

PromptDLA: A Domain-aware Prompt Document Layout Analysis Framework with Descriptive Knowledge as a Cue

Zirui Zhang*, Yaping Zhang*, Lu Xiang, Yang Zhao, Feifei Zhai, Yu Zhou and Chengqing Zong, *Fellow, IEEE*

Abstract—Document Layout Analysis (DLA) is crucial for document artificial intelligence and has recently received increasing attention, resulting in an influx of large-scale public DLA datasets. Existing work often combines data from various domains in recent public DLA datasets to improve the generalization of DLA. However, directly merging these datasets for training often results in suboptimal model performance, as it overlooks the different layout structures inherent to various domains. These variations include different labeling styles, document types, and languages. This paper introduces PromptDLA, a domain-aware Prompter for Document Layout Analysis that effectively leverages descriptive knowledge as cues to integrate domain priors into DLA. The innovative PromptDLA features a unique domain-aware prompter that customizes prompts based on the specific attributes of the data domain. These prompts then serve as cues that direct the DLA toward critical features and structures within the data, enhancing the model's ability to generalize across varied domains. Extensive experiments show that our proposal achieves state-of-the-art performance among DocLayNet, PubLayNet, M6Doc, and D⁴LA. Our code is available on GitHub.¹

Index Terms—Document Layout Analysis, PromptDLA, Document Understanding

I. INTRODUCTION

Document Layout Analysis (DLA) aims to distinguish the physical or logical layout structure of documents [1]–[6], identifying areas characterized by elements such as text, image, and table. It is fundamental to modern document artificial intelligence, significantly influencing subsequent document understanding tasks such as information extraction and digital transformation [7]–[13].

With growing research interest in DLA, large-scale datasets have emerged, such as PubLayNet [14], DocBank [15], DocLayNet [16], M6Doc [17] and D⁴LA [18]. To enhance generalizability in real-world scenarios, recent DLA datasets such as DocLayNet [16], M6Doc [17], and D⁴LA [18] have increased document diversity by combining data from various domains,

Y. Zhang, L. Xiang, Y. Zhao, F. Zhai, Y. Zhou and C. Zong are with the Institute of Automation, Chinese Academy of Sciences, and the University of the Chinese Academy of Sciences. Y. Zhou is also with the Fanyu AI Laboratory, Zhongke Fanyu Technology Co., Ltd. Z. Zhang is with Columbia University; this work was done while at the Fanyu AI Laboratory. Email: zz3093@columbia.edu, {yaping.zhang, lu.xiang, yang.zhao, yzhou, cqzong}@nlpr.ia.ac.cn. Y. Zhang is the corresponding author.

Y. Zhang and Z. Zhang contributed equally and are co-first authors.

This work is supported by the National Natural Science Foundation of China under Grant 62476275 and 62106265, and by the Young Scientists Fund of the State Key Laboratory of Multimodal Artificial Intelligence Systems.

¹<https://github.com/Zirui00/PromptDLA.git>

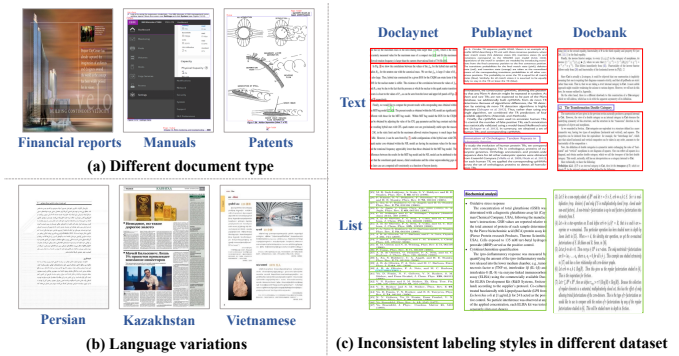


Fig. 1. Examples of different domain differences across (a) Different document types caused variations in layout structure and element distribution (financial report, manual, patent) (b) different language types, and (c) Inconsistent labeling styles. Note that the “text” and “list” items in DocLayNet are labeled as smaller individual units while they are integrated as a whole in DocBank.

including finance, law, and patents. However, merging data from these diverse domains introduces substantial distribution differences, both across and within datasets. Fig. 1 illustrates three critical yet overlooked domain differences encountered:

- **Different document types.** Document images from different types exhibit unique layout structures and label distributions. Fig. 1(a) visually contrasts the typical visual features of three document types: Financial reports (left) are characterized by well-designed color images and text-image overlays; manuals (center) frequently use functional screenshots of software interfaces to present data and controls; while patents (right) are dominated by structured, black-and-white technical line drawings with concise labels. These significant differences in layout highlight the need to incorporate domain priors into DLA.
- **Different languages.** Document images from various countries exhibit unique layout structures influenced by their respective languages. Fig. 1(b) demonstrates how language affects the layout of the document. Persian documents predominantly feature dense blocks of text arranged in continuous paragraphs. In contrast, documents from Kazakhstan integrate numerous small paragraphs interspersed with images, creating a visually diverse page layout. These variations highlight the correlation between language and document layout, as shown in Fig. 1.
- **Inconsistent labeling styles.** Different datasets often adopt disparate annotation guidelines leading to conflict

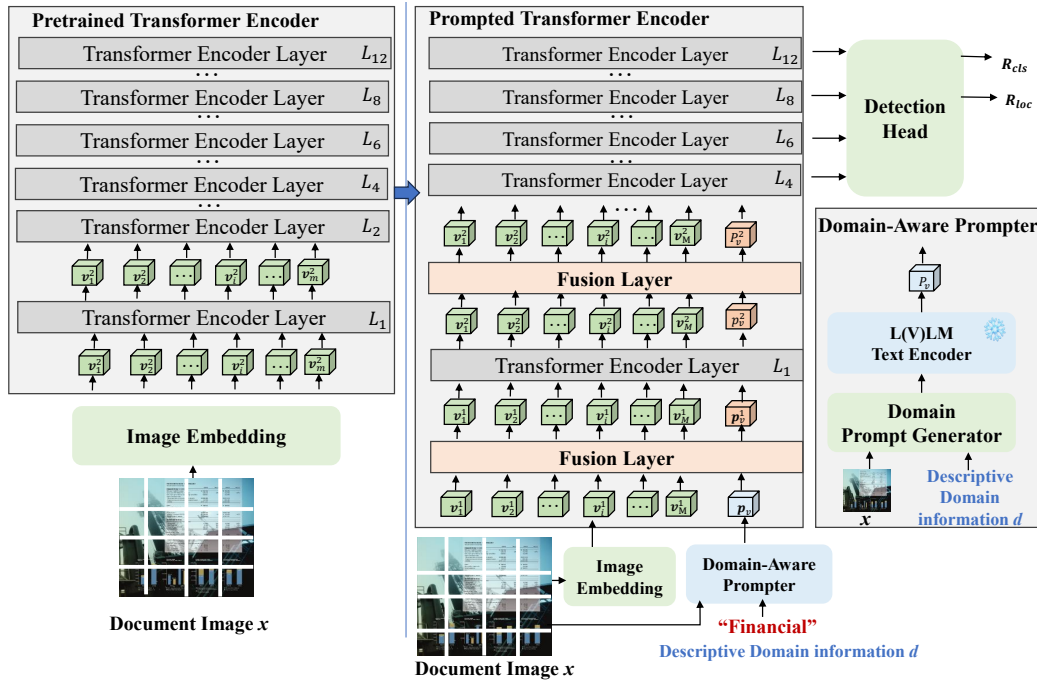


Fig. 2. Overview of the PromptDLA method for domain-aware layout prediction. A Domain-Aware Prompter encodes domain information into a prompt vector, which is prepended to the sequence of image patch embeddings. This augmented input is processed by a vision backbone. Multi-scale features are extracted from the backbone and refined by a FPN before being passed to a detection head for final layout prediction.

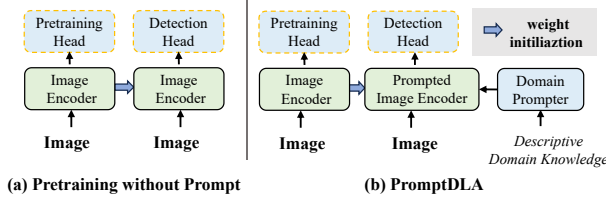


Fig. 3. Comparison with Pre-training Paradigms in DLA

labeling style, even for semantically similar elements. As shown in Fig. 1(c), DocLayNet annotates individual list items, whereas DocBank and PubLayNet group entire lists into single bounding boxes. Similarly, paragraph segmentation varies significantly across datasets. Such inconsistencies create conflicts during joint training and pose obstacles to building scalable, unified models.

These overlooked domain-related discrepancies can hinder the learning process and reduce the generalization capability of DLA models trained on combined datasets. A promising direction to mitigate these issues involves endowing DLA models with the capacity to adapt their analysis based on the specific characteristics of the input document's domain. Recent studies in large vision-language models (LVLMs) like CLIP [19] and large language models (LLMs) like LLaMA [20] have demonstrated the efficacy of prompt engineering for conditioning model behavior on domain-specific contexts across various tasks [21]–[26]. Inspired by this, we propose a novel framework for Domain-aware **Prompt** Document Layout Analysis, named **PromptDLA**. Unlike traditional pretraining-based DLA methods, PromptDLA integrates domain priors di-

rectly into the analysis process using LLMs like LLaMA [20] or LVLMs such as CLIP [19], BLIP2 [27]. Central to our approach is a prompted transformer encoder, fine-tuned with a novel domain-aware prompter (depicted in Fig. 2). This prompter uses descriptive knowledge from the domain information of corresponding images as cues, guiding the transformer encoder to recognize and adapt to the variability across different domains effectively. We evaluate our method on the DocLayNet, M6Doc, D⁴LA, and the integration of PubLayNet and DocLayNet datasets. The experimental results have shown that the domain-aware model has outperformed the current state-of-the-art method. In addition, due to the predominance of English in existing datasets, we have introduced a multilingual DLA dataset—MLDLA, which contains document images in seven different languages. Detailed experiments have demonstrated that our method can effectively generalize across scenarios where language serves as domain information. The contributions of this paper are summarized as follows:

- A novel domain-aware DLA framework, named PromptDLA, is proposed that explicitly introduces domain knowledge to DLA, enabling models to better handle variability across a variety of document domains.
- A unique and modular domain-aware prompter is proposed to generate customized prompts reflecting specific data attributes. It integrates easily with various backbone architectures, such as CNNs, ViTs, and Swin Transformers, using prompts derived from either human knowledge or LLMs generation.
- We conduct extensive experiments demonstrating the effectiveness of PromptDLA across multiple domain information types and datasets, including the newly introduced

TABLE I
THE DETAIL OF DOCUMENT TYPES IN DIFFERENT DLA DATASETS.

Dataset	Document Type	A.M.	Format	#Class	Language	#Images
PubLayNet [14]	Articles	Automatic	PDF	5	English	364232
DocLayNet [16]	Financial Reports, Manuals, Scientific Articles, Laws & Regulations, Patents, Government Tenders.	Manual	PDF	11	English, German, French, Japanese	80863
M6Doc [17]	Scientific articles, Textbooks, Books, Test papers, Magazines, Newspapers, Notes	Manual	PDF, Scanned, Photographed	74	English, Chinese	9080
D ⁴ LA [18]	Scientific report, Email, Form, Invoice, Letter, Specification, News article, Presentation, Resume, Scientific publication, Budget, Memo	Manual	PDF, Scanned,	27	English	11092
MLDLA(ours)	Magazine, Newspaper, Government Reports	Manual	PDF, Scanned, Photographed	5	Hindi, Kazakhstan, Vietnam, Turkey, Persia, Laos, Khmer	17505

MLDLA benchmark, achieving state-of-the-art results.

II. RELATED WORK

A. Document Layout Analysis

The current literature on document analysis has redefined document understanding as a broad term that covers various problems and tasks related to document intelligence systems. Based on whether pre-training on large-scale unlabeled document images can divide the Document Layout Analysis method into two categories: traditional object detection frameworks and document pre-training methods. Traditional object detection frameworks, such as Faster-RCNN [28], Mask-RCNN [29], and YOLO [30], typically train models directly on DLA datasets and may occasionally use ImageNet [31] pre-trained weights. In comparison, another approach involves training the transformer using the self-supervised method on a vast unlabeled document dataset and utilizing the pretrained transformer as the backbone of a two-stage Object Detection Framework. LayoutLMv3 [32] and Structxtv2 [33] are critical works in this area. It employs the multi-modal pre-training method, including Mask Image Modeling, Mask Language Modeling, and Word-Patch Alignment, using both the textual information on the image and the image itself as input to the transformer encoder. Other notable works in this area are DiT [34], DocFormer [35], UniDoc [36], and Self-Docseg [37], which relies solely on a vision model, closely aligning with the approach of BEiT [2], and directly applies a general pre-training framework to learn from large-scale document image data. Traditional approaches focus on enhancing the model's performance. As depicted in Fig. 3, PromptDLA diverges from conventional pre-training methods by directly injecting domain knowledge into the DLA framework. Instead of implicitly learning domain features, our approach leverages descriptive prompts from LLMs, offering a more flexible and resource-efficient paradigm for domain adaptation.

B. Prompt Engineering

Prompt engineering has significantly improved the adaptability of LLMs and LVLMs for specialized tasks. By introducing learnable tokens into vision transformers (ViTs), models can effectively focus on task-relevant features with minimal structural changes [38]. This strategy is similar to

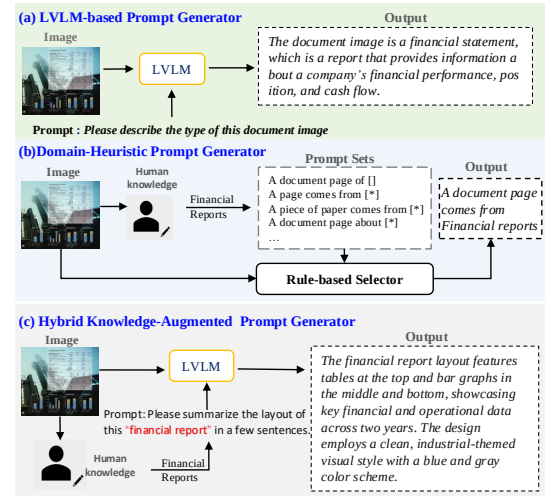


Fig. 4. Framework of Prompt Generator.

advances in multimodal learning, exemplified by CLIP, which jointly leverages text and image data to enhance robustness and versatility across various visual recognition tasks, particularly in zero-shot or few-shot settings [19]. Furthermore, prompt engineering more generally utilizes carefully designed prompts to effectively elicit contextually appropriate outputs from pre-trained models, greatly reducing the need for extensive retraining [39].

III. METHODOLOGY

Document layout analysis significantly depends on the data domain. A model that accurately recognizes domain-specific characteristics can adapt its outputs based on tailored prompts, efficiently handling variations across diverse document domains.

To facilitate this, we introduce a domain-aware prompter that enables the model to identify the domain of the input data. As illustrated in Fig. 2, our model is composed of four main components: an Image Embedding Module $\mathcal{F}_{patch}$, a Domain-aware Prompter $\mathcal{F}_{prompter}$, a Prompted Transformer Encoder $\mathcal{F}_{encoder}$, and a Detection Head $\mathcal{F}_{detect}$.

Given an input document image $x \in \mathbb{R}^{C \times H_{in} \times W_{in}}$, the Image Embedding Module extracts patch embeddings as visual

tokens:

$$v_1, v_2, \dots, v_M = \mathcal{F}_{patch}(x) \quad (1)$$

where $\{v_i\}_{i=1}^M$ represents the sequence of visual tokens. Currently, the Domain-Aware Prompter generates a domain-specific prompt embedding p_v :

$$p_v = \mathcal{F}_{prompter}(x, d) \quad (2)$$

where d represents descriptive knowledge as explicit domain information, provided by either a human expert or an LLM, as detailed in Fig. 4. Both the visual tokens and the prompt embedding are then processed by the Prompted Transformer Encoder:

$$f_1, f_2, \dots, f_L = \mathcal{F}_{encoder}(v_1, v_2, \dots, v_M, p_v) \quad (3)$$

yielding multilevel feature maps $\{f_l\}_{l=1}^L$. Finally, a subset of these feature maps is passed to the Detection Head. For instance, when using a transformer-based backbone, features from the 4th, 6th, 8th, and 12th layers are selected to predict the layout structure $\hat{y}$:

$$\hat{y} = \mathcal{F}_{detect}(f_4, f_6, f_8, f_{12}) \quad (4)$$

where $\hat{y}$ denotes the predicted refined bounding box $\hat{b}$ and a class label $\hat{l}$.

A. Image Embedding

We follow the patch embedding approach used in ViT [40] for image embedding. Specifically, the document image x is resized to $H \times W$, represented as $x \in \mathbb{R}^{C \times H \times W}$, where C , H , and W denote the channel, height, and width, respectively. The resized image is then divided into a sequence of non-overlapping patches of size $P \times P$. Each patch is linearly projected into a D -dimensional vector via $\mathcal{F}_{patch}$, as defined in Eq. 1, resulting in a flattened sequence:

$$v = [v_1, v_2, \dots, v_i, \dots, v_M], \quad \text{where } v_i \in \mathbb{R}^D. \quad (5)$$

Here, the sequence length M is determined as $M = HW/P^2$. Finally, learnable 1D positional embeddings are added to each vector to encode spatial information.

B. Domain-Aware Prompter

As shown in Fig. 2, the Domain-aware Prompter $\mathcal{F}_{prompter}$ consists of two components: a Text Encoder $\mathcal{F}_t$ and a Prompt Generator $\mathcal{F}_g$.

1) *Prompt Generator $\mathcal{F}_g$* : The Prompt Generator $\mathcal{F}_g$ is designed to dynamically produce natural language prompts pertinent to the document's domain. As illustrated in Fig. 4, $\mathcal{F}_g$ supports three distinct operational modes, ranging from direct utilization of curated human knowledge to automated generation using large models.:

- **LVLM-based Prompt Generator**: This mode leverages the generative capabilities of LVLMs, such as variants based on LLaMA-Adapter [41]. The input consists of the document image x paired with a general instruction (e.g., "Please describe the type of this document" or "Describe the primary use of this document"). The LVLM analyzes the image content and generates a textual description

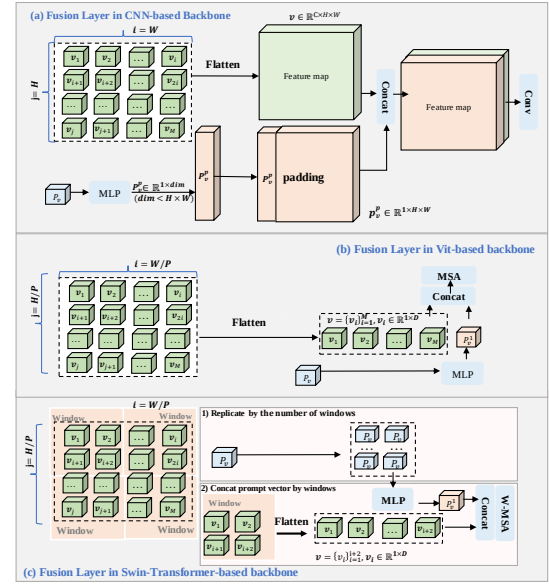


Fig. 5. Framework of Fusion Layer.

capturing the inferred document domain and characteristics. This approach offers high automation and zero-shot potential but incurs significant computational overhead.

- **Domain-Heuristic Prompt Generator**: This mode relies on curated human knowledge. It utilizes predefined 'Prompt Sets' containing various sentence templates designed to encapsulate domain information. Given an explicit domain class (e.g., 'invoice', 'scientific paper'), relevant templates are selected. We employ a rule-based selector, potentially augmented by CLIP's zero-shot classification capability, to refine template selection. For instance, multiple templates can be instantiated with specific document type names (e.g., from DocLayNet), embedded using CLIP's text encoder, and evaluated for their zero-shot classification accuracy on a relevant task. Templates yielding top-k performance are retained. This approach ensures that the generated prompts are interpretable and consistent with explicit rules and domain labels.
- **Hybrid Knowledge-Augmented Prompt Generator**: This hybrid approach combines aspects previous two. It uses an LVLM but guides its generation with more specific, human-provided knowledge compared to the general instructions. For example, instead of asking the LVLM to infer the type, a prompt like "Please describe the typical layout elements found in a financial report document" is provided, potentially along with the image x or just the domain label d . This allows the LVLM to generate more precise and contextually relevant descriptions tailored to a known document type, balancing automation with targeted knowledge injection.

2) *Text Encoder ($\mathcal{F}_t$)*: The Text Encoder $\mathcal{F}_t$ converts a natural language prompt describing the document domain into a fixed-dimensional embedding that provides domain-specific guidance to the layout analysis model. This Encoder is instantiated using a powerful pre-trained language model, either a

Algorithm 1: The PromptDLA Algorithm.

Require: $D(x^{(n)}, y^{(n)})_{n=1}^N$, y consists of target bounding boxes $\hat{b}$ and class labels l ;

Ensure: Prediction of layout

- 1: Initialize $\mathcal{F}_t \leftarrow \Theta_{llm}$ or Θ_{vlm} , $\mathcal{F}_e \leftarrow \Theta_{pretrain_e}$
- 2: Prompt Generator $\mathcal{F}_g$
- 3: **while** $t \leq \text{max iteration do}$
- 4: $(v_1^0, v_2^0, \dots, v_m^0) \leftarrow \mathcal{F}_{patch}(x)$
- 5: $prompt \leftarrow \mathcal{F}_g(x)$
- 6: $p_v \leftarrow \mathcal{F}_t(prompt)$
- 7: **for** $i=1; i \leq \text{layer nums do}$
- 8: $f_i \leftarrow (v_1^i, v_2^i, \dots, v_m^i)$
- 9: $in_i \leftarrow \mathcal{F}_{fuse}(f_i; \mathcal{F}_{mlp}(p_v))$
- 10: $(v_1^{i+1}, v_2^{i+1}, \dots, v_m^{i+1}) \leftarrow \mathcal{F}_e(in_i)$
- 11: **end for**
- 12: $f \leftarrow \mathcal{F}_{pn}(f_4, f_6, f_8, f_{12})$
- 13: $Loss \leftarrow R_{loc}(\mathcal{F}_{detect}(f, \hat{b}), \hat{b}) + \lambda R_{cls}(\mathcal{F}_{detect}(f), l)$
- 14: freeze $\mathcal{F}_t$
- 15: update $\mathcal{F}_{fuse}, \mathcal{F}_e, \mathcal{F}_{detect}$
- 16: $t \leftarrow t + 1$
- 17: **end while**

text encoder from a vision-language model (θ_{vlm} , e.g., CLIP, BLIP2) or a large language model (θ_{llm} , e.g., LLaMA [41]). To preserve the rich semantic knowledge acquired during pre-training, the $\mathcal{F}_t$ weights are frozen during training. The resulting embedding, denoted as $p_v \in \mathbb{R}^{D_{prompt}}$, serves as the domain-aware input for subsequent modules.

C. Prompted Transformer Encoder

The Prompted Transformer Encoder $\mathcal{F}_{encoder}$ integrates the visual tokens $\{v_i\}_{i=1}^M$ with the domain prompt embedding p_v . It typically comprises a standard Transformer Encoder backbone and dedicated Fusion Layers ($\mathcal{F}_{fuse}$) responsible for injecting the prompt information. As shown in Fig. 5, We explore compatibility with CNN, ViT and Swin Transformer [42] architectures, requiring slightly different fusion strategies:

- **CNN-based Encoder :** As illustrated in Fig. 5(a), for a CNN backbone (e.g., ResNet-50 [43]), the prompt embedding p_v is first projected to a target dimension (e.g., 512 or 768) using an MLP layer. This projected prompt vector is then spatially padded to match the height H and width W of the feature map $F^{(i)}$ at a specific layer or stage i . The resulting projection tensor p_v^p , now having dimensions $1 \times H \times W$, is concatenated channel-wise with the feature map $F^{(i)} \in \mathbb{R}^{C \times H \times W}$. In our implementation using ResNet-50 [43], this fusion operation is performed at the input of each of the four main residual stages, where the spatially expanded prompt features are concatenated with the stage's input feature map.
- **ViT-based Encoder:** As depicted in Fig. 5(b), for a ViT backbone, the fusion layer $\mathcal{F}_{fuse}$ typically employs an MLP ($\mathcal{F}_{mlp}$) to project the prompt embedding p_v to match the visual token dimension D . The projected prompt embeddings, denoted p_v^1 , are then concatenated with the sequence of position-aware visual tokens $\{\tilde{v}_i\}_{i=1}^M$ (where $\tilde{v}_i = v_i + e_i$, incorporating patch embedding v_i and positional embedding e_i). This combined sequence is the input to the standard Transformer Encoder layers ($\mathcal{F}_e$).
- **Swin Transformer-based Encoder:** For the hierarchical Swin Transformer backbone, illustrated in Fig. 5(c),

prompt fusion requires adaptation to its windowed attention and shifting window mechanisms [42]. Inspired by approaches like VPT [44], the prompt embedding p_v is processed by stage-specific MLPs within the fusion module $\mathcal{F}_{fuse}$ to generate dimension-matched embeddings $p_v^{(1)}$ for each stage. Within a stage, p_v is typically replicated N_{win} times (where N_{win} is the number of windows in that stage) and combined with the token sequence of each respective window before the windowed self-attention (W-MSA) computation. Appropriate masking or padding is applied to ensure dimensional consistency during attention calculations. The outputs from multiple stages (e.g., corresponding to features at 1/8, 1/16, 1/32 resolutions) are then typically fed into an FPN ($\mathcal{F}_{fpn}$) to generate multi-scale features for the detection head.

The output of $\mathcal{F}_{encoder}$ (potentially via $\mathcal{F}_{fpn}$) is a set of feature maps $f_1, f_2, \dots, f_L$ capturing both visual content and domain-specific context.

D. Detection Head

The Detection Head $\mathcal{F}_{detect}$ takes the contextualized features from the encoder and performs the final layout element prediction, including bounding boxes and class labels. Our framework is designed to be compatible with two detection head architectures:

- **RCNN-based Heads:** Standard two-stage detection frameworks like Faster R-CNN [28], Mask R-CNN [29], or Cascade R-CNN [1] can be readily employed. In this setup, the R-CNN detection head operates on the generated feature maps by the prompted transformer encoder, optimizing bounding box regression and classification objectives using standard loss functions. Specifically, the bounding box regression loss R_{loc} aims to minimize the discrepancy between predicted boxes $b_i = (b_{ix}, b_{iy}, b_{iw}, b_{ih})$ derived from proposals $\hat{b}_i$ and features f_i , and ground-truth boxes $\hat{b}_i$:

$$R_{loc} = \sum_{i=1}^N L_{loc}(r(f_i, b_i), \hat{b}_i), \quad (6)$$

where $r(\cdot)$ is the regression function, N is the number of proposals, and L_{loc} is typically the smooth L_1 loss. For Classification, a classifier $c(\cdot)$ assigns feature map patches to one of the classes, the classification loss R_{cls} is defined as:

$$R_{cls} = \sum_{i=1}^N L_{cls}(c(f_i), l_i), \quad (7)$$

where f_i and l_i denote the i -th object feature and class label, respectively. And L_{cls} is usually the cross-entropy loss. The total loss is a weighted sum:

$$R_{total} = R_{loc} + \lambda R_{cls}. \quad (8)$$

- **DETR-based Heads:** Our architecture also supports integration with DETR [45] and its variants. DETR employs an encoder-decoder structure where a set of learnable object queries interacts with the image features (from

TABLE II

COMPARISON WITH STATE-OF-THE-ART METHODS ON DocLAYNET. OUR METHOD USES DOMAIN PROMPTS FROM HUMAN KNOWLEDGE AND PROMPT ENGINEERING, WITH A ViT BACKBONE, CLIP TEXT ENCODER, AND CASCADE MASK R-CNN DETECTION HEAD. ALL BASELINE RESULTS ARE TAKEN DIRECTLY FROM PREVIOUS PAPERS.

Method	Pretraining	Caption	Footnote	Formula	List-item	Page-footer	Page-header	Picture	Section-header	Text	Table	Title	mAP
Mask R-CNN [17]	✗	71.5	71.8	63.4	80.8	59.3	70.0	72.7	69.3	82.9	85.8	80.4	73.5
Faster R-CNN [17]	✗	70.1	73.7	63.5	81.0	58.9	72.0	72.0	68.4	82.2	85.4	79.9	73.4
YOLOv5 [17]	✗	77.7	77.2	66.2	86.2	61.1	67.9	77.1	74.6	86.3	88.1	82.7	76.8
TransDLNet [17]	✗	68.2	74.7	61.6	81.0	54.8	68.2	68.5	69.8	82.4	83.8	81.7	72.3
SwinDocSegmenter [5]	✗	83.6	64.8	62.3	82.3	65.1	66.4	84.7	66.5	87.4	88.2	63.3	76.9
SelfDocSeg [5]	✓	-	-	-	-	-	-	-	-	-	-	-	74.3
LayoutLmV3 [32]	✓	73.1	77.5	69.0	79.8	61.3	61.3	74.0	69.0	86.3	85.9	84.4	75.7
DiT [34]	✓	75.0	76.2	68.1	83.5	62.1	74.0	74.5	71.2	86.4	86.6	83.0	76.4
PromptDLA(ViT, CLIP, Cascade)	✓	76.6	83.0	72.4	84.9	63.8	76.9	75.2	73.8	87.1	87.9	84.1	78.7
PromptDLA(ResNet, CLIP, DETR)	✓	91.8	84.1	54.7	69.4	82.9	39.5	82.0	87.3	70.3	83.0	91.3	77.7
PromptDLA(SwinTran, CLIP, DETR)	✓	92.5	85.5	57.6	71.3	84.2	40.2	83.1	88.5	73.1	83.8	92.1	79.6

$\mathcal{F}_{encoder}$, i.e., $\{\mathbf{f}_l\}_{l=1}^L$) via cross-attention mechanisms within the decoder. Feed-forward networks (FFNs) then directly predict the class and bounding box coordinates from the updated object queries. DETR is optimized end-to-end using a set-based bipartite matching loss that jointly considers classification and localization costs (e.g., cross-entropy for class, and a combination of L_1 and GloU loss for boxes).

This flexibility allows leveraging the strengths of different detection paradigms within our PromptDLA framework.

IV. EXPERIMENTS

This section validates PromptDLA's effectiveness through three sets of experiments. First, we examine the efficacy of PromptDLA compared to the state-of-art methods. Next, we evaluate our approach's generalization ability. Lastly, we perform extensive ablation studies and discussions on the model's design.

A. Experimental Settings

Datasets. We conduct extensive experiments to validate the proposed PromptDLA on 5 DLA benchmark datasets, including different document types, different languages, and different layer styles,

- **PubLayNet** [14] consists of 5 typical document layout elements: text, heading, list, graphic, and table. It contains over 364232 page samples, where the annotations were automatically generated by matching PDFs and XML formats of articles from the PubMed Central Open Access subset.
- **DocLayNet** [16] contains 6 document types (Financial Reports, Manuals, Scientific Articles, Laws Regulations, Patents, and Government Tenders) with 11 categories of annotations across four languages (with English documents comprising approximately 95%). The dataset contains about 80,863 manually annotated pages.
- **M6Doc** [17] contains a total of 9,080 modern document images, which are categorized into 7 document types (Scientific articles, Textbooks, Books, Test papers, Magazines, Newspapers, Notes) with 74 detailed categories.
- **D⁴LA** [18] contains a total of 11092 document images, which includes 12 diverse document types (Scientific report, Email, Form, Invoice, Letter, Specification, News

article, Presentation, Resume, Scientific publication, Budget, Memo) with 27 detailed categories.

- **MLDLA** is a Multi-Language DLA (MLDLA) dataset we constructed to evaluate the model generalization on more different languages. It comprises 175,000 images, which are manually labeled through a uniform labeling style, including 7 in different languages, such as Persian, Khmer, Kazakh, Lao, Turkish, Hindi, and Vietnamese.

Evaluation Metric. Our experiments are evaluated using the category-wise and overall mean average precision (mAP) @IOU[0.50:0.95] of bounding boxes following the literature [1]. This curve describes the relationship between precision and recall and is the most widely used evaluation metric for document layout analysis.

Implementation Details. We train our model on 8 3090 GPUs with a batch size 16 using a cosine learning rate schedule and a warm-up strategy with a 0.01 warm-up factor. We set the basic learning rate to $2e-4$. Additionally, we adopt the AdamW optimizer. Our study uses DiT [34] pre-trained weights and the Cascade-RCNN [1] detection head as our baseline method. The detailed training process is shown in Algorithm 1.

B. Generalization Ability

Generalization on Different Document Domains. We evaluate the performance of PromptDLA on different DLA datasets with more diverse document domains in Table VIII. The results show that the promptDLA can get consistent improvements on datasets with other domains, such as DocLayNet with 6 domains (2.3% over DiT), M6Doc with 7 domains(2.0% over DiT), and D4LA (1.4% over DiT), validating the generalization of PromptDLA on different document types.

Generalization on Multi-Language Datasets. We investigate the effects of domain-aware prompts by using different language types as domain information based on the MLDLA dataset. Firstly, we validate that CLIP can provide prior knowledge even for documents in different languages, including minority languages, through a zero-shot document classification task. More specifically, we insert the document's language into a prompt template as text and utilize CLIP to identify similarities between it and the associated image. Our experiments indicate that CLIP achieves a zero-shot classification accuracy of **47.53%** across seven languages in MLDLA. Furthermore, we apply PromptDLA to MLDLA

and present a comparison in Table III. Following the method described in Section 3.2 of the paper to construct prompt sets. Our approach improves precision by +1.0 compared to the DiT model without a prompt. DiT without a prompt achieves higher mAPs in categories like "List," which are less domain-relevant. We think that it may lack sufficient domain-specific information for accurate detection. On the contrary, our PromptDLA significantly improves other domain-specific details, such as "Fig." (from 54.5% to 57.3%) and "Table" (from 76.1% to 77.9%).

TABLE III
EXPERIMENTS ON A REAL MULTI-LANGUAGE DATASET.

Method	Text	Title	Figure	List	Table	mAP
DiT	77.9	58.6	54.5	75.6	76.1	68.5
PromptDLA	78.7	59.1	57.3	74.5	77.9	69.5
Δ	+0.8	+0.5	+2.8	-1.1	+1.8	+1.0

Generalization on Inconsistent Labeling Style. We investigate the impact of inconsistent labeling styles. We perform experiments to jointly train different datasets to enhance the model's performance in practical applications. However, we encounter challenges due to the inconsistent labeling styles observed in public Document Layout Analysis (DLA) datasets, particularly between DocLayNet and PubLayNet. As depicted in Fig. 6, the labeling styles of PubLayNet and DocLayNet exhibit notable differences. While PubLayNet's image and table align with DocLayNet's counterparts, PubLayNet's text corresponds to the set of DocLayNet's caption, footnote, and text. Similarly, PubLayNet's title matches the set of DocLayNet's title and section header. Notably, PubLayNet's list and DocLayNet's list items differ, with PubLayNet integrating multiple list items as a whole list, while DocLayNet labels each list item separately. Additionally, PubLayNet omits page-footer, page header, and formula, whereas DocLayNet includes these elements. To address these differences, we perform label mapping, which aligns DocLayNet labels with PubLayNet and retains the page footer, page header, and formula. As shown in Table IV, joint training of the datasets, even after label mapping, does not improve performance on DocLayNet. Instead, there is a decrease in performance attributed to annotation conflicts. To overcome this issue, we introduce domain prompts, observing a consistency improvement on both DocLayNet and PubLayNet. This confirms the model's adaptive learning capability to handle conflicts and effectively learn models tailored to the target domain. Notably, our method enhances mAP from 76.0 to 77.1 for DocLayNet and from 94.8 to 94.9 for PubLayNet.

TABLE IV
EXPERIMENTS ON INCONSISTENT LABELING STYLES.

	DocLayNet			PubLayNet		
	Baselines	Joint	PromptDLA	Baselines	Joint	PromptDLA
Text	86.9	87.0(+0.1)	87.5(+0.6)	94.4	94.5(+0.1)	94.5(+0.1)
Title	71.2	72.1(+0.9)	73.7(+2.5)	88.9	89.4(+0.5)	89.5(+0.6)
List	84.0	83.5(-0.5)	84.4(+0.4)	94.8	95.5(+0.7)	95.6(+0.8)
Table	87.1	87.2(+0.1)	87.7(+0.6)	97.6	97.8(+0.2)	97.8(+0.2)
Figure	77.2	75.7(-1.5)	76.3(-0.9)	96.9	96.9(+0.0)	97.0(+0.1)
Formula	68.4	69.1(+0.7)	68.9(+0.5)	-	-	-
Page-footer	64.0	61.7(-2.3)	64.1(+0.1)	-	-	-
Page-header	74.0	72.1(-1.9)	74.3(+0.3)	-	-	-
mAP	76.6	76.0(-0.6)	77.1(+0.5)	94.5	94.8(+0.3)	94.9(+0.4)

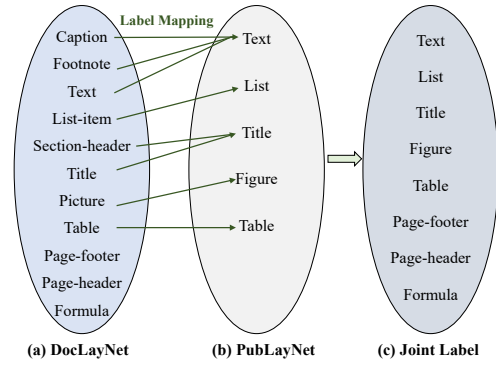


Fig. 6. Inconsistent labeling relationship between DocLayNet and PubLayNet.

Generalization on Out of Distribution Prompt. We explored the performance of PromptDLA in out-of-distribution (OOD) scenarios by splitting the DocLayNet dataset by document category, using 'manuals' as the test set, and training on the remaining categories of documents. As shown in Table V, the prompt works effectively in OOD situations.

TABLE V
OOD RESULT ON MANUALS FROM DOCLAYNET

Tag	Method	mAP	Δ
(a)	DiT	62.68	
(b)	DiT + Human knowledge	64.23	+1.55
(c)	DiT + LVLM	63.93	+1.25

Generalization Across Different Backbone Architectures. To evaluate the adaptability and robustness of our proposed PromptDLA framework, we integrated and tested it with diverse backbone architectures commonly used in vision tasks. Specifically, we assessed its performance with a standard Vision Transformer (ViT-Base), a hierarchical Vision Transformer (Swin-Transformer Base), and a widely adopted Convolutional Neural Network (ResNet-50). For the Transformer-based models (ViT and Swin-Transformer), the domain-aware prompts were incorporated as detailed in *Figure*. For the CNN-based ResNet-50, the prompt embeddings were integrated by concatenating them with the pooled features before the final classifier. The performance, measured by mean Average Precision (mAP), was evaluated on the DocLayNet dataset. Table VI compares these backbones with and without the PromptDLA module. PromptDLA consistently enhances the performance across all tested architectures. Notably, it improved +2.3 mAP points for ViT-Base, +1.0 mAP points for Swin-Base, and +0.7 mAP points for ResNet-50 compared to their respective baselines. These consistent gains underscore the versatility of PromptDLA, demonstrating that its effectiveness is not confined to a specific architectural paradigm and that it successfully leverages domain cues to benefit both Transformer and CNN models in document layout analysis.

Generalization Across Different Detection Heads. We further evaluate the generalizability of PromptDLA by integrating it with two representative detection heads: Faster R-CNN and DETR, both using a ResNet-50 backbone. As shown in Table VII, PromptDLA improves mAP by +1.6 on Faster

TABLE VI

PERFORMANCE COMPARISON OF DIFFERENT BACKBONE ARCHITECTURES WITH AND WITHOUT PROMPTDLA ON THE DocLAYNET DATASET. Δ INDICATES THE ABSOLUTE mAP IMPROVEMENT ACHIEVED BY ADDING PROMPTDLA COMPARED TO THE RESPECTIVE BASELINE MODEL

Tag	Method	mAP	Δ
(a)	ViT	76.4	
(b)	ViT + PromptDLA	78.7	+2.3
(c)	Swin	78.7	-0.12
(d)	Swin + PromptDLA	79.7	+1.0
(e)	ResNet50 (DETR)	77.0	
(f)	ResNet50 + PromptDLA	77.7	+0.7

R-CNN and +0.7 on DETR, demonstrating its compatibility with both traditional and Transformer-based detectors.

TABLE VII

EFFECT OF PROMPTDLA ON DIFFERENT DETECTION HEADS. Δ DENOTES THE ABSOLUTE mAP IMPROVEMENT. WE RE-IMPLEMENTED BASELINE MODELS FOR FAIR COMPARISON.

Tag	Method	mAP	Δ
(a)	Faster R-CNN (R50)	66.8	
(b)	Faster R-CNN + PromptDLA	68.4	+1.6
(c)	DETR (R50)	77.0	
(d)	DETR + PromptDLA	77.7	+0.7

C. Comparison with State-of-the-arts

We compare the proposed PromptDLA with (1) the DLA frameworks without pretraining: Faster-RCNN [28], Mask-RCNN [29], YOLOV5 [46], SwinDocSegmenter [5], and TransDLANet [17] and (2) the DLA framework with different pretraining models: DiT [34], LayoutLMv3 [32], and SelfDocSeg [5]. We evaluate PromptDLA on the challenging DocLayNet dataset [16], which provides diverse document types with significant domain variations. In our experimental setup, we systematically use the document type as the domain prompts, enabling our model to adapt to specific document characteristics. Table II presents that our PromptDLA outperforms the DLA framework with and without pretraining methods. We can observe that the promptDLA outperforms state-of-the-art SwinDocSegmenter [5] with 1.8% mAP. Although SwinDoc gets better mAPs in a few rows like "Table" and "Picture," we think these discernible categories are less relevant to the domain. Nevertheless, our PromptDLA exhibits substantial improvement over other domain-related detail categories, such as "Footnote" (from 64.8 to 83.0) and "Section-Header" (from 66.4 to 76.9).

D. Performance on Different Pretrained Models

We assess the performance of PromptDLA upon different pre-trained DLA models and datasets, validating that PromptDLA is easily plugged to enhance different DLA frameworks. As shown in Table VIII, our method can be applied to different pre-trained frameworks, including LayoutLMV3 and DiT. The performance of the PromptDLA is a further improvement on the pre-trained model. The stronger the performance of the pre-trained model, the better our method performed based on it. Excitingly, PromptDLA outperforms state-of-the-art models such as SwinDocSegmenter [5],

TABLE VIII

PERFORMANCE OF THE PROMPTDLA WITH DIFFERENT PRE-TRAINED MODELS ON DocLAYNET, M6DOC, AND D⁴LA, AND COMPARISON WITH THE CURRENT STATE-OF-THE-ART METHOD

Model	mAP@IOU[0.50:0.95]		
	DocLayNet	M6Doc	D ⁴ LA
TransDLA [17]	72.3	63.8	-
SwinDocSegmenter [5]	76.9	-	-
VGT [18]	-	-	68.8
LayoutLMv3	75.7	60.5	62.6
+PromptDLA	76.4(+0.7)	61.3(+0.8)	63.1(+0.5)
DiT	76.4	67.2	67.7
+PromptDLA	78.7(+2.3)	69.2(+2.0)	69.1(+1.4)

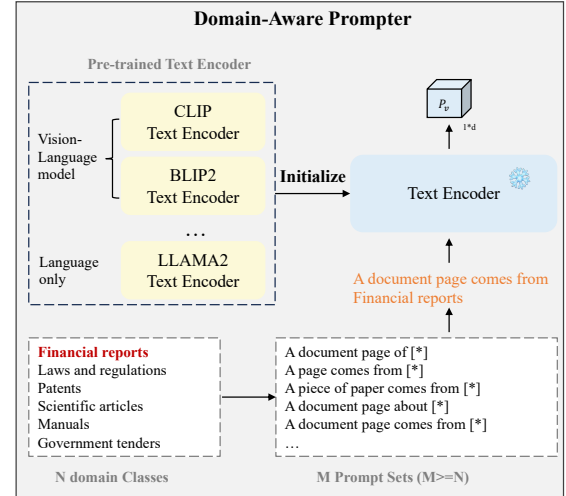


Fig. 7. Different pre-trained text encoder for PromptDLA.

TransDLA [17], and VGT [18] by 1.8%, 5.4%, and 0.3% on DocLayNet, M6Doc, and D⁴LA, respectively. Notably, compared to tailored models like SwinDocSegmenter and VGT, DiT with PromptDLA stands out for its simplicity and effectiveness.

E. Different Prompter

Different pre-trained text encoder for PromptDLA we delve deeper into the extent of prior knowledge that various large pre-trained models can provide about document images. We evaluate the performance of text encoders derived from different large pre-trained models, which fall into two main categories: those originating from vision-language pre-trained models and those from language-only pre-trained models. Specifically, we explore the capabilities of CLIP [19], BLIP2 [27], and LLAMA2 [20]. Notably, CLIP and BLIP2 are vision-language models, while LLAMA2 exclusively operates as a language model.

As illustrated in Table IX, first, compared to the baseline model DiT without any domain-aware prompt, "w/o Pre-trained Text Encoder" reduces the model's accuracy by 0.1. This suggests that utilizing domain information but randomly initializing it without a pre-trained text encoder doesn't effectively guide the model to differentiate between various domain documents. Conversely, using weights from pre-trained models improves performance for all three models, emphasizing a

solid correlation between document images in DLA datasets and their respective domains. It highlights the ability of the Text Encoder from Pre-trained models to provide valuable prior knowledge.

TABLE IX

EXPERIMENTS ON DIFFERENT PRE-TRAINED MODEL TEXT ENCODERS (W/O PRE-TRAINED TEXT ENCODER MEANS RANDOM INITIALIZING PROMPT WITHOUT USING A PRE-TRAINED TEXT ENCODER).

	mAP	Δ
DiT [34]	76.4	
w/o Pre-trained Text Encoder	76.3	-0.1
CLIP Text Encoder [19]	78.7	+2.3
BLIP2 Text Encoder [27]	79.0	+2.6
LLAMA2 Text Encoder [20]	77.8	+1.4

Moreover, CLIP and BLIP2 outperform LLAMA2, indicating that the Text Encoder from vision-language pre-trained models is superior to language-only pre-trained models. A vision-language pre-trained model can offer prior knowledge about both the relationship between document images and their corresponding text descriptions and general text representation. In contrast, a language-only pre-trained model only possesses knowledge concerning understanding human language.

Furthermore, BLIP2 outperforms CLIP by 0.3 when using the same prompt, suggesting that BLIP2 can more accurately find the relationship between document images and their text descriptions. Consequently, a visual language pre-training model with superior performance could provide an even greater boost to my approach. In our paper, we uniformly use CLIP to explore the role of other modules, but we can replace it with BLIP2 or other superior visual language pre-training models for more accurate results.

Analysis of Prompt Generation Strategies. We investigate different prompt generation strategies as illustrated in Fig. 4. As shown in Table X, we first compare human knowledge versus LVLM generation approaches. For real-world applications where domain labels are unavailable, human-knowledge prompts require a preceding classification step. We trained a ViT-based domain classifier that achieves 90% accuracy on DocLayNet. Integrating this classifier (c) results in a negligible mAP drop of only 0.12 compared to using ground-truth labels (a), confirming the approach's practical viability. In contrast, the LVLM-based generator (b) offers greater generality by eliminating the need for predefined categories, albeit at a higher computational cost. Despite this difference, its performance remains nearly identical to the human-knowledge approach. Second, we examine hybrid prompts and their impact on prompt quality. To combine the strengths of both methods, our hybrid strategy (d) leverages human knowledge to guide the LVLM, as shown in Fig. 4(c). This approach performs best, improving the mAP by 0.33 over the baseline, demonstrating that high-level human knowledge can effectively steer the LVLM toward more relevant feature descriptions. Furthermore, we explored concatenating prompts from methods (a) and (b) simultaneously (e). This concatenation resulted in almost no improvement, underscoring that the quality and relevance of a single, well-formed prompt are more critical than prompt

quantity.

TABLE X

ANALYSIS OF DIFFERENT PROMPT GENERATION STRATEGIES ON THE DocLayNet DATASET. THE Δ COLUMN MEASURES THE MAP CHANGE RELATIVE TO THE BASELINE HUMAN-KNOWLEDGE PROMPT (A).

Tag	Method	mAP	Δ
(a)	Human-Knowledge Prompt (Baseline)	78.69	-
(b)	LVLM-Generated Prompt	78.68	-0.01
(c)	Human-Knowledge Prompt w/ Classifier	78.57	-0.12
(d)	Hybrid Prompt (Human-Guided LVLM)	79.02	+0.33
(e)	Concatenated Prompts (a) + (b)	78.71	+0.02

F. Multi-Modalities

We finally explore whether the text of a document can improve layout analysis accuracy. We use Optical Character Recognition (OCR) to extract the text information from the document and then use the CLIP text encoder to retrieve text tokens. These tokens are concatenated into the PromptDLA backbone. The results, shown in Table XI, indicate that the text modality does not improve the layout analysis performance.

TABLE XI

COMPARISON OF VISUAL-ONLY VS TEXTUAL AND VISUAL MODALITIES

Tag	Method	mAP	Δ
(a)	PromptDLA	78.69	
(b)	PromptDLA + OCR	78.53	-0.16

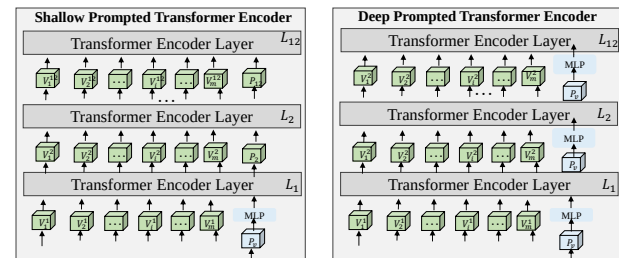


Fig. 8. Details of the shallow and deep prompt.

G. Ablation Studies

This section investigates the impact of different pretraining methods and the model's design.

Effect of Pretraining Methods. We analyze the model's accuracy in three situations: without pretraining on document images, with self-pretraining on document images using the single vision model, and with the DiT model being self-pretrained using a vision-language approach. Our baseline uses the pre-trained transformer encoder of DiT and the text

TABLE XII

ABLATION STUDY FOR PRETRAINING METHODS.

Tag	Method	mAP	Δ
(a)	PromptDLA	78.69	
(b)	w/o pretraining	70.66	-8.03
(c)	VLP-pretraining(Layoutlmv3)	76.38	-2.31
(d)	w/o CLIP text encoder	76.33	-2.36

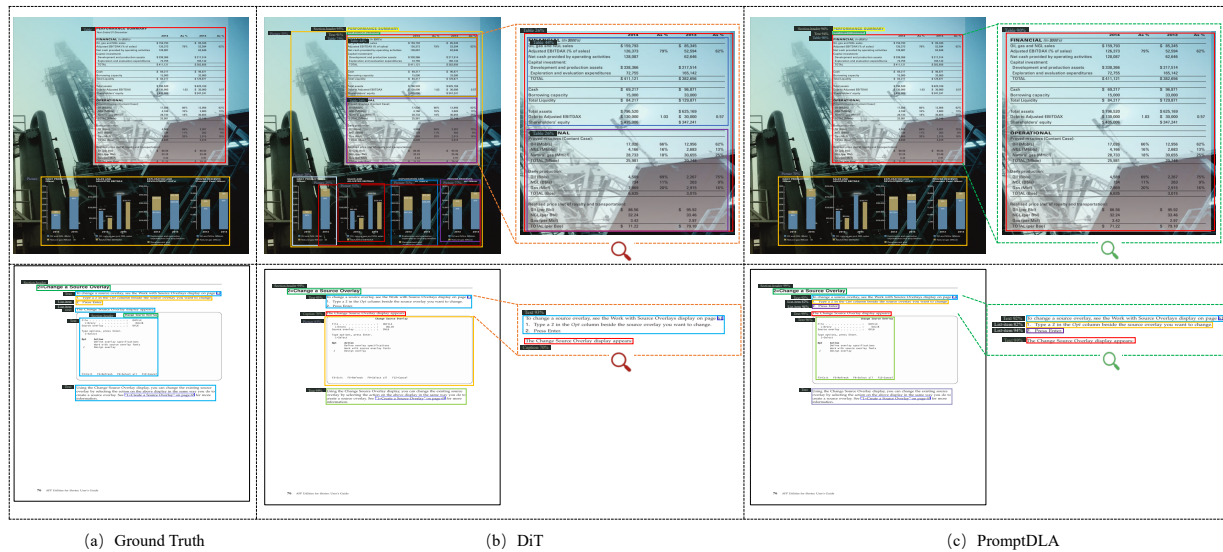


Fig. 9. Qualitative comparison between DiT and PromptDLA on Financial Reports (1st row) and Laws (2nd row) domain from DocLayNet. It is best viewed in color and zooming out.

encoder of CLIP. Compared to the baseline, the transformer encoder without pretraining reduces the mAP by 8.03, as illustrated in Table XII. Therefore, self-pretraining on large-scale document data can significantly enhance model accuracy by allowing the model to learn the generative document image representation via self-pretraining. Furthermore, the multi-modal pre-trained model from LayoutLMv3 is less effective than the single-model self-pre-trained transformer from DiT. Therefore, the single vision model's self-pre-training method is most suitable for dealing with layout analysis problems. The CLIP text encoder is well-pretrained and can provide a prior textual representation. Table XII demonstrates that removing the CLIP text encoder and randomly initializing the prompt vector decreases 2.36 in mAP. The CLIP text encoder is already trained on large-scale image-text pairs. Thus, it can provide domain prior knowledge effectively.

Effect of Prompt Location. We explore the impact of prompt location in the transformer encoder, employing two distinct prompt methods illustrated in Fig. 8: Shallow Prompted Encoder (SPE) and Deep Prompted Encoder (DPE). SPE exclusively incorporates a prompt at the first layer of the transformer encoder, whereas DPE integrates a prompt at every layer. As shown in Table XIII, both SPE and DPE models exhibit improvements compared to the baseline model without a prompt. Notably, the DPE method surpasses the baseline by 2.26 mAP. In the case of DPE, the CLIP text encoder generates 512-dimensional vectors, requiring MLP projection to the same 768-dimensional feature space as the image patch. We explore the design of the MLP for each transformer layer, specifically whether to use a shared MLP layer. Our observations indicate that the DPE method with a shared MLP layer results in an improvement of 2.01.

Effect of Prompt Design. This study aims to investigate how to design prompts for document domain priors, exploring the correlation between prompts and DLA model performance on DocLayNet. Table XIV presents various prompt templates

TABLE XIII
ABLATION STUDY FOR PROMPT LOCATION.

Tag	Method	mAP	Δ
(a)	Baselines	76.43	
(b)	SPE	78.41	+1.98
(c)	DPE	78.69	+2.26
(d)	DPE with share MLP Layer	78.44	+2.01

in the first column, the zero-shot document classification accuracy via CLIP in the second column, and the DLA results in the third column. The experiments demonstrate that prompts achieving higher accuracy in the CLIP classification task also lead to superior performance in our DLA model. This validates the rationale of our approach to creating prompt sets, focusing exclusively on the top-k accuracy rankings for the CLIP zero-shot classification task.

TABLE XIV
ABLATION STUDY FOR DIFFERENT DOMAIN PROMPTS.

Prompt	Accuracy	mAP
w/o prompt	-	76.43
A page comes from {Domain Class}	44.05	77.49
A document page of {Domain Class}	44.05	77.53
A piece of paper concerning with {Domain Class}	44.44	77.55
A piece of paper comes from {Domain Class}	46.99	77.97
A document page comes from {Domain Class}	48.45	78.12

H. Discussions

Results on Each Document Type in DocLayNet. As illustrated in Table XV, we trained PromptDLA and DiT using DocLayNet's entire training dataset and evaluated their performance separately on each document type in the test dataset. Compared to the baseline DiT model without any domain-aware prompt, the model with Prompt outperforms it in terms of mAP, except for "Laws and Regulations" and "Patents". Despite a decrease in our model's mAP in the

TABLE XV
EXPERIMENTS RESULT ON EACH DOCUMENT TYPE IN DocLayNet ("w/o" AND "w" DENOTE DiT BASELINE AND PROMPTDLA, RESPECTIVELY).

		Caption	Footnote	Formula	List-item	Page-footer	Page-header	Picture	Section-header	Table	Text	Title	mAP
Financial reports	w/o	55.7	10.6	-	69.4	52.6	46.1	64.0	64.4	90.4	85.1	54.5	59.3
	w	55.4	23.9	-	73.8	60.2	55.4	67.7	67.6	91.4	86.2	60.5	64.2
Government tenders	w/o	15.0	82.0	65.3	90.0	51.5	75.6	75.3	78.0	95.4	85.1	39.1	68.4
	w	27.1	88.3	100.0	91.8	50.9	80.4	77.8	81.7	95.0	88.3	61.3	76.6
Laws and regulations	w/o	28.0	91.6	16.7	81.6	42.9	71.8	49.5	68.0	70.4	84.1	84.4	62.6
	w	28.5	96.1	18.5	82.0	42.4	64.4	50.4	69.5	67.7	84.7	84.6	62.6
Manuals	w/o	85.7	33.8	-	82.0	76.1	87.6	76.7	78.3	70.3	83.5	59.4	73.3
	w	89.5	43.4	-	82.5	77.3	91.9	77.9	81.4	71.7	85.9	69.2	77.1
Patents	w/o	78.0	-	61.6	90.4	85.8	91.1	89.8	91.8	92.9	93.1	88.5	86.3
	w	81.1	-	51.8	91.7	87.3	92.0	87.7	91.9	92.8	93.6	86.6	85.7
Scientific articles	w/o	92.9	70.5	69.8	94.6	87.4	87.3	93.9	90.0	98.3	91.8	96.2	88.4
	w	94.2	81.2	74.6	95.4	90.2	89.9	93.9	90.8	98.2	92.8	96.0	90.7

"Patents" domain, it still surpassed the baseline model in most categories, such as "Caption" and "Page-header." We attribute the decline in the Formula and Picture categories to the limited correlation between Formula and Picture in the field of patents. **Validating CLIP's Prior Knowledge of Documents.** The motivation behind using CLIP as a prompter is its impressive ability to learn visual and textual representations. We validate our choice of CLIP as a prompter by evaluating its zero-shot document classification performance. As shown in Table XVI, CLIP achieves 48.45% accuracy on DocLayNet and 54.55% on M6Doc without any fine-tuning. This indicates that CLIP possesses prior information about the document layout image. **Computational Overhead.** PromptDLA is computationally efficient, adding negligible overhead. As shown in Table XVII, applying our method to a DiT backbone reduces inference speed by a mere 0.13 FPS (from 6.75 to 6.62 FPS) on an RTX 3090 GPU. This trend holds across various models.

TABLE XVI
ZERO-SHOT DOCUMENT CLASSIFICATION PERFORMANCE OF CLIP.

	DocLayNet	M6Doc
#Document Types	6	7
Accuracy	48.45%	54.55%

TABLE XVII
COMPUTATIONAL OVERHEAD OF PROMPTDLA.

Model	mAP@IOU[0.50:0.95]	FPS
DiT-Base	76.4	6.75
+PromptDLA	78.7(+2.3)	6.62(-0.13)
LayoutLMv3	75.7	4.44
+PromptDLA	76.4(+0.7)	4.41(-0.03)

Visualization. Fig. 9 presents the visualized results on the Financial Reports and Laws domain from DocLayNet comparison between ground truth, DiT, and PromptDLA is presented. For the sample of Financial in the 1st row, DiT misidentifies the background as "Figure" and recognizes the whole "Table" as two separates. At the same time, the PromptDLA removes the misclassification of "Figure" and produces a precise box of "Table." Moreover, the sample of Laws in 2nd row shows that our method precisely excludes the text box and identifies only the text inside it when processing manuals with text boxes. In contrast, DiT incorrectly identifies the text box as

a complete figure. These qualitative results demonstrate the ability of the PromptDLA to recognize ambiguous objects by domain prior. Besides, Fig. 10 in Appendix shows more visualization examples.

V. CONCLUSIONS

We propose a novel PromptDLA framework, which can explicitly introduce domain prior into the DLA frameworks and steer DLA automatically, distinguishing the variability of different domains. The PromptDLA features a unique domain-aware prompter that could customize prompts according to the specific attributes of the data domain. We underscore the significance of utilizing domain priors in DLA through extensive experiments. The results show a new state-of-the-art performance across multiple datasets, including DocLayNet (78.7), M6Doc (69.2), and D⁴LA (69.1). It's worth mentioning that the proposed domain-aware prompter is easily plugged into enhance different DLA frameworks. While PromptDLA demonstrates strong performance and adaptability, several avenues for future work remain. A key direction is efficiency optimization. Integrating large language or vision-language models, particularly in the prompter component, introduces computational overhead compared with baseline DLA models.

REFERENCES

- [1] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6154–6162.
- [2] H. Bao, L. Dong, S. Piao, and F. Wei, "Beit: Bert pre-training of image transformers," *arXiv preprint arXiv:2106.08254*, 2021.
- [3] Z. Zhang, J. Ma, J. Du, L. Wang, and J. Zhang, "Multimodal pre-training based on graph attention network for document understanding," *IEEE Transactions on Multimedia*, vol. 25, pp. 6743–6755, 2022.
- [4] H. Bi, C. Xu, C. Shi, G. Liu, Y. Li, H. Zhang, and J. Qu, "Srrv: A novel document object detector based on spatial-related relation and vision," *IEEE Transactions on Multimedia*, vol. 25, pp. 3788–3798, 2022.
- [5] A. Banerjee, S. Biswas, J. Lladós, and U. Pal, "Swindocsegformer: An end-to-end unified domain adaptive transformer for document instance segmentation," *arXiv preprint arXiv:2305.04609*, 2023.
- [6] J. Ma, J. Du, P. Hu, Z. Zhang, J. Zhang, H. Zhu, and C. Liu, "Hrdoc: Dataset and baseline method toward hierarchical reconstruction of document structures," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 2, pp. 1870–1877, Jun. 2023. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/25277>
- [7] C. Ma, Y. Zhang, M. Tu, Y. Zhao, Y. Zhou, and C. Zong, "Multi-teacher knowledge distillation for end-to-end text image machine translation," in *International Conference on Document Analysis and Recognition*. Springer, 2023, pp. 484–501.

- [8] J. He, L. Wang, Y. Hu, N. Liu, H. Liu, X. Xu, and H. T. Shen, "Icl-d3ie: In-context learning with diverse demonstrations updating for document information extraction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19485–19494.
- [9] C. Luo, Y. Shen, Z. Zhu, Q. Zheng, Z. Yu, and C. Yao, "Layoutlm: Layout instruction tuning with large language models for document understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 15630–15640.
- [10] Y. Wang, N. Lipka, R. A. Rossi, A. Siu, R. Zhang, and T. Derr, "Knowledge graph prompting for multi-document question answering," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, 2024, pp. 19206–19214.
- [11] Y. Liang, Y. Zhang, C. Ma, Z. Zhang, Y. Zhao, L. Xiang, C. Zong, and Y. Zhou, "Document image machine translation with dynamic multi-pre-trained models assembling," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024, pp. 7077–7088.
- [12] A. Hu, H. Xu, J. Ye, M. Yan, L. Zhang, B. Zhang, C. Li, J. Zhang, Q. Jin, F. Huang *et al.*, "mplug-docowl 1.5: Unified structure learning for ocr-free document understanding," *arXiv preprint arXiv:2403.12895*, 2024.
- [13] Z. Zhang, Y. Zhang, Y. Liang, L. Xiang, Y. Zhao, Y. Zhou, and C. Zong, "Layoutdit: Layout-aware end-to-end document image translation with multi-step conductive decoder," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 10043–10053.
- [14] X. Zhong, J. Tang, and A. J. Yepes, "Publaynet: largest dataset ever for document layout analysis," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 1015–1022.
- [15] M. Li, Y. Xu, L. Cui, S. Huang, F. Wei, Z. Li, and M. Zhou, "Docbank: A benchmark dataset for document layout analysis," *arXiv preprint arXiv:2006.01038*, 2020.
- [16] B. Pfizmann, C. Auer, M. Dolfi, A. S. Nassar, and P. Staar, "Doclaynet: A large human-annotated dataset for document-layout segmentation," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 3743–3751.
- [17] H. Cheng, P. Zhang, S. Wu, J. Zhang, Q. Zhu, Z. Xie, J. Li, K. Ding, and L. Jin, "M6doc: A large-scale multi-format, multi-type, multi-layout, multi-language, multi-annotation category dataset for modern document layout analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 15138–15147.
- [18] C. Da, C. Luo, Q. Zheng, and C. Yao, "Vision grid transformer for document layout analysis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19462–19472.
- [19] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.
- [20] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.
- [21] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Conditional prompt learning for vision-language models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 16816–16825.
- [22] D. Lee, S. Song, J. Suh, J. Choi, S. Lee, and H. J. Kim, "Read-only prompt optimization for vision-language few-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 1401–1411.
- [23] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24824–24837, 2022.
- [24] S. Liu, C. Chen, X. Qu, K. Tang, and Y.-S. Ong, "Large language models as evolutionary optimizers," in *2024 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2024, pp. 1–8.
- [25] B. Chen, Z. Zhang, N. Langrené, and S. Zhu, "Unleashing the potential of prompt engineering in large language models: a comprehensive review," *arXiv preprint arXiv:2310.14735*, 2023.
- [26] Z. Wang, H. Zhang, C.-L. Li, J. M. Eisenschlos, V. Perot, Z. Wang, L. Miculicich, Y. Fujii, J. Shang, C.-Y. Lee *et al.*, "Chain-of-table: Evolving tables in the reasoning chain for table understanding," *arXiv preprint arXiv:2401.04398*, 2024.
- [27] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," *arXiv preprint arXiv:2301.12597*, 2023.
- [28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.
- [29] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.
- [30] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
- [31] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [32] Y. Huang, T. Lv, L. Cui, Y. Lu, and F. Wei, "Layoutlmv3: Pre-training for document ai with unified text and image masking," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 4083–4091.
- [33] Y. Yu, Y. Li, C. Zhang, X. Zhang, Z. Guo, X. Qin, K. Yao, J. Han, E. Ding, and J. Wang, "Structextv2: Masked visual-textual prediction for document image pre-training," *arXiv preprint arXiv:2303.00289*, 2023.
- [34] J. Li, Y. Xu, T. Lv, L. Cui, C. Zhang, and F. Wei, "Dit: Self-supervised pre-training for document image transformer," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 3530–3539.
- [35] S. Appalaraju, B. Jasani, B. U. Kota, Y. Xie, and R. Manmatha, "Docformer: End-to-end transformer for document understanding," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 993–1003.
- [36] J. Gu, J. Kuen, V. I. Morariu, H. Zhao, R. Jain, N. Barmpalios, A. Nenkova, and T. Sun, "Unidoc: Unified pretraining framework for document understanding," *Advances in Neural Information Processing Systems*, vol. 34, pp. 39–50, 2021.
- [37] S. Maity, S. Biswas, S. Manna, A. Banerjee, J. Lladós, S. Bhattacharya, and U. Pal, "Selfdocseg: A self-supervised vision-based approach towards document segmentation," *arXiv preprint arXiv:2305.00795*, 2023.
- [38] R. Rezaei, M. J. Sabet, J. Gu, D. Rueckert, P. Torr, and A. Khakzar, "Learning visual prompts for guiding the attention of vision transformers," *arXiv preprint arXiv:2406.03303*, 2024.
- [39] T. B. Brown, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.
- [40] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.
- [41] R. Zhang, J. Han, C. Liu, A. Zhou, P. Lu, Y. Qiao, H. Li, and P. Gao, "Llama-adapter: Efficient fine-tuning of large language models with zero-initialized attention," in *The Twelfth International Conference on Learning Representations*, 2024.
- [42] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10012–10022.
- [43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [44] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim, "Visual prompt tuning," in *European conference on computer vision*. Springer, 2022, pp. 709–727.
- [45] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, 2020, pp. 213–229.
- [46] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 2778–2788.

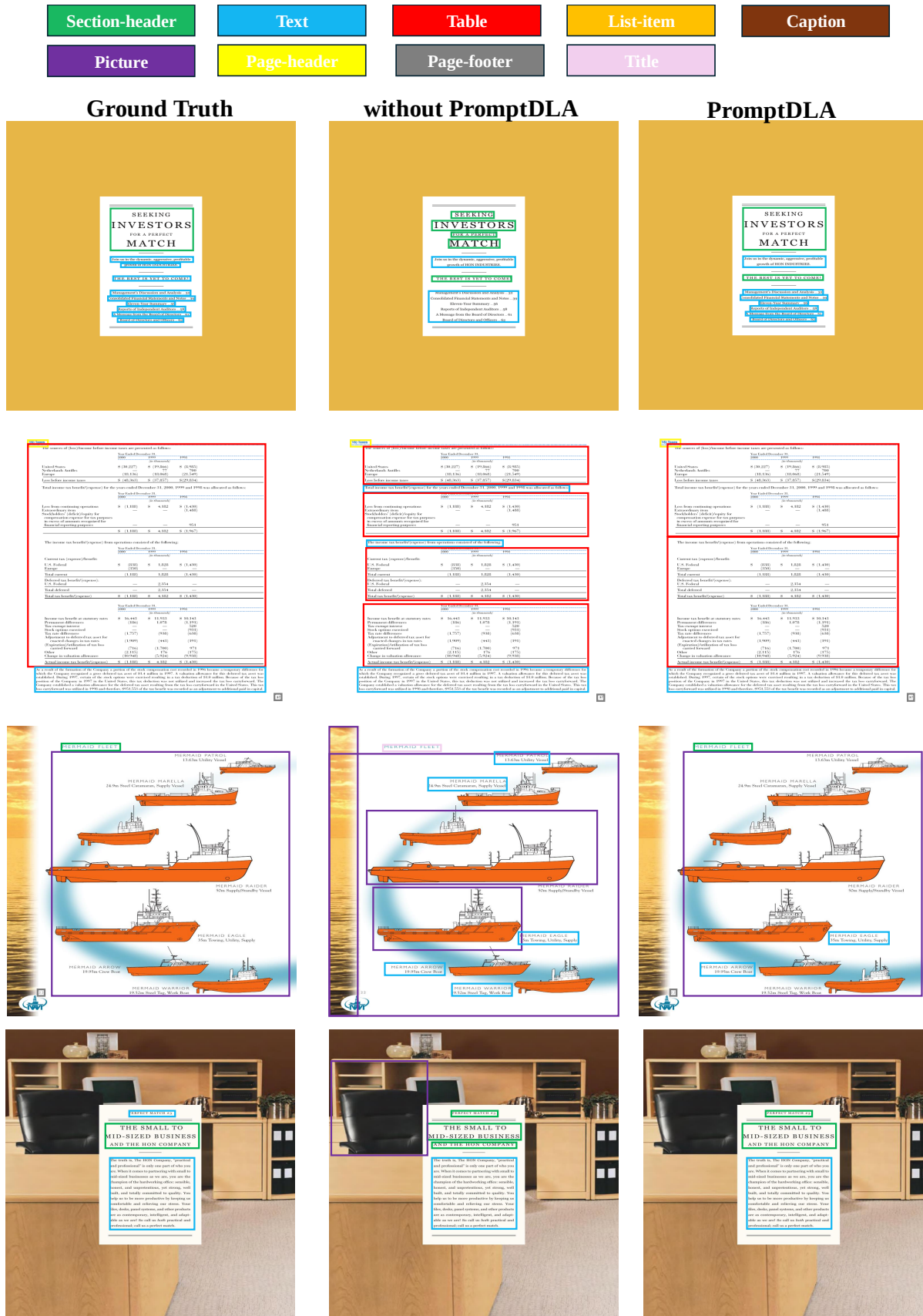


Fig. 10. Appendix: More visualization examples on the DocLayNet dataset. From left to right, columns show Ground Truth annotations, predictions without PromptDLA, and predictions with PromptDLA, highlighting document layout segmentation accuracy improvements.