts and redundant text, reducing readability and precision. As shown in Figure 1, common failure modes include rigid figure grids, detached captions, and limited interactivity. AlphaXiv uses LLMs for content condensation and layout optimization, yet still limits its author control over visual design, resulting in static presentations that underuse interactive capabilities. These issues stem from TeX–HTML pipelines emulating LaTeX behavior without a full TeX engine (Frankston et al., 2024), causing visual inconsistencies. Additionally, direct LLM-driven generation struggles with long contexts (Liu et al., 2024; Hsieh et al., 2024) and multimedia integration (Xiao et al., 2024).

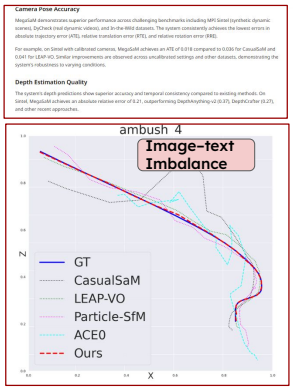
As shown in Figure 3, well-designed webpages enable broader research dissemination by presenting text, multimedia, and interactive components in a coordinated and navigable manner.

[1] https://github.com/YuhangChen1/Paper2All/tree/main/Paper2Web
[2] https://youtu.be/pcgeHoUBaWc

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 578–585



(a) Web page of arXiv HTML version.



(b) Web page generated by alphaXiv.

Figure 1: Problems in current scholar web page generation, including distorted layout and limited interactivity.

Our Work: PAPER2WEB. In this paper, we introduce PAPER2WEB to transform academic papers into structured, interactive, and knowledge-rich webpages. We first construct a dataset of paired academic papers and corresponding webpages. Specifically, we crawl accepted papers from conferences, parse their texts to extract reliable metadata such as authorship, and augment each record with citation counts from Semantic Scholar. We then perform multi-stage filtering, yielding PAPER2WEB, a dataset of 10,700 papers with the corresponding verified homepages.

Based on PAPER2WEB, we propose PWAGENT, a multi-agent framework for transforming scholarly documents into structured and interactive web content. PWAGENT first decomposes a paper into structured assets and organizes links and executable artifacts under a unified schema. It then constructs a semantically aligned resource repository enriched with relational metadata and exposed through standardized tools for downstream use. A content-aware allocation heuristic estimates each asset's layout budget. Finally, agent-driven iterative refinement drafts an initial website and improves it through optimization cycles.

We further construct a benchmark for PAPER2WEB. We introduce the first metric for measuring the interactivity and dynamic elements of generated webpages. Furthermore, we propose PaperQuiz to evaluate knowledge transfer from webpage screenshots through both verbatim and interpretive questions, incorporating a verbosity penalty to discourage overly text-heavy pages. On this benchmark, PWAGENT improves connectivity and completeness by roughly 12% on average across methods, achieving a 28% gain over the arXiv HTML baseline. It also yields an 18% average improvement under MLLM-as-a-Judge (Chen et al., 2024) evaluation and remains competitive with template-based variants.

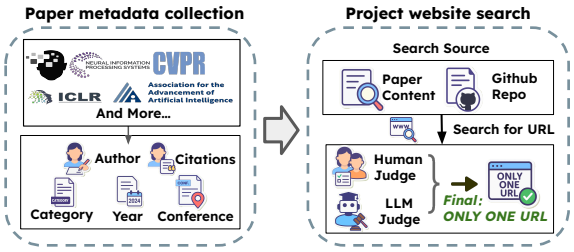


Figure 2: We collect the first paper-webpage dataset by crawling across multiple top-tier conferences and filtering by online search and human annotators.

2 Paper2Web Dataset

Since no dataset exists for analyzing academic website content and layout, we collect data from recent AI papers. We harvest project links from papers and code repositories, then crawl the corresponding webpage. Finally, we collect a comprehensive dataset covering multiple conferences and categories with 10,716 papers and their human-created project homepages. Figure 2 presents our data collection pipeline.

2.1 Data Collection

Paper Metadata Collection. We focus on AI papers as they are recent, peer-reviewed, cover diverse subfields with varied modalities, and attract attention that motivates high-quality dissemination. Using automated tools, we collect papers from major AI conferences (ICML, NeurIPS, WWW, ICLR, etc., 2020-2025). We extract source links, parse full texts for metadata (title, authors, venue, year), and retrieve citation counts from Semantic Scholar. Each paper's introduction is submitted to an LLM

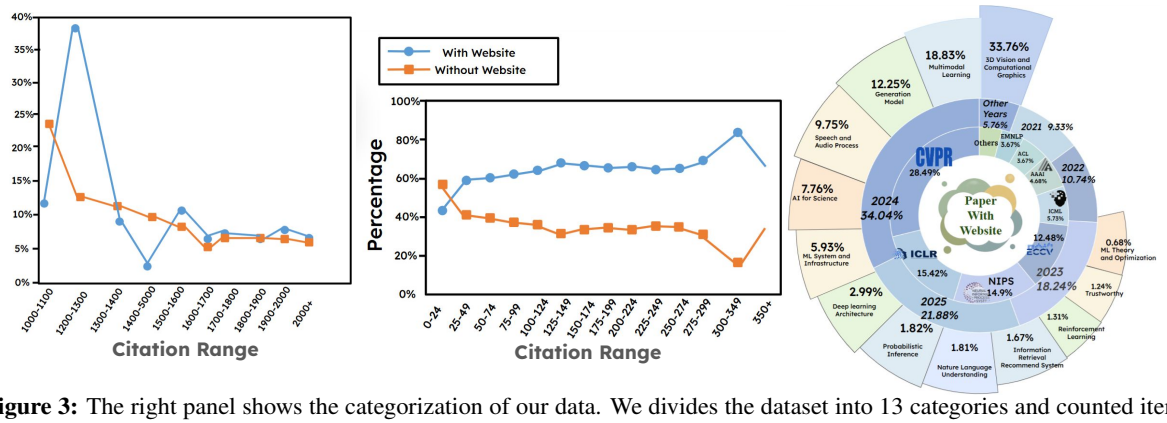


Figure 3: The right panel shows the categorization of our data. We divides the dataset into 13 categories and counted items in each. In addition, we show distributions by conference and by year. The left panel presents, for each category, the relative proportions of papers without and with a website among papers with low citation counts. The middle panel depicts the distribution of papers without and with a website restricted to highly cited papers (those with over 1,000 citations).

that assigns one of thirteen topical categories (Figure 3, right panel), enabling standardized analysis.

Our pipeline retrieves external links from each paper and its code repository, scanning the paper body and README files. We parse local context around each link, crawl the target HTML, and use an LLM to analyze the content. Human reviewers resolve ambiguous cases to ensure each paper maps to at most one canonical project website. Papers lacking relevant links in either source are defined as having no project homepage.

2.2 Data Characteristics

Finally, we curate a dataset comprising 10,716 papers with human-created project homepages and 85,843 without. We group papers into 13 categories following ICML/NeurIPS/ICLR conference taxonomies. The right panel of Figure 3 shows computer vision has the strongest demand for project websites, with homepage adoption rising steadily in recent years. To characterize webpage features, we manually audited 2,000 samples. We define interactive sites as pages with dynamic behaviors and explorable components responding to user intent; multimedia pages as those embedding rich media like videos; and static sites as pages delivering primarily text and still images in linear presentation.

3 Evaluation Metrics

3.1 Connectivity & Completeness

This metric assesses hyperlink quality and structural fidelity via LLM-based HTML analysis. Connectivity uses a dedicated URL parser to examine whether the webpage effectively links internal and external resources, while completeness measures the extent to which the webpage reproduces the core sections of the source paper. It incorporates quantitative priors including Image–Text Balance and Information Efficiency to ensure layout harmony and content density.

Image–Text Balance Prior. Let D denote the weighted deviation between the observed image–text ratio and the ideal 1:1 balance, and let $\gamma > 0$ be a scaling factor (Pang et al., 2025). We define the penalty term and score as:

$$\zeta = \frac{5}{1 + \gamma \cdot D}, \qquad S_{\text{img-txt}} = 5 - \zeta. \quad (1)$$

Information Efficiency Prior. To encourage concise, information-dense presentation, let $r = L/W$ denote the ratio between the generated text length L and the median human-designed length W, with $\beta > 0$ a scaling factor (e.g., β=0.6) (Tufte and Graves-Morris, 1983). We define the efficiency as:

$$p(r) = \frac{5}{1 + \beta \cdot \max(0, r - 1)}. \quad (2)$$

3.2 Holistic Evaluation with Human-Verified MLLM-as-a-Judge

To evaluate the overall effectiveness of web pages at a holistic level, we employ a MLLM as an automated judge, combined with human verification to mitigate bias. The model outputs a quantitative score ranging from 1 to 5 for each webpage. Specifically, it evaluates three key dimensions: *Interactive*, which measures element responsiveness, saliency emphasis, and overall usability; *Aesthetic*, which assesses element quality, layout balance, and visual appeal; and *Informative*, which evaluates the clarity and logical coherence of webpage content.

3.3 PaperQuiz

Inspired by Paper2Poster (Pang et al., 2025), we focus on the academic web page and acknowledge

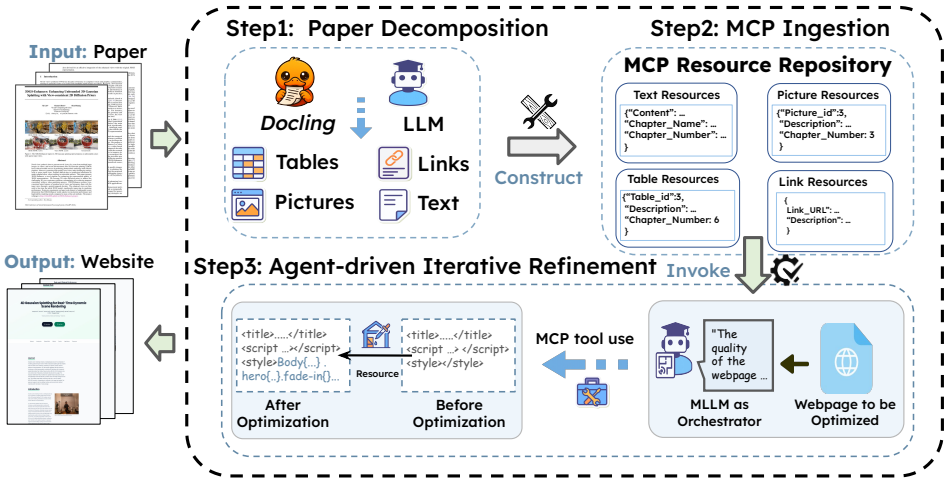


Figure 4: An overview of PWAGENT. Papers are deconstructed via Docling/Marker and LLMs into multiple assets and stored in an MCP repository. An agent drafts a page, then iteratively optimizes until layout and UX are solid.

its central role in communicating research as a dynamic bridge between authors and a broader audience. Therefore, we design an evaluation protocol that simulates this knowledge-transfer scenario. We first employ an LLM as an examiner to generate a comprehensive set of 50 questions from the source paper. These questions are divided into two types: 25 Verbatim questions, which are directly answerable from specific text, figures, or tables on the webpage, and 25 Interpretive questions, which require a higher-level comprehension of the paper's core contributions, methodology, and results. In the second stage, we present a screenshot of the rendered webpage to a diverse panel of MLLMs (including both open and closed source models). These models are tasked with answering the quiz based solely on the provided webpage content. By comparing the quiz scores across different generated web pages, we can quantitatively assess which one most effectively conveys the original paper's essential information. To prevent high scores resulting from excessive text transfer, we introduce a penalty term ζ, defined in Eq. 1.

4 PWAGENT: A Strong Baseline

To address the core challenges of the PAPER2WEB, we introduce PWAGENT, an automated pipeline for converting scientific papers into project homepages. The core of our approach involves parsing the paper's content into a structured format managed by an MCP server (Hou et al., 2025; Ehtesham et al., 2025; Krishnan, 2025). This server encapsulates key paper assets, along with predefined prompts for webpage generation and stylistic refinement, organizing them into a centralized resource repository. During this process, the agent leverages the tool-use capabilities of the MCP to access the resource repository, enabling a continuous optimization loop. The overall process includes the following key stages: (1) *Paper Decomposition*, which isolates key contributions from the paper. (2) *MCP Ingestion*, which encapsulates these contributions as a resource repository managed by the MCP server. (3) *Agent-driven Iterative Refinement*, which connects the MCP server to LLM-based agents that autonomously perform content matching and optimization through tool calls.

4.1 Paper Decomposition

We first deconstruct an unstructured scientific paper into structured intellectual assets that populate the MCP Resource Repository. Starting from the source PDF, the document is converted to Markdown using tools such as MARKER[3] or DOCLING[4]. An LLM then performs semantic decomposition that extracts metadata, reconstruct tables, and model detailed page layout and reading order, yielding a machine-readable representation like JSON or Markdown that captures key contents.

Instead of summarizing, the LLM analyzes the Markdown text against a predefined schema to identify, isolate, and organize the paper's key assets. These assets fall into three categories: (1) *Textual Assets:* each logical section is represented as a distinct resource object containing its title, LLM-generated synopsis, full text, and metadata; (2) *Visual Assets:* figures and tables are extracted as images and linked to their original captions, labels, and textual references to preserve context; and (3)

[3]https://github.com/datalab-to/marker
[4]https://github.com/docling-project/docling

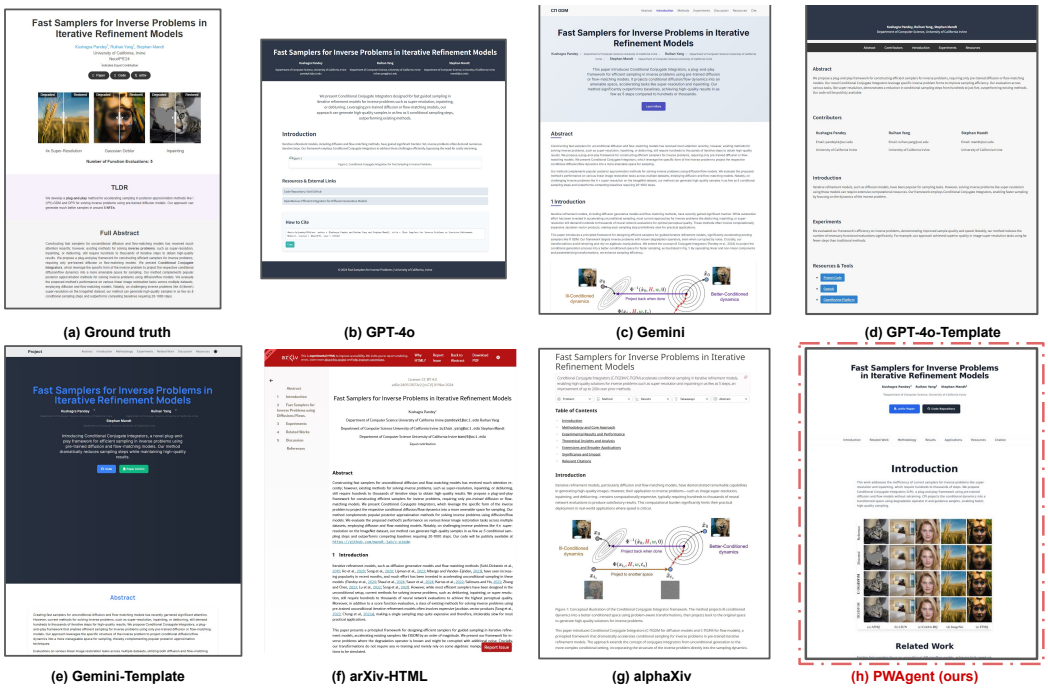


Figure 5: Illustration of website variants generated by different methods.

Link Assets: external URLs and internal citations are systematically captured and categorized to provide structured access to supplementary materials.

4.2 MCP Ingestion

Here we apply the MCP to the task of transforming scholarly papers into structured, queryable resources. We first instantiate a fully instrumented MCP server, which converts static assets into queryable resources with stable IDs and standardized tool access points. The server is responsible for resource construction, materializing assets with relational metadata and provisional layout budgets, and for tool registration, exposing consistent API for downstream retrieval, composition, and editing.

We enrich the parsed outputs with cross-modal semantics: (1) An LLM is used to align each visual element with its most relevant textual description and adds back-references to the citing paragraphs. (2) Link assets are typed by function to support structured cross-references. To achieve a coherent visual presentation, a content-aware spatial allocation heuristic estimates each asset's footprint and assigns a proportional layout budget to balance visual density across the page.

These enriched records are then committed to MCP server as MCP Resource Repository, where each resource is stored with a unique rid and fields for grounding and navigation. Concretely, the text resource stores the full paragraph and an LLM-generated synopsis; a Visual resource stores the image and its caption; and a Link resource stores the URL, its semantic role, and a short descriptor. Together, these resources form a structured, cross-referenced repository that serves as the foundation for webpage synthesis. Finally, the server registers a compact tool suite that provides enumeration of resource IDs, access to grounded content and metadata for rendering, typed references for connectivity placement, and initial layout allocation. This lightweight yet expressive interface is sufficient to synthesize a well-grounded HTML first draft for subsequent refinement by agentic workflow.

4.3 Agent-Driven Iterative Refinement

Finally, we propose an agent-driven iterative refinement mechanism to progressively enhance the layout, visual coherence, and semantic alignment of generated webpages. The process begins with initial page generation, where the agent retrieves essential metadata and relevant assets from the resource repository using MCP tools. Based on this information, it rapidly constructs a foundational webpage that serves as the baseline for refinement.

Following initialization, the system enters an iterative refinement loop that continues until no further corrective actions are needed or a predefined iteration limit is reached. At its core is an MLLM acting as the *Orchestrator Agent*, which conducts holistic visual assessments of the rendered webpage and invokes MCP tools to fix detected flaws. To address complex layout and visual consistency issues, the

Methods	Connectiveness				Completeness				Holistic Evaluation								
									Interactive			Aesthetic			Informative		
	Rule ↑	LLM ↑	Human ↑	Avg.↑	Rule↑	LLM↑	Human↑	Avg.↑	MLLM↑	Human↑	Avg.↑	MLLM↑	Human↑	Avg.↑	MLLM↑	Human↑	Avg.↑
Original Website	3.20	3.47	2.99	3.22	3.17	3.93	4.00	3.70	1.70	3.37	2.54	3.14	3.63	3.39	4.49	3.86	4.18
Model end-to-end methods																	
GPT-4o	1.81	2.07	2.05	1.98	2.11	3.15	3.43	2.90	0.53	1.85	1.19	2.61	2.13	2.37	2.01	2.56	2.29
Gemini-2.5-flash	2.26	2.11	2.16	2.18	2.72	3.56	3.43	3.24	1.30	2.15	1.73	2.80	2.41	2.61	3.63	2.68	3.16
DeepSeek-V3.2-Exp	1.83	2.09	2.16	2.03	2.09	3.21	3.51	2.94	0.54	2.01	1.28	2.63	2.20	2.42	2.21	2.61	2.41
Qwen3-Coder-480B-A35B	2.52	3.05	2.82	2.80	2.79	3.58	3.62	3.33	1.44	2.43	1.94	2.74	2.49	2.62	3.92	2.81	3.37
Model end-to-end methods + Template																	
GPT-4o-Template	1.83	2.26	2.77	2.29	2.25	3.37	3.54	3.05	0.56	1.47	1.02	2.63	2.35	2.49	3.87	2.58	3.23
Gemini-Template	2.47	2.87	2.78	2.71	2.73	3.72	3.78	3.41	**1.47**	1.58	1.53	2.75	2.46	2.61	4.28	2.67	3.48
DeepSeek-Template	2.38	2.91	2.80	2.70	2.75	3.68	3.84	3.42	1.45	1.60	1.53	2.74	2.46	2.60	4.26	2.67	3.47
Qwen-Template	3.01	3.21	2.87	**3.03**	2.88	3.90	3.80	3.53	**1.47**	1.58	1.53	2.77	2.93	2.85	**4.31**	3.22	3.77
Automated generation methods																	
arXiv (HTML)	**3.70**	2.23	1.34	2.42	2.49	3.81	3.75	3.35	1.05	1.51	1.28	2.72	2.65	2.69	4.01	3.06	3.54
alphaxXiv	3.43	3.01	2.91	3.12	2.88	3.95	3.85	**3.56**	1.25	1.61	1.43	2.73	2.80	2.77	4.20	3.46	3.83
PWAGENT (Our)	3.06	**3.30**	**2.94**	3.10	**2.91**	**4.02**	**3.86**	**3.56**	1.39	**3.16**	**2.28**	**2.82**	**3.35**	**3.09**	**4.31**	**3.56**	**3.93**

Table 1: Comparison between PAPER2WEB and other baselines across *Completeness*, *Connectivity* and MLLM evaluations.

Methods	Verbatim			Interpretive			Avg	Score with Penalty			
	open-source ↑	closed-source ↑	V-Avg ↑	open-source ↑	closed-source ↑	I-Avg ↑	Avg ↑	Penalty ↓	V_avg ↑	I_avg ↑	Avg ↑
Original Website	2.94	2.14	2.54	3.81	3.09	3.45	3.00	1.43	1.11	2.02	1.57
Model end-to-end methods											
GPT-4o	2.53	1.46	1.99	3.38	2.32	2.85	2.42	3.03	-0.93	-0.18	-0.56
Gemini-2.5-flash	2.60	1.59	2.10	3.14	2.72	2.93	2.52	2.18	-0.19	0.71	0.24
DeepSeek-V3.2-Exp	2.55	1.54	2.00	3.21	2.55	2.88	2.44	2.26	-0.26	0.62	0.18
Qwen3-Coder-480B-A35B	2.65	1.64	2.15	3.22	3.02	3.12	2.64	2.12	0.03	1.00	0.52
Model end-to-end methods + Template											
GPT-4o-Template	2.58	1.42	2.00	3.48	2.25	2.87	2.43	2.50	-0.50	0.37	-0.07
Gemini-Template	3.62	3.36	3.49	4.40	4.45	4.42	3.96	2.01	1.48	2.41	1.95
DeepSeek-Template	3.55	3.19	3.37	4.11	4.25	4.18	3.78	**1.96**	1.41	2.22	1.82
Qwen-Template	3.70	3.44	3.57	4.52	4.41	4.47	4.02	2.00	1.57	2.47	2.02
Automated generation methods											
arXiv (HTML)	3.62	3.42	3.52	4.52	4.43	4.47	4.00	2.87	0.65	1.60	1.13
alphaxXiv	3.57	**3.60**	3.58	**4.58**	**4.54**	**4.56**	**4.07**	1.97	**1.61**	**2.59**	**2.10**
PWAGENT (Our)	**3.76**	3.42	**3.59**	4.56	4.40	4.48	4.04	2.00	1.59	2.48	2.03

Table 2: PaperQuiz evaluation on the PAPER2WEB, based on open and closed-source MLLMs. The evaluation metrics include Raw Score and Score with Penalty under two settings: *"Verbatim"* and *"Interpretive"*.

Orchestrator performs joint global–local reasoning and coordinates targeted optimizations through tool calls. To reduce hallucinations during long-range reasoning, the agent segments the rendered page into independent visual tiles linked to their corresponding HTML fragments, sequentially analyzing each to detect imbalances and misalignments and propose precise edits. After each round of local refinement, adjacent tiles are merged, borrowing the spirit of merge sort. Therefore, neighboring regions can be jointly optimized by integrating their HTML and imagery. This aggregation allows the MLLM to capture inter-section dependencies and prevent visual artifacts such as overflow, occlusion, or cross-section drift. Finally, the Orchestrator performs a global pass to assess overall content completeness and visual harmony, realizing a part-to-whole optimization path that further mitigates hallucinations. The process terminates once optimization is complete or the maximum refinement cycles are reached.

5 How PWAGENT Makes Paper Alive?

5.1 Experiment Setups

We evaluate four distinct baseline methodologies to rigorously assess the performance of our proposed approach. These serve as crucial benchmarks for gauging information dissemination efficacy and human-centered friendliness. **(1) Oracle Method**, original websites created by authors. They serve as the gold standard for optimal presentation and content delivery; **(2) End-to-End Generation**, where GPT-4o, *Gemini-2.5-Flash* (Gemini), *DeepSeek-V3.2-Exp* (DeepSeek) and *Qwen3-Coder-480B-A35B* (Qwen) generate websites either through text-based rendering from scratch or by adapting the widely adopted Nerfies academic website template (Park et al., 2021) (The above models combined with a template will be referred to respectively as GPT-4o-Template, Gemini-Template,

DeepSeek-Template, and Qwen-Template); **(3) Existing HTML Versions**, where research papers from arXiv and alphaXiv provide public HTML versions; **(4) PWAGENT (Our)**, where Qwen3-30B-A3B is responsible for paper deconstruction and MCP ingestion, while the Orchestrator Agent is powered by the Qwen2.5-VL-32B model.

5.2 Main Results

Completeness & Connectivity. As shown in the left half of Table 1, we evaluate website completeness and connectivity. arXiv-HTML attains high rule-based connectivity but receives 64% lower human ratings, as it indiscriminately converts every citation into links, inflating metric scores while degrading user experience. alphaXiv shows balanced connectivity by selectively surfacing important links. For completeness, arXiv-HTML preserves verbose text with few images, scoring well with LLM and human judges but poorly on rule-based metrics. In contrast, our PWAGENT achieves 2% higher LLM-judged completeness than ground truth, demonstrating superior content condensation and balanced layout of text, images, and links.

Holistic Evaluation. As shown in the right half of Table 1, our PWAGENT achieves highest scores across all dimensions. While alphaXiv performs well in completeness and connectivity, it lacks interactive components, scoring 37% lower than our method in interactivity. Template-based methods effectively guide layout but constrain interactive element generation. Overall, PWAGENT outperforms all other methods, achieving 91% of ground truth quality in aesthetics and 94% in informativeness, with a 59% improvement in interactivity.

PaperQuiz. As shown in Table 2, we observe that: (1) Without the conciseness penalty, arXiv-HTML scores strongly; once applied, both arXiv-HTML and end-to-end GPT-4o receive large deductions, highlighting the value of concise, engineered sites and supporting website generation as effective context compression. (2) Gemini and Qwen are strong and generally outperform GPT-4o and DeepSeek; templates lift all models—DeepSeek-Template nears Gemini-Template, and Qwen-Template approaches the ground-truth site. (3) Across methods, open-source reader models consistently beat closed-source ones, indicating some open-source MLLMs (e.g., Qwen) can match or exceed closed models on certain visual tasks. (4) PWAGENT achieves best or near-best results across tasks and models, with total

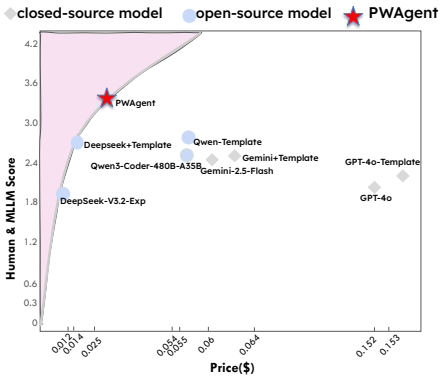


Figure 6: Pareto-front comparison of each website generation methods. Our PWAGENT achieve the highest quality with moderate and affordable cost.

information coverage rivaling arXiv-HTML; after the penalty, it still attains the highest overall score. (5) PWAGENT's penalty remains nontrivial, and the ground-truth site scores lower than expected, likely because it includes many videos and animations; in practice, authors can start from PWAGENT and add multimedia to reach the most desirable design.

5.3 In-depth Analysis

Efficiency Analysis. Figure 6 presents the average token cost per website. Our PWAGENT is highly token-efficient, requiring only $0.025 to produce a high-quality academic page. By contrast, end-to-end methods are costlier: GPT-4o is about $0.141 and Gemini about $0.054 per website. This yields 82% and 54% cost reductions, respectively, while maintaining strong page quality and usability. Even template-aided open models around $0.069 remain 2.8× more expensive, yet offer no clear advantage. Overall, PWAGENT delivers *state-of-the-art* cost efficiency with high presentation quality.

Case Study. In Figure 5, we present a qualitative comparison of different website baselines for a paper. Our PWAGENT not only preserves the structural integrity of the original paper but also achieves a well-balanced image-to-text ratio. Also, it offers versatile styling and superior aesthetic quality. However, there is still room for improvement compared to the human-designed version.

6 Conclusion

In this paper, we have introduced PAPER2WEB, a novel task and benchmark for generating project homepages from academic papers, and identified key challenges faced by current generative models and automated methods in handling long-context and layout-sensitive tasks.

References

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yaochen Wang, Yinuo Liu, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. 2024. Mllm-as-a-judge: Assessing multimodal llm-as-a-judge with vision-language benchmark. In *Forty-first International Conference on Machine Learning*.

Qiguang Chen, Zheng Yan, Mingda Yang, Libo Qin, Yixin Yuan, Hanjing Li, Jinhao Liu, Yiyan Ji, Dengyun Peng, Jiannan Guan, Mengkang Hu, Yantao Du, and Wanxiang Che. 2025. Autopr: Let's automate your academic promotion!

Christopher Clark and Santosh Divvala. 2016. Pdffigures 2.0: Mining figures from research papers. *Preprint*, arXiv:1606.01847.

Abul Ehtesham, Aditi Singh, Gaurav Kumar Gupta, and Saket Kumar. 2025. A survey of agent interoperability protocols: Model context protocol (mcp), agent communication protocol (acp), agent-to-agent protocol (a2a), and agent network protocol (anp). *arXiv preprint arXiv:2505.02279*.

Charles Frankston, Jonathan Godfrey, Shamsi Brinn, Alison Hofer, and Mark Nazzaro. 2024. Html papers on arxiv–why it is important, and how we made it happen. *arXiv preprint arXiv:2402.08954*.

Government Digital Service and Central Digital and Data Office. 2024. Publishing accessible documents. Last updated 2024-08-14.

Xinyi Hou, Yanjie Zhao, Shenao Wang, and Haoyu Wang. 2025. Model context protocol (mcp): Landscape, security threats, and future research directions. *arXiv preprint arXiv:2503.23278*.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*. COLM 2024.

Naveen Krishnan. 2025. Advancing multi-agent systems through model context protocol: Architecture, implementation, and applications. *arXiv preprint arXiv:2504.21030*.

Anukriti Kumar and Lucy Lu Wang. 2024. Uncovering the new accessibility crisis in scholarly pdfs. In *Proceedings of the 26th ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '24)*.

Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. 2020. Docbank: A benchmark dataset for document layout analysis. *arXiv preprint arXiv:2006.01038*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel S. Weld. 2020. S2orc: The semantic scholar open research corpus. In *Proceedings of ACL 2020*.

NHS Digital. 2025. Pdfs and other non-html documents — nhs digital service manual. Updated 2025-02.

Wei Pang, Kevin Qinghong Lin, Xiangru Jian, Xi He, and Philip Torr. 2025. Paper2poster: Towards multimodal poster automation from scientific papers. *arXiv preprint arXiv:2505.21497*.

Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. 2021. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5865–5874.

Jingwei Shi, Zeyu Zhang, Biao Wu, Yanjie Liang, Meng Fang, Ling Chen, and Yang Zhao. 2025. Presentagent: Multimodal agent for presentation video generation. *arXiv preprint arXiv:2507.04036*.

Dominika Tkaczyk, Paweł Szostek, Mateusz Fedoryszak, Piotr Jan Dendek, and Łukasz Bolikowski. 2015. Cermine: automatic extraction of structured metadata from scientific literature. *International Journal on Document Analysis and Recognition (IJDAR)*, 18(4):317–335.

Edward R Tufte and Peter R Graves-Morris. 1983. *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT.

W3C Web Accessibility Initiative. 2018. Understanding success criterion 1.4.10: Reflow (wcag 2.x). Accessed 2025-10-06.

Jingyu Xiao, Yuxuan Wan, Yintong Huo, Zhiyao Xu, and Michael R Lyu. 2024. Interaction2code: How far are we from automatic interactive webpage generation? *arXiv e-prints*, pages arXiv–2411.

Zeyu Zhu, Kevin Qinghong Lin, and Mike Zheng Shou. 2025. Paper2video: Automatic video generation from scientific papers. *Preprint*, arXiv:2510.05096.