PosterForest: Hierarchical Multi-Agent Collaboration for Scientific Poster Generation

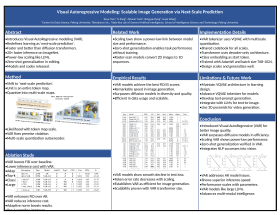
Jiho Choi[1*], **Seojeong Park**[1*], **Seongjong Song**[2], **Hyunjung Shim**[1†]
[1]Graduate School of Artificial Intelligence, KAIST, Republic of Korea
[2]School of Integrated Technology, Yonsei University, Republic of Korea
{jihochoi, seojeong.park, kateshim}@kaist.ac.kr, {bell}@yonsei.ac.kr

Abstract

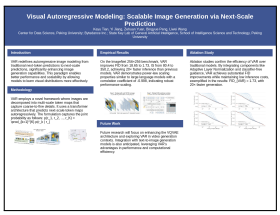
Automating scientific poster generation requires hierarchical document understanding and coherent content-layout planning. Existing methods often rely on flat summarization or optimize content and layout separately. As a result, they often suffer from information loss, weak logical flow, and poor visual balance. We present **PosterForest**, a training-free framework for scientific poster generation. Our method introduces the *Poster Tree*, a structured intermediate representation that captures document hierarchy and visual-textual semantics across multiple levels. Building on this representation, content and layout agents perform hierarchical reasoning and recursive refinement, progressively optimizing the poster from global organization to local composition. This joint optimization improves semantic coherence, logical flow, and visual harmony. Experiments show that PosterForest outperforms prior methods in both automatic and human evaluations, without additional training or domain-specific supervision. Code: https://github.com/kaist-cvml/poster-forest

1 Introduction

With the rapid advancement of science and technology, there has been an exponential increase (Hanson et al., 2024; Larsen and Von Ins, 2010) in the number of academic papers and technical reports with complex structures. As these documents are often difficult to interpret quickly, readers are required to invest significant time and cognitive resources to understand their main arguments. In this context, scientific posters have emerged as an effective medium for summarizing and presenting complex information in a concise and visually intuitive manner. By combining textual and visual elements, posters facilitate more accessible communication of technical content. However, manually crafting high-quality posters is a labor-intensive process that requires both domain knowledge and design expertise (Qiang et al., 2019; Wang et al., 2024). Automating scientific poster generation (SPG) is therefore a critical research problem, as it can accelerate the dissemination of specialized knowledge and reduce the burden on researchers.



(a) Paper2Poster



(b) P2P

Figure 1: **Limitations of Current SPG Methods.** Existing state-of-the-art scientific poster generation (SPG) methods, including P2P (Sun et al., 2025) and Paper2Poster (Pang et al., 2025), lack *hierarchical* document understanding, resulting in errors in both *content* and *layout*. (a) shows an example where an experiment table is incorrectly placed in the conclusion section. (b) illustrates an overly simplified poster, where paragraphs are merely summarized and assigned to fixed panels with fixed-sized figures.

Pioneering works such as PGM (Qiang et al., 2016, 2019), NCE (Xu and Wan, 2021), and PosterDoc (Jaisankar et al., 2024) approached automated poster generation by extracting text and figures from scientific documents and heuristically arranging them within poster panels. However, these approaches rely on fixed rules and struggle to handle the complexity of long, structured documents and the interplay between textual and visual content. To address this, more recent methods, including P2P (Sun et al., 2025) and Paper2Poster (Pang et al., 2025), adopt multi-agent pipelines that decompose the task into specialized sub-problems such as parsing, content summarization, layout planning, and rendering. This modular design improves flexibility and coordination across stages, but typically requires explicit model training, such as instruction

[*]Equal contribution
[†]Corresponding author

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 379–401

tuning or regressor-based optimization.

Despite these advances, current approaches suffer from several critical limitations. (1) **Shallow Document Understanding:** They primarily depend on surface-level text features, lacking a deep grasp of the hierarchical structure inherent to scientific documents and the semantic associations between textual and visual components. Consequently, they often exhibit an interrupted logical flow and weak integration of visual elements as in Figure 1 (a), ultimately reducing the effectiveness of posters in conveying information quickly and accurately. This limitation significantly increases users' cognitive load. (2) **Weak Content-Layout Integration:** Existing approaches often adopt a sequential pipeline in which the layout is determined before content placement. This decoupled strategy overlooks the intrinsic interdependence between content and layout, treating them as isolated components rather than pursuing their integrated organization. As a result, critical content may be truncated or misplaced, and the logical flow between textual and visual elements is frequently disrupted. Moreover, as shown in Figure 1 (b), this often results in posters that are overly simplified and fail to capture the complexity of the original document, diminishing their practical value. It diminishes the practical value of automated poster generation systems. (3) **Training Overhead:** Existing methods requiring instruction tuning or regression training pipelines add complexity and resource demands, limiting practical deployment.

In this study, we aim to address these limitations, which overlook both the hierarchical organization of scientific documents and the semantic alignment between content and layout. Such limitations hinder holistic understanding and often result in reduced clarity, visual incoherence, and discrepancies between visual elements and their explanatory context. To address these limitations, we present *PosterForest*, a novel framework with two core components. For limitation (1), we introduce the *Poster Tree*, a hierarchical intermediate representation. It prunes and merges document content across the section–subsection–paragraph hierarchy to preserve salient information while reducing redundancy, and explicitly links text with figures and tables. Each node jointly encodes **content** and **layout** attributes. This representation, tailored for scientific poster generation, preserves logical flow and strengthens text–visual associations.

For limitation (2), we propose hierarchical modification planning with multi-agent collaboration. Scientific poster generation requires balancing multiple interdependent objectives, such as content fidelity, layout efficiency, and visual coherence, which are difficult to optimize jointly within a single reasoning process. To address this, we decompose the task into specialized roles and employ multiple agents that focus on complementary aspects, including content summarization, layout planning, and visual material placement across different levels of the hierarchy. Through iterative coordination and feedback, these agents collaboratively refine the Poster Tree, enabling effective joint optimization of both content and structure. This results in visually balanced and semantically coherent posters with improved integration of textual and visual information. Finally, addressing limitation (3), the proposed pipeline is training-free and relies only on standard APIs and publicly available checkpoints, enabling practical deployment.

Overall, *PosterForest* delivers high-quality summarization, visual coherence, and consistent information flow, overcoming the limitations of prior methods. Extensive experiments show that our method consistently outperforms prior approaches in both automatic and human evaluations, achieving up to 59.2% preference in human studies (vs. 27.2% for prior work) while remaining training-free.

2 Related Work

Scientific Poster Generation. Although generic (i.e., movie, commercial) poster generation is a pervasively researched topic (Gupta et al., 2021; Li et al., 2020; Zheng et al., 2023; Inoue et al., 2023; Gao et al., 2025; Hsu and Peng, 2025), scientific poster generation is more challenging and crucial due to its deliverability of extensive information and reasoning. It has been studied as a layout-driven summarization problem (Qiang et al., 2016, 2019), where key elements such as panel size, position, and hierarchy are learned from examples. Earlier work, such as Xu and Wan (2021), emphasized the importance of content extraction, proposing a pipeline to select representative text and visuals. An instruction-tuning based P2P (Sun et al., 2025) and a regressor fitting-based Paper2Poster (Pang et al., 2025) are recent approaches of introducing LLM-based multi-agent frameworks to handle parsing, planning, and rendering in a modular way, extending the framework to generate slides such as

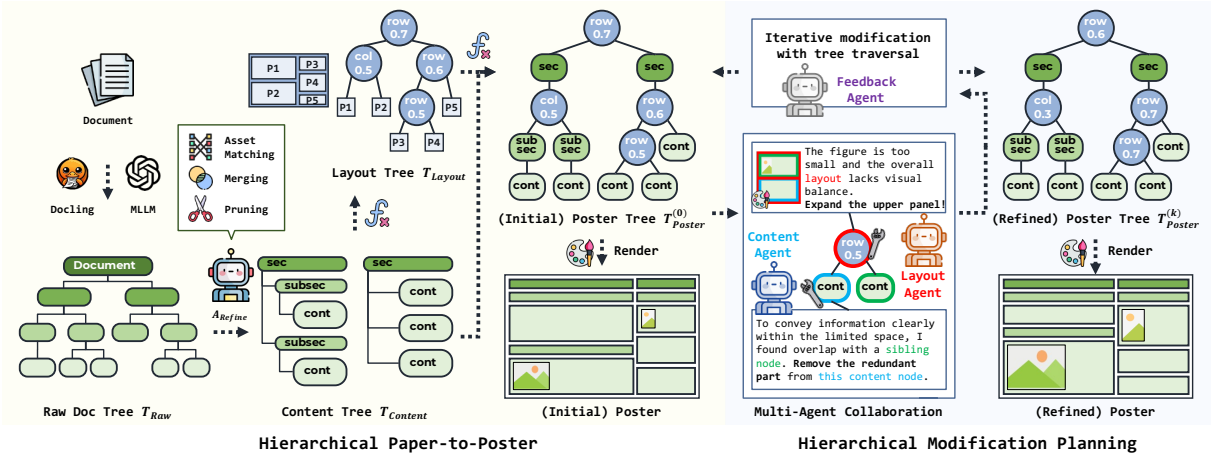


Figure 2: **Overview of PosterForest.** PosterForest first constructs a hierarchical Poster Tree that integrates document semantics and layout (Section 3.2), then iteratively refines it through collaboration between agents to optimize structure and visual coherence (Section 3.3).

PPTAgent (Zheng et al., 2025). These methods are supported by new benchmarks (Wang et al., 2024; Saxena et al., 2025) and evaluation protocols utilizing vision-language models (Lee et al., 2024), enabling fine-grained assessment of visual coherence and content fidelity. While these approaches achieve baseline-level automation, they often treat sections independently, simply mapping them to panels. As a result, the semantic flow and hierarchical connections across sections in the original paper are not well preserved in the generated poster.

Hierarchical Document Parsing and Understanding. Scientific poster generation requires a robust understanding of document structure and content, compared to naive rank-based text extraction (Mihalcea and Tarau, 2004). Early approaches to document understanding leveraged hierarchical parsing methods, such as DocParser (Rausch et al., 2021) and PDF-to-Tree (Zhang et al., 2024a), to recover logical section structures from rendered pages. These techniques enable semantic segmentation of long documents, which is crucial for downstream tasks like summarization and visualization. More recent works employ pre-trained multimodal models (Huang et al., 2022; Lin et al., 2023) or graph-based representations (Gemelli et al., 2022) to jointly model textual and visual elements. OWL (Hu et al., 2025) integrates multiple LLMs for reasoning and understanding documents. Such advances form the basis for extracting salient content needed for poster generation.

Multi-Agent Reasoning and Collaboration. Single-agent reasoning techniques such as Chain-of-Thought (CoT) (Wei et al., 2022) and its variations (Yao et al., 2023; Besta et al., 2024; Chen et al., 2022; Gao et al., 2023) enabled logical thinking of models, representatively academic (Team et al., 2024; Zhang et al., 2023) and mathematical (Shao et al., 2024) reasoning. Recent researches take a step forward and explore multi-agent collaboration to further enhance problem-solving capabilities. Instead of relying on a single reasoning path, multi-agent systems assign specialized roles to LLM agents and promote iterative feedback, critique, and coordination (Li et al., 2023; Zhang et al., 2024b; Hong et al., 2023; Tran et al., 2025; Li et al., 2024), successfully simulating collaborative software engineers (Qian et al., 2023), or peer-reviewers of scientific papers (Yu et al., 2024) even generating code from papers (Seo et al., 2025). In the poster generation, this paradigm enables specialized agents to perform document analysis, content summarization, and layout composition (Sun et al., 2025; Pang et al., 2025).

3 Proposed Method

3.1 Preliminaries

Recent advances in scientific poster generation (SPG), including P2P (Sun et al., 2025) and Paper2Poster (Pang et al., 2025), have introduced multi-agent pipelines that automatically synthesize posters from research papers. These methods leverage multimodal large language models (MLLMs), such as GPT-4 (Achiam et al., 2023) and Qwen (Bai et al., 2023), to extract textual and visual content, summarize key information, and organize it into structured panel layouts.

Among them, Paper2Poster adopts a modular approach. It comprises: (a) a parser that constructs an asset library of textual and visual elements, (b) a planner that matches text content with relevant figures and tables, and (c) a painter-commenter loop that iteratively refines contents inside the panel through vision-language feedback. P2P further leverages instruction tuning to enhance coordination among multiple agents, whereas Paper2Poster employs regressor-based learning to optimize content arrangement and visual composition.

Despite their effectiveness, existing approaches typically treat scientific papers as linear text sequences, disregarding structural relationships among textual units and semantic alignment between text and visuals. Consequently, they capture only shallow associations within and across modalities. Structural cues such as section and subsection boundaries, paragraph-level semantics, and cross-references to figures and tables are often underutilized or entirely ignored. These limitations result in logical discontinuities across the content and weakened correspondence among different elements, ultimately diminishing overall clarity and informativeness.

To address these limitations, we introduce **PosterForest**, a training-free framework for SPG. PosterForest operates in two main stages: (1) constructing a hierarchical **Poster Tree** that jointly encodes the document's semantic content and the poster's layout structure, and (2) iteratively refining this *Poster Tree* through multi-agent collaboration between specialized content and layout experts. The following sections provide a detailed description of each stage.

3.2 Hierarchical Paper-to-Poster

Given an input paper (or document) $\mathcal{D}$, we first parse it into a **Raw Doc Tree**, $\mathcal{T}_{\text{Raw}}$. We define $\mathcal{T}_{\text{Raw}} = (\mathcal{V}_{\text{Raw}}, \mathcal{E}_{\text{Raw}})$ as a rooted tree that represents the structural hierarchy of the document contents. Each node $v \in \mathcal{V}_{\text{Raw}}$ corresponds to a document element such as title, section, subsection, paragraph, figure, or table, which contains its raw semantic content. Each directed edge $(u \to v) \in \mathcal{E}_{\text{Raw}}$ denotes a parent–child relation, indicating that v is a subcomponent of u in the hierarchy. Let $\mathcal{A}_{\text{Parser}}$ be a parsing agent that extracts this structure as:

$$\mathcal{T}_{\text{Raw}} = \mathcal{A}_{\text{PARSE}}(D)\,. \tag{1}$$

The resulting tree explicitly captures both hierarchical organization and referential links, ensuring that figures and tables appear as children of the textual nodes that reference them.

We then refine $\mathcal{T}_{\text{Raw}}$ into a **Content Tree**, $\mathcal{T}_{\text{Content}} = (\mathcal{V}_{\text{Content}}, \mathcal{E}_{\text{Content}})$, which preserves the essential information to construct a scientific poster. Guided by the MLLM agent, this process involves *pruning* less important nodes, *merging* redundant or closely related content, and *summarizing* lengthy textual content, ensuring the resulting structure remains concise, coherent, and focused on key information. During this process, the node-edge relationships are updated to reflect the revised structure. Let $\mathcal{A}_{\text{Refine}}$ be a content refinement agent that performs these operations as:

$$\mathcal{T}_{\text{Content}} = \mathcal{A}_{\text{REFINE}}(\mathcal{T}_{\text{Raw}}). \tag{2}$$

In $\mathcal{T}_{\text{Content}}$, each node corresponds to a concise textual unit or a visual asset (e.g., figures and tables), with minor details removed. Each node $c \in \mathcal{V}_{\text{Content}}$ is represented as $c = (t, s)$ with $t \in \mathcal{T}_{\text{semantic}}$ denoting the semantic type (e.g., paragraph, figure, table) and $s \in \mathcal{S}_{\text{semantic}}$ denoting the semantic content (summarized text, caption, or visual data). This produces a compact and informative representation tailored for poster generation.

Based on the Content Tree, we establish a **Layout Tree**, $\mathcal{T}_{\text{Layout}} = (\mathcal{V}_{\text{Layout}}, \mathcal{E}_{\text{Layout}})$, which specifies the poster's spatial organization. The Layout Tree follows a widely adopted approach (Qiang et al., 2016, 2019; Pang et al., 2025) in poster layout modeling, where the canvas is hierarchically partitioned into regions organized by rows and columns. Unlike previous methods that derive such structures directly from the document layout, our Layout Tree is initialized from $\mathcal{T}_{\text{Content}}$, inheriting the hierarchical relationships among content elements, and aligning the layout with the intended content structure of the poster as:

$$\mathcal{T}_{\text{Layout}} = \mathcal{O}_{\text{LAYOUT\_INIT}}(\mathcal{T}_{\text{Content}}). \tag{3}$$

Each layout node $l \in \mathcal{V}_{\text{Layout}}$ is represented as $l = (r, x)$ with $r \in \mathcal{R}_{\text{spatial}}$ (region type: row split, column split, panel) and $x \in \mathcal{X}_{\text{spatial}}$ (spatial attributes: normalized position, width, height, aspect ratio). The initial allocation of regions is deterministically derived from content statistics.

Finally, we integrate content and layout into a unified representation, the **Poster Tree**, $\mathcal{T}_{\text{Poster}}$. We define $\mathcal{T}_{\text{Poster}} = (\mathcal{V}_{\text{Poster}}, \mathcal{E}_{\text{Poster}})$ by merging

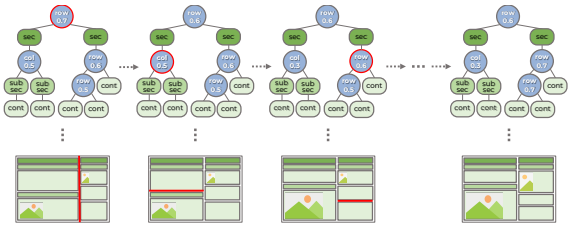


Figure 3: **Poster Tree Traversal (Node-level).** The Poster Tree and layout are iteratively updated through the shared decision of the layout and Content Agent.

the Content Tree, $\mathcal{T}_{\text{Content}}$, and the Layout Tree, $\mathcal{T}_{\text{Layout}}$, where each semantic node is mapped to a spatial region of the poster as:

$$\mathcal{T}_{\text{Poster}} = \mathcal{O}_{\text{MERGE}}\big(\mathcal{T}_{\text{Content}},\, \mathcal{T}_{\text{Layout}}\big). \quad (4)$$

Each poster node $w \in \mathcal{V}_{\text{Poster}}$ is a heterogeneous node that jointly encodes semantic attributes (e.g., a summarized paragraph or key visual element from $\mathcal{T}_{\text{Content}}$) and spatial attributes placement from $\mathcal{T}_{\text{Layout}}$. The merge operation aligns the hierarchical structure of the content with the corresponding layout partition, producing nodes that specify both *what* information is displayed and *where and how* it appears on the canvas. This unified tree representation provides an inductive bias tailored for poster generation, as it tightly couples the logical document hierarchy with the visual layout structure.

3.3 Hierarchical Modification Planning

After constructing the initial Poster Tree, $\mathcal{T}^{(0)}_{\texttt{poster}}$, we introduce a hierarchical refinement phase that is designed to jointly optimize both content quality and layout organization. In contrast to prior methods (Sun et al., 2025; Pang et al., 2025) that fix the layout and subsequently adjust only the content, our approach traverses the heterogeneous nodes in Poster Tree and performs node-specific updates by leveraging both local attributes and hierarchical context, while further incorporating global evaluation to achieve a more coherent and polished final result.

3.3.1 Poster Tree Traversal (Node-level)

Given the initialized Poster Tree, $\mathcal{T}^{(0)}_{\text{Poster}}$, refinement begins through a hierarchical traversal from the root toward the leaves. Each node is updated by jointly considering its intrinsic attributes, the propagated information from its parent, and the structural context defined by its descendants. This top-down propagation ensures that modifications applied at higher levels are consistently reflected throughout the tree.

Layout Agent. For each layout node $l_i \in \mathcal{V}_{\text{Poster}}$, the Layout Agent, $\mathcal{A}_{\text{LAYOUT}}$ optimizes geometric attributes such as region ratios, alignment, and spatial distribution. The optimization is performed by aggregating the structural and semantic statistics of all descendant nodes as:

$$l_i^{(t+1)} = \mathcal{A}_{\text{LAYOUT}}\big(l_i^{(t)},\, \tilde{\mathcal{P}}(l_i^{(t)}),\, \mathcal{D}(l_i^{(t)})\big), \quad (5)$$

where $\tilde{\mathcal{P}}(l_i)$ represents the updated information of the parent node after its refinement, and $\mathcal{D}(l_i)$ denotes the set of all descendants of l_i.

Content Agent. For each content node $c_i \in \mathcal{V}_{\text{Poster}}$, $\mathcal{A}_{\text{CONTENT}}$ refines textual density and semantic abstraction by referencing both the updated configuration of its parent layout node and the descendant layout context of that parent:

$$c_i^{(t+1)} = \mathcal{A}_{\text{CONTENT}}\big(c_i^{(t)},\, \tilde{\mathcal{P}}(c_i^{(t)}),\, \mathcal{D}(\mathcal{P}(c_i^{(t)}))\}\big), \quad (6)$$

where $\tilde{\mathcal{P}}(c_i)$ represents the updated information of the parent node after its refinement, and $\mathcal{D}(\mathcal{P}(c_i))$ denotes the set of all descendants of the parent node of c_i. This hierarchical dependency allows local content updates to reflect global layout constraints and parent-level refinements.

The traversal proceeds until every node has been updated once, yielding an intermediate tree $\mathcal{T}^{(t+1)}_{\text{Poster}}$ that captures coherent modifications across both spatial and semantic dimensions. The resulting representation serves as the basis for subsequent global evaluation.

3.3.2 Iterative Tree Refinement (Tree-level)

After completing full node-level traversal, PosterForest performs iterative refinement at the tree level to progressively enhance the overall poster structure. Each tree-level iteration corresponds to a complete pass of node-level updates, resulting in a refined Poster Tree that jointly enhances semantic clarity and spatial organization. This process may be repeated up to a maximum of K iterations to incrementally enhance layout quality and content coherence. In practice, we set $K = 2$, which empirically yields stable and visually balanced results with a single additional refinement step.

Global Feedback Agent. At each iteration t, the rendering of the current Poster Tree $\mathcal{T}^{(t)}_{\text{Poster}}$ is evaluated by a multimodal large language model

(MLLM) acting as a Global Feedback Agent, denoted as $\mathcal{A}_{\text{FEEDBACK}}$. This agent analyzes the poster's visual organization, textual structure, and hierarchical balance, and provides structured global feedback to determine whether an additional tree-level traversal should be executed:

$$[\hat{\mathcal{F}}^{(t)}_{\text{GLOBAL}}, \pi^{(t)}_{\text{CONTINUE}}] = \mathcal{A}_{\text{FEEDBACK}}(\mathcal{T}^{(t)}_{\text{Poster}}), \tag{7}$$

where $\hat{\mathcal{F}}^{(t)}_{\text{GLOBAL}}$ denotes the structured global feedback extracted from the MLLM, and $\pi^{(t)}_{\text{CONTINUE}} \in \{0, 1\}$ is a binary signal indicating whether another refinement iteration should be performed.

If $\pi^{(t)}_{\text{CONTINUE}} = 1$, the next tree-level traversal is triggered using the propagated feedback:

$$\mathcal{T}^{(t+1)}_{\text{Poster}} = \mathcal{O}_{\text{TRAVERSE}}(\mathcal{T}^{(t)}_{\text{Poster}}, \hat{\mathcal{F}}^{(t)}_{\text{GLOBAL}}), \tag{8}$$

where $\mathcal{O}_{\text{TRAVERSE}}$ denotes one complete pass of the propagation-based node-level refinement defined in Equation (5) and Equation (6). Otherwise, the iterative refinement loop terminates, and the final Poster Tree is obtained as $\mathcal{T}^{*}_{\text{Poster}} = \mathcal{T}^{(t)}_{\text{Poster}}$.

4 Experiments

4.1 Experimental Setup

Baselines. Following the evaluation protocols of P2P (Sun et al., 2025) and Paper2Poster (Pang et al., 2025), we compare four categories of baseline methods. First, `Oracle` methods represent upper bounds. The original `Paper` represents the upper bound for content fidelity, while the author-created `GT Poster` indicates the optimal layout and clarity achievable by human experts. Second, end-to-end methods employ GPT-4o to generate posters directly. Specifically, `GPT-4o-HTML` renders posters by converting the paper into HTML, whereas `GPT-4o-Image` produces poster images in a single step using GPT-4o. Third, multi-agent workflows encompass general-purpose converters and algorithmic generators. For this category, we evaluate the PDF-to-HTML conversion toolkit of `OWL` (Hu et al., 2025) and Python-pptx conversion results of `PPTAgent`. Finally, poster-specialized agents include `P2P` (Sun et al., 2025), Paper2Poster, and our proposed method. To ensure that visual factors did not influence qualitative evaluations and user studies, we standardized the color scheme and font across all posters.

Datasets. For quantitative evaluation, we used the 100 paper–poster pairs provided by the Paper2Poster benchmark (Pang et al., 2025), which is an extension of the PosterSum dataset (Saxena et al., 2025). For qualitative results and user studies, we additionally collected 15 recent paper–poster pairs from the AI conferences (e.g., NeurIPS, CVPR, and ACL). More comprehensive experimental details are provided in the Appendix.

Implementation Details. For further details regarding model architecture and evaluation protocols, please refer to the Appendix.

Model	TF	Aesthetic Score ↑				Information Score ↑				Overall ↑
		Elem.	Lay.	Eng.	Avg.	Cla.	Cont.	Logic	Avg.	
Paper	-	4.05	3.89	2.80	3.58	4.00	4.68	3.98	4.22	3.90
GT Poster	-	4.07	3.90	2.70	3.56	4.09	3.96	3.89	3.98	3.77
4o-HTML	✔	3.53	3.82	2.72	3.36	3.94	3.64	3.47	3.68	3.52
4o-Image	✔	2.93	3.02	2.75	2.90	1.05	2.04	2.22	1.77	2.33
OWL-4o	✔	2.76	3.62	2.56	2.98	3.92	2.89	3.36	3.39	3.19
PPTAgent-4o	✔	2.49	3.05	2.45	2.66	2.05	1.26	1.38	1.56	2.11
P2P-4o	✘	3.63	**4.01**	2.96	**3.91**	3.80	**3.99**	3.48	**3.94**	3.72
PosterAgent-Qwen	✘	3.93	3.67	2.89	3.50	3.95	3.85	3.68	3.83	3.66
PosterAgent-4o	✘	3.95	3.86	2.93	3.58	**4.03**	3.96	3.60	3.86	3.72
PosterForest-Qwen (Ours)	✔	**4.02**	3.85	**2.99**	3.62	3.98	3.93	3.54	3.82	3.72
PosterForest-4o (Ours)	✔	**4.02**	3.96	2.96	3.65	4.00	3.88	**3.71**	3.87	**3.76**

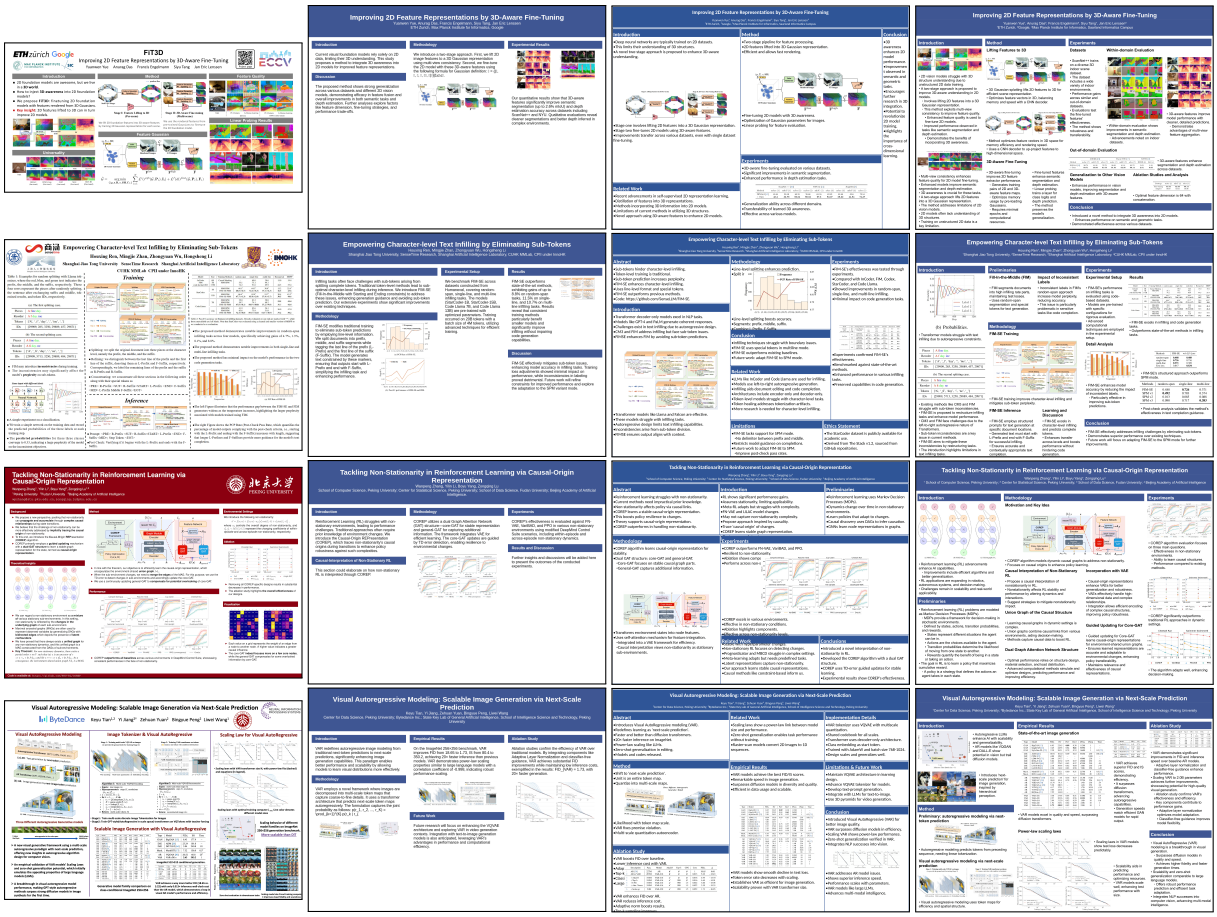
Table 1: **MLLM-as-a-Judge score across four categories of baselines.** The average score serves as a fine-grained assessment of 6 different perspectives. The best score is **bold**, and the second is underlined for each criterion. "TF" denotes *Training-free* methods.

Method	Content	Esthetics	Structure	Overall
4o-HTML	2.0 %	1.6 %	2.4 %	1.6 %
P2P	9.2 %	21.2 %	13.2 %	12.0 %
Paper2Poster	32.8 %	24.0 %	24.8 %	27.2 %
Ours	**56.0 %**	**53.2 %**	**59.6 %**	**59.2 %**

Table 2: **Human Evaluation.** Numbers represent the proportion of times each method was ranked first for each criterion.

4.2 Qualitative Evaluation

In Figure 4, we compare our method with state-of-the-art baselines, P2P and Paper2Poster, and the poster made by the authors (Ground Truth). All experiments were conducted using the GPT-4o framework to ensure consistency in model performance. Our approach dynamically adjusts column widths and panel sizes, resulting in a balanced distribution of content. Compared to P2P and Paper2Poster, our method makes more efficient use of space, prevents the inclusion of oversized or undersized figures, and avoids excessively long or verbose paragraphs through strategic hierarchical organization. Notably, our method excels at preserving information: for the VAR paper (4th row), both P2P and Paper2Poster omit either the result table or graph, whereas our method retains both, ensuring that critical information is maintained. Furthermore, our approach demonstrates robust performance across diverse paper formats and academic domains, as illustrated by the examples from 3D Vision-ECCV



(a) GT (b) P2P (c) Paper2Poster (d) PosterForest (Ours)

Figure 4: **Qualitative Comparison.** Posters generated by the SoTA baseline methods and *PosterForest*, based on papers from various AI fields (NLP, CV, RL), along with the original posters (GT) designed by the authors.

(1st row), Language Processing-ACL (2nd row), and Reinforcement Learning-ICML (3rd row).

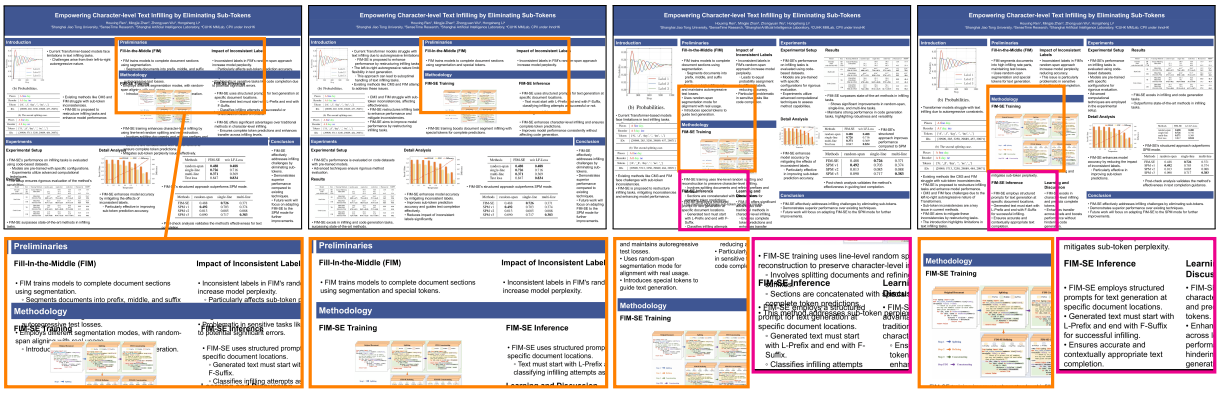
4.3 Quantitative Evaluation

Following the evaluation protocol introduced in P2P and Paper2Poster, we employ MLLM-as-a-Judge metrics for quantitative evaluation. The GPT-4o model is prompted to act as six independent judges, each assigning a score from 1 to 5 based on the following criteria: element quality, layout balance, engagement, clarity, content completeness, and logical flow. The first three criteria assess aesthetics, while the latter three evaluate the informativeness of the generated poster. As shown in Table 1, the judges indicate that our method is comparable to other baselines in terms of aesthetics, and demonstrates superior performance in informativeness. Importantly, these scores are the closest to those of the author-created ground truth posters (GT), demonstrating the effectiveness of our approach. Details of the MLLM-as-a-Judge are provided in the supplementary material.

While MLLM-based evaluation provides a scalable and objective means for poster assessment, it still has inherent limitations in fully capturing subjective preferences and subtle qualities valued by human readers. Therefore, to complement the quantitative results, we further conduct a user study to obtain human judgments and validate the practical effectiveness of our method.

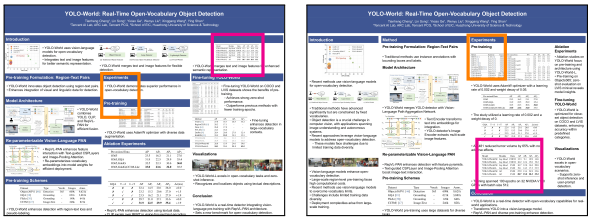
4.4 User Study

To conduct a user study to evaluate poster quality from a human perspective, we recruited 25 participants, all of whom were graduate students in the field of AI and had participated in scientific conferences. The study uses 10 sets (40 questions in total), each consisting of a group of posters and four evaluation questions. Each poster group is generated with four GPT-4o-based methods: 4o-HTML, P2P, Paper2Poster, and our proposed method. For each set, participants are asked to select one poster per



(a) Base (b) Only Content Agent (c) Only Layout Agent (d) Both Agents

Figure 5: **Effect of Content and Layout Agents.** Using both agents balances layout (orange) and removes redundancy (magenta), yielding well-organized posters with proper information density and strong visual harmony.



```
Root
 |-- Section 1
 |-- Subsection 1.1
 |-- Section 2
 |-- Subsection 2.1
 |-- Subsection 2.2
```

```
Root
 |-- Section 1
 |    `-- Subsection 1.1
 `-- Section 2
      |-- Subsection 2.1
      `-- Subsection 2.2
```

(a) w/o Hierarchical (b) w/ Hierarchical

Figure 6: **Effect of Hierarchical Content Tree.** With a hierarchical structure, logical order and spatial coherence are preserved (orange) and text–visual alignment improves; the performance table appears under Experiments rather than Introduction (magenta).

question based on the following criteria: (1) *content fidelity*, which poster best reflects the content of the paper; (2) *aesthetic quality*, which poster is the most visually harmonious; (3) *structural clarity*, which poster delivers information in the most structurally effective way; and (4) *overall quality*, which poster appears most complete and well-polished overall. As shown in Table 2, our proposed method is strongly preferred over the other SoTA baselines across all four criteria. Please refer to the Appendix for further details on the user study.

4.5 Ablation Study

4.5.1 Effect of Hierarchical Content Tree

We conducted an ablation study to analyze the impact of incorporating hierarchical structure into the Content Tree, $\mathcal{T}_{\texttt{content}}$, during content generation. When the hierarchical organization is omitted, as shown in Figure 6 (a), sections and subsections are often disordered or mixed together on the poster. This results in a loss of semantic grouping and spatial coherence between related elements such as figures and explanatory text. In contrast, applying hierarchical parsing, as in Figure 6 (b), preserves the logical relationships between sections and subsections, ensuring that related content is grouped and displayed in a consistent and interpretable manner. This hierarchical structure enhances both the readability and spatial cohesion of the generated poster, supporting more effective information delivery.

4.5.2 Effect of Content and Layout Agents

To evaluate the effectiveness of the multi-agent collaboration, we conducted an ablation study by comparing four configurations: (a) the base model with only the initial Poster Tree, (b) Content Agent only, (c) Layout Agent only, and (d) both agents combined. The Content Agent prunes cross-node redundancy and right-sizes the remaining text to the layout, as shown in Figure 5(b), but often leads to suboptimal panel arrangements and unbalanced layouts. In contrast, the Layout Agent focuses on optimizing the spatial arrangement at the layout level. As illustrated in Figure 5 (c), this configuration achieves improved visual organization, but suboptimal figure scaling and text overflow frequently

occur due to the lack of content adjustment. When both agents are used together, as shown in Figure 5 (d), the system effectively addresses both content redundancy and layout imbalance, producing well-organized posters with appropriate information density and visual harmony. These results confirm that the joint use of Content and Layout Agents is essential for achieving both semantic and structural quality in automated poster generation.

5 Conclusion

This work proposed PosterForest, a hierarchical multi-agent framework for scientific poster generation that explicitly models the interplay between document structure and layout design. The proposed Poster Tree serves as a unified intermediate representation, enabling integrated reasoning over semantic and spatial attributes. Through hierarchical refinement driven by Content and Layout Agents, our method dynamically balances information density and visual harmony without any training or dataset-specific tuning. Empirical results and user studies confirm that PosterForest substantially outperforms prior methods in informativeness, clarity, and structural quality.

Limitations

While PosterForest demonstrates significant improvements, certain limitations persist. First, generated posters may not always achieve optimal content density, which can lead to less efficient space utilization. Second, the lack of robust quality metrics may limit the comprehensiveness of quantitative evaluation, highlighting the need for further development of advanced evaluation methodologies in future research. Detailed failure cases and future directions are provided in the supplementary material.

Ethical Considerations

We rely only on officially released, publicly accessible models and APIs. In all experiments we call GPT-4o through OpenAI's official interface, and we also use publicly available Qwen checkpoints where noted. We do not fine-tune any models. Our framework is training-free and used strictly within the terms of the providers' licenses. Source papers and posters are analyzed solely for non-commercial research under practices consistent with academic fair use, and we include references to the original sources to respect creator rights. Our human evaluation recruits 25 graduate-level participants with prior conference experience; assessments concern posters only, and no personal attributes are collected or analyzed. An AI assistant was used for sentence-level drafting and refining to improve clarity.

Acknowledgments

This work was supported by Samsung Research, Samsung Electronics Co., Ltd.; the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Korea government (MSIT) (No. RS-2025-00520207); Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (Nos. RS-2024-00457882, 2022-0-01045, 2022-0-00680); and a grant partly supported by both IITP (MSIT) and Korea Evaluation Institute of Industrial Technology (KEIT) (MOTIE) (No. RS-2025-02217259).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and 1 others. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 17682–17690.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Yifan Gao, Zihang Lin, Chuanbin Liu, Min Zhou, Tiezheng Ge, Bo Zheng, and Hongtao Xie. 2025. Postermaker: Towards high-quality product poster generation with accurate text rendering. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8083–8093.

Andrea Gemelli, Sanket Biswas, Enrico Civitelli, Josep Lladós, and Simone Marinai. 2022. Doc2graph: a task agnostic document understanding framework based on graph neural networks. In *European Conference on Computer Vision*, pages 329–344. Springer.

Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry S Davis, Vijay Mahadevan, and Abhinav Shrivastava. 2021. Layouttransformer: Layout generation and completion with self-attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1004–1014.

Mark A Hanson, Pablo Gómez Barreiro, Paolo Crosetto, and Dan Brockington. 2024. The strain on scientific publishing. *Quantitative Science Studies*, 5(4):823–843.

Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, and 1 others. 2023. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 3(4):6.

HsiaoYuan Hsu and Yuxin Peng. 2025. Postero: Structuring layout trees to enable language models in generalized content-aware layout generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8117–8127.

Mengkang Hu, Yuhang Zhou, Wendong Fan, Yuzhou Nie, Bowei Xia, Tao Sun, Ziyu Ye, Zhaoxuan Jin, Yingru Li, Qiguang Chen, and 1 others. 2025. Owl: Optimized workforce learning for general multi-agent assistance in real-world task automation. *arXiv preprint arXiv:2505.23885*.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM international conference on multimedia*, pages 4083–4091.

Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. 2023. Layoutdm: Discrete diffusion model for controllable layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10167–10176.

Vijay Jaisankar, Sambaran Bandyopadhyay, Kalp Vyas, Varre Chaitanya, and Shwetha Somasundaram. 2024. Postdoc: Generating poster from a long multimodal document using deep submodular optimization. *arXiv preprint arXiv:2405.20213*.

Peder Larsen and Markus Von Ins. 2010. The rate of growth in scientific publication and the decline in coverage provided by science citation index. *Scientometrics*, 84(3):575–603.

Seongyun Lee, Seungone Kim, Sue Park, Geewook Kim, and Minjoon Seo. 2024. Prometheus-vision: Vision-language model as a judge for fine-grained evaluation. In *Findings of the association for computational linguistics ACL 2024*, pages 11286–11315.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36:51991–52008.

Jianan Li, Jimei Yang, Jianming Zhang, Chang Liu, Christina Wang, and Tingfa Xu. 2020. Attribute-conditioned layout gan for automatic graphic design. *IEEE Transactions on Visualization and Computer Graphics*, 27(10):4039–4048.

Long Li, Weiwen Xu, Jiayan Guo, Ruochen Zhao, Xingxuan Li, Yuqian Yuan, Boqiang Zhang, Yuming Jiang, Yifei Xin, Ronghao Dang, and 1 others. 2024. Chain of ideas: Revolutionizing research via novel idea development with llm agents. *arXiv preprint arXiv:2410.13185*.

Jiawei Lin, Jiaqi Guo, Shizhao Sun, Zijiang Yang, Jian-Guang Lou, and Dongmei Zhang. 2023. Layoutprompter: Awaken the design ability of large language models. *Advances in Neural Information Processing Systems*, 36:43852–43879.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

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*.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, and 1 others. 2023. Chatdev: Communicative agents for software development. *arXiv preprint arXiv:2307.07924*.

Yu-Ting Qiang, Yan-Wei Fu, Xiao Yu, Yan-Wen Guo, Zhi-Hua Zhou, and Leonid Sigal. 2019. Learning to generate posters of scientific papers by probabilistic graphical models. *Journal of Computer Science and Technology*, 34(1):155–169.

Yuting Qiang, Yanwei Fu, Yanwen Guo, Zhi-Hua Zhou, and Leonid Sigal. 2016. Learning to generate posters of scientific papers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Johannes Rausch, Octavio Martinez, Fabian Bissig, Ce Zhang, and Stefan Feuerriegel. 2021. Docparser: Hierarchical document structure parsing from renderings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4328–4338.

Rohit Saxena, Pasquale Minervini, and Frank Keller. 2025. Postersum: A multimodal benchmark for scientific poster summarization. *arXiv preprint arXiv:2502.17540*.

Minju Seo, Jinheon Baek, Seongyun Lee, and Sung Ju Hwang. 2025. Paper2code: Automating code generation from scientific papers in machine learning. *arXiv preprint arXiv:2504.17192*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Tao Sun, Enhao Pan, Zhengkai Yang, Kaixin Sui, Jiajun Shi, Xianfu Cheng, Tongliang Li, Wenhao Huang, Ge Zhang, Jian Yang, and 1 others. 2025. P2p: Automated paper-to-poster generation and fine-grained benchmark. *arXiv preprint arXiv:2505.17104*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, and 1 others. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D Nguyen. 2025. Multi-agent collaboration mechanisms: A survey of llms. *arXiv preprint arXiv:2501.06322*.

Hao Wang, Shohei Tanaka, and Yoshitaka Ushiku. 2024. Scipostlayout: A dataset for layout analysis and layout generation of scientific posters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8136–8141.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Sheng Xu and Xiaojun Wan. 2021. Neural content extraction for poster generation of scientific papers. *arXiv preprint arXiv:2112.08550*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Haofei Yu, Zhaochen Hong, Zirui Cheng, Kunlun Zhu, Keyang Xuan, Jinwei Yao, Tao Feng, and Jiaxuan You. 2024. Researchtown: Simulator of human research community. *arXiv preprint arXiv:2412.17767*.

Yue Zhang, Zhihao Zhang, Wenbin Lai, Chong Zhang, Tao Gui, Qi Zhang, and Xuan-Jing Huang. 2024a. Pdf-to-tree: Parsing pdf text blocks into a tree. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10704–10714.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Arik. 2024b. Chain of agents: Large language models collaborating on long-context tasks. *Advances in Neural Information Processing Systems*, 37:132208–132237.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2023. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*.

Zhiyuan Zhao, Hengrui Kang, Bin Wang, and Conghui He. 2024. Doclayout-yolo: Enhancing document layout analysis through diverse synthetic data and global-to-local adaptive perception. *arXiv preprint arXiv:2410.12628*.

Guangcong Zheng, Xianpan Zhou, Xuewei Li, Zhongang Qi, Ying Shan, and Xi Li. 2023. Layoutdiffusion: Controllable diffusion model for layout-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22490–22499.

Hao Zheng, Xinyan Guan, Hao Kong, Jia Zheng, Weixiang Zhou, Hongyu Lin, Yaojie Lu, Ben He, Xianpei Han, and Le Sun. 2025. Pptagent: Generating and evaluating presentations beyond text-to-slides. *arXiv preprint arXiv:2501.03936*.

Supplementary Material

A Additional Qualitative Results

Additional results for scientific poster generation are presented in Figure A1 and Figure A2.

B Experimental Details

B.1 Qualitative Experiments Setup

Standardization. To ensure a fair comparison across different methods, all posters were generated with a standardized width of 48 inches and height of 36 inches. In addition, the color schemes and font styles were unified for all posters, eliminating visual biases related to design choices. These controls allow the qualitative evaluation to focus solely on the content and layout quality produced by each model.

B.2 Quantitative Experiments Setup

Dataset Details. For all quantitative experiments, we use the Paper2Poster benchmark (Pang et al., 2025), which is the first large-scale dataset for scientific poster generation. The benchmark consists of 100 paper–poster pairs curated from recent AI conferences, including NeurIPS, ICML, and ICLR (2022–2024). Each pair includes a full-length research paper and its corresponding author-designed poster, enabling rigorous evaluation of poster generation models.

The dataset ensures high quality and diversity by selecting papers from a range of AI subfields such as computer vision, natural language processing, and reinforcement learning. The input papers average 22.6 pages and 12,156 words, with an average of 22.6 figures per paper. The corresponding posters contain about 774 words and 8.7 figures on average, resulting in a significant compression of both textual and visual content.

To avoid data leakage, only the test split of the POSTERSUM dataset (Saxena et al., 2025) was used when curating the benchmark. Papers were filtered to include the latest camera-ready versions and to ensure a broad distribution across years and venues. This benchmark provides a challenging setting for evaluating poster generation methods in terms of both informativeness and layout quality.

B.3 Implementation Details

All experiments were conducted on a server running Ubuntu 22.04 LTS, equipped with an AMD EPYC 7543 CPU and eight NVIDIA RTX A6000 GPUs (48 GB each). Four GPUs were allocated to Qwen-2.5-7B-Instruct and four to Qwen2.5-VL-7B-Instruct, using PyTorch 2.6.0. Random seeds were fixed for Python, NumPy, and PyTorch to ensure experimental reproducibility.

B.4 Hyperparameters

The maximum number of tree refinement iterations was set to $T_{\texttt{max}} = 2$. Pilot studies showed that additional exchanges between agents did not yield significant improvements. These hyperparameters were chosen to balance generation quality and computational efficiency.

B.5 Detailed Baseline Setup

For figure extraction in P2P, we adopted DocLayout-YOLO (Zhao et al., 2024). The PosterAgent model in Paper2Poster was implemented in two variants: one utilizing `GPT-4o` and the other employing open-source Qwen models, specifically Qwen-2.5-7B and Qwen-2.5-VL-7B.

C Evaluation Metric and Rationale

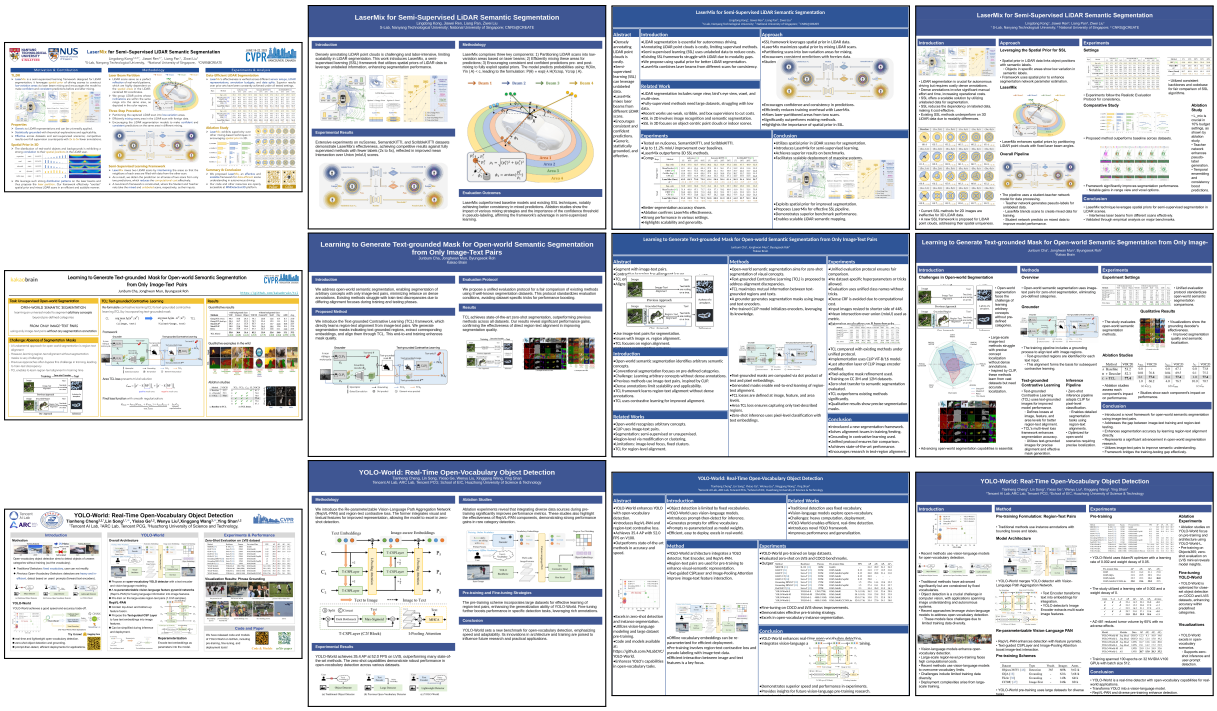
C.1 Reason for Metric Selection

Our evaluation metrics were selected to comprehensively assess the effectiveness of scientific poster generation from multiple perspectives. Human evaluation was conducted using four criteria: content fidelity, visual harmony, structural effectiveness, and overall completeness. These metrics reflect the need for posters to accurately summarize findings, present information attractively and logically, and maintain a professional appearance.

Although MLLM-as-a-judge metrics (e.g., GPT-4o) enable fine-grained assessment of informativeness and aesthetics (Pang et al., 2025), they have limitations. Discrepancies between ground truth and generated posters are often better detected by human observers than by automated metrics. However, with advances in multimodal evaluation, MLLM-based metrics are expected to become more reliable.

C.2 User Study Details

We recruited 25 graduate students with experience in academic conferences. Each participant evaluated 10 pairs of papers and posters generated by four methods: GPT-4o HTML, P2P (Sun et al., 2025), Paper2Poster (Pang et al., 2025), and PosterForest. Posters were standardized for font and color.



(a) GT (b) P2P (c) Paper2Poster (d) Ours

Figure A1: **Additional Qualitative Comparison [1/2].** Posters generated with the GPT-4o framework of baseline methods and *PosterForest*, based on papers spanning different AI conferences, along with the original posters (GT) created by the authors.

Participants viewed the original poster and four generated posters in randomized order, judging based on content fidelity, aesthetic quality, structural clarity, and overall quality.

D Limitations and Future Work

D.1 Failure Cases

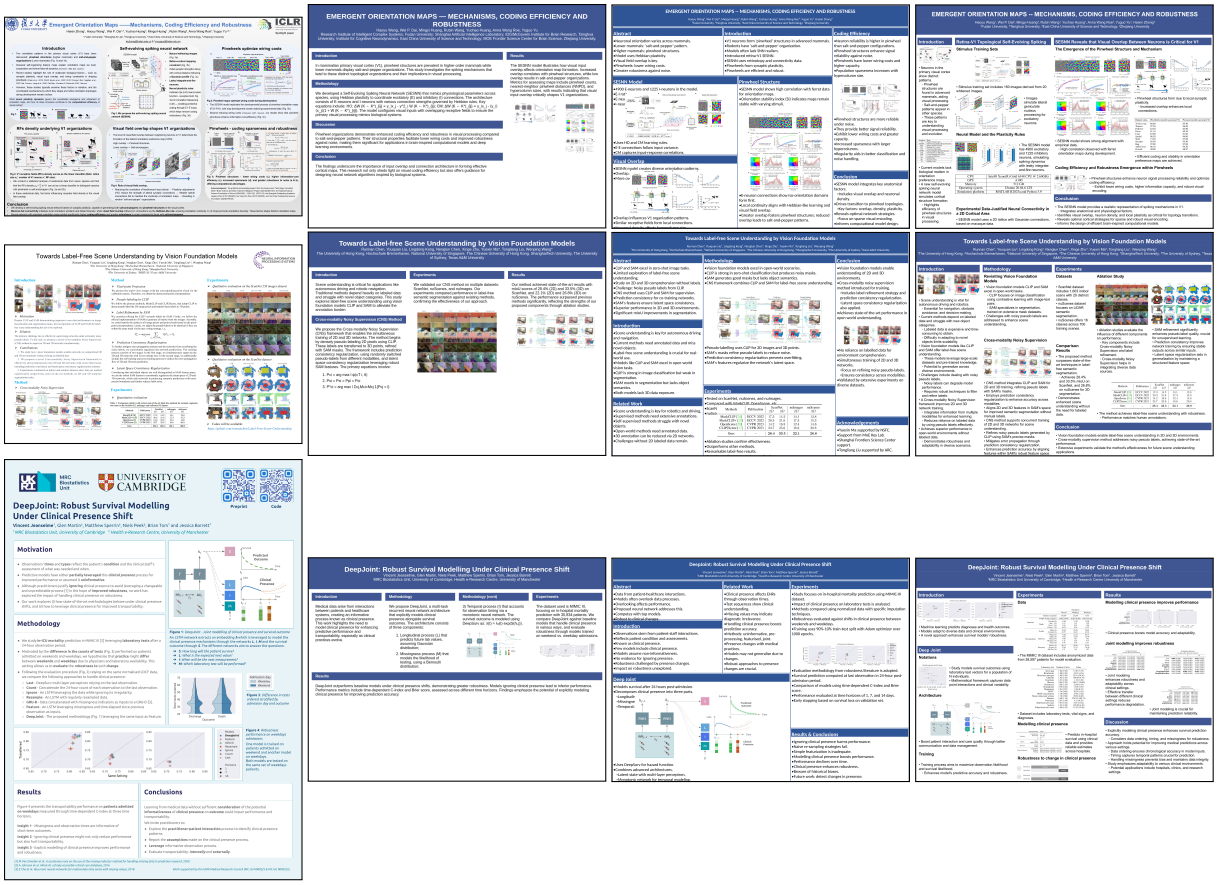
While our hierarchical approach improves global understanding of paper structure, there remain notable limitations compared to human designers. In particular, our framework struggles when processing papers containing a large number of figures or tables, especially when such assets are densely clustered or heavily interleaved within the text. These scenarios often result in errors during asset parsing and content matching, which can propagate through subsequent stages of the pipeline. As a result, missing or misaligned visual elements accumulate, further degrading the informativeness and visual coherence of the generated posters.

Figure A3a illustrates an example paper in which many figures are densely arranged within a short span of text. As shown in Figure A3b, the generated poster for this case exhibits clear failures, with several figures missing or incorrectly matched to their respective sections. This example demonstrates how dense and complex visual layouts present a significant challenge for the current parsing and matching pipeline, ultimately limiting the quality of the generated output.

While improving the performance of parsers such as Docling is not the primary focus of this work, enhancing asset parsing and extraction remains an important avenue for future research. Advances in these components could further improve the overall robustness and quality of the poster generation framework, especially when handling papers with complex and densely interleaved visual elements.

D.2 Future Directions

From a modeling perspective, future work will focus on enhancing the framework's ability to robustly parse and reason over complex layouts, particularly in papers with densely interleaved figures and tables. Incorporating more advanced hierarchical modeling and visual-semantic alignment techniques may further narrow the gap with human-designed posters. For example, human designers intuitively adjust the placement and scale of figures based on factors such as the amount of information



(a) GT (b) P2P (c) Paper2Poster (d) Ours

Figure A2: **Additional Qualitative Comparison [2/2].** Posters generated with the GPT-4o framework of baseline methods and *PosterForest*, based on papers spanning different AI conferences, along with the original posters (GT) created by the authors.

conveyed, perceived importance, and the context of each figure, as well as considerations like font size and spatial balance. However, our current framework does not yet fully capture the relative importance or semantic richness of individual figures and their optimal integration within the layout. Modeling these nuanced factors in future work could enable more practical and human-aligned poster generation.

From the perspective of evaluation, we plan to build upon the metrics introduced in P2P (Sun et al., 2025) and Paper2Poster (Pang et al., 2025) to develop more reliable and comprehensive automated metrics for poster quality assessment. Improving automated evaluation protocols is crucial for accurately measuring progress and guiding future research in scientific poster generation.

E Additional Materials

E.1 Parser Agent, $\mathcal{A}_{\texttt{Parser}}$.

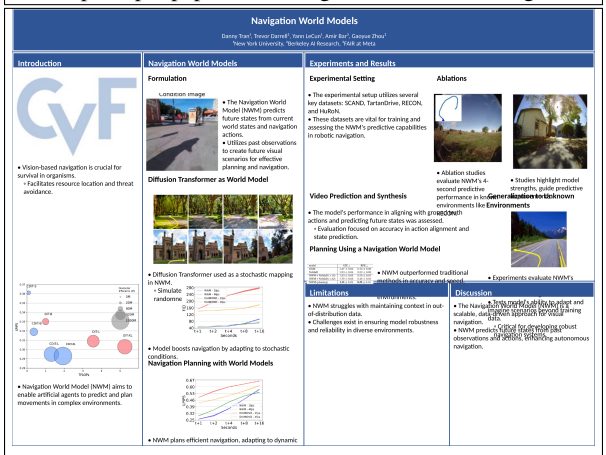
The Parser Agent parses the raw scientific document into a hierarchical tree structure, organizing components such as titles, sections, subsections, paragraphs, and visual assets for further processing by other agents. An example prompt is provided in Figure A4.

Collaborative Poster Optimization. This criterion assesses the overall arrangement, alignment, and spacing of text and graphics to ensure a coherent and readable poster structure.

E.2 Summarizing Agent, $\mathcal{A}_{\texttt{Summ}}$.

The Summarizing Agent is responsible for pruning, merging, and summarizing essential information from each section, subsection, and paragraph of the document. It removes nonessential details to improve clarity and density, producing a content

Example input paper containing a dense cluster of figures.



Generated poster with missing or misaligned visual elements due to parsing failures.

Figure A3: An example of a failure case arising from clustered figures in the input paper. The parser fails to correctly extract and match all figures, leading to incomplete or incorrectly structured visual content in the poster.

tree optimized for poster presentation. An example prompt is provided in Figure A5.

E.3 Content Agent, $\mathcal{A}_{\texttt{Content}}$.

The Content Agent evaluates and refines the quality and clarity of textual content at each node of the Poster Tree. It adjusts text volume, resolves redundancy, and selects appropriate information for each panel in collaboration with the Layout Agent. An example prompt is provided in Figure A6.

E.4 Layout Agent, $\mathcal{A}_{\texttt{Layout}}$.

The Layout Agent assesses and adjusts the spatial arrangement, visual balance, and aspect ratio of each panel. It collaborates with the Content Agent to ensure appropriate allocation of space and a visually coherent poster layout. An example prompt is provided in Figure A7.

E.5 Feedback Agent, $\mathcal{A}_{\texttt{Feedback}}$.

The feedback Agent reviews the rendered poster tree and judges three aspects—*visual organization*, *textual structure*, and *hierarchical balance*. It returns concise, structured feedback and a binary `continue` decision indicating whether another tree-level refinement should proceed. An example prompt is provided in Figure A8.

Parser Agent [1/2]

You are a document structure analysis expert who creates hierarchical tree representations of academic papers for poster generation.

Given a markdown document and extracted figure/table captions, analyze the structure and create a hierarchical tree representation focusing on the logical flow guided by visual elements.

Caption-Guided Strategy:

- Primary Guidance: Use figure/table captions to identify main sections and subsections.
- Section Identification: Group content around figures/tables with related captions.
- Logical Flow: Organize sections based on narrative flow indicated by visual elements.
- Content Mapping: Map text content to the section most relevant to nearby figures/tables.

Available Figure/Table Information:

Figures:

- {% if figures_info %}{% for fig_id, fig_info in figures_info.items() %}
  – Figure {{ fig_id }}{% if fig_info.caption_number %} (from "Figure {{ fig_info.caption_number }}" in caption){% endif %}: {{ fig_info.caption }}
  {% endfor %}
  {% endif %}

Tables:

- {% if tables_info %}{% for table_id, table_info in tables_info.items() %}
  – Table {{ table_id }}{% if table_info.caption_number %} (from "Table {{ table_info.caption_number }}" in caption){% endif %}: {{ table_info.caption }}
  {% endfor %}
  {% endif %}

Requirements:

- Preserve Paper Structure: Maintain the original paper's hierarchical structure without depth limitations.
- Caption-driven Sections: Create sections based on figure/table themes and captions.
- Preserve All Content: Include complete original text without truncation or summarization.
- No Depth Limits: Allow arbitrary nesting depth.
- Each node requires: id, title, type, content, assets, children.
- Asset Reference Format: Use actual figure/table IDs (e.g., "2", "3", "1").
- Asset Assignment: Assign figures/tables to most relevant sections.

Asset ID Reference Rules:

- {% if figures_info %}
  – For figures, use exact IDs in the "figures" array: [{% for fig_id in figures_info.keys() %}"fig_id"{% if not loop.last %}, {% endif %}{% endfor %}] {% endif %}
- {% if tables_info %} For tables, use exact IDs in the "tables" array: [{% for table_id in tables_info.keys() %}"table_id"{% if not loop.last %}, {% endif %}{% endfor %}] {% endif %}

Parser Agent [2/2]

Caption-Based Section Strategy:

1. Identify themes from figure/table captions.
2. Group related content around visual elements.
3. Create section titles reflecting caption themes.
4. Assign figures/tables to relevant sections using exact IDs.
5. Ensure logical flow (introduction→methodology→results).
6. Preserve hierarchical depth as in the original paper.

Structure Rules:

- Root: Contains title, authors, abstract, and metadata.
- Sections: May contain content and/or subsections.
- Subsections: Arbitrary depth, may contain content or further subsections.
- Content Preservation: All text content preserved entirely.
- Asset Assignment: Figures/tables assigned at any section/subsection level.

Example (Multi-level Hierarchy with Preserved Structure):

```
{
  "tree": {
    "id": "root",
    "title": "Paper Title: Subtitle",
    "type": "title",
    "content": "Complete paper title, authors, affiliations, and full
        abstract content...",
    "assets": {"figures": [], "tables": [], "references": []},
    "children": [
      {
        "id": "1",
        "title": "Introduction",
        "type": "section",
        "content": "Full introduction content...",
        "assets": {"figures": ["1"], "tables": [], "references": []},
        "children": [
          {
            "id": "1.1",
            "title": "Problem Statement",
            "type": "subsection",
            "content": "Detailed problem statement content...",
            "assets": {"figures": [], "tables": [], "references": []},
            "children": [
              {
                "id": "1.1.1",
                "title": "Current Limitations",
                "type": "subsubsection",
                "content": "Detailed limitations content...",
                "assets": {"figures": [], "tables": [], "references": []},
                "children": []
              }]
          }]
      }]
    }
}
```

Document to Analyze:

```
{{ markdown_document }}
```

Generate the hierarchical tree structure as JSON, preserving the original paper's complete structure and depth.

Figure A4: **Parser Agent** prompt example.

Summarizing Agent

You are a document content divider and extractor specialist, expert in dividing and extracting content from various types of documents and reorganizing it into a two-level JSON format for later PPT generation.

Based on the given markdown document, generate a JSON output for later PPT generation. Ensure the output is concise and focused.

Step-by-Step Instructions:

1. Identify Sections and Subsections: Detect sections and subsections based on heading levels and logical structure.
2. Divide Content: Organize content into sections and subsections, ensuring each subsection contains approximately 500 words.
3. Refine Titles: Use existing headings as titles, otherwise create relevant titles.
4. Remove Unwanted Elements: Eliminate headers, footers, and text surrounded by ~~ indicating deletion.
5. Refine Text: Remove unnecessary (citations) or trivial (repetitive, non-important) content for conciseness.

Example Output:

```
{
  "metadata": {
      "title": "title of document",
      "author": "name of authors",
      "publish date": "date of publication",
      "organization": "name of organization"
  },
  "sections": [
      {
          "title": "title of section1",
          "subsections": [
              {
                  "title": "title of subsection1.1",
                  "content": "content of subsection1.1"
              },
              {
                  "title": "title of subsection1.2",
                  "content": "content of subsection1.2"
              }
          ]
      },
      {
          "title": "title of section2",
          "subsections": [
              {
                  "title": "title of subsection2.1",
                  "content": "content of subsection2.1"
              }
          ]
      }
  ]
}
```

Document Input:

```
{{ markdown_document }}
```

Give your output in JSON format.

Figure A5: **Summarizing Agent** prompt example.

Content Agent [1/2]

System Prompt:

You are the *Content Agent*, an expert in scientific content clarity and efficiency. Your primary goal is to ensure content within a scientific poster is clear, concise, and logically structured.

Goal

- current_volume_ratio = current fill percentage of the space.
- Adjust the text so the final ratio is as close to 100% as possible, without exceeding it.
- Acceptable range: 80–100%, but closer to 100% is best.
- Always edit the given JSON (poster bullet schema) in place.
- Output only a JSON array (no explanations).

Hierarchical Awareness

1. Poster is hierarchical: parent → children; siblings share a level.
2. Use current_content as this node's main source only for expansion.
3. Use sibling_contents to avoid redundancy and ensure complementary focus.

Output rules

1. Return only a JSON array where each element is a bullet object.
2. Do NOT output plain strings, nested arrays, or any other format.
3. Each bullet object must exactly follow this schema (mandatory):

```
{
    "alignment": "left",
    "bullet": true,
    "level": <integer>,
    "runs": [ { "text": "<bullet text>" } ]
}
```

Expand Template:

Instructions:

1) Expand toward 100%.

- current_volume_ratio is `{{ current_volume_ratio }}`%.
- Target is 100%. Expand so the final length is about (100 - current_volume_ratio)% longer.
- Do not add all possible details at once; add gradually, starting with the most important points, and monitor length as you go.
- Stop once you estimate you've reached the target delta; if still under 95%, leave the rest for later passes.

2) Increase length by:

- Splitting existing bullets into smaller, clearer ones.
- Adding concise sub-bullets with extra details from current_content.
- Include key concepts, mechanisms, methods, assumptions, limitations, and results.
- Prefer precise or quantitative facts (counts, thresholds, time, cost, accuracy).

3) Avoid repeating siblings unless you add new perspective.

4) Output rule: Return only a JSON array of bullet objects following the schema.

- The format must exactly match previous_response (same keys).
- No strings, no explanations, no extra keys.

Content Agent [2/2]

Output Example

```
[
  { "alignment": "left", "bullet": true, "level": 0, "runs": [ { "text": "
      Our method boosts CIFAR-10 accuracy by 15% while cutting training
      cost 40%." } ] },
  { "alignment": "left", "bullet": true, "level": 0, "runs": [ { "text": "
      Key: adaptive LR and aggressive data augmentation." } ] }
]
```

previous_response:

```
{{ previous_response }}
```

current_content:

```
{{ current_content }}
```

sibling_contents:

```
{{ sibling_contents }}
```

current_volume_ratio:

```
{{ current_volume_ratio }}
```

Condense Template:

Instructions:

1) Condense toward 100%.

- current_volume_ratio is {{ current_volume_ratio }}%.
- Target is 100%. Condense so the final length is about (100 - current_volume_ratio)% shorter.
- Merge bullets into fewer ones, but keep each sentence short and crisp, until the ratio is near 100%.
- Stop once you estimate you've reached the target delta; if still above 105%, leave deeper cuts for later passes.

2) Remove overlap with siblings:

- Drop background/definitions if already covered in sibling_contents.

3) Structural compression:

- Keep only the most essential contributions, numbers, or distinctions.
- Sub-bullets should be folded into parent unless critical.

4) Output rule: Return only a JSON array of bullet objects following the schema.

- The format must exactly match previous_response (same keys).
- No strings, no explanations, no extra keys.

Output Example

```
[
  { "alignment": "left", "bullet": true, "level": 0, "runs": [ { "text": "
      Our method improves accuracy while cutting cost." } ] }
]
```

previous_response:

```
{{ previous_response }}
```

sibling_contents:

```
{{ sibling_contents }}
```

current_volume_ratio:

```
{{ current_volume_ratio }}
```

Figure A6: **Content Agent** prompt example.

Layout Agent [1/2]

System Prompt:

You are the *Layout Agent* for scientific posters. Your task is to determine which panel to enlarge and by how much (Δ), based on figure legibility balance within the red analysis box. Inside the red box, green boxes represent panels and blue boxes represent figures.

Goal

- Equalize figure legibility across panels using small, gradual changes.
- Do NOT optimize by raw figure size or panel area.

Legibility Definition

- Charts/Tables: judge by *perceived text size* (axis ticks, labels, legends, numbers).
- Method/Diagram/Schematic/Workflow: even with little/no text, judge by structural complexity (component count, branching, arrow/box density, nested sub-panels, tiny icons). Treat complex diagrams as if they contained implicit small text.
- A panel is "harder to read" if it has smaller text OR higher visual complexity.

Priorities

1) Shape Sanity

- Avoid creating extremely tall–narrow panels.

2) Legibility Balance (Hard Rule)

- If one panel is harder to read $\rightarrow$ enlarge that panel.
- Do NOT enlarge panels already comfortable to read.
- Ignore raw figure size/area.

Direction Mapping

- 'split_type='vertical'' $\rightarrow$ enlarge LEFT or RIGHT.
- 'split_type='horizontal'' $\rightarrow$ enlarge TOP or BOTTOM.

Δ (Delta): Magnitude of Enlargement Δ is a continuous scale from 0 to 5, representing how strongly to enlarge the harder-to-read panel.

- 0 $\rightarrow$ No adjustment (balanced or shape risk)
- 1–2 $\rightarrow$ Very subtle adjustment
- 3–4 $\rightarrow$ Moderate adjustment
- 5 $\rightarrow$ Maximum safe enlargement

Choose Δ proportionally to the severity of the legibility imbalance.

Output Format (strict JSON)

```
{
  "direction": "<LEFT|RIGHT|TOP|BOTTOM>",
  "delta": <0-5>,
  "reason": "Action: enlarge {LEFT|RIGHT|TOP|BOTTOM}. Rationale: describe
      legibility comparison (text-size + diagram-complexity) and shape
      consideration."
}
```

Layout Agent [2/2]

Template:

CONTEXT

- Split Type: `{{ split_type }}`

Task

1) Assess legibility for each panel:

- Charts/Tables → text-size
- Diagrams/Method figures → structural complexity (implicit small text)

2) Pick the panel to enlarge = harder-to-read panel.

3) Choose Δ (0–5) proportional to how severe the imbalance is.

- 0 = no change, 5 = strongest possible enlargement.
- No strings, no explanations, no extra keys.

4) Output strict JSON with fields ‘direction‘, ‘delta‘, ‘reason‘.

Example Outputs

LEFT enlarge (vertical)

```
{
    "direction": "LEFT",
    "delta": 5,
    "reason": "Action: enlarge LEFT. Retionale: left diagrams are visually
        complex with many branches; prevent tall-narrow shape."
}
```

RIGHT enlarge (vertical)

```
{
    "direction": "RIGHT",
    "delta": 4,
    "reason": "Action: enlarge UP. Rationale: right schematic is more
       complex and crowded, but difference is subtle; shape remains
       stable."
}
```

TOP enlarge (horizontal)

```
{
    "direction": "TOP",
    "delta": 3,
    "reason": "Action: enlarge TOP. Rationale: top panel contains dense
       multi-axis plots with fine labels; bottom is simpler; shape
       acceptable."
}
```

BOTTOM enlarge (horizontal)

```
{
    "direction": "BOTTOM",
    "delta": 1,
    "reason": "Action: enlarge BOTTOM. Rationale: bottom diagrams slightly
        denser than top; minimal adjustment applied to prevent tall-
       narrow distortion."
}
```

Figure A7: **Layout Agent** prompt example.

Feedback Agent

System Prompt:

You are a scientific poster evaluator. You will see a poster image.

Evaluate the poster against THREE criteria:

- Visual organization: alignment/grid consistency, whitespace, figure–text balance, crowding/overflow.
- Textual structure: redundancy, truncation/overflow, missing context, uneven verbosity.
- Hierarchical balance: section order, parent–child locality, consistent heading levels, grouping coherence.

Decision rule:

- Assess all three criteria and note any failures.
- Output ONLY:
  reason → 1–2 sentences summarizing the key issue(s) or that the poster is fine.
  continue → 1 if any criterion fails (further work needed), else 0.

Final Output Format:

```
{
    "reason": "<1-2 sentences>",
    "continue": 0 | 1
}
```

Template:

Insturction :

1. Examine the poster image.
2. Visual organization (fail if any apply):
   - Misaligned grid, cramped/excessive whitespace, figure–text size mismatch, or crowding/overflow.
3. Textual structure (fail if any apply):
   - Redundancy, truncated/overflowing text, missing context, or uneven verbosity across sections.
4. Hierarchical balance (fail if any apply):
   - Disordered section flow, inconsistent heading levels, weak parent–child locality, poor grouping coherence.
5. Decide "continue":
   - If any of the above fail → continue = 1
   - If none fail → continue = 0
6. Output only a single JSON object with keys: reason, continue

Output Example (issues found)

```
{
      "reason": "Figures overpower adjacent text and subsections are
          detached from their parent sections, disrupting flow.",
      "continue": 1
}
```

Output Example (no issues)

```
{
      "reason": "Content is well-explained with balanced figures and
          consistent hierarchy.",
      "continue": 0
}
```

Figure A8: **Feedback Agent** prompt example.