This CVPR Findings paper is the Open Access version, provided by the Computer Vision Foundation. Except for this watermark, it is identical to the accepted version; the final published version of the proceedings is available on IEEE Xplore.

SciPostLayoutTree: A Dataset for Structural Analysis of Scientific Posters

Shohei Tanaka Atsushi Hashimoto Yoshitaka Ushiku
OMRON SINIC X Corporation
{shohei.tanaka, atsushi.hashimoto, yoshitaka.ushiku}@sinicx.com

Abstract

Scientific posters play a vital role in academic communication by presenting ideas through visual summaries. Analyzing reading order and parent-child relations of posters is essential for building structure-aware interfaces that facilitate clear and accurate understanding of research content. Despite their prevalence in academic communication, posters remain underexplored in structural analysis research, which has primarily focused on papers. To address this gap, we constructed SciPostLayoutTree, a dataset of approximately 8,000 posters annotated with reading order and parent-child relations. Compared to an existing structural analysis dataset, SciPostLayoutTree contains more instances of spatially challenging relations, including upward, horizontal, and long-distance relations. As a solution to these challenges, we develop Layout Tree Decoder, which incorporates visual features as well as bounding box features including position and category information. The model also uses beam search to predict relations while capturing sequence-level plausibility. Experimental results demonstrate that our model improves the prediction accuracy for spatially challenging relations and establishes a solid baseline for poster structure analysis. The dataset and code are publicly available at https://github.com/omron-sinicx/scipostlayouttree.

1. Introduction

While the structure of documents and posters is designed to facilitate human understanding, it remains unclear whether ML models can analyze such structure. Reliable structural analysis is crucial for information access technologies, including structure-aware text-to-speech synthesis [7, 30], question answering [8, 23], and retrieval using tables of contents [11]. It would further aid layout assessment; discrepancies between model predictions and designer intentions could indicate a potentially confusing layout.

Existing work on structural analysis has primarily focused on textual documents, in what is known as document structure analysis (DSA) [1, 26]. However, information is also conveyed through visual media such as scientific posters, a primary medium in academic communication. Misinterpretation of posters may hinder access to the research content they convey. This study aims to establish a foundational framework that enables models to interpret and reason over the structural organization of posters.

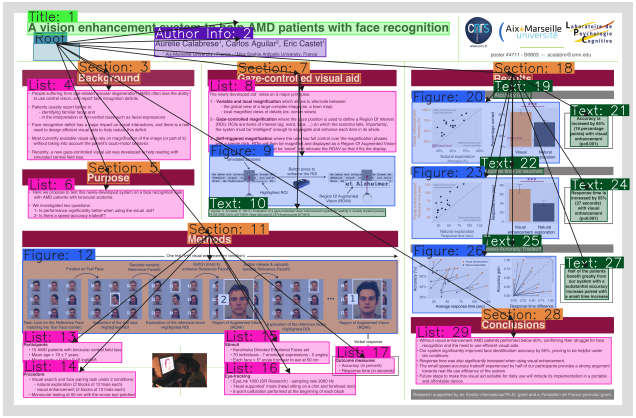


Figure 1. Example from SciPostLayoutTree. Each arrow denotes a parent-child relation, with the tail indicating the parent and the head indicating the child. The node labeled "Root" denotes the root of the DFS-ordered tree, corresponding to the poster. The number shown next to each BBox category indicates its reading order priority.

As a step toward this goal, we construct SciPostLayoutTree, a dataset containing structural annotations for approximately 8,000 posters. Figure 1 shows an example poster with our structure annotation: the reading order and parent-child relations among bounding boxes (BBoxes) are represented as a depth-first-search (DFS)-ordered tree. A comparative analysis with a DSA dataset reveals that SciPostLayoutTree contains more instances of spatially challenging relations, including upward, horizontal, and long-distance relations. The result indicates that poster structures present novel technical challenges beyond those found in document structures.

To address these technical challenges, we develop Layout Tree Decoder as an extension of an existing structural

analysis model. The model incorporates not only visual features but also BBox features including position and category information, which enhances its ability to capture spatially challenging relations. In addition, the model applies beam search during tree decoding to capture sequence-level plausibility. Experimental results show that our model achieves higher accuracy on spatially challenging relations compared to an existing model using only visual features and greedy decoding. This observation suggests that our model serves as a solid baseline for poster structure analysis.

Our contributions are summarized as follows:

- We present SciPostLayoutTree, a novel dataset of approximately 8,000 scientific posters annotated with reading order and parent-child relations, represented as DFS-ordered trees.
- We show that the posters pose novel spatially challenging relations, including upward, horizontal, and long-distance relations, which are infrequent in documents.
- We develop Layout Tree Decoder, which provides a solid baseline for poster structure analysis by extending an existing model with BBox features and beam search to capture spatially challenging relations.

2. Related Work

This section presents existing DSA and scientific poster datasets, contrasting them with our dataset.

2.1. Document Structure Analysis Dataset

Existing DSA datasets can be categorized into three types based on their objectives: predicting (i) only reading order [3, 29, 34], (ii) only hierarchical structure [14, 31], or (iii) both [1, 18, 26]. These datasets target textual documents and focus on structural analysis challenges such as domain diversity and the prediction of structures spanning multiple pages. In contrast, our dataset targets scientific posters and is characterized by frequent spatially challenging relations, including upward, horizontal, and long-distance relations.

Among the listed datasets, Comp-HRDoc [26], GraphDoc [1], and DocHieNet [31] share similar tasks with our work. Comp-HRDoc provides annotations for predicting the reading order and parent-child relations of BBoxes as a tree structure. However, Comp-HRDoc uses text lines as BBox units and focuses on paragraph structuring and section heading detection, whereas our work targets region-level BBoxes such as paragraphs and figures, with different prediction objectives.

GraphDoc provides structural annotations for region-level BBoxes on document pages from DocLayNet [18]. Since GraphDoc is not publicly available, we do not include it in our comparative analysis. Chen et al. also developed a model that predicts reading order and parent-child relations between BBoxes based on visual features, which we adopt as a baseline in our experiments.

DocHieNet includes documents of various domains such as scientific papers, financial reports, and legal documents. We compare DocHieNet with our dataset, as both contain annotations on region-level BBoxes, including reading order and parent-child relations. However, the model proposed by Xing et al. predicts a hierarchical structure from a given reading order, spanning multiple pages. This setting is not comparable to our task of jointly predicting reading order and parent-child relations within a single poster.

2.2. Scientific Poster Dataset

Although several datasets of scientific posters have been proposed [17, 21, 22, 32], they focus on poster generation and do not include annotations for structural analysis. SciPostLayout [22] contains BBox annotations for approximately 8,000 scientific posters collected from the web. We construct our dataset by extending SciPostLayout with annotations of reading order and parent-child relations among the BBoxes.

3. SciPostLayoutTree Dataset

This section presents the task, the annotation procedure, and the statistical analysis of the dataset.

3.1. Task Definition

The DSA task is first defined in Comp-HRDoc [26] as predicting a DFS-ordered tree based on reading order and parent-child relations from a set of BBoxes. We follow this setting for scientific posters. The input is a set of BBoxes denoted as $B = \{b_1, b_2, \ldots, b_N\}$, where each BBox b_i has center position and size (x_i, y_i, w_i, h_i) as well as category c_i. Here, N denotes the number of BBoxes in the poster. Each BBox is treated as both a layout element and a node in the tree. In addition, a virtual node b_0 representing the poster is introduced as the Root. The model takes the BBox set B as input and predicts the reading order sequence $\Phi = (\phi(0), \phi(1), \ldots, \phi(N))$. Here, $\phi(j)$ is a mapping function that outputs the BBox index i corresponding to the 0-based reading order index j. Since the Root node b_0 is always placed at the beginning of the reading order, the predicted sequence Φ is of length $N + 1$, starting with $\phi(0) = 0$. The model also predicts the parent BBox $\psi(i)$ for each BBox b_i. Here, $\psi(i)$ is a mapping function that outputs the index of the parent BBox of b_i. As a result, a DFS-ordered tree T is constructed on the BBox set B, where the reading order and parent-child relations are structurally consistent.

3.2. Dataset Annotation

We construct SciPostLayoutTree by adding structural annotations to SciPostLayout [22]. In contrast to business or

artistic posters, scientific posters have an explicit and consistent reading order intended by the authors. This property allows us to define a unique DFS-ordered tree structure for each poster. Each poster was annotated as a DFS-ordered tree rooted at the Root node[1]. The annotation was conducted by professional annotators from an external vendor, with one annotator assigned per poster. All annotations were reviewed by in-house supervisors and the authors with expertise in scientific poster layouts, ensuring rigorous quality control. To quantify annotation consistency, we evaluated 100 randomly selected posters by comparing the original annotations with independently annotated trees by two additional annotators, and observed a high agreement score (0.91)[2]. Posters containing only a single BBox were excluded as structurally trivial. As a result, we obtained 7,849 annotated posters, split into 6,853/498/498 for training, validation, and test sets. We confirmed that 98% of posters in the validation and test sets have non-overlapping authors. The test set size of 498 posters is sufficient compared to prior datasets such as DocHieNet.

3.3. Statistical Analysis

We highlight the unique statistical property of SciPostLayoutTree in comparison with DocHieNet [31], which brings us novel technical challenges.

3.3.1. Layout Element Statistics

Table 1 presents statistics on the layout elements in SciPostLayoutTree. Compared to DocHieNet documents, which contain approximately 12 BBoxes and fewer than one figure per page on average[3], our posters contain approximately 25 BBoxes and a greater number of figures. These observations indicate that structural analysis for posters requires predicting relations among a larger number of BBoxes per page than in document-based tasks, and entails greater reliance on visual features.

3.3.2. Layout Tree Statistics

Figure 2 summarizes statistics of the overall tree structures in our dataset. Figures 2 (a)-(b) show that the trees in our dataset are shallower and wider than those in DocHieNet. Figure 2 (c) shows that the number of children per node follows a skewed distribution, where a small number of nodes have many children. These characteristics indicate that our dataset contains more cases than DocHieNet in which the model must predict the reading order among many child nodes sharing the same parent.

3.3.3. Reading Order Statistics

Figure 3 uses a polar heatmap to visualize the distribution of directions and distances between consecutive pairs of BBoxes. Assuming the top-left corner of the image as the origin, we compute the direction class for each consecutive pair of BBoxes with center coordinates (x_i, y_i) and (x_j, y_j) as follows:

$$\theta_{i,j} = \arctan2(y_j - y_i,\, x_j - x_i), \quad \theta_{i,j} \in (-\pi, \pi]$$

$$a_{i,j} = \left\lfloor \frac{(\theta_{i,j} + 2\pi) \bmod 2\pi + \frac{\pi}{8}}{\frac{\pi}{4}} \right\rfloor \bmod 8$$

The angle $\theta_{i,j}$ increases in the clockwise direction, with 0 aligned to the horizontal axis. The floor function $\lfloor \cdot \rfloor$ returns the greatest integer less than or equal to its argument. Here, $a_{i,j} \in \{0, \ldots, 7\}$ denotes the direction class index in clockwise order, with 0 corresponding to Right and 7 to Top-Right. The distance axis is binned on a $\log_2$ scale. The distance d between BBoxes is defined as follows:

$$d_{i,j} = \begin{cases} \dfrac{|x_j - x_i|}{\max(w_i, w_j)} & \text{if } |x_j - x_i| \geq |y_j - y_i| \\ \dfrac{|y_j - y_i|}{\max(h_i, h_j)} & \text{otherwise} \end{cases}$$

The distance between BBox centers is normalized by the larger of their widths or heights, depending on the dominant transition axis, to capture scale-invariant patterns. Normalizing the Euclidean distance by the BBox diagonal may underestimate actual distance, particularly when horizontally elongated BBoxes are vertically arranged. Color intensity in the heatmap indicates transition frequency in each direction–distance bin, measured on a $\log_2$ scale.

The figure shows that, as in DocHieNet, most reading transitions in our dataset are directed downward and span short distances. However, our dataset includes more transitions toward Top and Top-Right, regardless of distance, and also exhibits frequent horizontal transitions to Right and Left. These observations indicate that reading order is not constrained to a single direction.

3.3.4. Parent-Child Relation Statistics

Figure 4 presents statistics on parent-child relations. In our dataset, parent-child relations are primarily directed down-

Table 1. Statistics of layout elements by category. Mean (Std.) denotes the mean and standard deviation per poster.

Category	Total Count	Mean (Std.)
Title	7,842	1.00 (± 0.04)
Author Info	7,692	0.98 (± 0.15)
Section	41,321	5.26 (± 2.43)
Text	52,725	6.72 (± 4.89)
List	23,761	3.03 (± 3.31)
Figure	38,625	4.92 (± 3.62)
Table	5,655	0.72 (± 1.26)
Caption	14,966	1.91 (± 2.66)
All	192,587	24.54 (± 9.58)

[1] See the Appendix for the annotation guideline.
[2] See the Appendix for the evaluation details.
[3] These values are derived from A.1 of the DocHieNet paper.

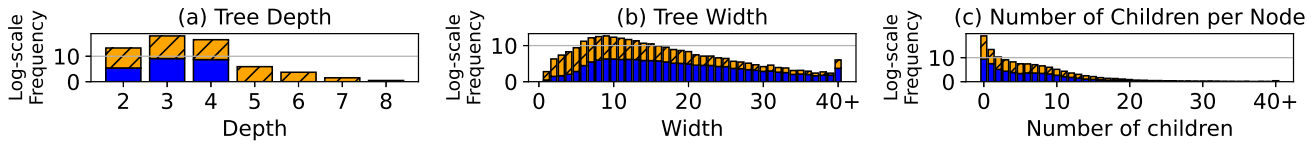


Figure 2. Distributions of tree depth, tree width, and number of children per node. SciPostLayoutTree (blue) and DocHieNet (orange-hatched) are displayed as stacked bars. All values are shown on a $\log_2(1 + \text{count})$ scale. Figures (a) and (b) show frequencies per 1,000 pages; Figure (c) shows frequencies per 1,000 nodes.

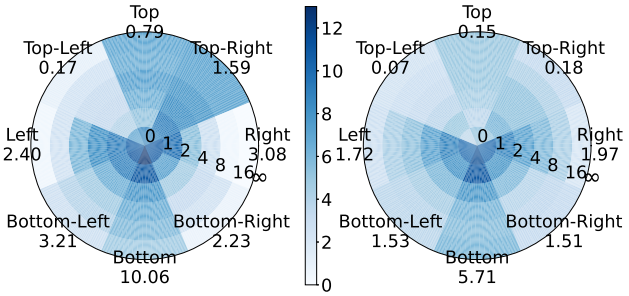


Figure 3. Reading order heatmaps by direction and distance. The heatmaps show results from SciPostLayoutTree (left) and DocHieNet (right), with counts normalized per 1,000 pages. Distances are binned on a $\log_2$ scale, and heatmap values represent $\log_2(1 + \text{count})$. The numbers below each direction indicate the mean count per page.

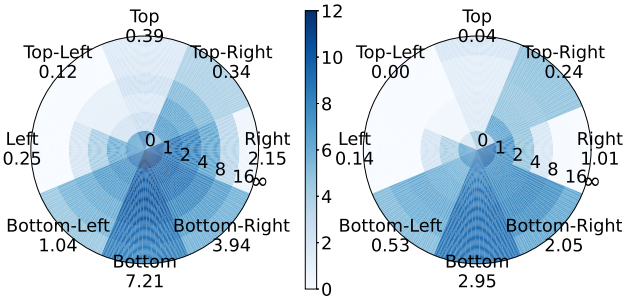


Figure 4. Parent-child relation heatmaps by direction and distance. The format follows Fig. 3.

ward, as in DocHieNet, which suggests that parents are typically positioned above their children. However, our dataset includes more upward and horizontal relations, indicating increased directional diversity. Compared to reading order transitions (Fig. 3), parent-child relations span longer distances, necessitating the capture of non-local relations for accurate prediction. Furthermore, as shown in Table 1, each poster comprises approximately 25 BBoxes, which necessitates the identification of the correct parent among many candidates.

4. Layout Tree Decoder

This section presents the model, the tree decoding algorithm, and the loss function.

4.1. Model Architecture

As shown in Table 1, our dataset contains many figures and thus exhibits strong dependence on visual features. We therefore build our model on Document Relation Graph Generator (DRGG) proposed by Chen et al. (2025), which predicts relations between BBoxes using visual features. Figure 5 presents an overview of our Layout Tree Decoder. Given an input poster image and BBoxes, the model first extracts a set of visual features $V = \{\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_N}\}$ using a Visual Feature Extractor (VFE). The VFE is constructed by stacking a visual backbone [2, 5, 28], a Feature Pyramid Network (FPN) [12], ROI Align [6], and a Box Head [19]. Each vector $\mathbf{v_i}$ represents the visual feature corresponding to the BBox b_i. The feature vector for the Root node b_0, denoted as $\mathbf{v_0}$, is obtained by applying a Multilayer Perceptron (MLP) to the FPN output. We then extend V to include $\mathbf{v_0}$ as $V = \{\mathbf{v_0}, \mathbf{v_1}, \ldots, \mathbf{v_N}\}$.

DRGG applies a Relation Feature Extractor (RFE), which consists of an average pooling layer and MLPs, to the visual features V, independently for row and column roles. The feature sequences used for relation prediction are defined as $V_r = \{\mathbf{v_0^r}, \ldots, \mathbf{v_N^r}\}$ and $V_c = \{\mathbf{v_0^c}, \ldots, \mathbf{v_N^c}\}$, corresponding to the row and column roles, respectively. V_r represents preceding or child nodes, whereas V_c represents subsequent or parent nodes in a DFS-ordered tree.

Following the DRGG design for visual feature encoding with RFE, we further integrate the spatial and categorical information of each BBox via a BBox Embedding. For each node b_i, the normalized coordinates of the top-left and bottom-right corners are encoded into a position embedding $\mathbf{z_i}$ via an MLP. The category c_i is embedded as $\mathbf{e_i}$. For the Root node b_0, the coordinate region is fixed to $(0, 0, 1, 1)$, representing the entire poster, and the category is embedded as a special class, "Root". We analyze the effect of combining BBox features with visual features on reading order and parent-child prediction in spatially challenging relations[4].

[4]Models incorporating OCR text features were also evaluated, but no improvement in prediction accuracy was observed. Details are provided in

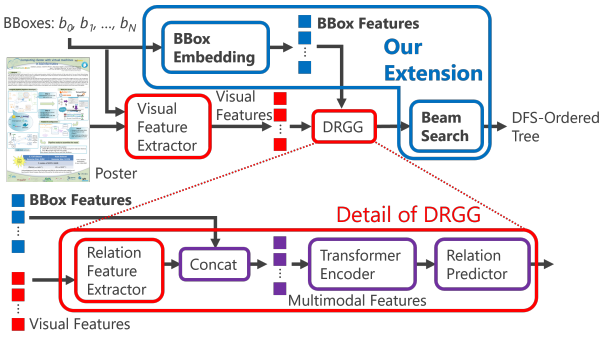


Figure 5. Overview of Layout Tree Decoder. We extend DRGG by incorporating BBox features and beam search.

The BBox features and visual features are combined to obtain the following multimodal features:

$$\mathbf{m_i^r} = \text{concat}(\mathbf{v_i^r}, \mathbf{z_i}, \mathbf{e_i}),\ \mathbf{m_i^c} = \text{concat}(\mathbf{v_i^c}, \mathbf{z_i}, \mathbf{e_i})$$

$\mathbf{m_i^r}$ and $\mathbf{m_i^c}$ are input to a Transformer encoder [24] to produce the node representations F_r and F_c for relation prediction. The relation predictor is an MLP that receives the vector $\text{concat}(F_r[i], F_c[j])$ for each node pair (b_i, b_j) and outputs a score tensor $G \in \mathbb{R}^{2\times(N+1)\times(N+1)}$. Here, $G[0]$ and $G[1]$ correspond to the following matrices:

- **Subsequent score matrix** $S \in \mathbb{R}^{(N+1)\times(N+1)}$: $S_{i,j}$ indicates the score that b_j follows b_i in the reading order.
- **Parent score matrix** $P \in \mathbb{R}^{(N+1)\times(N+1)}$: $P_{i,j}$ indicates the score that b_j is the parent of b_i.

4.2. Tree Decoding Algorithm and Beam Search

The model decodes a DFS-ordered tree T by applying Algorithm 1 to the score matrices S and P. This decoding algorithm is derived by reorganizing the reading order and parent-child prediction procedures proposed by Wang et al. (2024). The decoded tree is consistent with the reading order $\Phi = (\phi(0), \ldots, \phi(N))$ and the parent-child relations.

Previous studies predicted reading order and parent-child relations via greedy decoding [26, 27]. However, as shown in Figs. 3–4, our dataset contains a substantial number of spatially challenging relations, such as upward, horizontal, and long-distance relations. During greedy decoding, infrequent relations are often not captured due to their low local scores. To address this issue, we apply beam search to steps 3–5 (reading order prediction) and 9–15 (parent-child prediction) in Algorithm 1 to capture sequential-level plausibility. Beam search is widely used in natural language processing tasks such as machine translation [4, 24], text summarization [20], image captioning [25], dialogue generation [9], and named entity recognition [33]. At each decoding step, beam search retains multiple top-ranked candidates to favor sequences with higher total scores despite low local scores. We evaluate the effect of beam search on reading order and parent-child prediction, particularly for spatially challenging relations.

the Appendix.

Algorithm 1 Tree decoding from score matrices

Require: Subsequent score matrix $S \in \mathbb{R}^{(N+1)\times(N+1)}$, Parent score matrix $P \in \mathbb{R}^{(N+1)\times(N+1)}$
Ensure: DFS-ordered tree T over $\{0, 1, \ldots, N\}$, where b_0 represents the poster and $\{b_1, \ldots, b_N\}$ are its BBoxes
1: Initialize ϕ as an empty reading order mapping
2: Set $\phi(0) \leftarrow 0$
3: **for** $i = 1$ to N **do**
4: Select $\phi(i) \leftarrow \arg\max_{k \notin \{\phi(0),\ldots,\phi(i-1)\}} S_{\phi(i-1),\,k}$
5: **end for**
6: Initialize DFS-ordered tree T with the root node 0
7: Initialize ψ as an empty parent mapping
8: Initialize stack $Q \leftarrow [0]$ {The rightmost path from the root node to a leaf node}
9: **for** $j = 1$ to N **do**
10: $i \leftarrow \phi(j)$
11: Select parent $\psi(i) \leftarrow \arg\max_{p \in Q} P_{i,p}$
12: Add node i as the rightmost child of $\psi(i)$ in T
13: Truncate Q after $\psi(i)$
14: Push i to Q
15: **end for**
16: **return** T

4.3. Loss Function

We define cross-entropy losses based on the subsequent score matrix S and the parent score matrix P. Each BBox b_i has a unique ground-truth (GT) subsequent BBox b_j^{sub} and a unique parent BBox b_j^{par}. The losses are defined as follows:

$$\mathcal{L}_{\text{sub}} = \frac{1}{N}\sum_{i\in\mathcal{I}_{\text{sub}}} \text{CE}(S_i, j_i^{\text{sub}}),\ \mathcal{L}_{\text{par}} = \frac{1}{N}\sum_{i\in\mathcal{I}_{\text{par}}} \text{CE}(P_i, j_i^{\text{par}})$$

Here, $\text{CE}(*)$ denotes the cross-entropy loss for either a score vector S_i or P_i, using the GT index j as the label. The index set $\mathcal{I}_{\text{sub}}$ is $\{0, \phi(1), \ldots, \phi(N-1)\}$, which excludes the terminal node in the reading order sequence. The index set $\mathcal{I}_{\text{par}}$ is $\{1, 2, \ldots, N\}$, which excludes the Root node. The total loss is defined as $\mathcal{L} = \mathcal{L}_{\text{sub}} + \mathcal{L}_{\text{par}}$.

5. Experiments

This section investigates the effect of incorporating the BBox Embedding and beam search into DRGG.

5.1. Experimental Settings

We compared four DRGG variants, each with a different combination of the BBox Embedding and beam search [5].

[5] We also evaluated VLMs (GPT-5, Gemini 3 Pro), but they showed lower performance than DRGG-based models. Details are provided in the

DRGG predicts score matrices between BBoxes based on the visual features, and determines the reading order and parent-child relations through greedy decoding.
DRGG-BE (BBox Embedding) augments the input features with the BBox features.
DRGG-BS (Beam Search) replaces greedy decoding with beam search to predict the relations.
DRGG-BEBS (BBox Embedding and Beam Search) integrates DRGG-BE and DRGG-BS.

To confirm that performance differences among methods are not attributable to backbone choice, we employed five visual backbones: ResNet-50 [5], ViT [2], Swin [13], DiT [10], and InternImage [28]. The beam width was set to 20 for DRGG-BS and DRGG-BEBS[6].

5.2. Evaluation Metrics

Three metrics were used to evaluate the predicted DFS-ordered trees: Tree Edit Distance (TED) [15, 16], Semantic Tree Edit Distance Score (STEDS) [14], and Reading Edit Distance Score (REDS) [26]. TED is the edit distance between the predicted tree T_P and the GT tree T_G, defined as the minimum number of node insertions, deletions, and substitutions. STEDS is a normalized version of TED with respect to tree size, defined as follows:

$$\text{STEDS} = 100 \left(1 - \frac{\text{TED}(T_G, T_P)}{\max(|T_G|, |T_P|)}\right)$$

Here, $|T_G|$ and $|T_P|$ denote the number of nodes in the GT tree and the predicted tree. REDS evaluates the similarity between reading orders using the Levenshtein distance (string edit distance) $\text{LD}(\Phi_G, \Phi_P)$ between the GT order Φ_G and the predicted order Φ_P. The metric is defined as follows:

$$\text{REDS} = 100 \left(1 - \frac{\text{LD}(\Phi_G, \Phi_P)}{\max(|T_G|, |T_P|)}\right)$$

The accuracies were computed for each direction and distance. These accuracies are defined as the ratio of GT relations included in the predicted tree. Relations involving the Root were excluded from the accuracy computation, as direction and distance are undefined for them.

5.3. Evaluation Results

Table 2 shows that both the BBox Embedding and beam search consistently improve STEDS, REDS, and TED. The BBox Embedding improves both STEDS and REDS, with a more substantial effect on STEDS. The results suggest that incorporating the BBox Embedding is effective for improving the accuracy of the parent-child prediction.

Beam search contributes to improvements in REDS, while its effect on STEDS is less substantial than that of the BBox Embedding. This difference reflects the structural properties of the two prediction tasks. Reading order prediction is sequential, with each decision depending on the previous decisions. In contrast, parent-child prediction does not depend on previous decisions, except when constructing the rightmost path. Beam search biases the decoding process toward reading orders with higher sequence-level plausibility.

DRGG-BEBS, which integrates the BBox Embedding and beam search, achieves the highest scores across all metrics. This result suggests that the two components are complementary. Furthermore, these improvements are consistent across visual backbones, demonstrating that the effectiveness of our extension is not dependent on the backbone.

Despite these improvements, some posters remain challenging, as illustrated in Figure 6[7]. In this poster, all Figures and Tables are attached as children to the bottom-left Section, following a Z-shaped reading order. Accurate prediction of such relations requires capturing semantic grouping and structural plausibility over longer sequences.

5.3.1. Analysis of Reading Order Prediction Improvements by Beam Search

The directional analysis in Table 3 shows that downward relations have high baseline accuracy, which limits the im-

Table 2. Comparison of DRGG and its extensions across visual backbones. $*$ indicates significant improvement over DRGG at $p < 0.005$ according to Wilcoxon signed-rank test.

Backbone	Decoder	STEDS (↑)	REDS (↑)	TED (↓)
ResNet-50	DRGG	68.74	75.07	8.83
	DRGG-BE	84.24*	86.44*	4.41*
	DRGG-BS	76.79*	83.14*	6.65*
	DRGG-BEBS	88.45*	90.40*	3.22*
ViT	DRGG	80.40	85.94	5.45
	DRGG-BE	86.38*	88.24*	3.85*
	DRGG-BS	83.89*	89.46*	4.46*
	DRGG-BEBS	90.04*	91.73*	2.78*
Swin	DRGG	79.90	85.77	5.69
	DRGG-BE	86.73*	88.42*	3.69*
	DRGG-BS	83.11*	89.02*	4.74*
	DRGG-BEBS	89.26*	90.88*	2.95*
DiT	DRGG	78.33	84.81	6.09
	DRGG-BE	85.32*	87.36*	4.22*
	DRGG-BS	82.12*	88.72*	5.07*
	DRGG-BEBS	88.69*	90.63*	3.24*
InternImage	DRGG	80.19	85.53	5.53
	DRGG-BE	86.89*	88.56*	3.70*
	DRGG-BS	83.75*	89.25*	4.55*
	DRGG-BEBS	89.34*	90.97*	2.93*

Appendix.

[6]See the Appendix for experimental details.

[7]See the Appendix for additional challenging examples.

Table 3. Per-direction accuracy of the reading order prediction with DRGG-BEBS. Values in parentheses indicate the accuracy improvement over DRGG-BE. $\star$ and $\star\star$ indicate significant improvements over DRGG-BE at $p < 0.005$ and $p < 0.05$, respectively, according to McNemar's test.

Backbone	Right	Bottom-Right	Bottom	Bottom-Left	Left	Top-Left	Top	Top-Right
ResNet-50	86.0 (+4.0)*	94.5 (+0.6)	96.7 (+2.0)*	92.7 (+2.4)*	84.2 (+3.6)*	73.3 (-1.7)	87.5 (+2.9)	79.8 (+5.5)*
ViT	87.7 (+2.8)*	95.2 (+0.0)	97.2 (+1.3)*	93.1 (+2.0)*	85.4 (+2.6)*	75.0 (+5.0)	89.3 (+2.3)	81.4 (+5.7)*
Swin	87.9 (+2.4)*	95.0 (-0.5)	97.0 (+1.1)*	92.7 (+1.0)**	85.5 (+2.8)*	73.3 (-5.0)	89.0 (+2.0)	81.1 (+3.7)*
DiT	85.3 (+2.8)*	94.6 (+0.5)	96.6 (+1.6)*	92.9 (+2.7)*	83.3 (+3.3)*	71.7 (-3.3)	88.1 (+2.3)	79.3 (+5.5)*
InternImage	87.5 (+2.1)*	94.8 (+0.0)	96.4 (+0.7)*	92.9 (+1.3)*	85.9 (+2.9)*	73.3 (+3.3)	88.1 (+2.3)**	80.1 (+3.0)*

Table 4. Per-distance accuracy of the reading order prediction. The format and experimental settings follow Table 3.

Backbone	(0, 1]	(1, 2]	(2, 4]	(4, 8]	(8, 16]	(16, ∞)
ResNet-50	94.7 (+1.9)*	84.0 (+4.1)*	77.4 (+6.4)*	85.4 (+5.6)**	86.3 (+5.3)**	89.4 (+3.3)
ViT	95.3 (+1.3)*	85.5 (+3.1)*	80.5 (+4.4)*	86.4 (+8.1)*	87.0 (+2.3)	89.4 (+4.9)**
Swin	95.2 (+1.0)*	85.5 (+2.6)*	78.7 (+2.2)	84.3 (+3.0)	87.0 (+1.5)	90.2 (+4.9)**
DiT	94.4 (+1.6)*	84.1 (+3.8)*	77.2 (+5.1)*	83.3 (+3.5)	87.8 (+3.8)	91.1 (+0.8)
InternImage	94.8 (+0.9)*	85.1 (+2.1)*	78.0 (+4.0)*	88.9 (+2.5)	86.3 (+2.3)	87.8 (+3.3)

Table 5. Per-direction accuracy of the parent-child prediction with DRGG-BEBS. Values in parentheses indicate the accuracy improvement over DRGG-BS. $\star$ and $\star\star$ indicate significant improvements over DRGG-BS at $p < 0.005$ and $p < 0.05$, respectively, according to McNemar's test.

Backbone	Right	Bottom-Right	Bottom	Bottom-Left	Left	Top-Left	Top	Top-Right
ResNet-50	93.3 (+13.3)*	95.3 (+8.4)*	98.0 (+9.2)*	92.6 (+13.6)*	74.5 (+19.1)*	90.7 (+18.6)**	88.8 (+14.0)*	83.4 (+10.4)**
ViT	94.3 (+5.3)*	97.1 (+3.7)*	98.4 (+5.5)*	94.7 (+9.3)*	77.3 (+7.1)	86.0 (+7.0)	89.9 (+20.8)*	87.6 (+12.4)*
Swin	94.8 (+7.9)*	97.1 (+6.1)*	98.5 (+6.6)*	92.1 (+8.5)*	73.0 (+9.9)*	88.4 (+14.0)**	91.0 (+21.3)*	90.2 (+14.0)*
DiT	92.8 (+7.7)*	96.3 (+6.5)*	97.8 (+7.2)*	94.7 (+12.1)*	80.1 (+14.9)*	83.7 (+23.3)**	90.4 (+32.6)*	84.5 (+9.3)*
InternImage	93.8 (+6.1)*	96.9 (+5.5)*	98.5 (+6.1)*	94.3 (+10.8)*	75.9 (+7.1)**	88.4 (+11.6)	90.4 (+13.5)*	92.2 (+16.6)*

Table 6. Per-distance accuracy of the parent-child prediction. The format and experimental settings follow Table 5.

Backbone	(0, 1]	(1, 2]	(2, 4]	(4, 8]	(8, 16]	(16, ∞)
ResNet-50	97.8 (+6.4)*	92.2 (+13.1)*	92.0 (+14.7)*	92.3 (+13.3)*	91.1 (+16.5)*	93.7 (+28.3)*
ViT	98.3 (+5.1)*	92.8 (+4.2)*	94.8 (+6.6)*	94.9 (+5.7)*	94.0 (+9.5)*	95.6 (+23.4)*
Swin	98.5 (+5.8)*	93.4 (+7.6)*	94.2 (+9.0)*	93.5 (+7.9)*	93.4 (+12.3)*	94.6 (+20.5)*
DiT	97.9 (+7.2)*	92.9 (+7.6)*	92.4 (+9.5)*	93.1 (+7.7)*	93.0 (+15.2)*	94.6 (+20.0)*
InternImage	98.4 (+5.2)*	93.1 (+7.1)*	93.5 (+8.0)*	94.6 (+7.5)*	95.6 (+13.3)*	95.6 (+17.6)*

provement obtained through beam search[8]. In contrast, upward and horizontal relations have lower baseline accuracy, and beam search leads to substantial improvements in these cases. The distance-based analysis in Table 4 reveals a similar pattern. Short-distance relations show high baseline accuracy, resulting in limited improvement, whereas long-distance relations benefit significantly from beam search. These results suggest that beam search mitigates the bias toward downward and short-distance relations, allowing the model to predict correct reading orders even when relations are spatially challenging.

[8]See the Appendix for results on all backbones and decoders.

However, the error analyses in Figures 7–8 revealed that even the models with beam search frequently mispredicted reading orders biased toward downward and short-distance relations. These limitations indicate that further progress in reading order prediction requires decoding strategies that reduce systematic directional and distance biases.

5.3.2. Analysis of Parent-Child Prediction Improvements by BBox Embedding

Tables 5–6 show that accuracy for spatially challenging relations improved more than that for downward or short-distance relations[8]. These results indicate the effectiveness

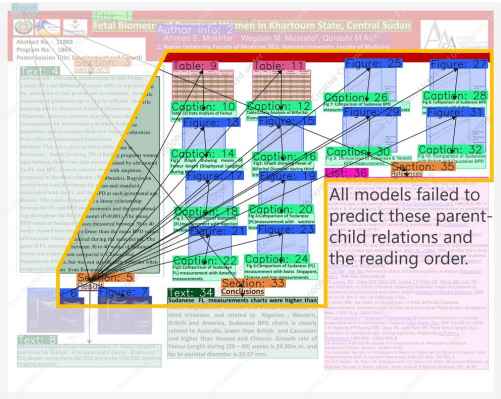


Figure 6. Example of a poster with the GT annotation, where the predicted tree receives a low STEDS (42.70).

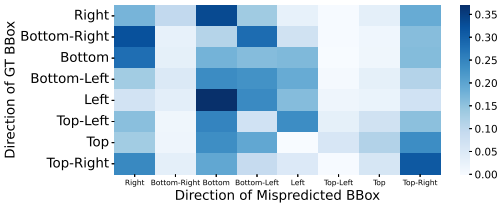


Figure 7. Distribution of reading order prediction failures for each GT BBox direction. Failures from all backbones using DRGG-BEBS are accumulated in this visualization. These failures account for 7.00% of all predictions.

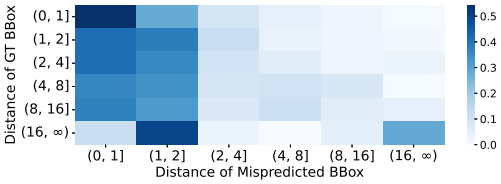


Figure 8. Distribution of reading order prediction failures for each GT BBox distance. These failures account for 7.00% of all predictions. The format follows Fig. 7.

of BBox information in improving performance on such challenging cases.

In contrast, the error analysis in Figures 9–10 revealed that the Root node was frequently mispredicted as the parent, regardless of the correct parent's direction or distance. Figure 11 shows that most of these errors involved nodes such as Text, List, Figure, or Table, which should have had Section as their parent. Our analysis revealed that in 60% of the cases, the correct Section node was present in the constructed tree but not on the rightmost path, and was therefore excluded from the selectable parent candidates. These errors arise from incorrect predictions in earlier steps and are difficult to resolve in subsequent decoding. The observed limitations indicate that progress in parent-child prediction depends on a model architecture capable of capturing not only pairwise relations but also the poster's structural organization.

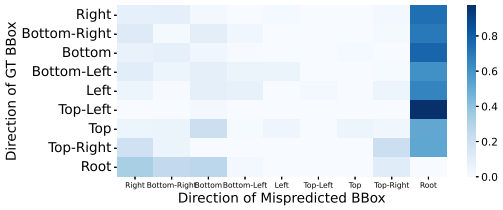


Figure 9. Distribution of parent-child prediction failures for each GT BBox direction. These failures account for 3.59% of all predictions. The format follows Fig. 7.

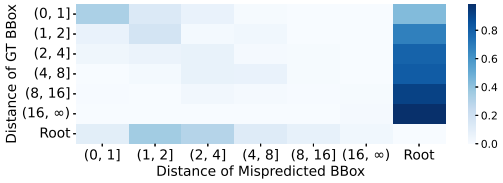


Figure 10. Distribution of parent-child prediction failures for each GT BBox distance. These failures account for 3.59% of all predictions. The format follows Fig. 7.

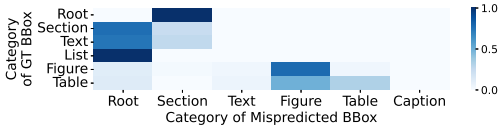


Figure 11. Distribution of parent-child prediction failures for each GT BBox category. These failures account for 3.59% of all predictions. The format follows Fig. 7.

6. Conclusion

We constructed SciPostLayoutTree, a dataset of approximately 8,000 scientific posters annotated with reading order and parent-child relations in the form of DFS-ordered trees. This dataset contains more instances of spatially challenging relations, such as upward, horizontal, and long-distance relations, compared to documents. We extended a relation prediction model that uses visual features and greedy decoding by incorporating the BBox Embedding and beam search. Experimental results show that the BBox Embedding improves the accuracy of spatially challenging parent-child prediction, while beam search improves the accuracy of spatially challenging reading order prediction. Furthermore, error analysis revealed the importance of developing models that better capture the structural organization of posters while mitigating bias toward downward and short-distance relations.

Acknowledgments. This work was supported by the JST Moonshot R&D Program (Grant Number JPMJMS2236) and by JSPS KAKENHI (Grant Number 25H01149).

References

[1] Yufan Chen, Ruiping Liu, Junwei Zheng, Di Wen, Kunyu Peng, Jiaming Zhang, and Rainer Stiefelhagen. Graph-based document structure analysis, 2025. 1, 2, 4

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 4, 6

[3] Hao Feng, Shu Wei, Xiang Fei, Wei Shi, Yingdong Han, Lei Liao, Jinghui Lu, Binghong Wu, Qi Liu, Chunhui Lin, Jingqun Tang, Hao Liu, and Can Huang. Dolphin: Document image parsing via heterogeneous anchor prompting, 2025. 2

[4] Markus Freitag and Yaser Al-Onaizan. Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*. Association for Computational Linguistics, 2017. 5

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 4, 6

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018. 4

[7] Sieun Hyeon, Kyudan Jung, Nam-Joon Kim, Hyun Gon Ryu, and Jaeyoung Do. Mathreader : Text-to-speech for mathematical documents, 2025. 1

[8] Jordy Van Landeghem, Rubén Tito, Łukasz Borchmann, Michał Pietruszka, Paweł Józiak, Rafał Powalski, Dawid Jurkiewicz, Mickaël Coustaty, Bertrand Ackaert, Ernest Valveny, Matthew Blaschko, Sien Moens, and Tomasz Stanisławek. Document understanding dataset and evaluation (dude), 2023. 1

[9] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models, 2016. 5

[10] Junlong Li, Yiheng Xu, Tengchao Lv, Lei Cui, Cha Zhang, and Furu Wei. Dit: Self-supervised pre-training for document image transformer, 2022. 6

[11] Xinze Li, Zhenghao Liu, Chenyan Xiong, Shi Yu, Yu Gu, Zhiyuan Liu, and Ge Yu. Structure-aware language model pretraining improves dense retrieval on structured data, 2023. 1

[12] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection, 2017. 4

[13] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021. 6

[14] Jiefeng Ma, Jun Du, Pengfei Hu, Zhenrong Zhang, Jianshu Zhang, Huihui Zhu, and Cong Liu. Hrdoc: Dataset and baseline method toward hierarchical reconstruction of document structures, 2023. 2, 6

[15] Mateusz Pawlik and Nikolaus Augsten. Efficient computation of the tree edit distance. *ACM Trans. Database Syst.*, 40 (1), 2015. 6

[16] Mateusz Pawlik and Nikolaus Augsten. Tree edit distance: Robust and memory-efficient. *Information Systems*, 56:157–173, 2016. 6

[17] Yu-Ting Qiang, Yan-Wei Fu, Xiao Yu, Yan-Wen Guo, Zhi-Hua Zhou, and Leonid Sigal. Learning to Generate Posters of Scientific Papers by Probabilistic Graphical Models. *Journal of Computer Science and Technology*, 34(1):155–169, 2019. 2

[18] Johannes Rausch, Octavio Martinez, Fabian Bissig, Ce Zhang, and Stefan Feuerriegel. Docparser: Hierarchical structure parsing of document renderings, 2021. 2

[19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016. 4

[20] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks, 2017. 5

[21] Tao Sun, Enhao Pan, Zhengkai Yang, Kaixin Sui, Jiajun Shi, Xianfu Cheng, Tongliang Li, Wenhao Huang, Ge Zhang, Jian Yang, and Zhoujun Li. P2p: Automated paper-to-poster generation and fine-grained benchmark, 2025. 2

[22] Shohei Tanaka, Hao Wang, and Yoshitaka Ushiku. Scipostlayout: A dataset for layout analysis and layout generation of scientific posters. In *35th British Machine Vision Conference 2024, BMVC 2024, Glasgow, UK, November 25-28, 2024*. BMVA, 2024. 2

[23] Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. Hierarchical multimodal transformers for multi-page docvqa, 2023. 1

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 5

[25] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator, 2015. 5

[26] Jiawei Wang, Kai Hu, Zhuoyao Zhong, Lei Sun, and Qiang Huo. Detect-order-construct: A tree construction based approach for hierarchical document structure analysis, 2024. 1, 2, 5, 6

[27] Jiawei Wang, Kai Hu, and Qiang Huo. Unihdsa: A unified relation prediction approach for hierarchical document structure analysis, 2025. 5

[28] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, Xiaogang Wang, and Yu Qiao. Internimage: Exploring large-scale vision foundation models with deformable convolutions, 2023. 4, 6

[29] Zilong Wang, Yiheng Xu, Lei Cui, Jingbo Shang, and Furu Wei. Layoutreader: Pre-training of text and layout for reading order detection, 2021. 2

[30] Yujia Xiao, Shaofei Zhang, Xi Wang, Xu Tan, Lei He, Sheng Zhao, Frank K. Soong, and Tan Lee. Contextspeech: Expressive and efficient text-to-speech for paragraph reading. In *INTERSPEECH 2023*, page 4883–4887. ISCA, 2023. 1

[31] Hangdi Xing, Changxu Cheng, Feiyu Gao, Zirui Shao, Zhi Yu, Jiajun Bu, Qi Zheng, and Cong Yao. DocHieNet: A large and diverse dataset for document hierarchy parsing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1129–1142, Miami, Florida, USA, 2024. Association for Computational Linguistics. 2, 3

[32] Sheng Xu and Xiaojun Wan. Neural Content Extraction for Poster Generation of Scientific Papers, 2021. 2

[33] Chong Zhang, Ya Guo, Yi Tu, Huan Chen, Jinyang Tang, Huijia Zhu, Qi Zhang, and Tao Gui. Reading order matters: Information extraction from visually-rich documents by token path prediction, 2023. 5

[34] Chong Zhang, Yi Tu, Yixi Zhao, Chenshu Yuan, Huan Chen, Yue Zhang, Mingxu Chai, Ya Guo, Huijia Zhu, Qi Zhang, and Tao Gui. Modeling layout reading order as ordering relations for visually-rich document understanding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9658–9678, Miami, Florida, USA, 2024. Association for Computational Linguistics. 2