DocLayout-YOLO: Axis-Attentive Document Layout Detector with Mesh-Guided Synthetic Pretraining

Anonymous authors
Paper under double-blind review

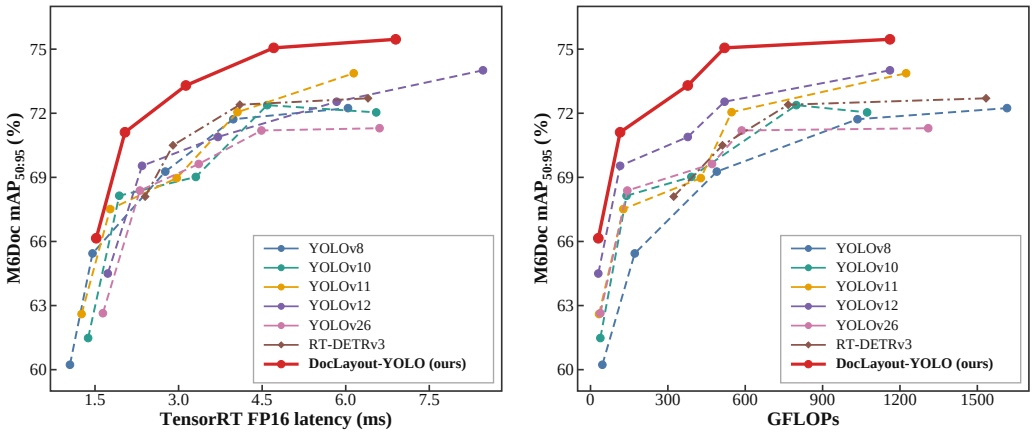


Figure 1: Comparisons with popular real-time detectors in terms of latency-accuracy (left) and FLOPs-accuracy (right) trade-offs on M^{6}Doc.

Abstract

Accurate and efficient document layout analysis is critical for modern document parsing systems. However, although the YOLO framework represents a leading paradigm for the real-time detector, its adaptation to heterogeneous documents remains underexplored. To address this gap, we propose DocLayout-YOLO, a document-specialized real-time detector that integrates diverse synthetic document pretraining and page-axis context modeling. From the data perspective, we introduce MeshLayout-1M, a synthetic pretraining corpus generated by the proposed Mesh-candidate BestFit algorithm, which provides large-scale and diverse document layouts for pretraining and improves adaptation to heterogeneous document scenarios. From the model perspective, we propose Page-Axis Layout Attention (PALA), which captures long-range dependencies along the horizontal and vertical page axes while remaining computationally efficient. Extensive experiments demonstrate that DocLayout-YOLO achieves state-of-the-art performance on four heterogeneous document benchmarks, obtaining 72.4%/75.5%/96.0%/73.1% mAP$_{50:95}$ on D^{4}LA, M^{6}Doc, HJ-Dataset, and PRIMA-LAD, respectively. Compared with existing real-time detectors, DocLayout-YOLO establishes a superior accuracy-efficiency trade-off in document layout analysis.

1 Introduction

Document Layout Analysis (DLA) (Zhong et al., 2019; Pfitzmann et al., 2022; Cheng et al., 2023) aims to locate and classify semantic regions such as text, tables, and figures, providing essential structural information for document parsing and understanding (Wang et al., 2026; Cui et al., 2026; Li et al., 2026; Yu et al., 2026). With the rapid development of large language models (LLMs) (OpenAI, 2023; Yang et al., 2025) and retrieval-augmented generation (RAG) (Yu et al., 2025; Cho et al., 2024; Chen et al., 2025), the demand for high-quality structured document data has increased substantially, requiring real-time document detectors to achieve both higher accuracy and efficiency.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Among existing real-time object detectors, the YOLO series (Jocher et al., 2026; Tian et al., 2025; Jocher et al., 2023) achieves a strong accuracy-efficiency trade-off, yet its potential for document layout analysis remains largely unexplored. However, adapting YOLO detectors to document scenarios is non-trivial due to two critical challenges. At the data level, adapting detectors to document scenarios requires effective document-domain pretraining, which relies on large-scale and structurally diverse corpora covering rich layout variations. However, existing datasets face a trade-off among dataset scale, layout diversity, annotation quality, and so on (as shown in Tab. 1), while constructing large-scale annotated document corpora with fine-grained layouts remains costly (Zhong et al., 2019; Li et al., 2020). At the model level, accurately localizing document elements requires effective structural context modeling, as document layouts contain complex long-range dependencies among text lines, paragraph columns, table grids, and figure-caption pairs (Wang et al., 2024b; Da et al., 2023). However, conventional convolutional YOLO architectures lack explicit mechanisms to capture such dependencies, while the Area Attention recently introduced in YOLOv12 (Tian et al., 2025) is primarily designed for efficient generic visual modeling and does not explicitly exploit the horizontal and vertical organization of document pages.

To overcome these challenges, we propose **DocLayout-YOLO**, a document-specialized real-time detector that improves detection performance through diverse synthetic document pretraining and page-axis context modeling. At the data level, we introduce **MeshLayout-1M**, a large-scale synthetic pretraining dataset generated by our proposed Mesh-candidate BestFit algorithm based on the principles of 2D Bin-packing (Lodi et al., 2002). By synthesizing diverse document layouts with varying elements, scales, and layout structure, MeshLayout-1M provides a more diverse pretraining corpus, allowing better adaptation to heterogeneous document scenarios.

At the model level, we propose **Page-Axis Layout Attention (PALA)**, which explicitly captures long-range dependencies along the horizontal and vertical axes of document pages. By leveraging inherent page structures, PALA enables efficient context aggregation among layout elements while avoiding the computational overhead of global attention. By combining structure-diverse synthetic pretraining with page-axis context modeling, DocLayout-YOLO achieves a favorable accuracy-efficiency trade-off, as shown in Fig. 1. The contributions of this paper are summarized as follows:

- We introduce MeshLayout-1M, a structure-diverse synthetic pretraining dataset generated by the proposed Mesh-candidate BestFit algorithm, which enables large-scale document-domain pretraining with diverse element and layout organizations.
- We propose Page-Axis Layout Attention (PALA), an efficient context modeling mechanism that captures long-range dependencies along horizontal and vertical page axes, explicitly exploiting the inherent structural organization of document layouts while maintaining efficient computation.
- We present DocLayout-YOLO, a document-specialized real-time detector integrating synthetic layout pretraining and page-axis context modeling. It achieves state-of-the-art performance on four heterogeneous benchmarks while delivering superior accuracy-efficiency trade-offs over existing real-time detectors.

2 Related Work

2.1 Document Layout Analysis Datasets

Current document datasets are summarized in Tab. 1. Existing datasets have advanced document layout analysis by providing large-scale annotations, such as PubLayNet (Zhong et al., 2019) and DocBank (Li et al., 2020), or diverse document types and fine-grained categories, such as DocLayNet (Pfitzmann et al., 2022), D^4LA (Da et al., 2023), and M^6Doc (Cheng et al., 2023). However, existing datasets still face limitations in scale, layout diversity, element coverage, or annotation cost. Meanwhile, existing document generation approaches (Zhang et al., 2023b; Inoue et al., 2023; Hui et al., 2023; Jiang et al., 2023; Kong et al., 2022; Gupta et al., 2021) mainly focus on academic-style layouts, limiting their ability to support diverse document-domain pretraining.

2.2 Document Layout Analysis Approaches

Existing DLA methods can be broadly categorized into multimodal and unimodal approaches. Multimodal methods, such as LayoutLMv3 (Huang et al., 2022), VGT (Da et al., 2023), recently

Table 1: Comparison of existing document pretraining datasets.

Dataset	#Pages	#Elements	Element Types	Layout Diversity	Annotation Type
IIT-CDIP (Lewis et al., 2006)	42M	- (None)	- (None)	11 (Diverse)	Unannotated
PRIMA-LAD (Antonacopoulos et al., 2009)	478	7,453	5	5 (Diverse)	Automatic
PubLayNet (Zhong et al., 2019)	360K	3,311,660	5	1 (Academic)	Automatic
HJDataset (Shen et al., 2020)	2,271	250,000+	7	4 (Historical)	Semi-automatic
DocBank (Li et al., 2020)	500K	6,715,907	13	1 (Academic)	Automatic
DocLayNet (Pfitzmann et al., 2022)	80,863	1,107,470	11	6 (Diverse)	Manual
M^{6}Doc (Cheng et al., 2023)	9,080	237,116	74	7 (Diverse)	Manual
D^{4}LA (Da et al., 2023)	11,092	293,692	27	12 (Diverse)	Manual
IndicDLP (Nath et al., 2025)	121,198	1,856,241	42	12 (Diverse)	Manual
MeshLayout-1M (Ours)	**1M**	**14,549,846**	**11**	**Highly Diverse**	**Automatic**

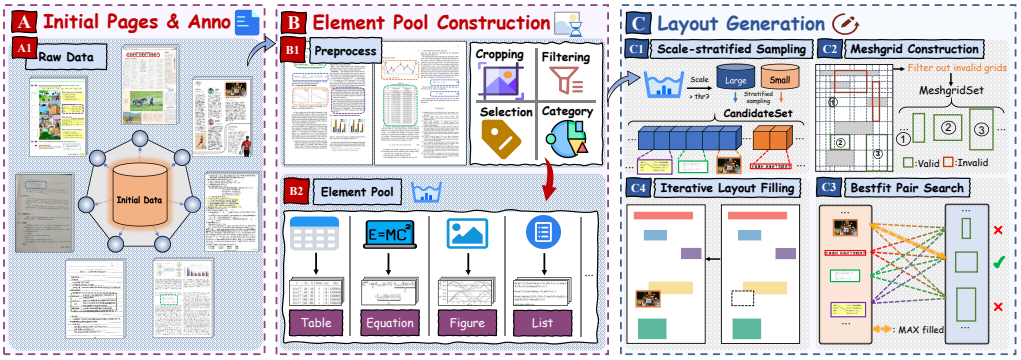


Figure 2: Illustration of Mesh-candidate BestFit. Given initial pages and annotations (**A**), a category-wise element pool is constructed (**B**). Consequently, our method generates diverse and coherent layouts via iterative candidate-grid matching and synthetic pages are formed (**C**).

M2Doc (Zhang et al., 2024), and DoPTA-HR (R. et al., 2024), improve performance by leveraging text-visual interactions. In contrast, unimodal methods (Shen et al., 2021; Banerjee et al., 2023; 2024a; Biswas et al., 2022; Li et al., 2022; Chen et al., 2024b) rely primarily on visual features and generic detection architectures. However, these methods either introduce additional multimodal complexity or lack explicit modeling of document-specific structural dependencies. As for real-time detectors, recent YOLO-based real-time detectors (Jocher et al., 2023; Wang et al., 2024a; Tian et al., 2025; Jocher et al., 2026) have achieved strong accuracy-efficiency trade-offs through advances in detection architectures and attention mechanisms. Nevertheless, they are primarily optimized for natural images and do not explicitly exploit the structural characteristics of document layouts.

3 Diverse Pretraining Documents Synthesis

The diversity of pretraining documents arises mainly from two aspects: (1) element diversity, which covers different element categories and scales; and (2) layout diversity, which captures various page organizations such as single-column, multi-column, and magazine-style layouts. To construct a diverse and scalable document pretraining corpus, we propose **Mesh-candidate BestFit**, a synthesis framework that jointly considers element diversity and layout diversity (as shown in Fig. 2). Unlike random placement or existing generative approaches, our method explicitly controls element composition, scale distribution, and spatial arrangement to generate realistic and diverse document layouts.

3.1 Preprocessing: Ensuring Element Diversity

In the preprocessing phase, to ensure the inclusion of a diverse range of document elements, we utilize an initial data corpus that includes document pages and annotations as element source. In the selection of initial data, we use DocBank (Li et al., 2020) as the element source due to its diverse element-level annotations, covering various document components, appearances, and scales. Consequently, we first construct an element pool from the original annotations. Regions are extracted and noisy excessively small elements are filtered out. Finally, we organize previously retained crops into category-specific pools for subsequent sampling.

3.2 Layout Generation: Ensuring Layout Diversity

Given an empty page, Mesh-candidate BestFit iteratively places elements through scale-stratified sampling and mesh-based bestfit matching. Unlike random placement or existing generative approaches, our method balances layout diversity and structural plausibility.

1. ***Scale-stratified Candidate Sampling*** Directly sampling elements from the entire pool may overpopulate layouts with small elements, preventing the formation of stable global structures. To address this issue, we partition elements into large- and small-scale groups based on their normalized sizes and sample candidates with a predefined ratio. This strategy preserves fine-grained elements while ensuring sufficient large regions for coherent page structures and establishing major page regions.
2. ***Meshgrid Construction*** Construct the meshgrid based on the existing layout and filter out the invalid grids that overlap with inserted elements. Only the remaining grids will be able to participate in matching with the candidate in subsequent steps.
3. ***BestFit Pair Search*** For each candidate, traverse all grids that meet the size requirement and search for the candidate–grid pair with the highest fill rate. Subsequently, remove the optimal candidate from the candidate set and update the layout.
4. ***Iterative Layout Filling*** Finally, the selected candidate is resized and center-placed to the assigned grid and composited onto the page. Repeat steps 2∼4 until no valid candidate–grid pair satisfies the size requirement or the maximum number of elements is reached.

Through the process above, elements are progressively placed into suitable regions, producing more regular and compact document layouts, as shown in Fig. 3. Meanwhile, the generated layouts also exhibit improved alignment and higher page occupancy. The generated pages from MeshLayout-1M, a large-scale synthetic pretraining corpus containing diverse element and layout organizations, enables pretrained models to adapt to a variety of real-world document types effectively. More detailed analysis and explanations are available in Appendix E, F.

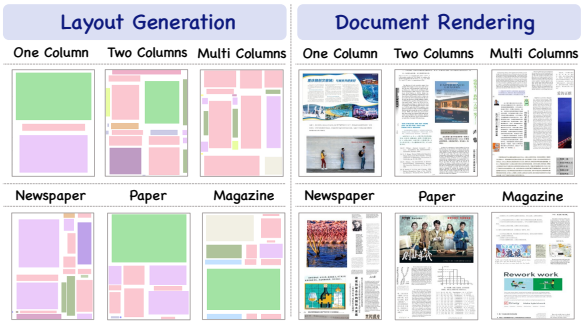


Figure 3: Examples of synthetic document pages.

4 Page-Axis Layout Attention

Document layouts exhibit strong structural regularities along the horizontal and vertical page axes, including text lines, paragraph columns, table structures, and cross-region correspondences. Capturing such long-range axis-aware context is important for accurate document element detection. Although global attention (Wang et al., 2018) can model these dependencies, its computational cost becomes prohibitive for high-resolution document features. Area Attention in YOLOv12 (Tian et al., 2025) improves efficiency by restricting interactions within local regions, but such partitioning may interrupt long-range dependencies across region boundaries. Motivated by the characteristics of document layout, we propose **Page-Axis Layout Attention (PALA)**, which models long-range document dependencies along the page axes more effectively.

Specifically, given the projected query, key, and value features $Q, K, V \in \mathbb{R}^{B\times C\times H\times W}$, PALA reorganizes them along the two page axes and models horizontal and vertical long-range context through two parallel attention branches. For the vertical branch, features are reorganized along the vertical axis as $Q_{ver}, K_{ver}, V_{ver} \in \mathbb{R}^{(BW)\times M\times H\times d}$, where M is the number of attention heads and $d = C/M$. Each sequence contains the H spatial positions sharing the same horizontal coordinate, allowing every token to attend to all positions along the vertical extent. The vertical attention and feature aggregation are computed as:

$$Y_{ver} = \mathrm{Softmax}\left(\frac{Q_{ver}K_{ver}^{T}}{\sqrt{d}}\right)V_{ver}. \tag{1}$$

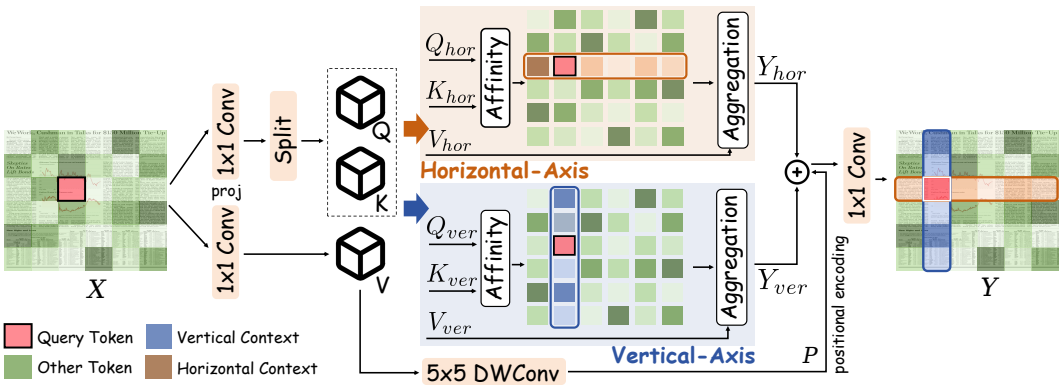


Figure 4: Illustration of Page-Axis Layout Attention (PALA).

Similarly, for the horizontal branch, the features are reorganized as: $Q_{hor}, K_{hor}, V_{hor} \in \mathbb{R}^{(BH)\times M\times W\times d}$, where each sequence corresponds to one feature-map row, enabling direct context aggregation across the full horizontal extent. The horizontal attention and feature aggregation are computed as:

$$Y_{hor} = \text{Softmax}\left(\frac{Q_{hor}K_{hor}^{T}}{\sqrt{d}}\right)V_{hor}. \tag{2}$$

Following the projection and positional encoding design of Area Attention, Q and K are obtained by splitting $QK = \phi_{qk}(X)$ along the channel dimension, while $V = \phi_v(X)$ and local spatial cues are encoded as $P = \text{DWConv}_{5\times5}(V)$. Finally, the two axis-aware features are reshaped and fused as:

$$Y = \text{Proj}(Y_{ver} + Y_{hor} + P). \tag{3}$$

Unlike Area Attention, which restricts interactions within predefined regions, PALA preserves long-range context along the full horizontal and vertical axes. Meanwhile, its parallel branches and independent normalization design (as shown in Fig. 4) avoid directional-order dependency and cross-axis competition, allowing complementary structural context to be effectively modeled.

We further analyze the attention complexity of PALA: omitting same linear projections, the attention-aggregation complexities of global attention, Area Attention with A equal-sized regions, and PALA are, respectively:

$$\begin{aligned} \mathcal{O}_{\text{Global}} &= O(BH^2W^2C), \quad \mathcal{O}_{\text{AA}} = O(B\frac{H^2W^2}{A}C), \\ \mathcal{O}_{\text{PALA}} &= O(BWH^2C + BHW^2C) = O(BHW(H+W)C). \end{aligned} \tag{4}$$

For square feature maps ($H = W = S$), PALA reduces the quadratic complexity of global attention from $O(BS^4C)$ to $O(BS^3C)$ while preserving full horizontal and vertical interactions. Compared with Area Attention, PALA avoids restricting context aggregation within predefined regions and provides a more efficient axis-aware attention mechanism for document page.

5 Experiments

5.1 Experimental Setup

Benchmarks and comparison methods We evaluate DocLayout-YOLO on four heterogeneous DLA benchmarks: D^{4}LA (Da et al., 2023), M^{6}Doc (Cheng et al., 2023), HJDataset (Shen et al., 2020), and PRIMA-LAD (Antonacopoulos et al., 2009). These datasets cover diverse document scenarios, spanning modern and historical sources, multiple languages, and diverse page structures. DocLayout-YOLO is compared with two types of methods: (1) real-time detectors, including YOLO-based models (Jocher et al., 2023; Wang et al., 2024a; Khanam & Hussain, 2024; Tian et al., 2025; Jocher et al., 2026) and RT-DETRv3 (Wang et al., 2025); and (2) specialized DLA approaches, including representative unimodal and multimodal methods. Details of the comparison methods and benchmarks are provided in the Appendix B.

Table 2: **Comparison with efficient real-time detectors on four document layout analysis benchmarks.** Accuracy is reported as $mAP_{50:95}$ (%) and FLOPs are measured at an input resolution of 1600×1600. TensorRT FP16 latency is measured on an NVIDIA H200 GPU with batch size 1.

Method	FLOPs (G)	#Param. (M)	Latency (ms)	D^4LA	M^6Doc	HJDataset	PRIMA-LAD
YOLOv8-N (Jocher et al., 2023)	45.9	2.8	1.1	64.6	60.2	87.8	61.3
YOLOv10-N (Wang et al., 2024a)	38.2	2.1	1.4	66.9	61.5	92.9	58.7
YOLOv11-N (Khanam & Hussain, 2024)	33.1	2.3	1.3	65.3	62.6	88.3	60.5
YOLOv12-N (Tian et al., 2025)	30.2	2.2	1.7	66.5	64.5	89.6	60.4
YOLOv26-N (Jocher et al., 2026)	38.2	2.6	1.6	65.2	62.6	91.5	59.5
DocLayout-YOLO-N	**30.2**	**2.2**	**1.5**	**67.2**	**66.2**	**93.6**	**67.7**
YOLOv8-S (Jocher et al., 2023)	171.5	10.8	1.5	65.7	65.4	88.1	63.2
YOLOv10-S (Wang et al., 2024a)	139.8	7.4	1.9	69.1	68.1	95.9	61.2
YOLOv11-S (Khanam & Hussain, 2024)	127.1	9.1	1.8	67.8	67.5	91.4	62.6
YOLOv12-S (Tian et al., 2025)	114.8	8.8	2.3	69.2	69.5	91.7	64.1
YOLOv26-S (Jocher et al., 2026)	142.7	10.0	2.3	67.7	68.4	92.2	63.7
DocLayout-YOLO-S	**114.8**	**8.8**	**2.0**	**70.6**	**71.1**	**94.8**	**70.3**
YOLOv8-M (Jocher et al., 2023)	489.8	25.7	2.8	69.3	69.3	94.7	67.0
RT-DETRv3-R18 (Wang et al., 2025)	322.5	21.1	2.4	69.6	68.1	94.1	65.7
RT-DETRv3-R34 (Wang et al., 2025)	511.7	32.4	2.9	70.5	70.5	94.9	68.8
YOLOv10-M (Wang et al., 2024a)	391.0	16.1	3.3	70.6	69.0	94.1	63.8
YOLOv11-M (Khanam & Hussain, 2024)	427.7	20.1	3.0	69.0	69.0	94.9	66.0
YOLOv12-M (Tian et al., 2025)	377.1	19.7	3.7	70.2	70.9	93.4	66.3
YOLOv26-M (Jocher et al., 2026)	471.2	21.9	3.4	68.7	69.6	94.2	65.3
DocLayout-YOLO-M	**377.1**	**19.7**	**3.1**	**70.7**	**73.3**	**95.3**	**71.3**
YOLOv8-L (Jocher et al., 2023)	1035.3	43.7	4.0	70.9	71.7	95.7	68.6
RT-DETRv3-R50 (Wang et al., 2025)	766.5	43.7	4.1	72.1	72.4	94.3	70.4
YOLOv10-L (Wang et al., 2024a)	798.1	25.9	4.6	71.6	72.4	95.6	65.7
YOLOv11-L (Khanam & Hussain, 2024)	547.0	25.4	4.1	71.4	72.1	94.9	67.9
YOLOv12-L (Tian et al., 2025)	519.9	26.5	5.8	71.1	72.5	94.9	66.9
YOLOv26-L (Jocher et al., 2026)	586.2	26.3	4.5	69.8	70.9	95.1	64.6
DocLayout-YOLO-L	**519.9**	**26.5**	**4.7**	**72.0**	**75.1**	**95.7**	**73.4**
YOLOv8-X (Jocher et al., 2023)	1615.2	68.2	6.0	71.0	72.2	95.4	69.6
RT-DETRv3-R101 (Wang et al., 2025)	1533.8	84.6	6.4	72.2	72.7	93.0	70.3
YOLOv10-X (Wang et al., 2024a)	1072.6	31.8	6.5	72.1	72.0	94.0	61.1
YOLOv11-X (Khanam & Hussain, 2024)	1223.9	56.9	6.1	71.7	73.9	95.4	65.5
YOLOv12-X (Tian et al., 2025)	1161.1	59.4	8.5	70.8	74.0	94.0	67.0
YOLOv26-X (Jocher et al., 2026)	1309.6	59.0	6.6	70.8	71.3	95.7	65.8
DocLayout-YOLO-X	**1161.1**	**59.4**	**6.9**	**72.4**	**75.5**	**96.0**	**73.1**

Implementation Details DocLayout-YOLO is built on YOLOv12 (Tian et al., 2025), with FlashAttention (Dao et al., 2022; Dao, 2024) integrated into PALA. Each model is pretrained on MeshLayout-1M and subsequently finetuned on the downstream benchmarks using an input resolution of 1600×1600 on $8\times$ H200 GPUs. All models are trained for 500 epochs using the SGD optimizer with an initial learning rate of 0.02, consistent with YOLOv12. A linear learning rate decay schedule is adopted, with a linear warm-up for the first three epochs. The latencies of all models are tested on an H200 GPU using TensorRT FP16. See Appendix A for full implementation details.

5.2 Comparisons with State-of-the-Art Real-time Detectors

We compare DocLayout-YOLO with existing real-time detectors in Tab. 2. **N-scale:** DocLayout-YOLO-N improves over the baseline YOLOv12-N by 0.7%/1.7%/4.0%/7.3% mAP on four benchmarks, while reducing latency by 11.8%. It further exceeds YOLOv8-N, YOLOv10-N, and YOLOv26-N by up to 9.0% at comparable inference speed. **S-scale:** DocLayout-YOLO-S delivers gains of 1.4%/1.6%/3.1%/6.2% over YOLOv12-S with 13.0% lower latency. Compared with YOLOv26-S and RT-DETRv3-R18, it achieves improvements of up to 6.6% and 4.6%, respectively. **M-scale:** With a latency of 3.1 ms, DocLayout-YOLO-M surpasses YOLOv12-M by 0.5%/2.4%/1.9%/5.0% and provides a 16.2% speedup. It also outperforms RT-DETRv3-R34 by up to 2.8% while using 26.3% fewer FLOPs. **L-scale:** DocLayout-YOLO-L improves upon YOLOv12-L by 0.9%/2.6%/0.8%/6.5% while reducing latency from 5.8 ms to 4.7 ms. **X-scale:** DocLayout-YOLO-X achieves the best overall results of 72.4%/75.5%/96.0%/73.1% mAP on four benchmarks. Compared with YOLOv12-X, it yields gains of 1.6%/1.5%/2.0%/6.1% together with an 18.8% latency reduction, and surpasses RT-DETRv3-R101 by up to 3.0% with comparable inference latency. In addition, PALA retains the same feature projections and positional encoding as Area Attention in YOLOv12, while standard FLOPs profiling does not account for attention aggregation, resulting in identical parameter and GFLOP counts; PALA nevertheless achieves consistently lower latency through more efficient axis-wise aggregation.

Table 3: Performance comparison with existing state-of-the-art (SOTA) methods on public benchmarks. **Best** and Second Best results are highlighted. † indicates that ground truth text-level bounding boxes and OCR annotations are used during prediction.

Method	Venue	Backbone	Modality V	Modality L	Params	D^{4}LA	M^{6}Doc	HJDataset	PRIMA-LAD
LayoutParser (Shen et al., 2021)	ICDAR 2021	ResNext (Xie et al., 2017)	✓	✗	6.6M	-	-	81.6	64.7
DocSegTr (Biswas et al., 2022)	arXiv 2022	ResNet (He et al., 2016)	✓	✗	84.2M	-	60.3	83.1	42.5
DiT-L (Li et al., 2022)	ACM MM 2022	ViT-L (Dosovitskiy et al., 2021)	✓	✗	361.6M	68.2	70.2	-	-
SelfDocSeg (Maity et al., 2023)	ICDAR 2023	ResNet-50 (He et al., 2016)	✓	✗	63.7M	-	-	78.8	52.1
SwinDocSegmenter (Banerjee et al., 2023)	ICDAR 2023	Swin (Liu et al., 2021)	✓	✗	220.9M	-	47.1	84.6	54.4
GraphKD (Banerjee et al., 2024a)	ICDAR 2024	ResNet (He et al., 2016)	✓	✗	136.6M	-	-	79.7	41.9
SemiDocSeg (Banerjee et al., 2024b)	IJDAR 2024	Swin-L (Liu et al., 2021)	✓	✗	220.9M	-	-	81.1	45.9
RoDLA (Chen et al., 2024b)	CVPR 2024	InternImage (Wang et al., 2023)	✓	✗	338.9M	-	70.0	-	-
PARL (Liu et al., 2026)	arXiv 2026	HRNetv2 (Sun et al., 2019)	✓	✗	65M	70.9	72.9	-	-
HybriDLA (Chen et al., 2026)	AAAI 2026	InternImage (Wang et al., 2023)	✓	✗	392M	-	71.4	-	-
LayoutLMv3 (Huang et al., 2022)	ACM MM 2022	RoBERTa-B (Liu et al., 2019)	✓	✓	138.4M	60.0	64.3	82.7	40.3
VGT (Da et al., 2023)	ICCV 2023	ViT-B (Dosovitskiy et al., 2021)	✓	✓	266.8M	68.8	-	-	-
M2Doc† (Zhang et al., 2024)	AAAI 2024	ResNet (He et al., 2016)	✓	✓	266.5M	-	69.9	-	-
DoPTA-HR (R. et al., 2024)	arXiv 2024	ViT-B (Dosovitskiy et al., 2021)	✓	✓	85M	70.7	69.5	-	-
DocSAM (Li et al., 2025)	CVPR 2025	Swin-L (Liu et al., 2021)	✓	✓	317M	55.5	50.0	90.5	-
PromptDLA (Zhang et al., 2026)	TMM 2026	ViT-B (Dosovitskiy et al., 2021)	✓	✓	-	69.2	69.1	-	-
DocLayout-YOLO-X	**-**	**v12x w/ PALA**	✓	✗	**59.4M**	**72.4**	**75.5**	**96.0**	**73.1**

Table 4: Ablation study of MeshLayout-1M pretraining and PALA across different model sizes.

Size	Components		Benchmarks			
	MeshLayout-1M	PALA	D^{4}LA	M^{6}Doc	HJDataset	PRIMA-LAD
N			66.5	64.5	89.6	60.4
		✓	67.0	65.3	91.0	62.6
	✓		66.9	65.4	92.1	65.8
	✓	✓	**67.2**	**66.2**	**93.6**	**67.7**
S			69.2	69.5	91.7	64.1
		✓	70.2	70.1	92.9	66.8
	✓		69.8	70.4	93.9	68.6
	✓	✓	**70.6**	**71.1**	**94.8**	**70.3**
M			70.2	70.9	93.4	66.3
		✓	70.6	71.5	95.2	69.2
	✓		70.6	72.6	95.0	70.0
	✓	✓	**70.7**	**73.3**	**95.3**	**71.3**
L			71.1	72.5	94.9	66.9
		✓	71.6	73.7	95.3	69.1
	✓		71.7	74.5	95.4	71.9
	✓	✓	**72.0**	**75.1**	**95.7**	**73.4**
X			70.8	74.0	94.0	67.0
		✓	71.8	74.5	95.2	67.8
	✓		71.5	75.0	95.8	71.8
	✓	✓	**72.4**	**75.5**	**96.0**	**73.1**

5.3 Comparisons with State-of-the-Art Specialized DLA Methods

We compare DocLayout-YOLO with specialized document layout analysis methods in Tab. 3. DocLayout-YOLO-X establishes state-of-the-art performance on all four benchmarks, achieving 72.4%/75.5%/96.0%/73.1% mAP on D^{4}LA, M^{6}Doc, HJDataset, and PRIMA-LAD, respectively. Notably, DocLayout-YOLO relies solely on visual inputs and consistently surpasses both visual-only unimodal and visual-language multimodal approaches. Compared with the previous best-performing unimodal methods, DocLayout-YOLO improves performance on D^{4}LA and M^{6}Doc by 1.5% and 2.6% while using only 59.4M parameters, substantially fewer than most competitive DLA models. Furthermore, unlike multimodal methods such as LayoutLMv3, VGT, and DoPTA-HR, DocLayout-YOLO does not require OCR annotations or semantic language features during inference, demonstrating that strong document layout perception can be achieved through efficient visual-only modeling. Moreover, many approaches rely on large backbones, DocLayout-YOLO maintains a lightweight YOLO-based architecture and achieves superior performance with significantly lower model complexity.

Table 5: Comparison of different document pretraining datasets across M^{6}Doc document types.

Pretraining Data	Volume	M^{6}Doc						
		Book	Magazine	Newspaper	Note	Paper	Test Paper	Textbook
None	–	53.5	74.2	74.4	64.9	71.5	71.5	65.7
DocBank (Li et al., 2020)	400K	53.9 (+0.4)	74.7 (+0.5)	74.8 (+0.4)	65.5 (+0.6)	72.7 (+1.2)	71.1 (-0.4)	66.1 (+0.4)
PubLayNet (Zhong et al., 2019)	340K	53.7 (+0.2)	74.0 (-0.2)	74.5 (+0.1)	64.0 (-0.9)	70.5 (-1.0)	71.6 (+0.1)	65.9 (+0.2)
DocSynth (Biswas et al., 2021)	300K	53.2 (-0.3)	73.7 (-0.5)	73.5 (-0.9)	62.7 (-2.2)	69.9 (-1.6)	69.4 (-2.1)	65.0 (-0.7)
RanLayNet (Anand et al., 2023)	7K	53.9 (+0.4)	73.2 (-1.0)	72.8 (-1.6)	62.3 (-2.6)	70.7 (-0.8)	70.3 (-1.3)	65.2 (-0.5)
LayoutDM (Inoue et al., 2023)	300K	53.3 (-0.2)	74.7 (+0.5)	74.5 (+0.1)	62.3 (-2.6)	69.7 (-1.8)	69.9 (-1.6)	64.5 (-1.3)
LACE (Chen et al., 2024a)	300K	54.0 (+0.5)	74.0 (-0.2)	74.3 (-0.1)	64.2 (-0.7)	71.8 (+0.3)	71.3 (-0.2)	65.7 (0.0)
MeshLayout	300K	53.9 (+0.4)	74.9 (+0.7)	75.1 (+0.7)	65.7 (+0.8)	72.2 (+0.7)	72.0 (+0.5)	65.5 (-0.2)
MeshLayout	500K	54.7 (+1.2)	75.2 (+1.0)	75.3 (+0.9)	65.6 (+0.7)	72.9 (+1.4)	72.4 (+0.9)	66.4 (+0.7)
MeshLayout	**1M**	**55.6 (+2.1)**	**76.0 (+1.8)**	**75.9 (+1.5)**	**66.7 (+1.8)**	**74.6 (+3.1)**	**72.6 (+1.1)**	**66.9 (+1.2)**

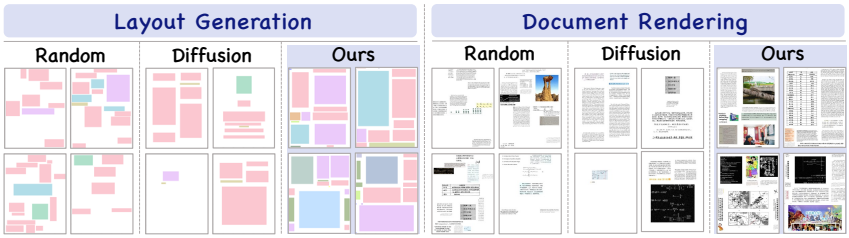


Figure 5: Qualitative comparison of random placement, diffusion, and Mesh-candidate BestFit.

5.4 Ablation Studies

Component Effectiveness Analysis. Tab. 4 further disentangles the contributions of MeshLayout-1M pretraining and PALA. Specifically, the MeshLayout-only setting corresponds to the original YOLOv12 pretrained on MeshLayout-1M, while the PALA-only replaces Area Attention with PALA without pretraining. Averaged across five model sizes, PALA alone improves mAP by 0.7%/0.7%/1.2%/2.2% on D^{4}LA, M^{6}Doc, HJDataset, and PRIMA-LAD, respectively, while MeshLayout-1M pretraining alone yields gains of 0.5%/1.3%/1.7%/4.7%. Combining both components consistently achieves the best results, with average improvements of 1.0%/2.0%/2.4%/6.2% over the YOLOv12 baseline. These results demonstrate that the gains of DocLayout-YOLO arise from both the architectural design of PALA and document-domain pretraining with MeshLayout-1M, with the two components providing complementary benefits.

Comparison of Pretraining Datasets. Tab. 5 compares MeshLayout-1M with representative real-world and synthetic document pretraining datasets. Real-world datasets include PubLayNet (Zhong et al., 2019) and DocBank (Li et al., 2020), which provide annotated real-world scientific pages. Synthetic alternatives include DocSynth (Biswas et al., 2021) which synthesizes pages using layouts derived from PubLayNet, while RanLayNet (Anand et al., 2023) adopts random placement. For diffusion-based methods, we use LayoutDM (Inoue et al., 2023) and LACE (Chen et al., 2024a) trained on DocBank to generate layouts and then render with DocBank elements. Fig. 5 qualitatively compares synthetic layouts and pages generated by random placement, diffusion-based generation, and Mesh-candidate BestFit. For a fair comparison, we pretrain DocLayout-YOLO-S on different datasets under identical settings and evaluate downstream performance. Existing datasets show inconsistent transfer across document types, whereas MeshLayout yields increasingly consistent gains with larger pretraining scales. With 1M pages, it improves all seven document types by 1.1%$\sim$3.1% mAP, demonstrating scalable and robust transfer to heterogeneous layouts.

Ablations on MeshLayout Components. Tab. 6 ablates each component of Mesh-candidate BestFit using DocLayout-YOLO-S. For every variant, we generate the same number of pretraining pages (1M) and compare downstream fine-tuning performance. 1) Removing meshgrid construction reduces synthesis to unconstrained random placement, producing disordered layouts (Fig. 5) and degrading performance by 1.2%/2.6%/3.1% on M^{6}Doc, HJDataset, and PRIMA-LAD, respectively. 2) Without scale-stratified sampling, small elements become overrepresented, weakening layout structure and reducing performance by 0.3%/0.4%/1.1%. 3) Without BestFit matching, random candidate-region assignment causes size mismatches and excessive blank areas, leading to drops

Table 6: Ablation study on the core components of Mesh-candidate BestFit.

Method	Meshgrid Construction	Scale-stratified Sampling	BestFit Matching	M6Doc	HJDataset	PRIMA-LAD
Baseline (w/o MeshLayout pretraining)	–	–	–	70.1	92.9	66.8
w/o Meshgrid Construction	✗	✓	–	69.9	92.2	67.2
w/o Scale-stratified Sampling	✓	✗	✓	70.8	94.4	69.2
w/o BestFit Matching	✓	✓	✗	70.4	94.3	68.5
Full MeshLayout	✓	✓	✓	**71.1**	**94.8**	**70.3**

Table 7: Local attentions comparison.

Size	Attention Type	Mem. (G)	CUDA FP32	CUDA FP16	M^{6}Doc
N	Global (Wang et al., 2018)	0.295	9.498	2.123	64.5
	Area (Tian et al., 2025)	0.295	5.250	1.656	64.5
	Criss-Cross (Huang et al., 2019)	0.301	3.187	1.847	64.7
	Swin (Liu et al., 2021)	0.295	2.988	1.656	62.3
	CSWin (Dong et al., 2022)	0.295	3.587	1.755	63.6
	Axial (Wang et al., 2020)	0.295	3.690	1.643	64.2
	Horizontal-only	0.295	2.867	1.378	64.1
	Vertical-only	0.295	2.901	1.398	63.9
	PALA (ours)	**0.295**	**3.596**	**1.500**	**65.3**
L	Global (Wang et al., 2018)	0.514	61.524	9.184	73.0
	Area (Tian et al., 2025)	0.512	30.956	5.637	71.1
	Criss-Cross (Huang et al., 2019)	0.516	19.399	6.588	73.2
	Swin (Liu et al., 2021)	0.512	18.847	5.485	71.7
	CSWin (Dong et al., 2022)	0.514	21.559	5.905	72.6
	Axial (Wang et al., 2020)	0.512	22.266	5.223	72.4
	Horizontal-only	0.512	17.862	4.325	72.7
	Vertical-only	0.512	17.912	4.378	71.4
	PALA (ours)	**0.512**	**20.709**	**4.694**	**73.7**

Table 8: Speed, memory, and efficiency.

Model	FLOPs (G)	Mem. (GiB)	CUDA Latency (ms) FP32	FP16	M^{6}Doc
YOLOv8-N (Jocher et al., 2023)	45.9	0.29	2.4	1.1	60.2
YOLOv10-N (Wang et al., 2024a)	38.2	0.31	2.8	1.4	61.5
YOLOv12-N (Tian et al., 2025)	30.2	0.30	5.3	1.7	64.5
DocLayout-YOLO-N	**30.2**	**0.30**	**3.6**	**1.5**	**66.2**
YOLOv8-S (Jocher et al., 2023)	171.5	0.34	5.5	1.5	65.4
YOLOv10-S (Wang et al., 2024a)	139.8	0.48	5.6	1.9	68.1
YOLOv12-S (Tian et al., 2025)	114.8	0.35	9.9	2.3	69.5
DocLayout-YOLO-S	**114.8**	**0.35**	**6.7**	**2.0**	**71.1**
YOLOv8-M (Jocher et al., 2023)	489.8	0.44	13.2	2.8	69.3
YOLOv10-M (Wang et al., 2024a)	391.0	0.53	12.0	3.3	69.0
YOLOv12-M (Tian et al., 2025)	377.1	0.49	19.4	3.7	70.9
DocLayout-YOLO-M	**377.1**	**0.49**	**14.0**	**3.1**	**73.3**
YOLOv8-L (Jocher et al., 2023)	1035.3	0.57	22.3	4.0	71.7
YOLOv10-L (Wang et al., 2024a)	798.1	0.61	19.0	4.6	72.4
YOLOv12-L (Tian et al., 2025)	519.9	0.51	31.2	5.8	72.5
DocLayout-YOLO-L	**519.9**	**0.51**	**20.7**	**4.7**	**75.1**
YOLOv8-X (Jocher et al., 2023)	1615.2	0.69	36.3	6.0	72.2
YOLOv10-X (Wang et al., 2024a)	1072.6	0.71	27.4	6.5	72.0
YOLOv12-X (Tian et al., 2025)	1161.1	0.70	53.1	8.5	74.0
DocLayout-YOLO-X	**1161.1**	**0.70**	**37.9**	**6.9**	**75.5**

of 0.7%/0.5%/1.8%. These results demonstrate that the three components jointly ensure structural plausibility, scale consistency, and effective downstream transfer.

Page-Axis Layout Attention. Tab. 7 compares different attention mechanisms on M^{6}Doc. Latencies are obtained on identical H200 to ensure fairness, while Memory usage is measured with the TensorRT model (bs = 1). Following YOLOv12 (Tian et al., 2025), we adjust the number of channels to ensure that all models have similar parameters and FLOPs and disable Flash-Attention. Compared with Global Attention, PALA achieves higher accuracy with substantially lower latency, suggesting that unrestricted global interactions may introduce cross-region interference, while axis-wise constraints better match document structures. Swin performs worse because its local windows limit direct interactions across long document structures, while Criss-Cross and CSWin extend attention along horizontal and vertical directions to better capture such dependencies. PALA further combines both page axes and achieves the best 65.3%/73.7% mAP at N/L scales. Horizontal-only consistently outperforms Vertical-only, while combining both yields further gains, indicating the importance and complementarity of horizontal and vertical dependencies in document layouts. More explanations and analysis are available in Appendix D.

Speed, memory, and efficiency. Tab. 8 evaluates the speed, memory usage, and accuracy-efficiency trade-off of DocLayout-YOLO across five model scales under identical H200 GPU hardware settings. Despite identical parameter counts to YOLOv12, DocLayout-YOLO consistently achieves lower latency, highlighting the practical efficiency of PALA's attention aggregation. Compared with YOLOv12, DocLayout-YOLO reduces FP32 latency by 1.7/3.2/5.4/10.5/15.2 ms and FP16 latency by 0.2/0.3/0.6/1.1/1.6 ms from N to X scales, while maintaining comparable peak memory usage. Meanwhile, compared with YOLOv8 and YOLOv10, DocLayout-YOLO further provides a favorable balance between inference speed, memory usage, and detection accuracy.

6 Conclusion

This work presents DocLayout-YOLO, a real-time detector specifically designed for document layout analysis. DocLayout-YOLO adapts the YOLO paradigm to heterogeneous document layouts by combining large-scale synthetic document pretraining with MeshLayout-1M and efficient long-range page-axis modeling with PALA. Extensive experiments show that DocLayout-YOLO achieves state-of-the-art results across multiple benchmarks and consistently delivers a superior accuracy–efficiency trade-off over existing real-time detectors.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language editing, wording refinement, and LaTeX formatting. Generative AI was not used to design the proposed methods, determine the experimental methodology, or make scientific claims on behalf of the authors. All AI-assisted text and suggestions were manually reviewed, revised, and verified by the authors for technical correctness and consistency with the actual implementation and experimental results. We take full responsibility for the final content of this work, including any text or artifacts produced with the aid of generative AI.

REFERENCES

Avinash Anand, Raj Jaiswal, Mohit Gupta, Siddhesh S. Bangar, Pijush Bhuyan, Naman Lal, Rajeev Singh, Ritika Jha, Rajiv Ratn Shah, and Shin'ichi Satoh. Ranlaynet: A dataset for document layout detection used for domain adaptation and generalization. In *ACM Multimedia Asia*, 2023.

Apostolos Antonacopoulos, David Bridson, Christos Papadopoulos, and Stefan Pletschacher. A realistic dataset for performance evaluation of document layout analysis. In *ICDAR*, 2009.

Ayan Banerjee, Sanket Biswas, Josep Lladós, and Umapada Pal. Graphkd: Exploring knowledge distillation towards document object detection with structured graph creation. In *ICDAR*, 2024a.

Ayan Banerjee, Sanket Biswas, Josep Lladós, and Umapada Pal. Semidocseg: Harnessing semi-supervised learning for document layout analysis. *International Journal on Document Analysis and Recognition*, 2024b.

Ayan Banerjee et al. Swindocsegmenter: An end-to-end unified domain adaptive transformer for document instance segmentation. In *ICDAR*, 2023.

Sanket Biswas, Pau Riba, Josep Lladós, and Umapada Pal. Docsynth: A layout guided approach for controllable document image synthesis. In *ICDAR*, 2021.

Sanket Biswas et al. DocSegTr: An instance-level end-to-end document image segmentation transformer. *arXiv preprint arXiv:2201.11438*, 2022.

Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *CVPR*, 2018.

Jian Chen, Ruiyi Zhang, Yufan Zhou, and Changyou Chen. Towards aligned layout generation via diffusion model with aesthetic constraints. In *ICLR*, 2024a.

Jun Chen, Dannong Xu, Junjie Fei, Chun-Mei Feng, and Mohamed Elhoseiny. Document haystacks: Vision-language reasoning over piles of 1000+ documents. In *CVPR*, 2025.

Yufan Chen, Omar Moured, Ruiping Liu, Junwei Zheng, Kunyu Peng, Jiaming Zhang, and Rainer Stiefelhagen. Hybridla: Hybrid generation for document layout analysis. In *AAAI*, 2026.

Yufan Chen et al. Rodla: Benchmarking the robustness of document layout analysis models. In *CVPR*, 2024b.

Hiuyi Cheng, Peirong Zhang, Sihang Wu, Jiaxin Zhang, Qiyuan Zhu, Zecheng Xie, Jing Li, Kai Ding, and Lianwen Jin. M^6doc: A large-scale multi-format, multi-type, multi-layout, multi-language, multi-annotation category dataset for modern document layout analysis. In *CVPR*, 2023.

Jaemin Cho, Debanjan Mahata, Ozan Irsoy, Yujie He, and Mohit Bansal. M3docrag: Multi-modal retrieval is what you need for multi-page multi-document understanding. *arXiv preprint arXiv:2411.04952*, 2024.

Cheng Cui, Ting Sun, Suyin Liang, Tingquan Gao, Zelun Zhang, Jiaxuan Liu, Xueqing Wang, Changda Zhou, Hongen Liu, Manhui Lin, Yue Zhang, Yubo Zhang, Jing Zhang, Jun Zhang, Xing Wei, Yi Liu, Dianhai Yu, and Yanjun Ma. Boosting document parsing efficiency and performance with coarse-to-fine visual processing. In *CVPR*, 2026.

Cheng Da, Chuwei Luo, Qi Zheng, and Cong Yao. Vision grid transformer for document layout analysis. In *ICCV*, 2023.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *ICLR*, 2024.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *NeurIPS*, volume 35, 2022.

Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. CSWin transformer: A general vision transformer backbone with cross-shaped windows. In *CVPR*, 2022.

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. In *ICLR*, 2021.

Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry Davis, Vijay Mahadevan, and Abhinav Shrivastava. Layouttransformer: Layout generation and completion with self-attention. In *ICCV*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, 2017.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. Layoutlmv3: Pre-training for document AI with unified text and image masking. In *ACM MM*, 2022.

Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. CCNet: Criss-cross attention for semantic segmentation. In *ICCV*, 2019.

Mude Hui, Zhizheng Zhang, Xiaoyi Zhang, Wenxuan Xie, Yuwang Wang, and Yan Lu. Unifying layout generation with a decoupled diffusion model. In *CVPR*, 2023.

Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Layoutdm: Discrete diffusion model for controllable layout generation. In *CVPR*, 2023.

Zhaoyun Jiang, Jiaqi Guo, Shizhao Sun, Huayu Deng, Zhongkai Wu, Vuksan Mijovic, Zijiang James Yang, Jian-Guang Lou, and Dongmei Zhang. Layoutformer++: Conditional graphic layout generation via constraint serialization and decoding space restriction. In *CVPR*, 2023.

Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLO, 2023. URL `https://github.com/ultralytics/ultralytics`.

Glenn Jocher, Jing Qiu, Mengyu Liu, Shuai Lyu, Fatih Cagatay Akyon, and Muhammet Esat Kalfaoglu. Ultralytics YOLO26: Unified real-time end-to-end vision models. *arXiv preprint arXiv:2606.03748*, 2026.

Rahima Khanam and Muhammad Hussain. YOLOv11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*, 2024.

Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Constrained graphic layout generation via latent optimization. In *ACM MM*, 2021.

Xiang Kong, Lu Jiang, Huiwen Chang, Han Zhang, Yuan Hao, Haifeng Gong, and Irfan Essa. BLT: bidirectional layout transformer for controllable layout generation. In *ECCV*, 2022.

David D. Lewis, Gady Agam, Shlomo Argamon, Ophir Frieder, David A. Grossman, and Jefferson Heard. Building a test collection for complex document information processing. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006.

Gengluo Li, Pengyuan Lyu, Chengquan Zhang, Huawen Shen, Liang Wu, Xingyu Wan, Gangyan Zeng, Han Hu, Can Ma, and Yu Zhou. Towards real-world document parsing via realistic scene synthesis and document-aware training. In *CVPR*, 2026.

Jianan Li, Jimei Yang, Jianming Zhang, Chang Liu, Christina Wang, and Tingfa Xu. Attribute-conditioned layout GAN for automatic graphic design. *IEEE TVCG*, 2021.

Junlong Li, Yiheng Xu, Tengchao Lv, Lei Cui, Cha Zhang, and Furu Wei. Dit: Self-supervised pre-training for document image transformer. In *ACM MM*, 2022.

Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. Docbank: A benchmark dataset for document layout analysis. In *International Conference on Computational Linguistics*, 2020.

Xiao-Hui Li, Fei Yin, and Cheng-Lin Liu. DocSAM: Unified document image segmentation via query decomposition and heterogeneous mixed learning. In *CVPR*, 2025.

Fuyuan Liu, Dianyu Yu, He Ren, Nayu Liu, Xiaomian Kang, Delai Qiu, Fa Zhang, Genpeng Zhen, Shengping Liu, Jiaen Liang, et al. Parl: Position-aware relation learning network for document layout analysis. *arXiv preprint arXiv:2601.07620*, 2026.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

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. In *ICCV*, 2021.

Andrea Lodi, Silvano Martello, and Daniele Vigo. Recent advances on two-dimensional bin packing problems. *Discrete Applied Mathematics*, 2002.

Subhajit Maity, Sanket Biswas, Siladittya Manna, Ayan Banerjee, Josep Lladós, Saumik Bhattacharya, and Umapada Pal. Selfdocseg: A self-supervised vision-based approach towards document segmentation. In *ICDAR*, 2023.

Oikantik Nath, Sahithi Kukkala, Mitesh Khapra, and Ravi Kiran Sarvadevabhatla. IndicDLP: A foundational dataset for multi-lingual and multi-domain document layout parsing. In *ICDAR*, 2025.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S. Nassar, and Peter W. J. Staar. Doclaynet: A large human-annotated dataset for document-layout segmentation. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.

Nikitha S. R., Tarun Ram Menta, and Mausoom Sarkar. Dopta: Improving document layout analysis using patch-text alignment. *arXiv preprint arXiv:2412.12902*, 2024.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

Zejiang Shen, Kaixuan Zhang, and Melissa Dell. A large dataset of historical japanese documents with complex layouts. In *CVPRW*, 2020.

Zejiang Shen et al. Layoutparser: A unified toolkit for deep learning based document image analysis. In *ICDAR*, 2021.

Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514*, 2019.

Yunjie Tian, Qixiang Ye, and David Doermann. YOLOv12: Attention-centric real-time object detectors. In *NeurIPS*, 2025.

Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Yolov10: Real-time end-to-end object detection. In *NeurIPS*, 2024a.

Bin Wang, Tianyao He, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Tao Chu, Yuan Qu, Zhenjiang Jin, Weijun Zeng, Ziyang Miao, Bangrui Xu, Junbo Niu, Mengzhang Cai, Jiantao Qiu, Qintong Zhang, Dongsheng Ma, Yuefeng Sun, Hejun Dong, Wenzheng Zhang, Jutao Xiao, Jiayong Shi, Pengyu Liao, Xiaomeng Zhao, Huaping Zhong, Liqun Wei, Jing Yu, Jie Yang, Wei Li, Shasha Wang, Qianqian Wu, Xuanhe Zhou, Weijia Li, Zhenxiang Li, Zhongying Tu, Jiang Wu, Lijun Wu, Chao Xu, Kai Chen, Wentao Zhang, Yu Qiao, Bowen Zhou, Dahua Lin, and Conghui He. Mineru2.5-pro: Pushing the limits of data-centric document parsing at scale. *arXiv preprint arXiv:2604.04771*, 2026.

Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan L. Yuille, and Liang-Chieh Chen. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In *ECCV*, 2020.

Jiawei Wang, Kai Hu, and Qiang Huo. DLAFormer: An end-to-end transformer for document layout analysis. In *ICDAR*, 2024b.

Shuo Wang, Chunlong Xia, Feng Lv, and Yifeng Shi. RT-DETRv3: Real-time end-to-end object detection with hierarchical dense positive supervision. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2025.

Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. InternImage: Exploring large-scale vision foundation models with deformable convolutions. In *CVPR*, 2023.

Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and Maosong Sun. Visrag: Vision-based retrieval-augmented generation on multi-modality documents. In *ICLR*, 2025.

Xinlei Yu, Chengming Xu, Zhangquan Chen, Yudong Zhang, Shilin Lu, Cheng Yang, Jiangning Zhang, Shuicheng Yan, and Xiaobin Hu. Visual document understanding and reasoning: A multi-agent collaboration framework with agent-wise adaptive test-time scaling. In *CVPR*, 2026.

Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. In *ICLR*, 2023a.

Junyi Zhang, Jiaqi Guo, Shizhao Sun, Jian-Guang Lou, and Dongmei Zhang. Layoutdiffusion: Improving graphic layout generation by discrete diffusion probabilistic models. In *ICCV*, 2023b.

Ning Zhang, Hiuyi Cheng, Jiayu Chen, Zongyuan Jiang, Jun Huang, Yang Xue, and Lianwen Jin. M2doc: A multi-modal fusion approach for document layout analysis. In *AAAI*, 2024.

Zirui Zhang, Yaping Zhang, Lu Xiang, Yang Zhao, Feifei Zhai, Yu Zhou, and Chengqing Zong. Promptdla: A domain-aware prompt document layout analysis framework with descriptive knowledge as a cue. *IEEE TMM*, 2026.

Xu Zhong, Jianbin Tang, and Antonio Jimeno-Yepes. Publaynet: Largest dataset ever for document layout analysis. In *ICDAR*, 2019.

Zhuofan Zong, Guanglu Song, and Yu Liu. DETRs with collaborative hybrid assignments training. In *ICCV*, 2023.

Table of Contents

A More Implementation Details **15**

B Benchmark & Comparison Methods Details **17**

B.1 Public Benchmark Details 17

B.2 Comparison Methods 18

C More Experiments **19**

C.1 Results on More Comparison Datasets 19

C.2 Robustness Evaluation 19

C.3 Heatmap Visualization 20

C.4 Design-heuristic Metric Evaluation Evaluation on MeshLayout-1M 20

C.5 Ablation on Pretraining Data Volume 21

C.6 Effects of MeshLayout-1M Pretraining on Long-tail Distribution 21

D Additional Analysis of Page-Axis Layout Attention (PALA) **23**

D.1 More Analysis & Comparisons of Different Local Attentions 23

D.2 Distinguishing PALA from Criss-Cross and Axial Attention 24

D.3 Area Attention vs. PALA: Is PALA Always More Efficient? 25

E Mesh-candidate BestFit Details **27**

E.1 DocBank-based Element Pool Construction 27

E.2 Scale-stratified Candidate Sampling 27

E.3 Mesh-candidate BestFit Algorithm 27

F Visual Diagnosis of MeshLayout-1M **29**

F.1 Synthetic Documents Illustration 29

F.2 Layout Diversity 29

F.3 Element Scale and Shape Distributions 31

G Layout Analysis Examples **32**

H Limitations & Future Work **35**

A More Implementation Details

Training Details. All DocLayout-YOLO variants are first pretrained on MeshLayout-1M and subsequently finetuned on each downstream benchmark. Both stages use an input resolution of 1600×1600, a global batch size of 64 distributed across $8\times$ H200 GPUs. We adopt SGD with a momentum of 0.9 and a weight decay of 5×10^{-4}. For downstream finetuning, the initial learning rate is set to 0.02 and linearly decayed to 2×10^{-4}. Pretraining is performed for 50 epochs, and downstream finetuning is performed for 500 epochs. Following the scale-dependent augmentation strategy of YOLOv12, Mixup ratios of 0/0/0/0.15/0.20 and copy-paste ratios of 0/0/0/0.50/0.60 are used for the N/S/M/L/X variants during downstream fine-tuning. Detailed hyperparameters are summarized in Tab. S1. For comparisons with real-time detectors, YOLOv8, YOLOv10, YOLOv11, baseline YOLOv12, are trained from scratch on each downstream dataset using the exact same and identical input resolution, optimizer and learning rate with minor adaptation to ensure a fair comparison. For specialized DLA methods, we report the results provided in their original publications.

Table S1: Hyperparameters for MeshLayout-1M pretraining and downstream fine-tuning of the DocLayout-YOLO.

Hyperparameters	**MeshLayout-1M Pretraining**	**Downstream Fine-tuning**
	Training Configuration	
Model scales	N/S/M/L/X	N/S/M/L/X
Input resolution	1600×1600	1600×1600
Maximum epochs	50	500
Optimizer	SGD	SGD
Momentum	0.9	0.9
Global batch size	64	64
Number of GPUs	8	8
Weight decay	5×10^{-4}	5×10^{-4}
Warm-up epochs	3	3
Warm-up momentum	0.8	0.8
Warm-up bias learning rate	0.0	0.0
Initial learning rate	0.02	0.02
Final learning rate	2×10^{-4}	2×10^{-4}
Learning-rate schedule	Linear decay	Linear decay
	Loss Parameters	
Box loss gain	7.5	7.5
Classification loss gain	0.5	0.5
DFL loss gain	1.5	1.5
	Augmentation Parameters	
HSV hue augmentation	0.015	0.015
HSV saturation augmentation	0.7	0.7
HSV value augmentation	0.4	0.4
Translation augmentation	0.1	0.1
Scale augmentation	0.5	0.5
Mosaic augmentation	1.0	1.0
Mixup augmentation	0	0/0/0/0.15/0.20
Copy-paste augmentation	0	0/0/0/0.50/0.60
Close-mosaic epochs	10	10

Architecture Details. DocLayout-YOLO preserves the macro-architecture and scaling configurations of the baseline YOLOv12, such as the R-ELAN feature aggregation. The number and placement of attention blocks also remain unchanged. Within each block, we retain the original query, key, and value projections, output projection, positional encoding branch, MLP, normalization layers, and residual connections. The only architectural modification is the replacement of area attention with PALA.

YOLOv26 Implementation. We carefully adjust YOLOv26 for optimal performance and a fair comparison. Specifically, each model is trained for 500 epochs at an input resolution of 1600×1600 using $8\times$ NVIDIA H200 GPUs and a global batch size of 64. Following YOLOv26, we use MuSGD with an initial learning rate of 0.02, momentum of 0.937, and weight decay of 5×10^{-4}. The learning rate is linearly decayed to 2×10^{-4}, with a three-epoch warm-up. AMP is enabled during training. We use the same data-augmentation settings as YOLOv12.

RT-DETRv3 Implementation. We also carefully adjust RT-DETR settings for optimal performance and a fair comparison. Following deault settings, all models are trained on $8\times$ NVIDIA H200 GPUs with eight images per GPU, giving a global batch size of 64, and initialized with their corresponding ImageNet-pretrained weights. The evaluation resolution is fixed at 1600×1600, while training uses multi-scale inputs from 1216 to 1984 pixels, obtained by proportionally scaling the official RT-DETRv3 augmentation range. The training pipeline includes random photometric distortion, expansion, cropping, horizontal flipping, and multi-scale resizing. We adopt AdamW with a base learning rate of 4×10^{-4}, a weight decay of 1×10^{-4}, gradient clipping with a maximum norm of 0.1, and EMA with a decay of 0.9999. The backbone learning-rate multiplier is 0.1 for R18/R34/R50 and 0.01 for R101. AMP is enabled, and the number of object queries is fixed to 300. To account for the substantially different dataset sizes, D^4LA, M^6Doc, HJDataset, and PRIMA-LAD are trained for 96/120/240/500 epochs with 700/450/120/50 warm-up iterations, respectively.

TensorRT Details. For TensorRT details, GPU latency is measured on a single NVIDIA H200 GPU using TensorRT 10.16.1.11 with batch size one and a static input shape of $1 \times 3 \times 1600 \times 1600$. Models are exported as simplified FP32 ONNX graphs with opset 17 and compiled into separate FP32 and FP16 engines. After 30 warm-up iterations, synchronized engine execution is measured for at least 5,000 iterations and 60 seconds, excluding data transfer, preprocessing, decoding, and NMS. We report mean latency, whose 95% confidence-interval half-width is below 0.10 ms for all reported measurements.

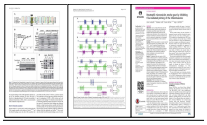
B BENCHMARK & COMPARISON METHODS DETAILS

B.1 PUBLIC BENCHMARK DETAILS

To evaluate the generalization ability of DocLayout-YOLO across diverse document scenarios, we conduct experiments on four main DLA benchmarks, including M^{6}Doc (Cheng et al., 2023), D^{4}LA (Da et al., 2023), HJDataset (Shen et al., 2020), and PRIMA-LAD [1] (Antonacopoulos et al., 2009). These benchmarks cover a wide range of document types, including modern and historical documents, multilingual content, and diverse layout structures.

In addition, we provide supplementary evaluations on other three widely used document layout datasets, including PubLayNet (Zhong et al., 2019), DocLayNet (Pfitzmann et al., 2022), and IndicDLP (Nath et al., 2025), to further examine the robustness of our method across different document domains. Detailed statistics of all benchmarks, including data volume, document types, categories, and example images, are summarized in Tab. S2.

Table S2: Details of public benchmarks used for evaluation.

Benchmark	# Training	# Testing	Document Types	Categories	Example Documents
M^{6}Doc	5,448	2,724	Book, Magazine, Newspaper, Note, Paper, Test Paper, Textbook	74 fine-grained categories	
D^{4}LA	8,868	2,224	Budget, Email, Form, Invoice, Letter, Memo, News, Presentation, Resume	DocTitle, ListText, LetterHead, Question, RegionList, TableName, FigureName, Footer, Number, ParaTitle, RegionTitle, LetterDear, OtherText, Abstract, Table, Equation, PageHeader, Catalog, ParaText, Date, LetterSign, RegionKV, Author, Figure, Reference, PageFooter, PageNumber	
HJDataset	1,589	341	Main, Advertisement, Index, Other	Body, Row, Title, Bio, Name, Position, Other	
PRIMA-LAD	382	96	Magazine, Technical Article	Figure, Table, Paragraph, Header, Page-number, Floating, Caption, Credit, Drop-capital, Heading, Footer	
PubLayNet	335,703	11,245	Academic	Text, Title, List, Table, Figure	
DocLayNet	69,103	6,480	Financial, Scientific, Patents, Manuals, Laws, Tenders	Caption, Footnote, Formula, List-item, Page-footer, Page-header, Picture, Section-header, Table, Text, Title	
IndicDLP	95,824	11,994	Acts & Rules, Brochures, Forms, Magazines, Manuals, Newspapers, Notices, Novels, Question Papers, Research Papers, Syllabi, Textbooks	42 fine-grained categories	

[1] For PRIMA-LAD, we use the 11-class annotation configuration adopted in our experiments, containing 8,942 valid regions over 478 pages.

B.2 Comparison Methods

We compare DocLayout-YOLO with representative methods from three categories: (1) real-time object detectors, which focus on achieving favorable accuracy–efficiency trade-offs; (2) specialized unimodal DLA methods, which rely primarily on visual information for document layout understanding; and (3) specialized multimodal DLA methods, which leverage additional OCR annotation and language semantic representation through vision-language modeling. The complete list of comparison methods is summarized in Tab. S3.

Table S3: Summary of comparison methods.

Category	Methods
Real-time Detectors	YOLOv8 (Jocher et al., 2023), YOLOv10 (Wang et al., 2024a), YOLOv11 (Khanam & Hussain, 2024), YOLOv12 (Tian et al., 2025), YOLOv26 (Jocher et al., 2026), RT-DETRv3 (Wang et al., 2025)
Specialized DLA	*Unimodal* LayoutParser (Shen et al., 2021), DocSegTr (Biswas et al., 2022), DiT-L (Li et al., 2022), SelfDocSeg (Maity et al., 2023), SwinDocSegmenter (Banerjee et al., 2023), GraphKD (Banerjee et al., 2024a), SemiDocSeg (Banerjee et al., 2024b), RoDLA (Chen et al., 2024b), PARL (Liu et al., 2026), HybriDLA (Chen et al., 2026), DocSAM (Li et al., 2025)
Specialized DLA	*Multimodal* LayoutLMv3 (Huang et al., 2022), VGT (Da et al., 2023), M2Doc (Zhang et al., 2024), DoPTA-HR (R. et al., 2024), PromptDLA (Zhang et al., 2026)

C MORE EXPERIMENTS

C.1 RESULTS ON MORE COMPARISON DATASETS

As shown in Tab. S4, we further evaluate DocLayout-YOLO on three additional document layout benchmarks, including PubLayNet, DocLayNet, and IndicDLP. Compared to YOLOv12-X, DocLayout-YOLO-X consistently improves mAP by 0.6%/1.3%/0.6%, respectively. The consistent gains across datasets with different document domains and layout characteristics further demonstrate the generalization ability of our proposed method.

Table S4: Comparison results on PubLayNet, DocLayNet, and IndicDLP.

Method	PubLayNet	DocLayNet	IndicDLP
YOLOv12-X	96.9	81.0	56.3
DocLayout-YOLO-X	**97.5**	**82.3**	**56.9**

C.2 ROBUSTNESS EVALUATION

Tabs. S5 and S6 evaluate robustness under 12 document perturbations at three severity levels on PubLayNet-P and M^{6}Doc-P, respectively (Chen et al., 2024b). P-Avg denotes the average accuracy over all perturbations, while mRD measures normalized performance degradation, with lower mRD and higher P-Avg indicating stronger robustness. On PubLayNet-P, DocLayout-YOLO-X achieves 97.5% clean mAP, 85.8% P-Avg, and 48.7 mRD, improving over baseline YOLOv12-X by 0.6%/9.1% in clean mAP/P-Avg and reducing mRD by 30.7. It also surpasses the previous best RoDLA by 15.8% on P-Avg while lowering mRD by 67.3. On M^{6}Doc-P, DocLayout-YOLO-X achieves 75.5% clean mAP, 68.0% P-Avg, and 118.8 mRD, outperforming the baseline YOLOv12-X by 1.5%/0.3% in clean mAP/P-Avg with a 2.6 reduction in mRD. Compared with the previous best results, it further improves clean mAP/P-Avg by 5.3%/6.3% and reduces mRD by 28.8. These results suggest that DocLayout-YOLO provides improved robustness to diverse document degradations without noticeably compromising clean-document performance.

Table S5: Robustness evaluation on the PubLayNet-P dataset. V, L, and T denote visual, layout, and textual modalities, respectively. **Best** and second-best results are highlighted.

Method	Backbone	Modality V	L	T	Clean	Spatial P1	P2	P3	Content P4	P5	Inconsistency P6	P7	P8	Blur P9	P10	Noise P11	P12	P-Avg↑	mRD↓
LayoutParser (Shen et al., 2021)	ResNeXt (Xie et al., 2017)	✓	✗	✗	89.0	35.8	78.1	68.0	84.8	45.9	79.0	85.8	82.6	51.4	31.1	53.6	0.3	58.0	212.7
Faster R-CNN (Ren et al., 2015)	ResNet (He et al., 2016)	✓	✗	✗	90.2	**44.2**	75.3	74.3	78.9	53.2	81.4	82.7	80.2	80.6	63.2	55.7	24.3	66.2	175.5
DocSegTr (Biswas et al., 2022)	ResNet (He et al., 2016)	✓	✗	✗	90.4	28.3	76.0	71.2	85.2	46.2	69.3	86.2	68.9	50.8	19.8	60.2	0.4	55.2	233.0
Mask R-CNN (He et al., 2017)	ResNet (He et al., 2016)	✓	✗	✗	91.0	40.0	74.0	71.8	76.9	48.7	79.0	80.6	77.4	78.2	54.2	55.1	31.9	64.0	192.7
SwinDocSegmenter (Banerjee et al., 2023)	Swin (Liu et al., 2021)	✓	✗	✗	93.7	39.0	83.3	61.3	88.9	46.6	87.9	88.1	86.6	37.3	29.2	36.1	0.5	57.1	214.4
Cascade R-CNN (Cai & Vasconcelos, 2018)	DiT (Li et al., 2022)	✓	✗	✗	94.5	31.9	**86.5**	**82.2**	92.1	79.6	87.2	92.0	91.6	93.8	71.3	69.9	41.8	76.7	95.8
Cascade R-CNN (Cai & Vasconcelos, 2018)	LayoutLMv3 (Huang et al., 2022)	✓	✓	✓	95.1	32.7	85.9	79.8	92.3	68.5	86.5	93.1	86.7	82.9	47.0	82.1	45.1	73.6	116.2
Co-DINO (Zong et al., 2023)	Swin (Liu et al., 2021)	✓	✗	✗	94.3	22.4	43.0	24.7	92.6	56.8	86.4	72.6	75.8	35.0	20.2	55.4	14.9	50.0	254.1
Cascade R-CNN (Cai & Vasconcelos, 2018)	InternImage (Wang et al., 2023)	✓	✗	✗	94.1	27.7	80.2	79.7	89.6	56.6	83.8	91.6	89.6	84.9	54.0	55.2	0.2	66.1	141.9
DINO (Zhang et al., 2023a)	InternImage (Wang et al., 2023)	✓	✗	✗	95.4	34.4	82.2	**82.2**	92.3	57.7	91.7	92.8	92.1	90.9	63.4	47.3	1.3	69.0	120.7
Co-DINO (Zong et al., 2023)	InternImage (Wang et al., 2023)	✓	✗	✗	94.2	19.1	48.3	35.7	91.3	60.6	87.9	81.4	87.6	37.9	27.9	49.7	11.2	53.2	230.3
RoDLA (Chen et al., 2024b)	InternImage (Wang et al., 2023)	✓	✗	✗	96.0	31.6	79.3	80.6	92.9	61.6	91.6	92.6	91.6	91.3	67.7	58.8	0.3	70.0	116.0
YOLOv12-X (Tian et al., 2025)	v12-X	✓	✗	✗	96.9	29.4	79.5	81.7	95.9	87.1	95.0	96.7	95.9	96.2	72.9	76.9	13.2	76.7	79.4
DocLayout-YOLO-X	**v12-X + PALA**	✓	✗	✗	**97.5**	21.2	85.7	80.3	**97.4**	**96.4**	**97.5**	**97.2**	**96.3**	**96.4**	**76.3**	**89.3**	**95.9**	**85.8**	**48.7**

Table S6: Robustness evaluation on the M^{6}Doc-P dataset.

Method	Backbone	Modality V	L	T	Clean	P-Avg↑	mRD↓
DocSegTr (Biswas et al., 2022)	ResNet (He et al., 2016)	✓	✗	✗	60.3	43.2	212.6
Mask R-CNN (He et al., 2017)	ResNet (He et al., 2016)	✓	✗	✗	61.9	48.9	216.7
Faster R-CNN (Ren et al., 2015)	ResNet (He et al., 2016)	✓	✗	✗	62.0	49.6	192.2
SwinDocSegmenter (Banerjee et al., 2023)	Swin (Liu et al., 2021)	✓	✗	✗	47.1	39.7	239.2
Cascade R-CNN (Cai & Vasconcelos, 2018)	DiT (Li et al., 2022)	✓	✗	✗	70.2	60.6	164.8
Cascade R-CNN (Cai & Vasconcelos, 2018)	LayoutLMv3 (Huang et al., 2022)	✓	✓	✓	64.3	57.0	176.1
RoDLA (Chen et al., 2024b)	InternImage (Wang et al., 2023)	✓	✗	✗	70.0	61.7	147.6
YOLOv12-X (Tian et al., 2025)	v12-X	✓	✗	✗	74.0	67.7	121.4
DocLayout-YOLO-X	**v12-X + PALA**	✓	✗	✗	**75.5**	**68.0**	**118.8**

C.3 HEATMAP VISUALIZATION

To further examine the learned spatial representations, we visualize the third-stage (S3) feature maps of the X-scale YOLOv12 baseline and DocLayout-YOLO. As shown in Fig. S1, the baseline YOLOv12 produces relatively diffuse responses, particularly on pages with dense text, tabular structures, and multi-column layouts. In contrast, DocLayout-YOLO generates more structured activations that better align with semantically meaningful regions, including text blocks, table cells, figures, and their surrounding boundaries. This improvement is especially evident in layouts with strong horizontal and vertical dependencies, where PALA maintains coherent responses across related elements while suppressing irrelevant background activation. These observations qualitatively support the effectiveness of PALA in enhancing long-range page-context modeling and element-level representation for document layout analysis.

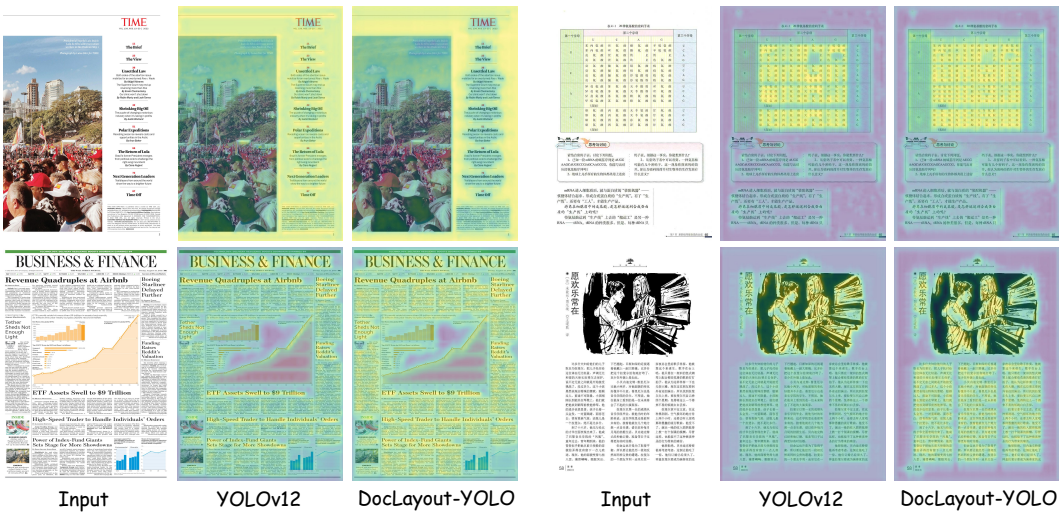


Figure S1: Comparison of heatmaps between YOLOv12 and the proposed DocLayout-YOLO. DocLayout-YOLO exhibits clearer object perception than YOLOv12.

C.4 DESIGN-HEURISTIC METRIC EVALUATION EVALUATION ON MESHLAYOUT-1M

In this section, we quantitatively evaluate whether the synthetic document page produced by Mesh-Candidate BestFit aligns with the human design principle. The evaluation employs the *Align* and *Density* metrics, which respectively measure the aesthetic quality of layouts in terms of document alignment and density. For *Align*, we utilize the LayoutGAN++ (Kikuchi et al., 2021; Li et al., 2021) metric which measures the alignment of elements in the document:

$$L_{alg} = \sum_{i=1}^{N} \min \begin{pmatrix} g(\Delta x_i^L), g(\Delta x_i^C), g(\Delta x_i^R) \\ g(\Delta y_i^T), g(\Delta y_i^C), g(\Delta y_i^B) \end{pmatrix}. \tag{5}$$

where $x_i^*(* = L, C, R), y_i^*(* = T, C, B)$ denotes the x-axis left/center/right and y-axis top/center/bottom of i-th elements in document, $g(x) = -\log(1-x)$, and $\Delta x_i^*(* = L, C, R)$ is computed as:

$$\Delta x_i^* = \min_{\forall j \neq i} |x_i^* - x_j^*| \tag{6}$$

$\Delta y_i^*(* = T, C, B)$ can be computed similarly. For *Density*, we calculate the ratio of filled area in the layout:

$$L_{dst} = \frac{\sum_{i=1}^{N} |e_i|}{|L|} \tag{7}$$

where $|e_i|$ denotes area of element e_i in L, and $|L|$ denotes area of the whole layout. For *Align*, a lower value denotes a more aligned document. For *Density*, a larger value denotes a more compact and dense layout. The experimental results, as shown in Tab. S7. Results indicate that the Mesh-candidate BestFit method significantly outperforms diffusion and random methods in both alignment and density. Visual results further confirm that the layouts produced by Mesh-candidate BestFit better conform to the standards of human aesthetics and design principles.

Table S7: Quantitative comparison between different layout generation methods.

Layout Generation	Align↓	Density↑
Random	0.0171	0.259
Diffusion (LACE)	0.0032	0.476
Mesh-candidate BestFit (ours)	**0.0009**	**0.645**

C.5 Ablation on Pretraining Data Volume

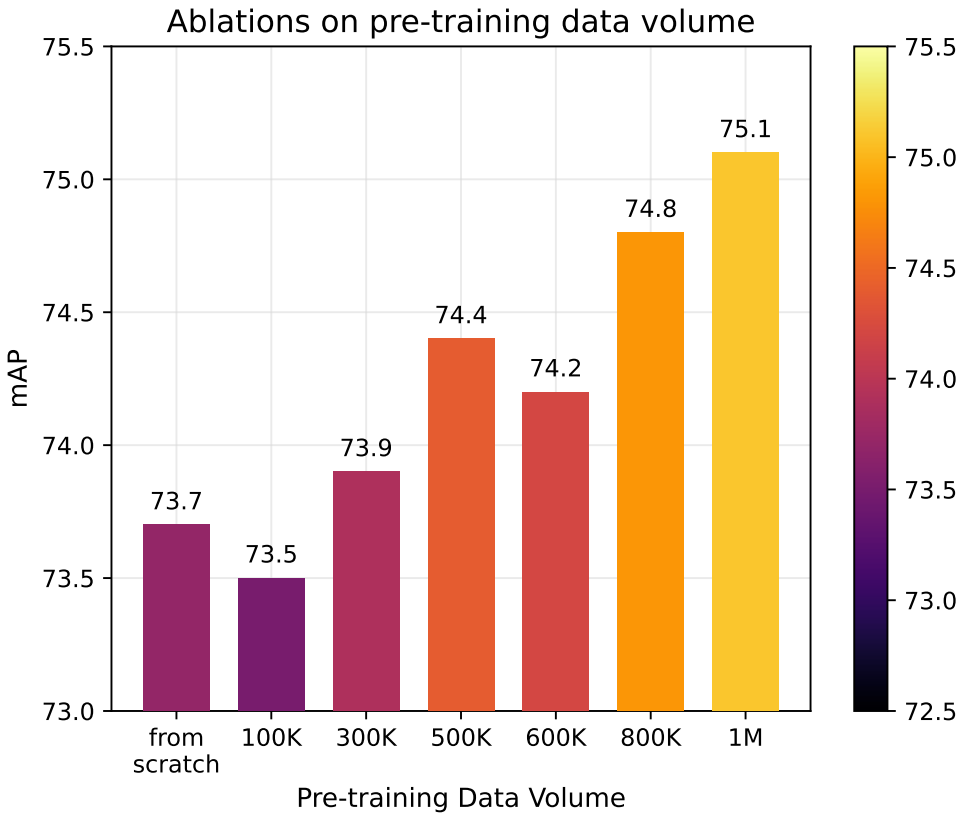


Figure S2: Ablations on pretraining data volume.

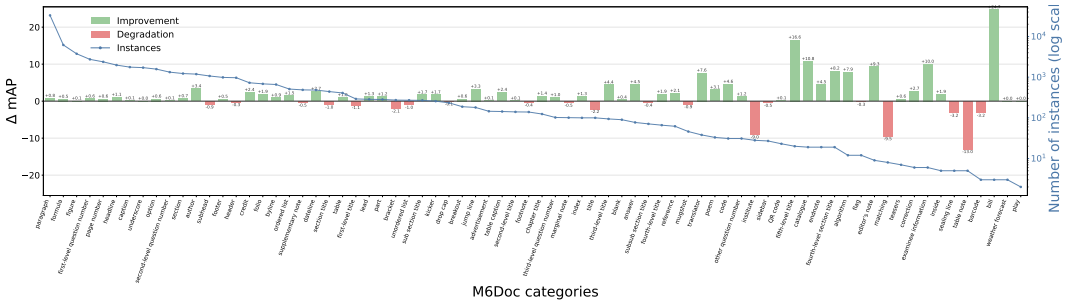
We conduct ablation experiments on the pretraining data volume of MeshLayout-1M. Specifically, we pretrain DocLayout-YOLO-L using different amounts of Mesh-candidate generated data, ranging from 0 to 1M pages, and subsequently fine-tune it on M^{6}Doc. Results are shown in Fig. S2. Overall, model performance improves as the pretraining data volume increases, despite minor fluctuations at intermediate scales. The mAP rises from 73.7% without pretraining to 74.4% with 500K pages, reaches 74.8% at 800K, and further improves to 75.1% with 1M pages. These results indicate a clear benefit from scaling the synthetic pretraining corpus, while the marginal gains become smaller at larger data volumes.

C.6 Effects of MeshLayout-1M Pretraining on Long-tail Distribution

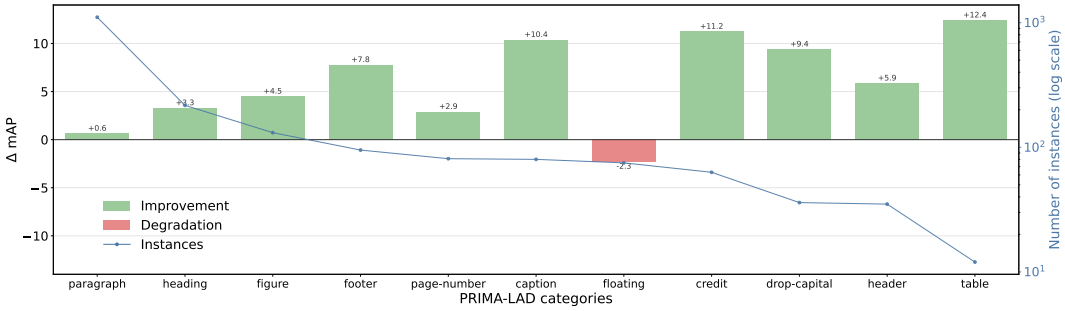
We analyze the effect of MeshLayout-1M pretraining under long-tailed category distributions. The class-wise results on M^{6}Doc and PRIMA-LAD are shown in Fig. S3a and Fig. S3b, respectively. The left y-axis reports the AP change for each category, while the right y-axis shows the number

of instances. Δ mAP denotes the performance difference between DocLayout-YOLO-L with and without MeshLayout-1M pretraining.

Overall, MeshLayout-1M yields improvements across a broad range of downstream categories, although a small number of classes exhibit performance degradation. The gains are particularly pronounced for several low-frequency categories, suggesting that diverse synthetic pretraining can alleviate long-tail effects to some extent. For example, on M^{6}Doc, the improvements reach 7.6% for translator, 10.8% for catalog, and 9.3% for editor's note. On PRIMA-LAD, caption, credit, and table improve by 10.4%, 11.2%, and 12.4%, respectively.



(a) M^{6}Doc.



(b) PRIMA-LAD.

Figure S3: Class-wise metric analysis. Left Y-Axis indicate class-wise AP and right Y-Axis indicate number of instances for each category.

D Additional Analysis of Page-Axis Layout Attention (PALA)

D.1 More Analysis & Comparisons of Different Local Attentions

Comparison with Different Attention Mechanisms. Fig. S4 illustrates the structural differences between PALA and representative local attention mechanisms. Swin Attention restricts interactions within local windows, while Area Attention partitions the feature map into disjoint regions, preventing direct context aggregation across region boundaries. Criss-Cross Attention extends interactions along both spatial axes in parallel, but jointly normalizes horizontal and vertical attention weights. Axial Attention instead performs separately normalized axis-wise attention sequentially, making the second direction dependent on the output of the first. In contrast, PALA performs horizontal and vertical attention in parallel with separate normalization, allowing the two directional contexts to be modeled independently while preserving long-range dependencies along both page axes. This design preserves complementary horizontal and vertical structural cues without local partitioning, cross-axis competition, or directional-order dependency.

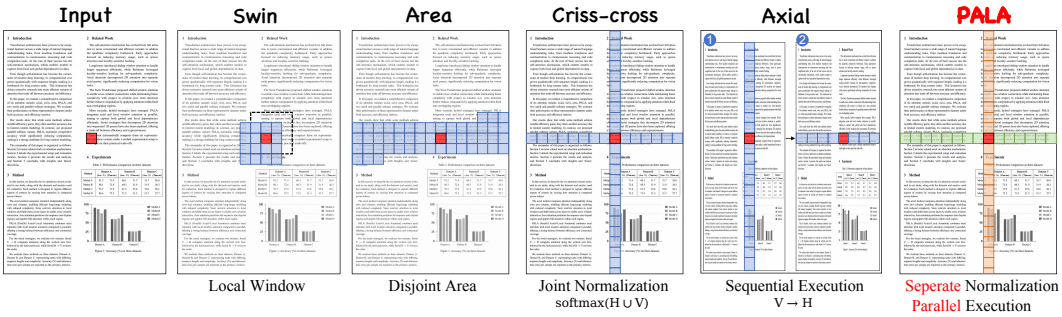


Figure S4: Comparison of representative local attention mechanisms with our proposed PALA.

Horizontal vs. Vertical Context Modeling. Following Tab. 7, we further isolate the horizontal and vertical branches of PALA in Tab. S8. Horizontal-only consistently outperforms Vertical-only across both datasets and model scales, with gains of 0.2%/1.3% mAP on M^{6}Doc and 0.5%/2.1% mAP on PRIMA-LAD for the N/L models, respectively. We notice that this asymmetry is consistent with the geometry of document elements. As shown in Tab. S9, 89.0% and 83.8% of elements in M^{6}Doc and PRIMA-LAD are wider than tall, while 63.9% and 58.2% have aspect ratios of at least 2:1, respectively. Horizontal attention therefore better covers elongated structures such as text lines, headings, captions, and table rows. Row-wise context may also be more semantically coherent, as tokens along the same row often belong to the same layout element. Nevertheless, vertical attention remains complementary for modeling paragraph columns, table columns, vertically stacked regions, and cross-region correspondences.

Table S8: Comparison of Horizontal-only, Vertical-only, and complete PALA.

Dataset	Size	Horizontal-only	Vertical-only	PALA	$\Delta(H-V)$
M^{6}Doc	N	64.1	63.9	65.3	+0.2
M^{6}Doc	L	72.7	71.4	73.7	+1.3
PRIMA-LAD	N	60.6	60.1	62.6	+0.5
PRIMA-LAD	L	67.0	64.9	69.1	+2.1

Why Global Attention Underperforms Criss-Cross and PALA. Following Tab. 7, we conduct a diagnostic analysis of Global Attention as shown in Tab. S10. Global Attention achieves the highest recall (76.8%) among the three variants, but its precision is substantially lower than Criss-Cross and PALA (83.0% vs. 86.7%/88.1%), resulting in the lowest mAP (64.5% vs. 64.7%/65.3%). The higher recall but lower precision indicates that the performance degradation of Global Attention primarily stems from additional false positives rather than missed detections. This suggests that unrestricted global interactions may introduce interference between visually similar but spatially

Table S9: Geometric statistics of valid document elements in the official dataset splits. Aspect ratios are computed from pixel-space axis-aligned bounding boxes.

Dataset	# Boxes	Wider than tall	Aspect ratio $\geq$ 2:1	Aspect ratio $\geq$ 4:1	Mean aspect ratio	Geometric-mean aspect ratio
M^{6}Doc	237,116	89.0%	63.9%	41.5%	5.22	3.19
PRIMA-LAD	8,942	83.8%	58.2%	40.1%	6.05	3.02

unrelated document regions. In contrast, Criss-Cross and PALA constrain long-range interactions along the page axes, reducing irrelevant cross-region context while preserving structurally relevant dependencies.

Table S10: Detection diagnostics for Global, Criss-Cross, and PALA.

Dataset	Attention	Precision	Recall	mAP$_{50:95}$
M^{6}Doc	Global	83.0	76.8	64.5
M^{6}Doc	Criss-Cross	86.7	75.5	64.7
M^{6}Doc	PALA	88.1	76.3	65.3

D.2 Distinguishing PALA from Criss-Cross and Axial Attention

To clarify the relationship between PALA, Criss-Cross Attention, and Axial Attention, we compare their formulations along two key dimensions. Although all three attentions model long-range context along the horizontal and vertical axes, they differ in *axis execution* (parallel vs. sequential) and *attention normalization* (joint vs. separate), as summarized in Tab. S11.

Table S11: Structural differences between Criss-Cross Attention, Axial Attention, and PALA.

Method	Axis Execution	Attention Normalization
Criss-Cross Attention	Parallel	Joint
Axial Attention	Sequential	Separate
PALA	Parallel	Separate

PALA vs. Criss-Cross Attention. Both PALA and Criss-Cross Attention aggregate horizontal and vertical context in parallel. Their main difference lies in how attention weights along the two axes are normalized. For a given query, let $\mathbf{s}_{hor}$ and $\mathbf{s}_{ver}$ denote the attention logits over its horizontal and vertical context, respectively. Criss-Cross Attention concatenates the two sets of logits and applies a joint softmax:

$$\left[A_{hor}^{\mathrm{CC}}, A_{ver}^{\mathrm{CC}}\right] = \mathrm{Softmax}\left(\left[\mathbf{s}_{hor}, \mathbf{s}_{ver}\right]\right). \tag{8}$$

Thus, horizontal and vertical context compete within a single attention distribution, such that strong responses along one axis can suppress attention along the other.

In contrast, PALA normalizes the two axes independently:

$$A_{hor}^{\mathrm{PALA}} = \mathrm{Softmax}(\mathbf{s}_{hor}), \qquad A_{ver}^{\mathrm{PALA}} = \mathrm{Softmax}(\mathbf{s}_{ver}). \tag{9}$$

This design treats horizontal and vertical context as two independent sources of structural information. Such separation is well suited to document layouts, where the two axes often represent distinct relationships, e.g., text lines and table rows horizontally, versus paragraph columns and vertically stacked regions vertically.

PALA vs. Axial Attention. Both PALA and Axial Attention normalize horizontal and vertical attention independently, but differ in how the two axes are executed. Axial Attention applies them sequentially:

$$Y^{\mathrm{Axial}} = \mathcal{A}_{hor}\left(\mathcal{A}_{ver}(X)\right), \tag{10}$$

where the second attention operation receives features already transformed by the first. PALA instead computes both directions in parallel from the same projected features:

$$Y_{hor} = \mathcal{A}_{hor}(X), \qquad Y_{ver} = \mathcal{A}_{ver}(X). \tag{11}$$

Neither branch depends on the output of the other, avoiding sequential dependency and directional-order sensitivity while preserving independently normalized context along both page axes.

Experimental Results As shown in Tab. S12, PALA achieves the best performance in three out of four settings. Compared with Sequential + Separate, parallel execution consistently improves mAP by 0.8%–3.2%, indicating the benefit of avoiding sequential dependency between the two axes. Compared with Parallel + Joint, separate normalization yields comparable performance on M^{6}Doc and improves PRIMA-LAD results by 0.2%–2.8%, suggesting that avoiding cross-axis competition generally provides more robust structural context modeling.

Table S12: Ablation study of the execution and normalization strategies in PALA.

Dataset	Size	Parallel + Separate (PALA)	Parallel + Joint	Sequential + Separate
M^{6}Doc	N	65.3	**65.5**	63.4
	L	**73.7**	73.1	72.9
PRIMA-LAD	N	**62.6**	59.8	59.6
	L	**69.1**	67.6	65.9

Summary. PALA can therefore be distinguished from the two related designs by combining *parallel axis execution* with *separate axis normalization*. Compared with Criss-Cross Attention, PALA removes competition between horizontal and vertical attention distributions; compared with Axial Attention, PALA removes the sequential dependency between the two directional operations.

D.3 Area Attention vs. PALA: Is PALA Always More Efficient?

A natural question is whether PALA is always more efficient than Area Attention, or whether Area Attention can achieve lower latency by using finer area partitioning. We investigate this question from both theoretical complexity and practical inference performance.

Theoretical Analysis. Ignoring linear projections, the attention aggregation complexities of Global Attention, Area Attention with A equal-sized areas, and PALA are $O(BH^2W^2C)$, $O(BH^2W^2C/A)$, and $O(BHW(H+W)C)$, respectively. Equating the latter two gives the complexity boundary

$$A^* = \frac{HW}{H+W}. \tag{12}$$

When $A < A^*$, PALA has lower attention aggregation complexity than Area Attention, whereas sufficiently fine-grained partitioning ($A > A^*$) can make Area Attention theoretically more efficient. Therefore, PALA is not theoretically more efficient for arbitrary area configurations. We further examine their practical accuracy–efficiency trade-off under different area settings.

Experimental Setting and Results. For a fair comparison, we train the YOLOv12-N model with different numbers of areas under the same 1600×1600 input resolution, training recipe, and TensorRT inference protocol. The accuracy-latency trade-off results are shown in Fig. S5. For Area Attention, the notation A_{P4}/A_{P5} denotes the numbers of areas used at the P4 and P5 feature levels, respectively; for example, $4/1$ is the default YOLOv12 configuration, while $16/4$ applies finer partitioning at both levels. At an input resolution of 1600×1600, the theoretical boundaries A^* are 50 and 25 for the A_{P4} and A_{P5}, respectively. Considering the default $4/1$ configuration in YOLOv12, we further evaluate $1/1$, $4/2$, $8/4$, and $16/4$ configurations to examine the accuracy–efficiency trade-off under different levels of area partitioning.

As shown in Fig. S5, increasing the number of areas reduces the size of each attention region and generally leads to lower computation latency. Accordingly, the FP32 latency of Area Attention

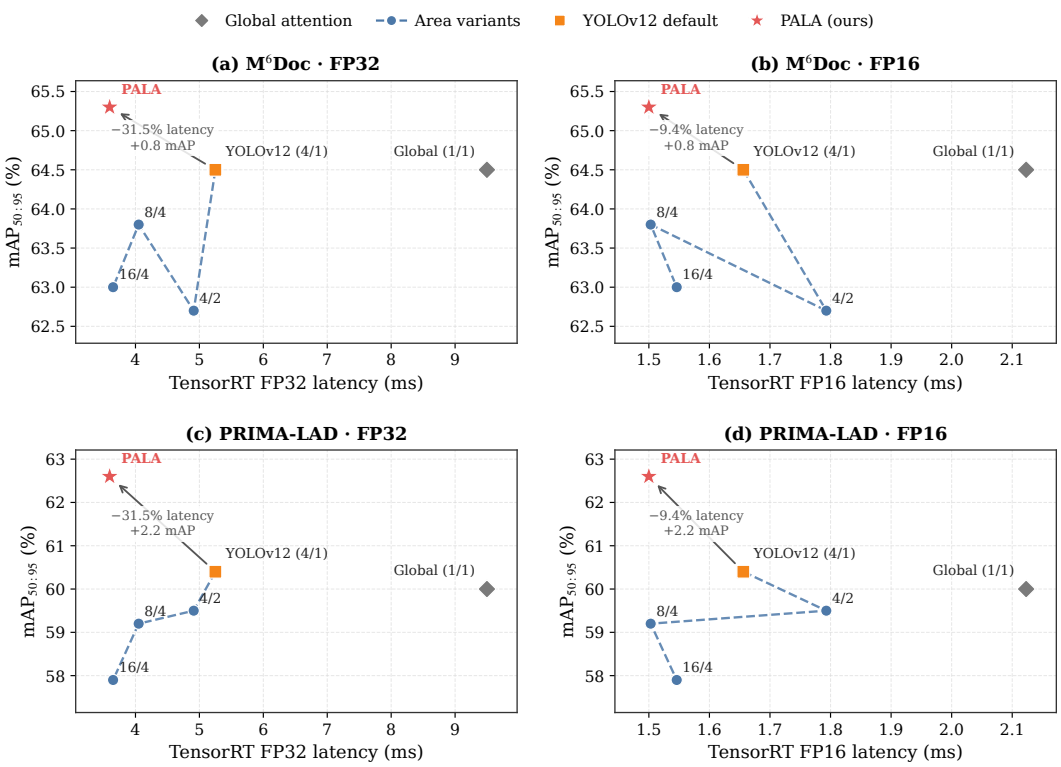


Figure S5: Accuracy–latency trade-offs of Global Attention, Area Attention with different (A_{P4}/A_{P5}) configurations, and PALA.

decreases from 5.250 ms at $4/1$ to 3.650 ms at $16/4$. However, finer partitioning also restricts context to increasingly smaller regions and leads to clear accuracy degradation: the $16/4$ configuration achieves only 63.0%/57.9% mAP on M^{6}Doc/PRIMA-LAD, compared with 64.5%/60.4% for the default $4/1$ configuration. PALA instead achieves 65.3%/62.6% mAP with 3.596/1.500 ms FP32/FP16 latency, preserving full horizontal and vertical interactions while remaining faster than the Area Attention configurations with comparable accuracy.

Summary. In conclusion, Area Attention is not inherently less efficient than PALA: sufficiently fine partitioning can further reduce its computational cost, but at the expense of restricted context and detection accuracy. In summary, PALA provides a better speed-accuracy trade-off by combining efficient axis-wise attention with long-range document context modeling.

E MESH-CANDIDATE BESTFIT DETAILS

E.1 DOCBANK-BASED ELEMENT POOL CONSTRUCTION

We construct the element pool exclusively from the training split of DocBank (Li et al., 2020), rather than extracting elements from any downstream evaluation benchmark. We exclude the *Date* category because it is rare and visually homogeneous, being largely derived from LaTeX `\date...`, `\today`, or journal-template `Dated:` fields and thus prone to source-specific overfitting, and the *Section* category because its weakly supervised annotations exhibit substantial fragmentation and label noise. For each annotated region, we crop the corresponding visual content together with its bounding-box geometry and retain its semantic label under the DocBank-based taxonomy. Invalid boxes are discarded, and elements whose normalized width or height is smaller than $\epsilon = 0.01$ are removed. This filtering suppresses annotation fragments and extremely thin crops that provide little useful visual or structural information. The retained instances are organized into category-specific pools, from which the page-level candidate set is subsequently constructed.

E.2 SCALE-STRATIFIED CANDIDATE SAMPLING

Directly sampling from the complete element pool may overrepresent small or narrow elements, causing the page to be filled by fragmented regions before major content areas establish a stable structure. We therefore partition the filtered elements into large- and small-scale group according to their normalized scales. An element is assigned to the large-scale group when both its normalized width and height are no smaller than $\tau_s = 0.05$; all remaining valid elements form the small-scale group. For each synthetic page, a fixed-size candidate set of $K = 500$ elements is constructed by independently sampling the two strata with a predefined large-to-small ratio of $\rho : (1 - \rho)$, where $\rho = 0.99$. The sampled subsets are then merged, and one candidate element is randomly drawn to initialize the page. This strategy preserves fine-grained elements while prioritizing the formation of major layout regions. Restricting BestFit matching to a fixed-size candidate set also bounds its search overhead and introduces page-to-page compositional diversity.

E.3 MESH-CANDIDATE BESTFIT ALGORITHM

After initialization, Mesh-candidate BestFit iteratively updates the available page regions and searches for the most compatible element–region pair. Algorithm 1 summarizes the whole process of Mesh-candidate BestFit. Given the current layout L, MESHENGINE constructs a set of non-overlapping empty grids from the page boundaries and the horizontal and vertical boundaries of the placed elements. For a candidate element e_i and an empty grid g_j, the matching score is defined as

$$\mathrm{match}(e_i, g_j) = \begin{cases} \dfrac{w_i h_i}{w_j h_j}, & w_i \le w_j \text{ and } h_i \le h_j, \\ 0, & \text{otherwise}, \end{cases} \tag{13}$$

where (w_i, h_i) and (w_j, h_j) denote the width and height of the element and grid, respectively. The score favors candidates that occupy the available region compactly while satisfying the size constraint. At each iteration, the candidate–grid pair with the highest score is selected, the candidate is centered within the matched grid, and the mesh is recomputed. Generation stops when the layout contains $N = 35$ elements or no valid pair exceeds $fr_{\mathrm{thr}} = 10^{-4}$.

Once the layout is finalized, a blank document canvas is created by sampling its width as $W \sim \mathcal{U}_{\mathrm{int}}(1200, 2000)$ and its aspect ratio as $r = H/W \sim \mathcal{U}(0.7, 1.5)$, with H capped at 3000 pixels. Each placed candidate box is mapped from normalized coordinates to the canvas and independently contracted about its center along the width and height using factors sampled from $\mathcal{U}(0.8, 0.95)$. The corresponding element crop is then resized to fit the resulting box and composited onto the page, while its coordinates and semantic label are retained as annotations.

Algorithm 1: Mesh-candidate BestFit Algorithm

Input: Element pool P; candidate-set size K; maximum layout size N; filtering threshold ϵ; scale threshold τ_s; large-element ratio ρ; matching threshold fr_{thr}
Output: Generated layout L

/* **Stage I: Element Filtering and Scale-aware Candidate Sampling** */
$P \leftarrow \text{Filter}(P, \epsilon)$;
$P_{\text{large}}, P_{\text{small}} \leftarrow \text{ScalePartition}(P, \tau_s)$;
$C_{\text{large}} \leftarrow \text{Sample}(P_{\text{large}}, \lfloor \rho K \rfloor)$;
$C_{\text{small}} \leftarrow \text{Sample}(P_{\text{small}}, K - \lfloor \rho K \rfloor)$;
$C_{\text{set}} \leftarrow C_{\text{large}} \cup C_{\text{small}}$;
/* **Stage II: Layout Initialization** */
$e^* \leftarrow \text{RandomSample}(C_{\text{set}})$;
$L \leftarrow \text{Initialize}(e^*)$;
$C_{\text{set}} \leftarrow C_{\text{set}} \setminus \{e^*\}$;
/* **Stage III: Iterative Mesh Construction and BestFit Matching** */
while $|L| < N \ \wedge \ C_{\text{set}} \neq \emptyset$ **do**
  // Construct candidate regions from the current layout
  $M_{\text{set}} \leftarrow \text{MeshEngine}(L)$;
  // Initialize the best element--region match
  $fr_{\max} \leftarrow 0$;
  $e_{\text{best}} \leftarrow \varnothing$;
  $g_{\text{best}} \leftarrow \varnothing$;
  // Search for the globally optimal BestFit match
  foreach $e_i \in C_{\text{set}}$ **do**
    foreach $g_j \in M_{\text{set}}$ **do**
      $fr \leftarrow \text{Match}(e_i, g_j)$;
      if $fr > fr_{\max}$ **then**
        $fr_{\max} \leftarrow fr$;
        $e_{\text{best}} \leftarrow e_i$;
        $g_{\text{best}} \leftarrow g_j$;
      end
    end
  end
  // Terminate if no valid match is available
  if $e_{\text{best}} = \varnothing \ \vee \ fr_{\max} < fr_{\text{thr}}$ **then**
    break;
  end
  // Place the selected element and update the candidate set
  $L \leftarrow L \cup \{\text{PlaceCenter}(e_{\text{best}}, g_{\text{best}})\}$;
  $C_{\text{set}} \leftarrow C_{\text{set}} \setminus \{e_{\text{best}}\}$;
end
return L;

F VISUAL DIAGNOSIS OF MESHLAYOUT-1M

In this section, we conduct a series of visual analyzes to demonstrate both the diversity of the synthesized pretraining dataset and the effectiveness of our proposed document synthesis method.

F.1 SYNTHETIC DOCUMENTS ILLUSTRATION

Firstly, a richer visualization of the Mesh-candidate BestFit generated document is shown in Fig. S6. S, M, L respectively denote small, medium, and large elements, indicating the components that are relatively abundant on the page. It is evident that the data we generate is rich in categories and possesses strong diversity. It can generate dense layouts containing many small elements and produce sparse layouts composed of a few large elements, similar to the layouts generated by diffusion-based models.

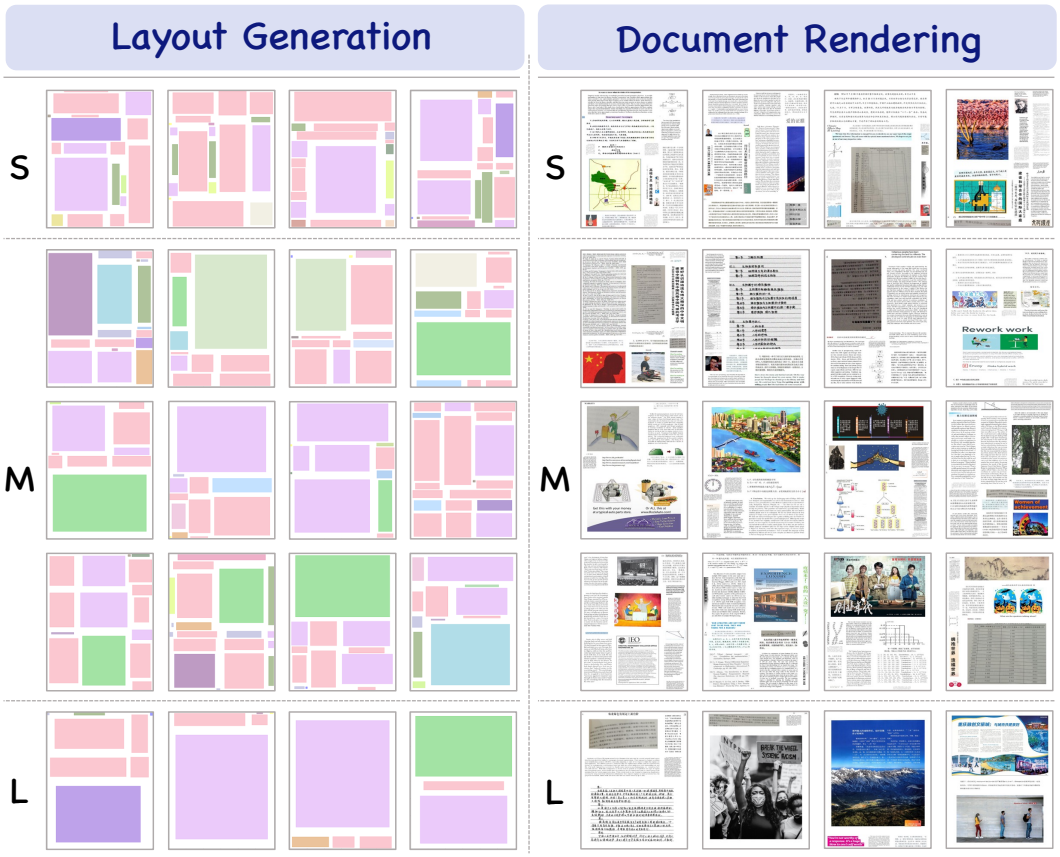
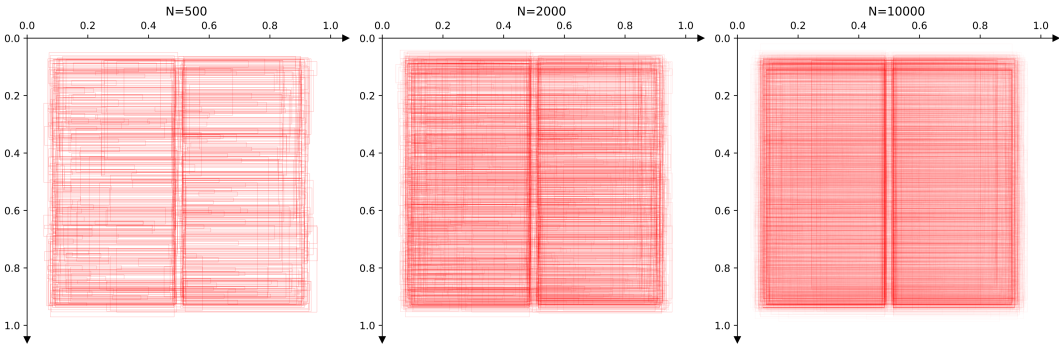


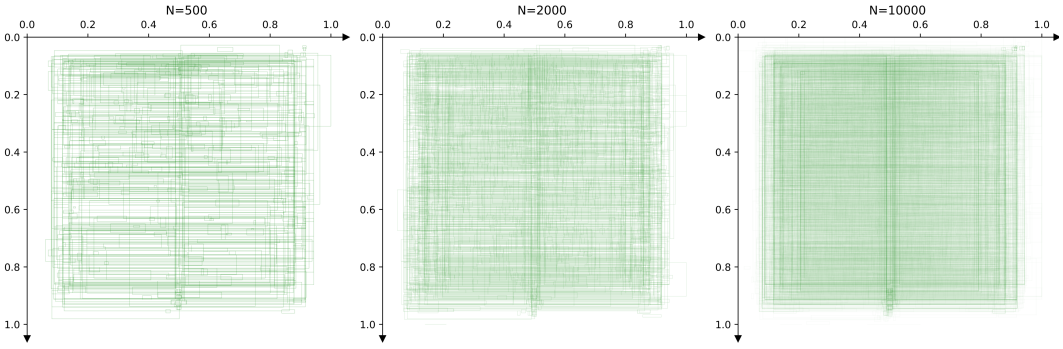
Figure S6: Visualization of generated diverse layouts and corresponding pages after rendering. *S, M, L* respectively denote small, medium, and large elements, indicating the components that are relatively abundant on the page.

F.2 LAYOUT DIVERSITY

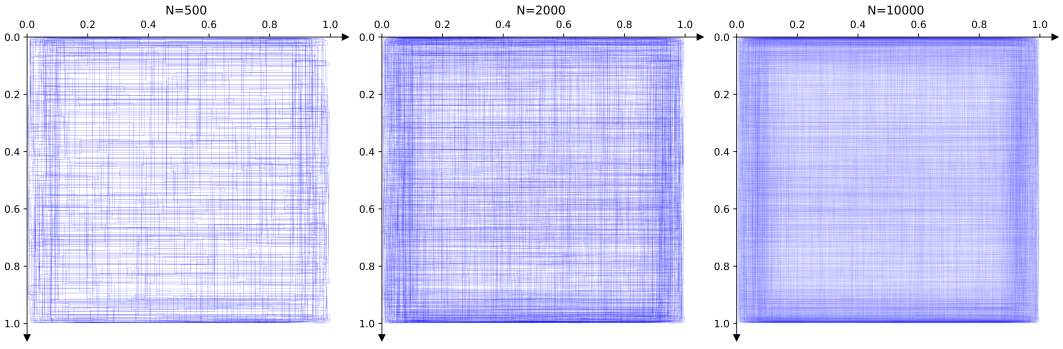
Next, we visualize and compare the document layout diversity of PubLayNet (Fig.S7a), DocBank (Fig.S7b), and MeshLayout-1M (Fig.S7c). Based on the visualization results, the document layouts in PubLayNet and DocBank clearly exhibit the two-column format and Manhattan layout typical of academic papers, lacking diversity. In contrast, the document layouts generated by Mesh-candidate BestFit demonstrate significant variation and diversity.



(a) Document layout distribution of PubLayNet.



(b) Document layout distribution of DocBank.



(c) Document layout distribution of MeshLayout-1M.

Figure S7: Document layout distribution of (a) PubLayNet, (b) DocBank and (c) MeshLayout-1M. N indicates number of documents used in visualization. Compared with two-column academic style layout in PubLayNet or DocBank, document layout synthesized by Mesh-candidate BestFit in MeshLayout-1M demonstrate significant variation and diversity.

F.3 Element Scale and Shape Distributions

We characterize the geometry of all 14,549,846 annotations in the complete MeshLayout-1M training split using normalized axis-aligned bounding boxes. Fig. S8 reports the class-wise marginal distributions of relative area and width-to-height ratio. These statistics describe the geometric coverage of the pretraining corpus without assuming that any single class follows a unimodal or canonical template.

Scale coverage. Median element area ranges from 0.62% of the page for Paragraph to 27.15% for Author, spanning more than a factor of 40 across classes. Paragraph and Title have broad 5th–95th percentile intervals (0.022%–27.29% and 1.73%–29.59%, respectively), while Caption, Table, and Abstract occupy more concentrated scale bands. The multimodal distributions of frequent classes such as Paragraph and Reference further show that the corpus contains several recurring size regimes rather than a single element scale.

Class-dependent shapes. The shape distributions differ substantially across semantic classes. Caption, Table, and Abstract are predominantly horizontal, with median width-to-height ratios of 11.50, 15.26, and 16.08, respectively. In contrast, Figure and Author are approximately square (median ratios 1.17 and 1.02), with 58.1% and 67.2% of their instances lying in the near-square range $[2/3, 1.5]$. We report these class-specific profiles as properties of the generated corpus; synthetic-to-real correspondence is not inferred from the marginal distributions alone.

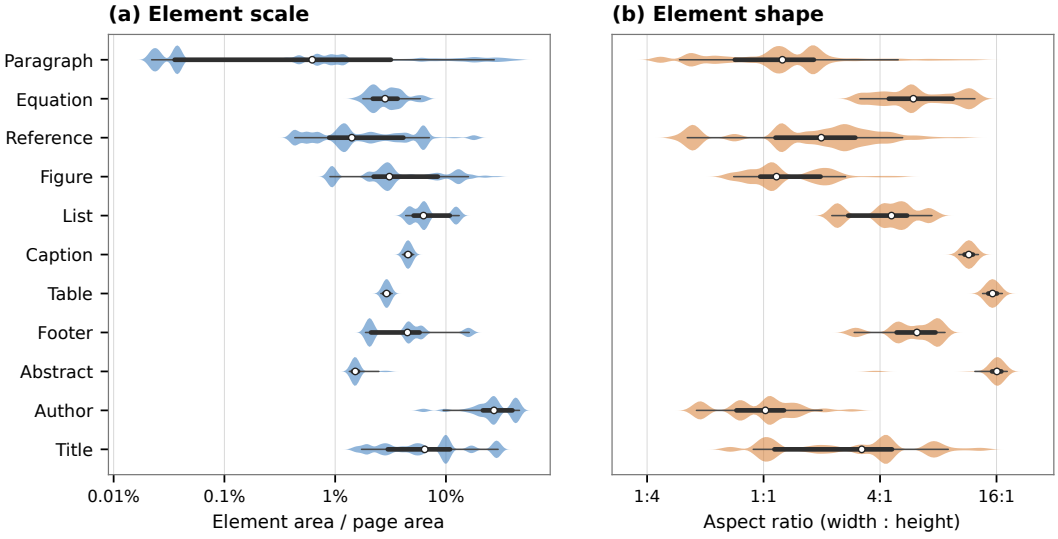


Figure S8: Class-wise element geometry in the complete MeshLayout-1M training split. (a) Bounding-box area relative to page area. (b) Width-to-height ratio.

G LAYOUT ANALYSIS EXAMPLES

Here, qualitative layout analysis examples on D^{4}LA (Fig. S9), M^{6}Doc (Fig. S10), HJ-Dataset (Fig. S11), PRIMA-LAD (Fig. S12), PubLayNet (Fig. S13), DocLayNet (Fig. S14), IndicDLP (Fig. S15) are illustrated.

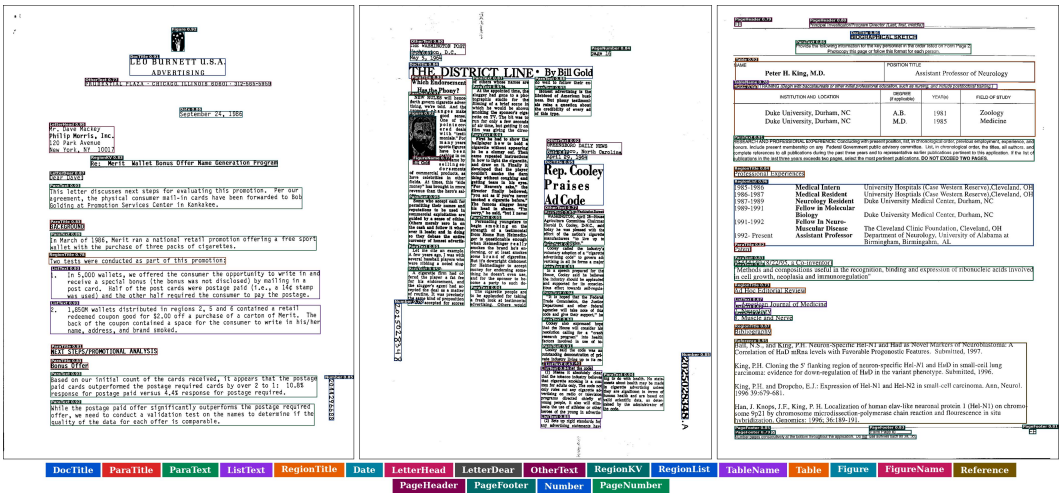


Figure S9: Layout analysis examples on D^{4}LA.



Figure S10: Layout analysis examples on M^{6}Doc.

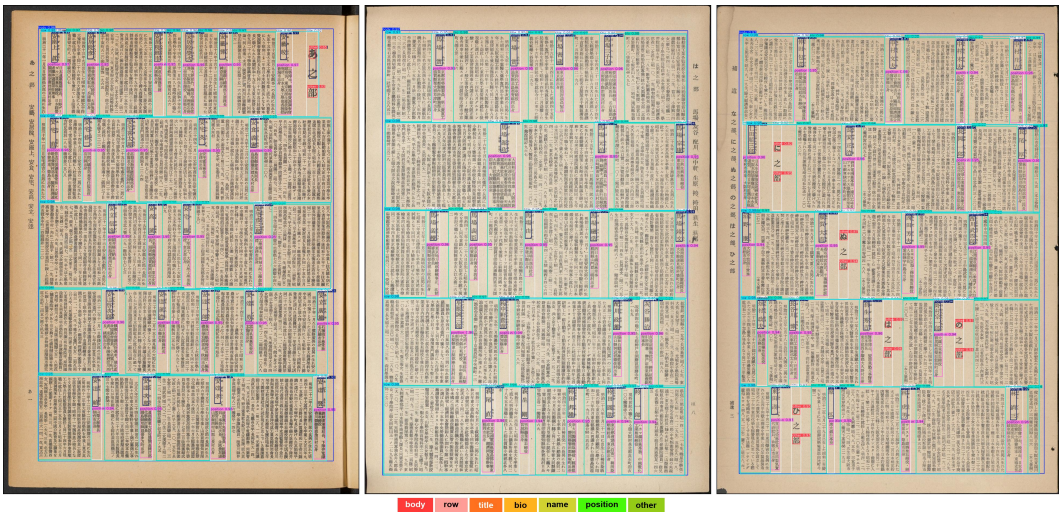


Figure S11: Layout analysis examples on HJDataset.



Figure S12: Layout analysis examples on PRIMA-LAD.

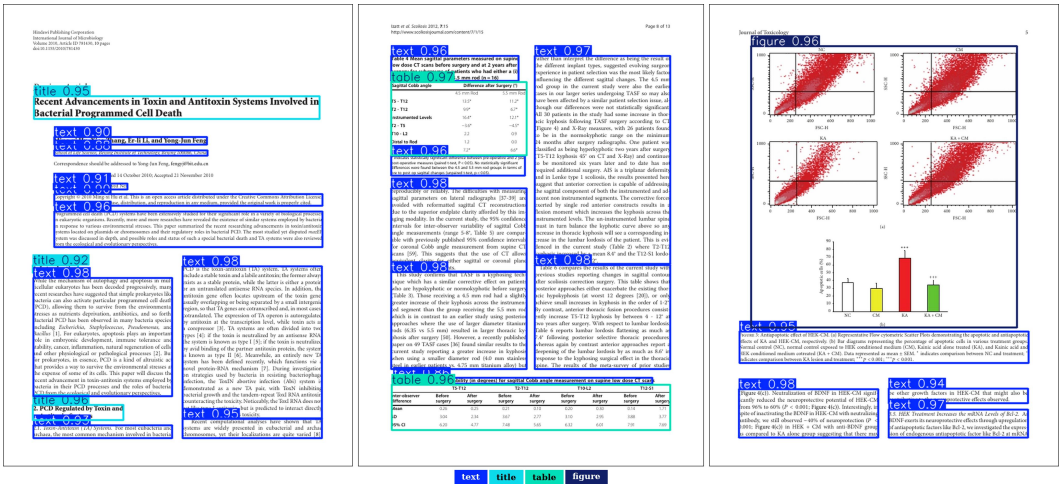


Figure S13: Layout analysis examples on PubLayNet.

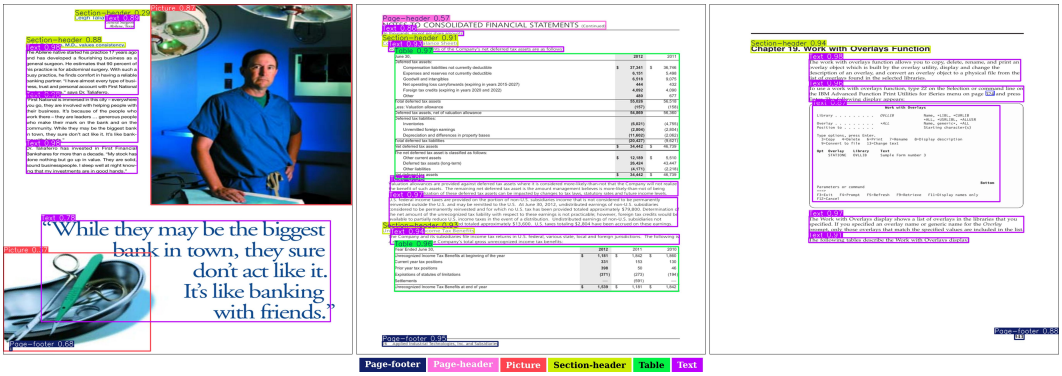


Figure S14: Layout analysis examples on DocLayNet.

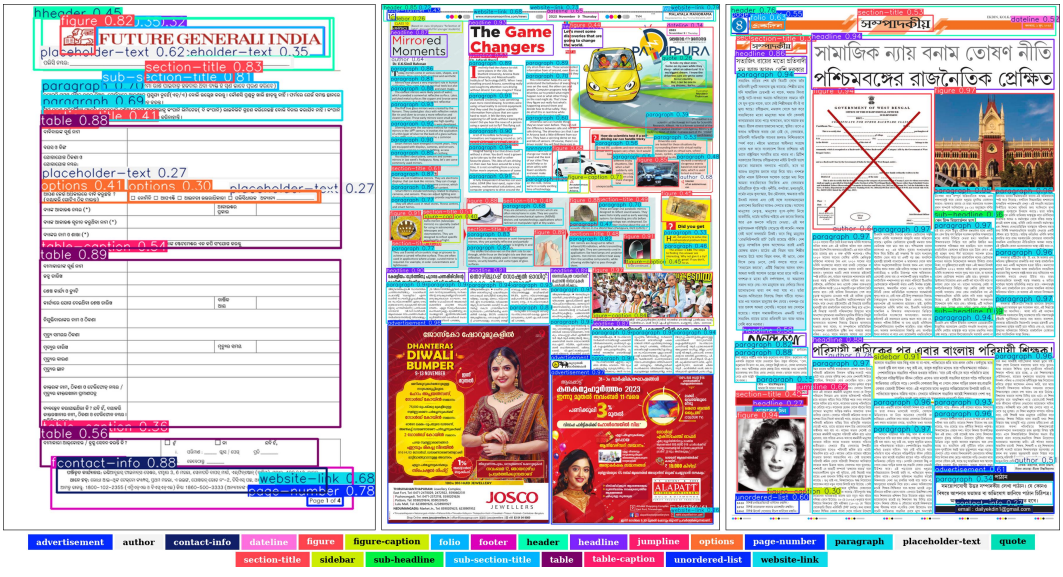


Figure S15: Layout analysis examples on IndicDLP.

H LIMITATIONS & FUTURE WORK

Efficiency across deployment environments. PALA achieves favorable practical efficiency with optimized attention implementations. Its actual speedup may vary across hardware platforms and inference backends due to differences in kernel optimization and memory access. Further operator-level optimization could improve its portability and efficiency across broader deployment environments.

Modeling scope of PALA. PALA is designed around the dominant horizontal and vertical organization of document layouts, enabling efficient long-range context modeling along both page axes. While this design covers common document structures well, highly irregular layouts with strong rotation, perspective distortion, or non-axis-aligned arrangements may benefit from more flexible interaction patterns. Extending PALA with adaptive directional modeling is a promising direction for such scenarios.

Synthesis scope of MeshLayout. MeshLayout focuses on increasing structural diversity through controllable geometric layout synthesis, with its current element pool primarily constructed from DocBank. Expanding the element sources could further increase diversity in content, language, and visual appearance. In addition, incorporating richer semantic relationships and realistic document variations could further narrow the gap between synthetic and real-world documents.

Future Work. Future work will explore more portable implementations of PALA and adaptive context modeling for irregular document structures. For MeshLayout, we plan to expand multilingual and multi-source element pools and incorporate richer semantic and visual variations, further improving the diversity and realism of synthetic document pretraining.