Appendix of 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

A. Dataset Description

This section presents the annotation guideline and detailed statistics of the dataset.

A.1. Annotation Guideline

The following guideline defines the procedure for annotating SciPostLayout [9] posters as DFS-ordered trees. It was developed through discussions between the in-house supervisors of the external vendor and the authors.

- The Root node represents the poster.
- Each BBox is assigned one of eight categories: Title, Author Info, Section, Text, List, Table, Figure, or Caption.
- Title, Author Info, and Section are always assigned the Root as their parent.
- Text, List, Table, and Figure are assigned the Section that contains them, if such a Section exists; otherwise, their parent is the Root.
- A Caption is assigned its corresponding Figure or Table as its parent.
- When a node has multiple children, they are ordered according to their reading priority. The DFS traversal follows this order.

A.2. Annotation Consistency

This subsection describes the evaluation procedure for the annotation consistency reported in Section 3.2. We evaluated agreement on 100 randomly selected posters from the test set by comparing the original annotations with independently annotated trees by two additional annotators. Agreement was computed for the entire tree, parent–child relations, and reading-order relations. Tree agreement was measured using STEDS defined in Section 5.2. Parent–child agreement was defined as the proportion of nodes whose assigned parent matches between two annotations. Reading-order agreement was defined as the proportion of consecutive node pairs in the DFS traversal that match between two annotations. Each score was computed per poster and averaged over the 100 posters. As a result, STEDS = 0.91, parent–child pair agreement = 0.97, and reading-order pair agreement = 0.94, indicating high agreement.

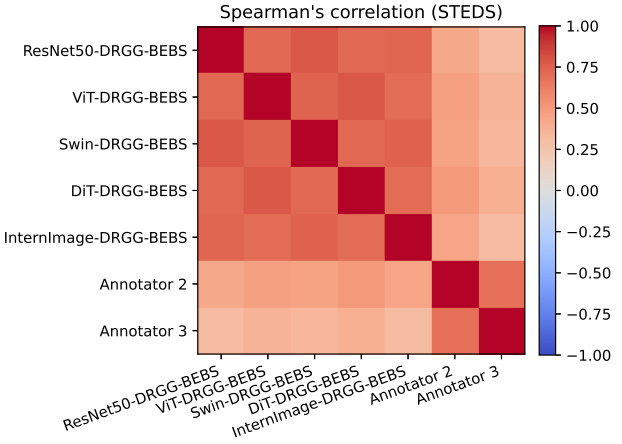


Figure 12. Heatmap of Spearman's rank correlation coefficients between the STEDS of DRGG-BEBS models and those of the annotators.

In addition, both the model STEDS in the evaluation experiments and the inter-annotator STEDS are approximately 0.9, raising the possibility that model errors may stem from inconsistent annotations. To investigate this hypothesis, we computed the Spearman rank correlation between per-poster model STEDS and inter-annotator STEDS. The average correlation is moderate (Spearman's ρ = 0.40), indicating that model errors are not primarily attributable to inconsistent annotations. The full correlation matrix is shown in Figure 12. As shown in the figure, correlations among models and among annotators are high, whereas correlations between models and annotators are low.

A.3. Detailed Statistics of Dataset

This subsection provides detailed statistics that supplement the figures and tables presented in the main paper.

Table 7. Statistics of the tree structures. Tree depth denotes the maximum number of nodes from the root to any leaf. Tree width denotes the maximum number of nodes at any single depth level. Children per node denotes the average number of child nodes per parent node.

Dataset	Tree Depth	Tree Width	Children per Node
SciPostLayoutTree	3.37 (± 0.56)	15.24 (± 7.49)	0.96 (± 2.47)
DocHieNet	3.16 (± 0.90)	9.41 (± 6.60)	0.93 (± 2.78)

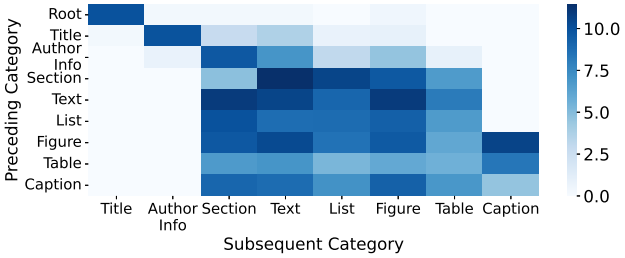


Figure 13. Heatmap of reading order frequencies in SciPostLayoutTree, aggregated by transitions between categories, with counts normalized per 1,000 pages. Heatmap values represent $\log_2(1 + \text{count})$.

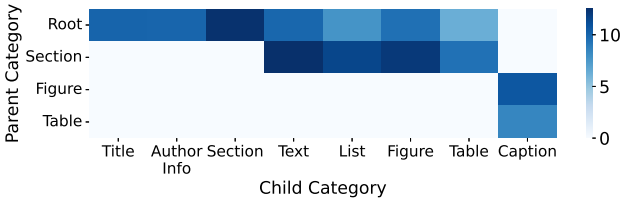


Figure 14. Heatmap of parent-child relation frequencies in SciPostLayoutTree, aggregated by transitions between categories. The format follows Fig. 13.

A.3.1. Layout Tree Statistics

Table 7 summarizes statistics of the tree structures in SciPostLayoutTree and DocHieNet [12]. This result is consistent with the conclusion drawn from Fig. 2.

A.3.2. Reading Order Statistics

Tables 8–9 provide the underlying counts for the heatmaps shown in Fig. 3.

The category transition patterns in Figure 13 and Table 10 indicate a consistent introductory sequence (Root → Title → Author Info), followed by diverse transitions such as Section → Text and Text → Figure.

A.3.3. Parent-Child Relation Statistics

Tables 11–12 provide the underlying counts for the heatmaps shown in Fig. 4.

Figure 14 and Table 13 reveal that the category transition patterns follow the annotation guideline, for example in transitions such as Section → Text and Figure → Caption, indicating that parent-child relations depend on the categories.

Table 8. Reading order frequency in SciPostLayoutTree, aggregated by direction and normalized distance bins, with counts normalized per 1,000 pages.

Direction \ Distance	(0, 1]	(1, 2]	(2, 4]	(4, 8]	(8, 16]	(16, ∞)
Right	2,143.97	889.16	41.15	5.48	0.38	0.00
Bottom-Right	2,045.61	150.72	23.57	7.64	4.46	0.76
Bottom	7,800.48	1,913.36	200.03	78.74	48.16	22.93
Bottom-Left	2,602.24	483.50	78.99	22.81	13.38	5.35
Left	1,568.61	524.53	247.29	57.20	4.59	0.38
Top-Left	130.97	14.27	8.28	7.01	4.20	1.15
Top	390.50	35.42	72.37	120.52	78.74	96.70
Top-Right	390.24	514.33	261.18	190.60	117.08	117.47

Table 9. Reading order frequency in DocHieNet, aggregated by direction and normalized distance bins, with counts normalized per 1,000 pages.

Direction \ Distance	(0, 1]	(1, 2]	(2, 4]	(4, 8]	(8, 16]	(16, ∞)
Right	1,358.60	412.46	127.70	49.80	17.91	3.20
Bottom-Right	1,207.44	219.20	50.32	13.01	9.87	7.97
Bottom	3,322.14	2,036.66	192.67	58.62	43.13	52.41
Bottom-Left	1,177.77	246.45	57.71	20.06	12.42	11.11
Left	1,136.85	398.86	128.10	35.49	18.10	5.36
Top-Left	10.98	28.56	11.50	7.06	5.82	4.31
Top	35.88	11.37	19.02	26.40	26.53	27.45
Top-Right	30.39	63.59	34.31	32.81	15.69	7.91

Table 10. Reading order frequency in SciPostLayoutTree, aggregated by transitions between categories, with counts normalized per 1,000 pages.

Preceding \ Subsequent	Title	Author Info	Section	Text	List	Figure	Table	Caption
Root	998.85	0.25	0.25	0.25	0.00	0.38	0.00	0.00
Title	0.25	978.98	6.24	11.98	0.76	0.89	0.00	0.00
Author Info	0.00	0.76	820.61	125.62	7.39	23.19	0.89	0.00
Section	0.00	0.00	26.88	2,813.73	1,498.53	802.78	101.54	0.00
Text	0.00	0.00	1,946.24	1,491.02	538.03	1,991.97	283.48	0.00
List	0.00	0.00	1,000.89	427.83	439.55	646.83	101.67	0.00
Figure	0.00	0.00	826.22	1,274.30	364.12	766.85	65.23	1,555.61
Table	0.00	0.00	106.77	128.55	38.35	64.85	47.39	327.81
Caption	0.00	0.00	530.39	444.13	140.53	623.26	120.27	23.32

Table 11. Parent-child relation frequency in SciPostLayoutTree, aggregated by direction and normalized distance bins, with counts normalized per 1,000 pages.

Direction \ Distance	(0, 1]	(1, 2]	(2, 4]	(4, 8]	(8, 16]	(16, ∞)
Right	1,173.53	568.73	312.01	89.69	10.19	0.51
Bottom-Right	2,256.08	734.74	501.08	269.08	121.29	57.84
Bottom	3,367.94	1,016.94	1,144.86	851.83	495.35	329.72
Bottom-Left	565.17	199.77	130.34	70.96	50.07	25.35
Left	159.38	67.40	16.94	3.19	0.00	0.00
Top-Left	114.79	3.95	1.27	1.15	0.00	0.00
Top	369.35	6.24	2.42	4.71	2.42	2.04
Top-Right	42.81	140.02	85.36	41.92	15.93	15.03

Table 12. Parent-child relation frequency in DocHieNet, aggregated by direction and normalized distance bins, with counts normalized per 1,000 pages.

Direction \ Distance	(0, 1]	(1, 2]	(2, 4]	(4, 8]	(8, 16]	(16, ∞)
Right	764.85	211.82	27.78	3.07	0.13	0.00
Bottom-Right	1,054.83	223.65	223.97	260.83	199.53	85.62
Bottom	523.82	365.73	471.28	564.41	549.51	480.10
Bottom-Left	219.01	58.04	63.66	80.58	68.88	42.28
Left	118.95	16.08	3.66	0.07	0.00	0.00
Top-Left	3.86	0.46	0.33	0.07	0.00	0.00
Top	25.95	2.29	1.83	1.57	1.50	3.01
Top-Right	9.74	104.76	30.06	27.25	31.24	32.09

Table 13. Parent-child relaiton frequency in SciPostLayoutTree, aggregated by transitions between categories, with counts normalized per 1,000 pages.

Parent \ Child	Title	Author Info	Section	Text	List	Figure	Table	Caption
Root	999.11	980.00	5,264.49	902.41	206.14	669.51	75.42	0.00
Section	0.00	0.00	0.00	5,815.01	2,821.12	4,251.50	645.05	0.00
Figure	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1,577.27
Table	0.00	0.00	0.00	0.00	0.00	0.00	0.00	329.47

B. Experimental Details

This section provides the implementation details and the experimental setup.

B.1. Implementation Details

We conducted all experiments using publicly available pretrained visual backbones, including ResNet-50 [3], ViT-Base [2], Swin-Base [5], DiT-Base [4], and InternImage-Base [11]. All backbone parameters were fine-tuned during training. All model parameters, except those in the pretrained backbones, were initialized using PyTorch's default initialization schemes. Random seeds were not explicitly fixed, and deterministic computation settings were not enforced; therefore, results may exhibit minor variations across runs due to stochastic training dynamics.

We employed the AdamW optimizer [7] with an initial learning rate of 2×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 0.05. A uniform learning rate scaling factor of 1.0 was applied to all layers. The learning rate was scheduled using a linear warmup [10] over the first 500 iterations with a warmup factor of 0.01, followed by cosine decay [6] to zero over the remaining 10,500 iterations, totaling 11,000 training steps (approximately 24 epochs). Gradient clipping with a maximum norm of 1.0 and L^2-norm type was applied to stabilize training.

Training was performed on 8 GPUs with a total batch size of 16, corresponding to 2 samples per GPU. All hyperparameters were selected based on performance on the validation set. Each model was evaluated on the test set in a single run with the final model checkpoint. Statistical significance testing was performed to assess performance differences.

B.2. Experimental Setup

All experiments were conducted on a computing node equipped with 8 NVIDIA A100-SXM4-80GB GPUs (total 640 GB GPU memory) and 2 AMD EPYC 7713 64-core processors, resulting in 128 physical CPU cores. The system had 2.0 TiB of RAM, of which approximately 1.9 TiB was available. The operating system was Ubuntu 22.04.5 LTS (Jammy Jellyfish). The GPU driver version was 535.161.08 with CUDA 12.2, while PyTorch was built with CUDA 12.1 support. The software environment consisted of Python 3.10.10, PyTorch 2.5.1+cu121, transformers 4.52.3, detectron2 0.6, and detrex 0.3.0.

C. Additional Experiments

This section presents additional figures, tables, and experiments that supplement the analysis in the main paper.

C.1. Effect of Text Embedding

We explored a model that incorporates textual content in posters as additional features. For each BBox b_i other than the Figure BBoxes, OCR text t_i was extracted using Tesseract [8]. Each t_i was then encoded into a text feature vector $\mathbf{u_i}$ using a pretrained SciBERT model [1], where $\mathbf{u_i}$ denotes the mean-pooled output of the last layer. For the Figure BBoxes, the special token `[FIGURE]` was fed into SciBERT to obtain corresponding text features. The text feature of the Root node, $\mathbf{u_0}$, was set to the zero vector. Finally, each text feature $\mathbf{u_i}$ was concatenated with the corresponding visual feature $\mathbf{v_i^r}$ or $\mathbf{v_i^c}$, as well as the BBox features $\mathbf{z_i}$ and $\mathbf{e_i}$, to obtain the multimodal features as follows:

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

The subsequent procedure is identical to that in the main paper.

We compare the models evaluated in the main paper with additional variants incorporating the text features.

DRGG-TE (Text Embedding) augments the input features with the text features.

DRGG-TEBS (Text Embedding and Beam Search) integrates DRGG-TE and DRGG-BS.

DRGG-BETE (BBox Embedding and Text Embedding) integrates DRGG-BE and DRGG-TE.

DRGG-BETEBS (BBox Embedding, Text Embedding, and Beam Search) integrates DRGG-BE, DRGG-TE, and DRGG-BS.

Table 14 shows that models incorporating text features perform comparably to, or slightly worse than, those using only visual and BBox features. This observation may be attributed to the following factors. First, humans typically interpret the layout structure of a poster prior to reading its textual content. Analogously, even when the language is unreadable, humans can often infer the overall layout structure if basic properties such as reading direction (e.g., left-to-right) are known. These considerations suggest that the contribution of textual content to structural analysis for posters is limited. Second, as shown in Table 1, scientific posters often contain many figures, which reduces the importance of textual content. Third, the current models predict reading order and parent-child relations using greedy or beam search, which are inherently limited to local decisions based on one or two decoding steps. Discourse-level structural understanding based on textual content requires a lookahead decoding strategy that can capture longer dependencies.

Table 14. Comparison of DRGG, its extensions, and their text-feature-enhanced variants across visual backbones. $*$ and $**$ indicate significant improvements over DRGG at $p < 0.005$ and $p < 0.05$, respectively, 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*
	DRGG-TE	71.21*	77.23*	8.21*
	DRGG-TEBS	78.46*	84.19*	6.15*
	DRGG-BETE	82.66*	85.08*	4.90*
	DRGG-BETEBS	87.15*	89.20*	3.59*
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*
	DRGG-TE	80.32	86.50	5.56
	DRGG-TEBS	83.35*	89.33*	4.73*
	DRGG-BETE	85.74*	87.82*	4.00*
	DRGG-BETEBS	89.04*	90.93*	3.06*
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*
	DRGG-TE	80.31	85.77	5.52
	DRGG-TEBS	84.18*	89.51*	4.46*
	DRGG-BETE	84.72*	86.51	4.22*
	DRGG-BETEBS	88.37*	90.11*	3.23*
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*
	DRGG-TE	79.25	85.52	5.83
	DRGG-TEBS	82.02*	88.40*	5.13*
	DRGG-BETE	85.56*	87.71*	4.10*
	DRGG-BETEBS	88.81*	90.82*	3.16*
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*
	DRGG-TE	81.62**	86.74	5.15
	DRGG-TEBS	85.05*	90.02*	4.21*
	DRGG-BETE	84.69*	86.63	4.26*
	DRGG-BETEBS	88.19*	89.94*	3.24*

C.2. VLM Results

We explored PromptTree, a model for decoding DFS-ordered trees using the latest VLMs such as GPT-5[1] and Gemini 3 Pro[2]. Figure 15 provides an overview of Prompt-

[1] https://openai.com/gpt-5/

[2] https://deepmind.google/models/gemini/pro/

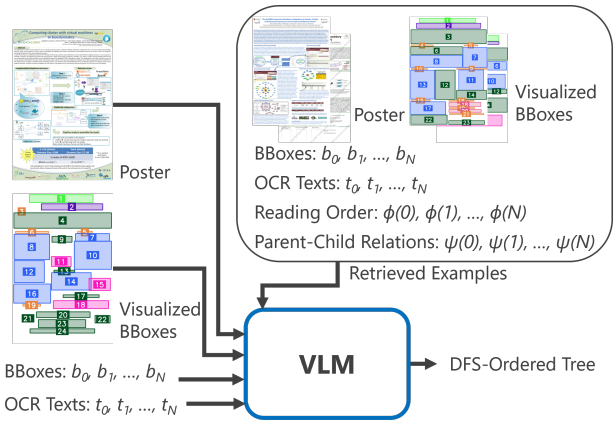


Figure 15. Overview of PromptTree.

Table 15. Comparison between DRGG variants and PromptTree, a VLM-based model.

Backbone	Decoder	STEDS (↑)	REDS (↑)	TED (↓)
ResNet-50	DRGG	68.74	75.07	8.83
	DRGG-BEBS	88.45	90.40	3.22
ViT	DRGG	80.40	85.94	5.45
	DRGG-BEBS	90.04	91.73	2.78
Swin	DRGG	79.90	85.77	5.69
	DRGG-BEBS	89.26	90.88	2.95
DiT	DRGG	78.33	84.81	6.09
	DRGG-BEBS	88.69	90.63	3.24
InternImage	DRGG	80.19	85.53	5.53
	DRGG-BEBS	89.34	90.97	2.93
GPT-5		76.96	83.13	6.34
Gemini 3 Pro		81.96	89.66	5.01

Tree. The model takes as input the following components: the poster image; a visualization of the BBoxes rendered on a white canvas of the same size as the poster, where BBoxes are color-coded by category and assigned indices after sorting in y–x order; the metadata of each BBox (including position, category, and BBox index); and OCR-extracted text. In addition, two retrieved examples are provided as supplementary inputs. These examples are selected from the training set by ranking candidate examples in descending order of Intersection over Union (IoU), which is computed via Hungarian algorithm between the BBoxes of the target image and those of each candidate. Each retrieved example contains the same input information as the target image, along with the corresponding ground-truth reading order and parent–child relations. The prompt template provided to the VLM is shown in Prompt 1.

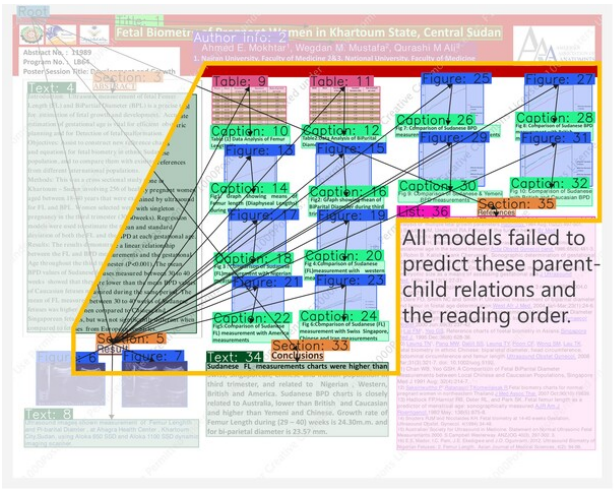
Table 15 presents a comparison between DRGG variants and PromptTree. PromptTree shows lower performance than all DRGG-BEBS variants when evaluated with either GPT-5 or Gemini 3 Pro, indicating that poster structure analysis remains challenging for the latest VLMs.

C.3. Additional Challenging Examples

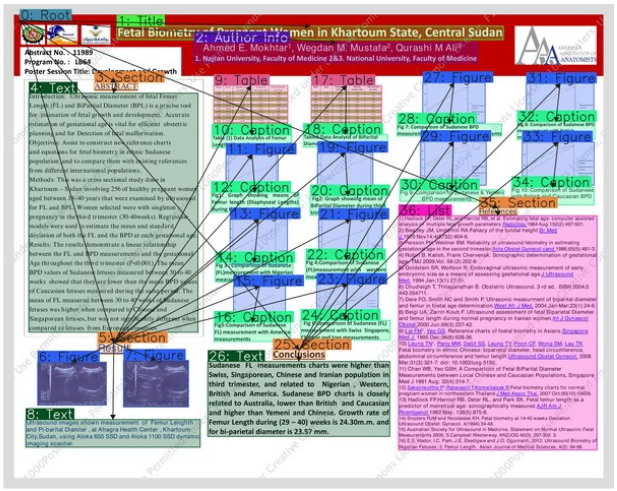
Figures 16–19 present additional challenging examples as a supplement to Fig. 6. In all cases, the model fails to predict inter-element relations due to irregular spatial arrangements. Accurate prediction of such relations requires capturing semantic grouping and structural plausibility over longer sequences.

```
You are an expert in document structure analysis.
You are given bounding boxes from a scientific poster and one or more images of
the poster.
Your task is to infer:
1) the reading order of the bboxes,
2) the parent-child relationships forming a rooted tree.

Example 1:

Input bboxes:
- bbox_number=<bbox_id>, category=<category_name>, x=<x_norm>, y=<y_norm>,
w=<w_norm>, h=<h_norm>, text="<ocr_text_if_any>"
- ...

Images for Example 1:
[POSTER_IMAGE]
[BBOXES_IMAGE]

Output:
reading_order = <reading_order_list>
tree = <tree_list_of_{bbox_number,parent}_objects>

Example 2:

Input bboxes:
- bbox_number=<bbox_id>, category=<category_name>, x=<x_norm>, y=<y_norm>,
w=<w_norm>, h=<h_norm>, text="<ocr_text_if_any>"
- ...

Images for Example 2:
[POSTER_IMAGE]
[BBOXES_IMAGE]

Output:
reading_order = <reading_order_list>
tree = <tree_list_of_{bbox_number,parent}_objects>

Now solve the following example.

Input bboxes:
- bbox_number=<bbox_id>, category=<category_name>, x=<x_norm>, y=<y_norm>,
w=<w_norm>, h=<h_norm>, text="<ocr_text_if_any>"
- ...

Use both the textual description of the bboxes and the visual layout in the
images.
Output the result in the same format as the previous examples, using JSON only.
Do not include any explanations.

Images for the target example:
[POSTER_IMAGE]
[BBOXES_IMAGE]
```

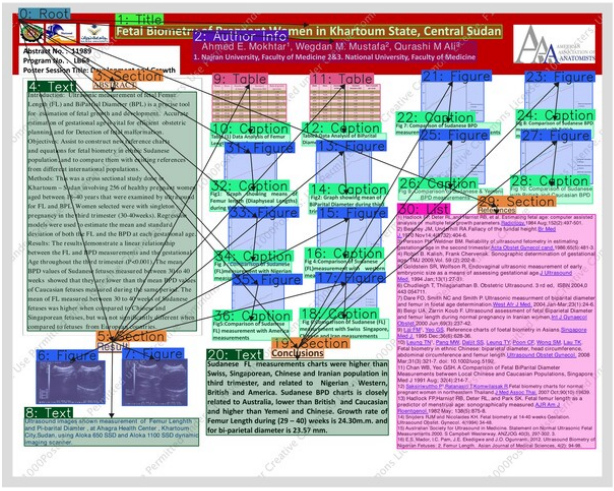
Prompt 1. Prompt used for decoding a DFS-ordered tree with PromptTree.



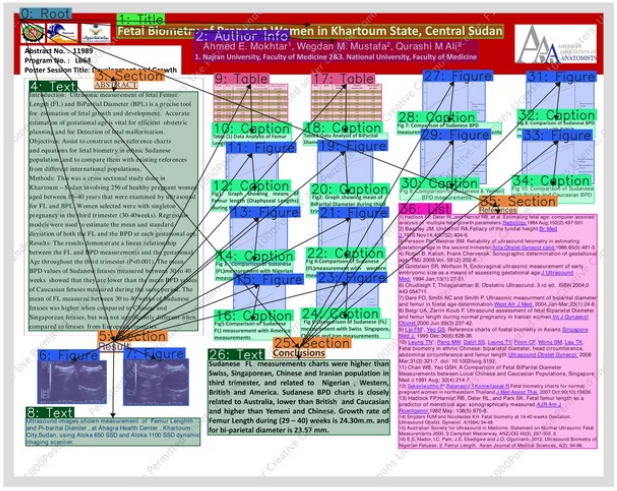
(a) GT Annotation



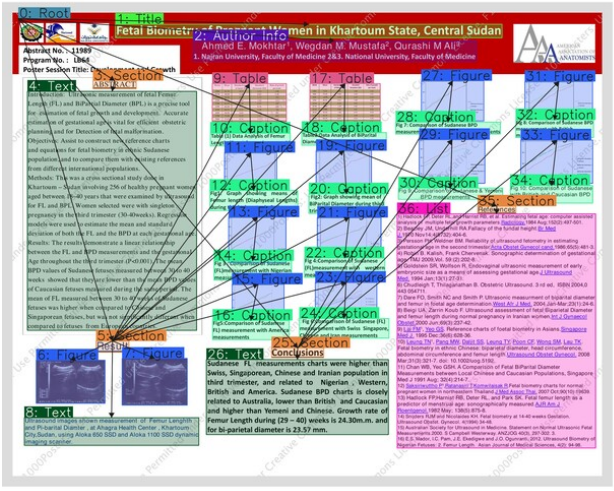
(b) ResNet50-DRGG-BEBS



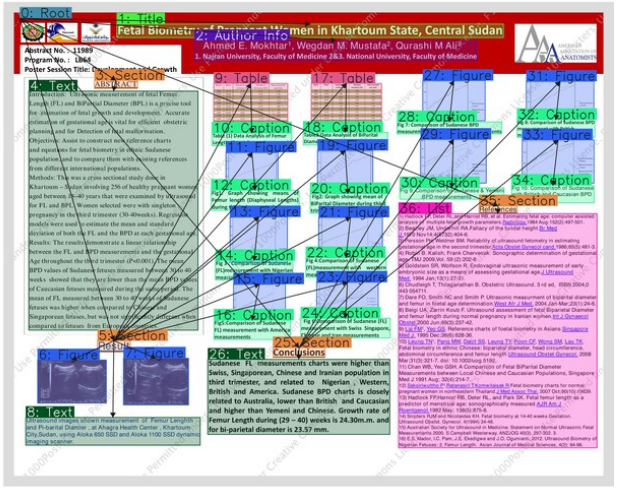
(c) ViT-DRGG-BEBS



(d) Swin-DRGG-BEBS

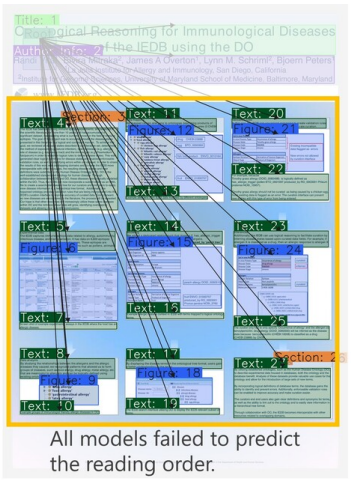


(e) DiT-DRGG-BEBS

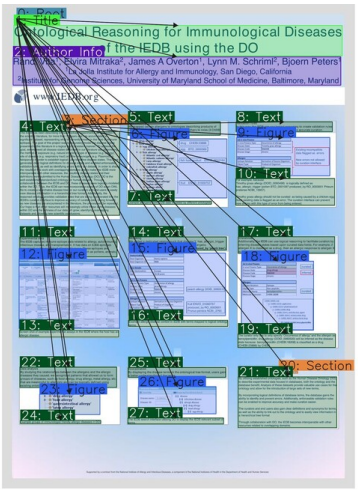


(f) InternImage-DRGG-BEBS

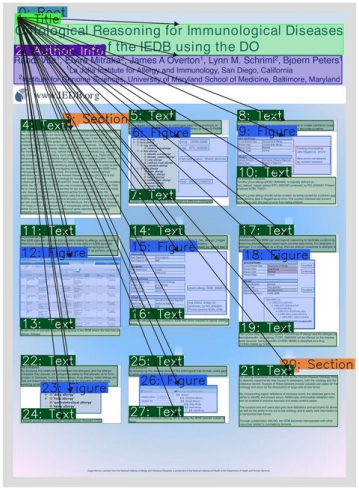
Figure 16. Example of a poster with the GT annotation and the predicted trees. The predicted trees received low STEDS (42.70).



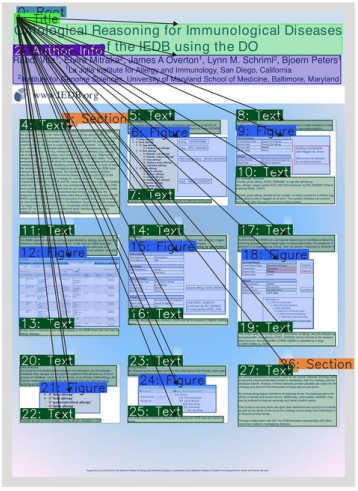
(a) GT Annotation



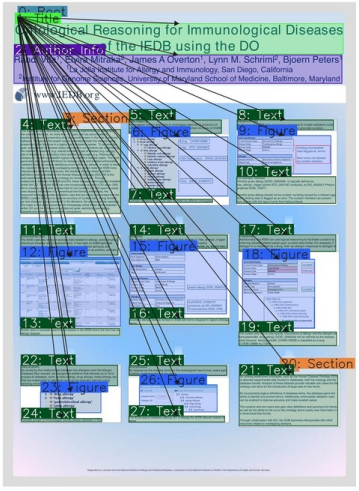
(b) ResNet50-DRGG-BEBS



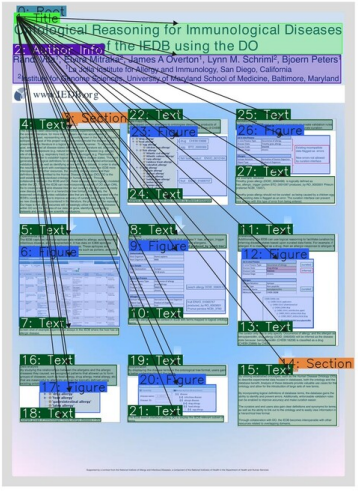
(c) ViT-DRGG-BEBS



(d) Swin-DRGG-BEBS

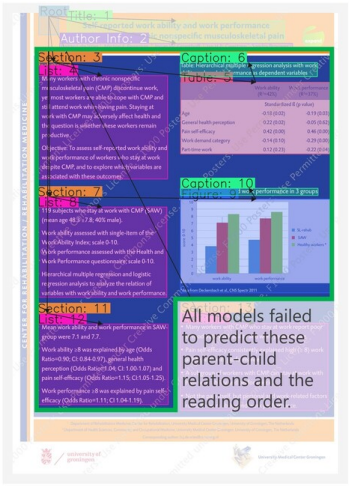


(e) DiT-DRGG-BEBS

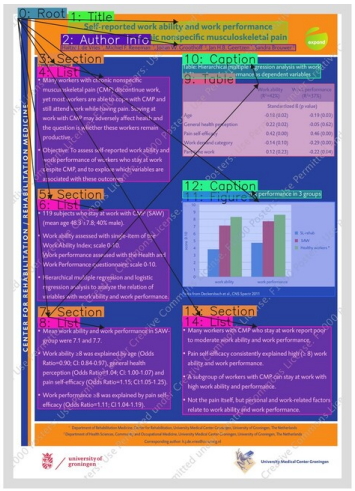


(f) InternImage-DRGG-BEBS

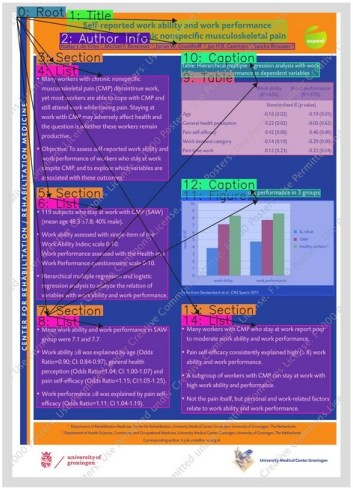
Figure 17. Example of a poster with the GT annotation and the predicted trees. The predicted trees received low STEDS (25.71).



(a) GT Annotation

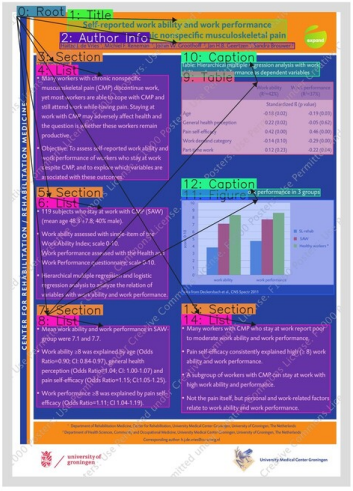


(b) ResNet50-DRGG-BEBS

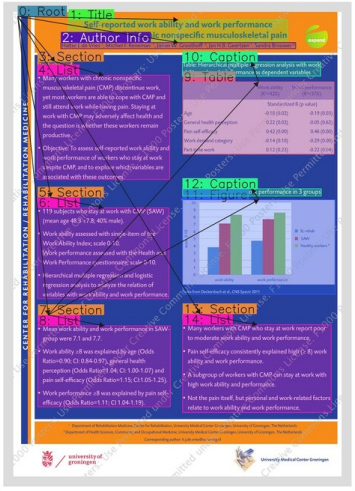


(c) ViT-DRGG-BEBS

(d) Swin-DRGG-BEBS

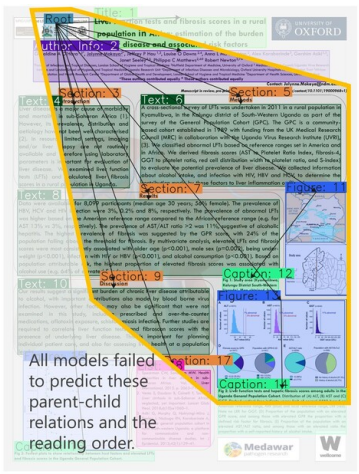


(e) DiT-DRGG-BEBS

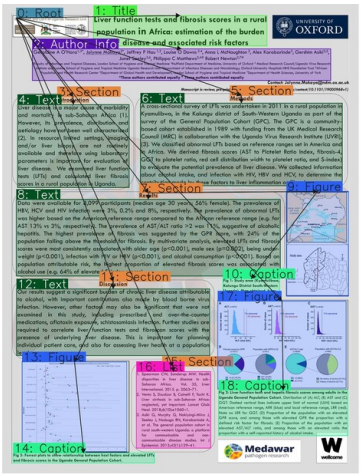


(f) InternImage-DRGG-BEBS

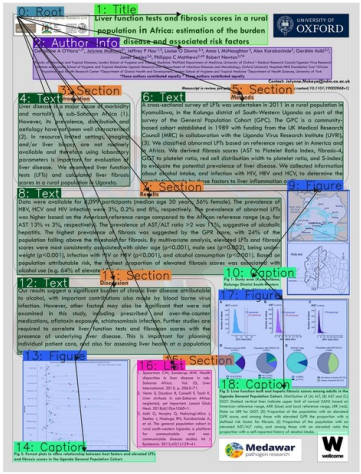
Figure 18. Example of a poster with the GT annotation and the predicted trees. The predicted trees received low STEDS (46.67).



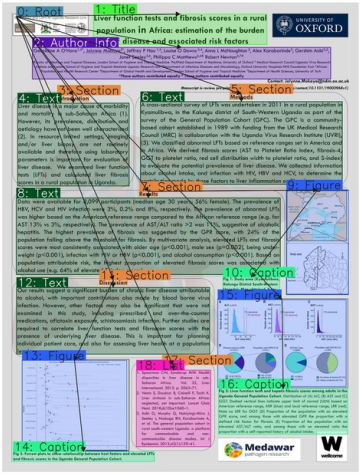
(a) GT Annotation



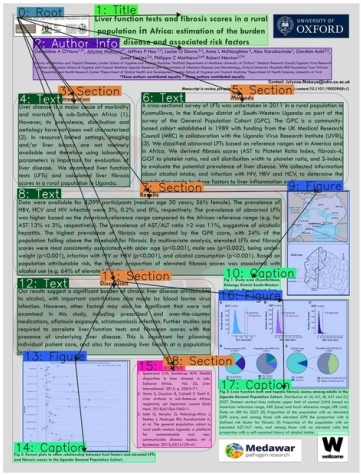
(b) ResNet50-DRGG-BEBS



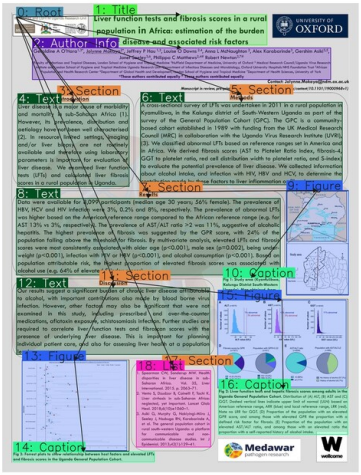
(c) ViT-DRGG-BEBS



(d) Swin-DRGG-BEBS



(e) DiT-DRGG-BEBS



(f) InternImage-DRGG-BEBS

Figure 19. Example of a poster with the GT annotation and the predicted trees. The predicted trees received low STEDS (53.68).

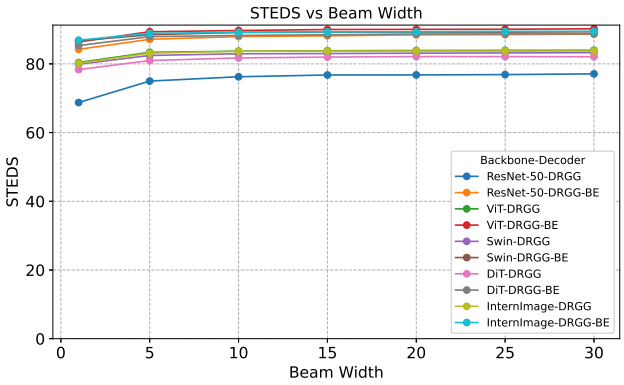


Figure 20. Effect of beam width on STEDS performance

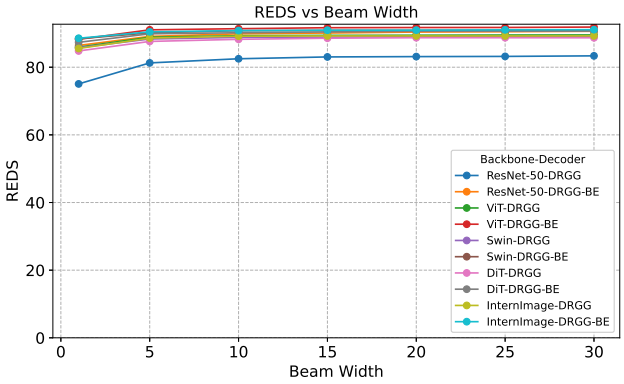


Figure 21. Effect of beam width on REDS performance

C.4. Supplement to Analysis of Reading Order Prediction Improvements by Beam Search

Tables 16–17 extend Tables 3–4 by including the results on all backbones and decoders. These results are consistent with the conclusion drawn from Tables 3–4.

C.5. Supplement to Analysis of Parent-Child Prediction Improvements by BBox Embedding

Tables 18–19 extend Tables 5–6 by including the results on all backbones and decoders. These results are consistent with the conclusion drawn from Tables 5–6.

C.6. Effect of Beam Width

Figures 20–22 show the performance variations of each evaluation metric when the beam width is varied among $\{1, 5, 10, 15, 20, 25, 30\}$. Across all metrics, the performance saturates around a beam width of 15 to 20. This behavior can be attributed to the fact that the average number of BBoxes in SciPostLayoutTree is approximately 25. Increasing the beam width beyond this point results in only marginal performance improvements.

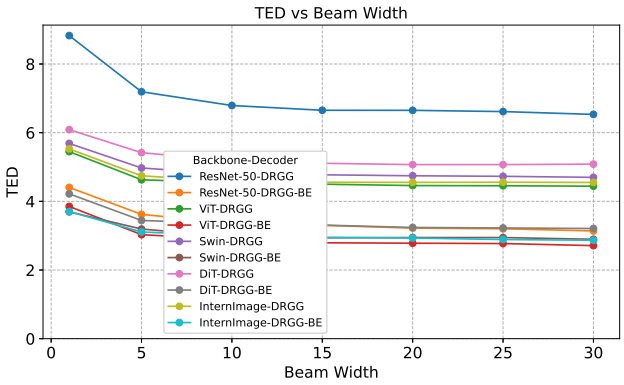


Figure 22. Effect of beam width on TED performance

C.7. Histogram of Scores

Figures 23–25 show histograms of each evaluation metric with a bin width of 10. Although the models often decode trees similar to the GT trees, low scores are still observed in a non-negligible number of cases. These results indicate that the models remain suboptimal.

Table 16. Per-direction accuracy of the reading order prediction with DRGG-BS or DRGG-BEBS. Values in parentheses indicate the accuracy improvement over DRGG or DRGG-BE. $\star$ and $\star\star$ indicate significant improvements over DRGG or DRGG-BE at $p < 0.005$ and $p < 0.05$, respectively, according to McNemar's test.

Backbone	Decoder	Right	Bottom-Right	Bottom	Bottom-Left	Left	Top-Left	Top	Top-Right
ResNet-50	DRGG-BS	79.4 (+5.0)*	91.5 (+1.7)**	93.9 (+4.0)*	87.7 (+6.1)*	77.8 (+6.5)*	58.3 (+0.0)	73.6 (+5.8)*	67.4 (+8.2)*
	DRGG-BEBS	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	DRGG-BS	84.6 (+2.9)*	93.6 (+1.2)**	95.3 (+1.8)*	91.3 (+2.6)*	82.8 (+2.1)*	63.3 (+5.0)	74.5 (+4.1)*	78.0 (+5.2)*
	DRGG-BEBS	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	DRGG-BS	85.6 (+4.0)*	93.9 (+1.3)*	95.2 (+1.4)*	91.7 (+2.6)*	83.5 (+3.2)*	60.0 (+5.0)	77.1 (+4.1)**	77.2 (+4.7)*
	DRGG-BEBS	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	DRGG-BS	82.4 (+3.1)*	91.7 (+0.8)	94.5 (+1.8)*	90.1 (+3.4)*	81.6 (+3.8)*	50.0 (-5.0)	69.6 (+2.9)	75.8 (+5.9)*
	DRGG-BEBS	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	DRGG-BS	85.1 (+2.6)*	93.4 (+0.2)	95.8 (+1.5)*	92.1 (+2.4)*	83.4 (+2.1)*	63.3 (+3.3)	80.9 (+5.2)*	77.8 (+4.7)*
	DRGG-BEBS	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 17. Per-distance accuracy of the reading order prediction. The format and experimental settings follow Table 16.

Backbone	Decoder	(0, 1]	(1, 2]	(2, 4]	(4, 8]	(8, 16]	(16, ∞)
ResNet-50	DRGG-BS	91.3 (+4.1)*	76.5 (+5.9)*	62.1 (+6.7)*	71.7 (+13.1)*	71.0 (+9.9)*	78.9 (+8.1)**
	DRGG-BEBS	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	DRGG-BS	93.2 (+2.0)*	82.6 (+2.7)*	72.1 (+4.4)*	83.8 (+8.6)*	77.1 (+3.1)	85.4 (+3.3)
	DRGG-BEBS	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	DRGG-BS	93.3 (+2.0)*	83.2 (+3.1)*	74.1 (+4.7)*	81.3 (+6.1)**	76.3 (+3.8)	87.8 (+4.9)
	DRGG-BEBS	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	DRGG-BS	91.8 (+2.0)*	80.6 (+4.1)*	72.5 (+6.0)*	77.3 (+4.0)	74.8 (+3.1)	86.2 (+2.4)
	DRGG-BEBS	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	DRGG-BS	93.7 (+1.5)*	83.3 (+3.6)*	74.9 (+3.1)	84.8 (+6.1)*	76.3 (+1.5)	87.0 (+5.7)**
	DRGG-BEBS	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 18. Per-direction accuracy of the parent-child prediction with DRGG-BE or DRGG-BEBS. Values in parentheses indicate the accuracy improvement over DRGG or DRGG-BS. $\star$ and $\star\star$ indicate significant improvements over DRGG or DRGG-BS at $p < 0.005$ and $p < 0.05$, respectively, according to McNemar's test.

Backbone	Decoder	Right	Bottom-Right	Bottom	Bottom-Left	Left	Top-Left	Top	Top-Right
ResNet-50	DRGG-BE	89.2 (+18.2)*	91.4 (+12.1)*	95.7 (+11.5)*	90.2 (+19.5)*	69.5 (+16.3)*	88.4 (+18.6)**	86.5 (+18.0)*	82.4 (+22.8)*
	DRGG-BEBS	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	DRGG-BE	91.3 (+5.2)*	94.2 (+4.8)*	96.6 (+5.9)*	91.7 (+11.0)*	75.2 (+12.1)*	83.7 (+9.3)	89.9 (+26.4)*	83.4 (+15.0)*
	DRGG-BEBS	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	DRGG-BE	92.0 (+9.3)*	94.6 (+6.1)*	96.9 (+7.2)*	89.4 (+9.1)*	70.2 (+6.4)**	88.4 (+20.9)*	91.6 (+24.7)*	90.7 (+20.7)*
	DRGG-BEBS	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	DRGG-BE	89.2 (+7.6)*	93.2 (+7.0)*	96.0 (+7.4)*	91.1 (+11.3)*	73.0 (+9.9)**	88.4 (+25.6)*	88.2 (+34.8)*	78.2 (+9.8)*
	DRGG-BEBS	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	DRGG-BE	89.9 (+5.2)*	94.2 (+5.5)*	97.2 (+6.7)*	91.1 (+10.2)*	70.9 (+4.3)	83.7 (+11.6)	87.6 (+16.3)*	89.1 (+19.2)*
	DRGG-BEBS	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 19. Per-distance accuracy of the parent-child prediction. The format and experimental settings follow Table 18.

Backbone	Decoder	(0, 1]	(1, 2]	(2, 4]	(4, 8]	(8, 16]	(16, ∞)
ResNet-50	DRGG-BE	96.3 (+7.7)*	88.0 (+17.4)*	86.2 (+22.1)*	88.7 (+21.4)*	86.4 (+22.5)*	88.3 (+32.2)*
	DRGG-BEBS	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	DRGG-BE	97.2 (+5.7)*	89.8 (+4.7)*	90.7 (+8.7)*	91.8 (+8.5)*	87.0 (+8.2)*	91.2 (+23.9)*
	DRGG-BEBS	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	DRGG-BE	97.2 (+5.6)*	90.5 (+8.5)*	91.4 (+10.7)*	90.7 (+9.7)*	90.8 (+15.8)*	91.7 (+26.8)*
	DRGG-BEBS	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	DRGG-BE	96.5 (+7.5)*	88.1 (+6.6)*	88.7 (+10.8)*	90.5 (+8.8)*	88.6 (+14.9)*	87.8 (+15.6)*
	DRGG-BEBS	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	DRGG-BE	97.0 (+4.8)*	90.3 (+7.1)*	90.0 (+8.9)*	91.5 (+8.7)*	89.9 (+12.7)*	91.7 (+23.9)*
	DRGG-BEBS	98.4 (+5.2)*	93.1 (+7.1)*	93.5 (+8.0)*	94.6 (+7.5)*	95.6 (+13.3)*	95.6 (+17.6)*

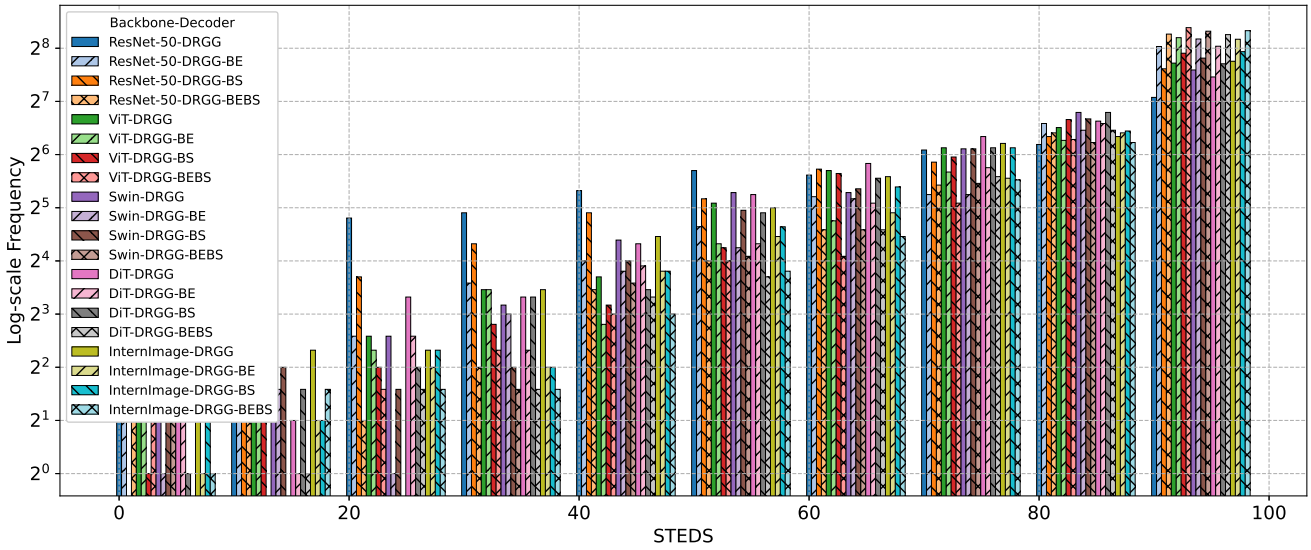


Figure 23. Histogram of STEDS with a bin width of 10

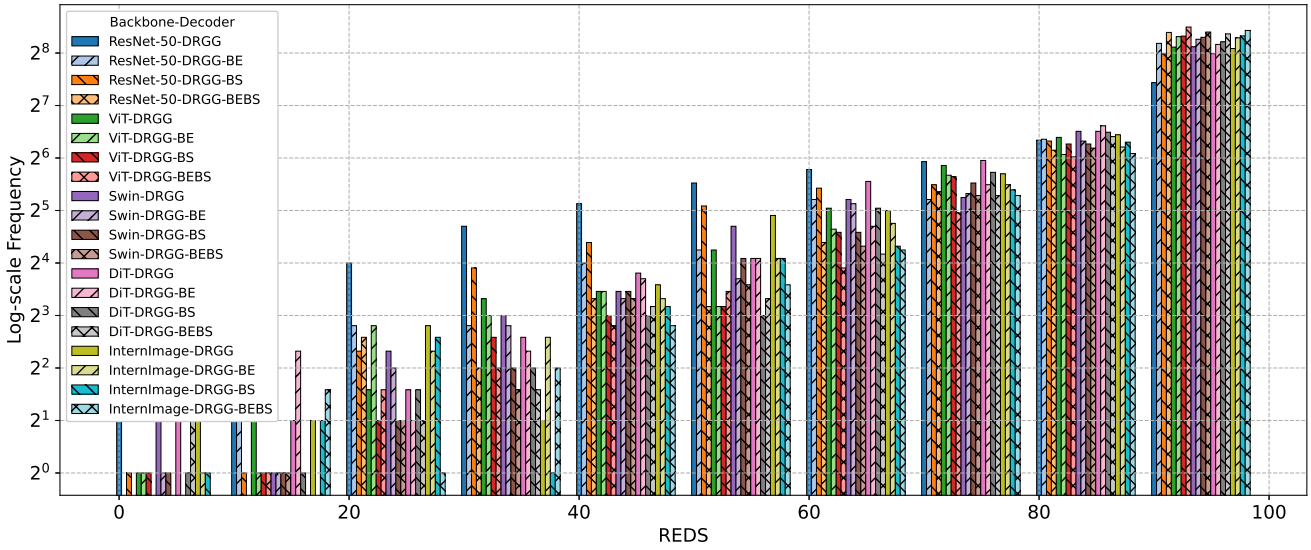


Figure 24. Histogram of REDS with a bin width of 10

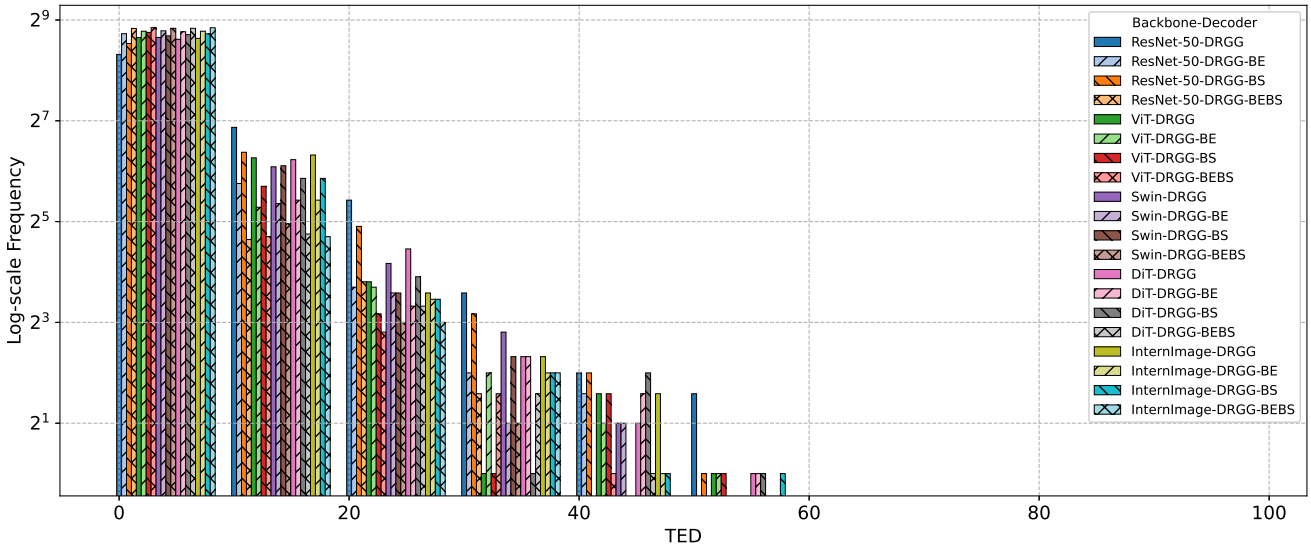


Figure 25. Histogram of TED with a bin width of 10

References

[1] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text, 2019. 6

[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. 5

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

[4] Junlong Li, Yiheng Xu, Tengchao Lv, Lei Cui, Cha Zhang, and Furu Wei. Dit: Self-supervised pre-training for document image transformer, 2022. 5

[5] 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. 5

[6] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2017. 5

[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 5

[8] R. Smith. An overview of the tesseract ocr engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, pages 629–633, 2007. 6

[9] 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. 1

[10] 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

[11] 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. 5

[12] 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