DocFusion: A Unified Framework for Document Parsing Tasks

Anonymous ACL submission

Abstract

Document parsing involves layout element detection and recognition, essential for analyzing complex structures and extracting key information. However, existing methods often employ multiple models for these tasks, leading to increased system complexity and maintenance overhead. While some models attempt to unify detection and recognition, they often fail to address the intrinsic differences in data representations, thereby limiting performance in document processing. Our research reveals that recognition relies on discrete tokens, whereas detection relies on continuous coordinates, leading to challenges in gradient updates and optimization. To bridge this gap, we propose the Gaussian-Kernel Cross-Entropy Loss (GK-CEL), enabling generative frameworks to handle both tasks simultaneously. Building upon GK-CEL, we propose DocFusion, a unified document parsing model with only 0.28B parameters. Additionally, we construct the DocLatex-1.6M dataset to provide high-quality training support. Experimental results demonstrate that DocFusion leverage GK-CEL effectively exploits the benefits of multi-task learning and achieves state-of-the-art performance across four key tasks.

1 Introduction

Document parsing plays a significant role in extracting structured data from documents, making it foundational for various downstream applications. For example, in Retrieval-Augmented Generation (RAG) workflows (Ren et al., 2023; Zhang et al., 2022), extracting well-organized and contextually rich information from documents can improve the performance of large language models (LLMs) (Jiang et al., 2023; Zhao et al., 2024a; Gao et al., 2024). However, real-world documents often embed information in complex structures, such as hierarchical layouts, mathematical expressions, and tables, which pose considerable challenges for automated parsing.

Tool Type	Size	DLA	MER	TR	OCR
System					
open-parse (2024)	-	✓	✗	✓	✓
LlamaParse (2024)	-	✓	✓	✓	✓
DeepDoc (2024)	-	✓	✗	✓	✓
MinerU (2024)	-	✓	✓	✓	✓
Model					
DocYOLO(2024c)	20M	✓	✗	✗	✗
ViTLP (2024)	253M	✓	✗	✓	✓
UniMER (2024b)	325M	✗	✓	✗	✗
Nougat (2023)	350M	✗	✓	✓	✓
GOT (2024)	580M	✗	✓	✓	✓
StructTable (2024)	938M	✗	✗	✓	✗
DocFusion(Ours)	289M	✓	✓	✓	✓

Table 1: Capabilities of document parsing tools. **Model** refers to a single model, while **System** integrates multiple models. **DLA**: Document Layout Analysis. **MER**: Math Expression Recognition. **TR**: Table Recognition. **OCR**: Optical Character Recognition. Compare with multi-model systems, DocFusion achieves all four tasks within a single model, requiring only 289M parameters.

Existing methods can be categorized into two main approaches: multi-module pipeline systems and end-to-end page-level OCR models. Multi-module pipeline systems decompose document parsing tasks into independent modules, allowing each module to adopt the best model. For example, DocLayout-YOLO (Zhao et al., 2024c) has demonstrated excellent performance in Layout analysis, while UniMERNet (Wang et al., 2024b) achieves SOTA results in Math Expression Recognition. Although this approach improves performance for specific tasks, integrating multiple models into a single system increases overall complexity. Moreover, these systems fail to fully exploit task-level collaboration, leading to inefficiencies in parameter usage. In contrast, end-to-end page-level OCR models, such as Nougat (Blecher et al., 2023) and GOT (Wei et al., 2024), can seamlessly integrate multiple recognition tasks.

While the outputs of these models demonstrate a well-organized logical structure, the models lack the ability to generate bounding boxes for layout elements. As a result, they fail to preserve the spatial relationships between documents and their layouts, which is crucial for interpretability in RAG workflows. Additionally, while these models perform well on page-level images, it struggles with specific layout elements, limiting their flexibility in application. To address these issues, this research focused on four key tasks: document layout analysis (**DLA**), mathematical expression recognition **(MER)**, table recognition **(TR)**, and optical character recognition **(OCR)**.

Several studies have attempted to apply generative frameworks to integrate object detection and content recognition, achieving promising results on natural images (Xiao et al., 2023). However, extending such frameworks to document images presents significant challenges due to the inherent structural and representational differences between these domains. Through experiments, we identify the primary issue as the fundamental conflict between the continuous nature of coordinate data and the discrete nature of token generation, which disrupts gradient updates during multi-task training (discussed in Section 3.2). In natural scene images, small deviations in coordinates and text are generally tolerable. However, in document parsing, even minor errors in LaTeX code can critically impact compilation success rates. This imposes stricter accuracy requirements when applying such frameworks to document understanding tasks. To address these challenges, we propose Gaussian-Kernel Cross-Entropy Loss (GK-CEL), an improved objective function designed to mitigate the inconsistencies between discrete and continuous data representations, enhancing the performance of generative frameworks in document parsing.

MER and TR are essential for LaTeX-based document processing, but existing datasets suffer from inconsistent formatting and redundant characters, where different writing styles generate identical compiled outputs, introducing noise that hinders model training and generalization (details are provided in Appendix A.3). To address this, we propose DocLatex-1.6M, a large-scale, high-quality dataset that enhances annotation consistency and improves model training efficiency. Experiments demonstrate that DocFusion trained on this dataset outperforms task-specific models with fewer parameters.

Our contributions are summarized as follows:

- We propose DocFusion, a unified generative multi-task model that standardizes task formulations and achieves SOTA performance across four key document parsing tasks.
- We propose GK-CEL to resolve the conflict between continuous coordinate and discrete token in the generative framework, enhancing document parsing and offering a reference for similar frameworks in other domains.
- Experimental results demonstrate that incorporating multi-task data significantly outperforms single-task setups, providing insights into the benefits of multi-task learning in document parsing.
- We constructed DocLatex-1.6M, a large-scale, high-quality dataset with 1.5M LaTeX-annotated math expressions and 100K tables, offering a valuable resource for advancing document parsing research.

2 Related Work

Document Parsing Models. Document parsing models have seen remarkable progress across various tasks. DLA has evolved from vision-based methods (Wick and Puppe, 2018; Bao et al., 2021) to multimodal approaches integrating textual features (Xu et al., 2022; Huang et al., 2022). OCR has transitioned from template matching (Smith, 2007) to deep learning-based solutions (Bušta et al., 2017; Chen et al., 2021; Mosbah et al., 2024). MER progressed from symbol segmentation (Miller and Viola, 1998) to CNN-RNN hybrids (Le et al., 2019) and Transformer-based models (Wang et al., 2024b). Similarly, TR now employs methods like grid segmentation and image-to-sequence techniques to reconstruct structured data (Qasim et al., 2019; Huang et al., 2023; Xia et al., 2024). Page-level end-to-end OCR models like Nougat (Blecher et al., 2023) and GOT (Wei et al., 2024) simplify workflows by integrating multi recognition tasks.

Modular Pipeline Systems. The advancements in task-specific models have driven the development of modular pipeline systems, which process complex document structures through specialized modules. For instance, Open-Parse(Filimonov,

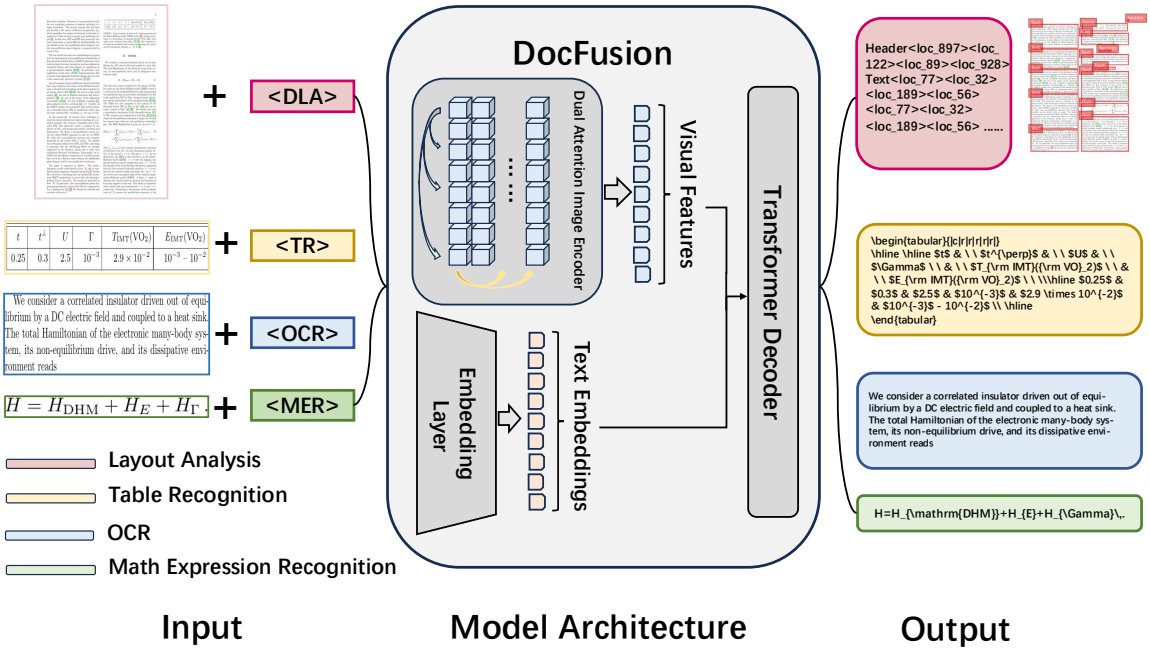


Figure 1: The model comprises three key components: a visual encoder, a text embedding layer and a Transformer decoder. The image features extracted by the visual encoder and the instruction embeddings are combined and then passed to the Transformer decoder, which produces the final output sequence.

2024) performs well in incrementally parsing complex layouts but lacks support for MER. Other systems, such as DeepDoc(Yu, 2024) and Llama-Parse(Liu, 2024), extend the scope of modular pipelines to handle more diverse tasks. In particular, MinerU(Zhao et al., 2024b) stands out by supporting advanced features such as complex layout parsing and Markdown conversion.

3 Method

We introduce the model architecture (3.1) and explain how detection tasks are represented into the generative framework. Then, we discuss the challenges (3.2) of detection tasks within this framework. Next, we explain the Gaussian-Kernel Cross-Entropy Loss(3.3)

3.1 Architecture

As shown in Figure 1, the architecture of DocFusion consists of three main components: a vision encoder, a text embedding layer, and a Transformer decoder. Since the task instructions are limited and predefined, no Transformer encoder is included, task-specific prompts are directly embedded, simplifying the architecture.

To unify the representation of object detection and text recognition tasks, we adopt a coordinate quantization representation (Xiao et al., 2023). Specifically, images are quantized into a fixed resolution (e.g., 1000×1000), and coordinates are represented as discrete tokens (e.g., <loc_1>, <loc_2>, ..., <loc_1000>). This approach enables the use of a unified generative framework for detection tasks. To address the challenges posed by densely structured content, the vision encoder incorporates a Dual Attention mechanism (Ding et al., 2022), which captures interactions across channel and spatial dimensions, enhancing feature extraction for intricate document layouts. Additionally, the traditional feed-forward network (FFN) is removed, reducing both parameter count and computational cost, further improving model efficiency.

The vision encoder processes input images $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ into visual features, flattened as token embeddings $\mathbf{V} \in \mathbb{R}^{N_v \times D_v}$. These embeddings are projected to D_t, resulting in $\mathbf{V}' \in \mathbb{R}^{N_v \times D_t}$, to match the task-specific prompt embeddings $\mathbf{T}\text{prompt} \in \mathbb{R}^{N_t \times D_t}$. The combined input $\mathbf{X} = [\mathbf{V}'; \mathbf{T}\text{prompt}]$ is then passed to the Transformer decoder to generate predictions.

3.2 Challenges and Motivations

While representing object detection as text generation enables joint training of layout analysis and page element recognition under a unified cross-entropy-based framework, it inherently forces continuous coordinates into discrete token spaces. This

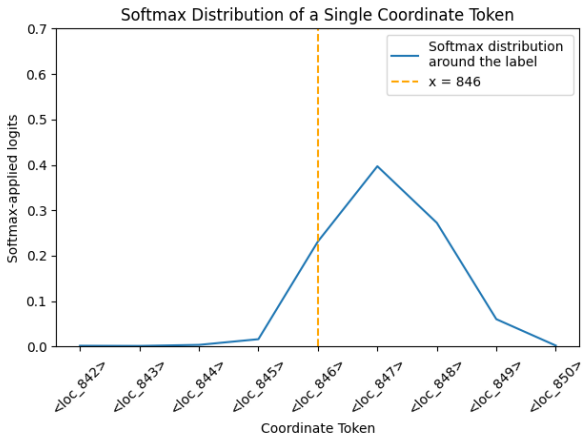


Figure 2: The distribution of logits for a target token after the loss has stabilized when using the Common CE Loss.

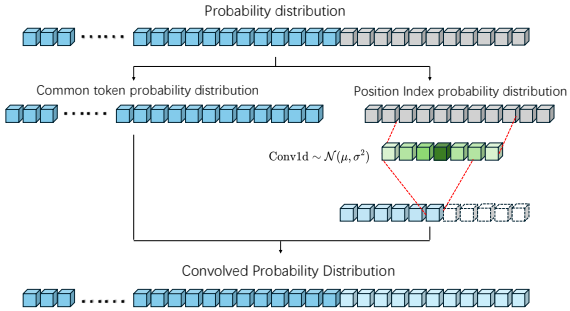


Figure 3: Illustration of Gaussian-Kernel Cross-Entropy Loss.

mismatch creates several challenges, especially in fine-tuning small coordinate adjustments, where the model struggles to produce accurate gradients, reducing training stability. As shown in Figure 2, small unavoidable deviations in coordinate labels smooth out the softmax distribution, preventing the target token's probability from forming a sharp peak. This makes it harder for the model to escape local optima and limits its learning capacity. Additionally, traditional cross-entropy loss, which is designed for discrete classification tasks, does not handle continuous changes well, further increasing inaccuracies during training.

In multi-task settings, these issues become even more challenging. The conflict between discrete loss functions and continuous coordinate optimization can skew gradients, causing one task to dominate at the cost of others. This imbalance reduces performance in other tasks and harms the model's ability to predict coordinates accurately, limiting its overall effectiveness in complex document parsing tasks. Solving these problems is critical to improving both localization accuracy and training stability across tasks.

3.3 Gaussian-Kernel Design

To address these challenges, we propose the Gaussian-Kernel Cross-Entropy Loss (GK-CEL). As shown in Figure 3, it applies a one-dimensional convolution with Gaussian-distributed weights over the probability distribution, fine-tuning the model's sensitivity to small coordinate changes while preserving the discrete treatment of cross-entropy. This approach alleviates the mismatch between discrete tokens and continuous coordinates, improves gradient quality, and prevents the coordinate prediction task from dominating the optimization process. As a result, it enhances localization accuracy and supports stable multi-task training.

Let the model's output logits be denoted as $\mathbf{Z} \in \mathbb{R}^{B \times L \times V}$, where B is the batch size, L is the sequence length, and V is the vocabulary size. The target labels are denoted as $\mathbf{T} \in \mathbb{N}^{B \times L}$. The range of indices corresponding to coordinate tokens is defined as $[s, e]$, representing their positions in the vocabulary.

The standard softmax probability distribution is first computed as:

$$\mathbf{P} = \text{softmax}(\mathbf{Z}) \tag{1}$$

A mask is then applied to zero out probabilities outside the range $[s, e]$, creating a modified probability tensor $\mathbf{P}'$:

$$\mathbf{P}'_{ijk} = \begin{cases} \mathbf{P}_{ijk}, & \text{if } k \in [s, e] \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where i represents the batch index, j represents the sequence position, and k represents the vocabulary index.

Next, a one-dimensional convolution kernel $\mathbf{K} \in \mathbb{R}^{1 \times 1 \times n}$ is constructed based on a Gaussian distribution, where n is the kernel size (an odd integer greater than 1), σ is the standard deviation and p represents the position of each element in the convolution kernel, measured as the offset from the center, where the center is located at $\frac{n+1}{2}$. The range of $p \in [1, n]$. The kernel weights of each index are computed as:

$$\mathbf{K}_p = \exp\left(-\frac{(p - \frac{n+1}{2})^2}{2\sigma^2}\right) \tag{3}$$

The kernel is then applied to $\mathbf{P}'$ via one-dimensional convolution:

$$\mathbf{C} = \text{conv1d}(\mathbf{P}', \mathbf{K}) \tag{4}$$

The convolution result $\mathbf{C}$ is integrated back into the original probability distribution $\mathbf{P}$ within the index range $[s, e]$, while retaining the original probabilities outside this range:

$$\mathbf{P}''_{ijk} = \begin{cases} \mathbf{C}_{ijk}, & \text{if } k \in [s, e] \\ \mathbf{P}_{ijk}, & \text{otherwise} \end{cases} \tag{5}$$

The final objective function is computed as:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{B} \sum_{j=1}^{L} \mathbf{M}_{ij} \log \mathbf{P}''_{ij\mathbf{T}_{ij}} \tag{6}$$

where $\mathbf{M}_{ij}$ is a mask matrix that indicates whether the target label at position (i, j) should contribute to the loss calculation.The normalization factor $\mathbf{N}$ is defined as the total number of valid targets.

4 Experiments

4.1 Training Datasets

In the training phase, the DLA task uses the DocLayNet (Pfitzmann et al., 2022) dataset, which contains 80,863 pages from 7 document types and is manually annotated with 11 categories. The images are split into 69,103/6,481/4,999 for training/validation/testing, respectively. The OCR dataset is also sourced from DocLayNet, which offers comprehensive annotations for layout elements and their corresponding text, and is widely regarded as a reliable resource in the academic community. For the TR and MER tasks, we used the DocLatex-1.6M dataset, which was constructed in this work. Additionally, although this work primarily focuses on document images, we introduced the HME100K(Yuan et al., 2022), a handwritten math expression dataset to enhance the generalization ability of the MER task.

4.2 Evaluation Metrics

4.2.1 Evaluation Metrics for Recognition

We employ BLEU (Papineni et al., 2001) and Edit Distance (Levenshtein, 1966) to evaluate sequences. Additionally, CDM (Wang et al., 2024c) and CSR were used to better assess the quality of LaTeX-based outputs.

BLEU is used for evaluating generated text, measuring n-gram overlap with reference texts.

Edit distance measures the minimum number of operations insertions, deletions, or substitutions required to transform one string into another.

CSR refers to the percentage of generated LaTeX outputs that can be successfully compiled into PDF.

ExpRate (Li et al., 2022) measures the proportion of samples where the predicted text matches the reference text without any errors.

CDM evaluates MER by comparing image-rendered expression at the character level with spatial localization, ensuring fairness and accuracy over text-based metrics like BLEU.

4.2.2 Evaluation Metrics for Detection

Since DocFusion adopts a novel approach in the DLA task without relying on confidence scores, we did not use the widely adopted Average Precision but instead focus on the following metrics:

Precision measures the proportion of correctly identified positive instances among all predicted positives.

Recall measures the proportion of correctly identified positive instances among all actual positives.

F1-score balances precision and recall, serving as their harmonic mean.

FPS measures the number of frames processed by the model per second.

4.3 Selection of Baseline Models

For the MER task, we selected UniMERNet (Wang et al., 2024b), the current state-of-the-art (SOTA) model, and Texify (Paruchuri, 2023), which has shown strong competitive performance in recent evaluations. In the OCR task, we compared several models, including the large-scale model TextMonkey (Liu et al., 2024) and smaller models such as Nougat (Blecher et al., 2023), for a multi-scale evaluation. For the TR task, we evaluated our approach against StructEqTable (Xia et al., 2024), one of the most representative models in current Table-to-Sequence methods. In the DLA task, we compared our method with two major object detection frameworks, YOLO and DETR (e.g., DocLayout-YOLO (Zhao et al., 2024c), Deformable-DETR (Zhu et al., 2020)). Although GOT (Wei et al., 2024) is not capable of handling the DLA task, it performs well in the other three recognition tasks, making it a relevant model for comparison.

Model	size	OCR		MER			TR	
		BLEU↑	EditDis↓	CDM↑	ExpRate↑	CSR↑	F1↑	CSR↑
UReader (2023)	7B	38.6	47.3	-	-	-	-	-
LLaVA-NeXT (2024)	34B	69.1	27.2	-	-	-	-	-
Nougat (2023)	250M	71.6	21.4	-	-	-	-	-
TextMonkey (2024)	7B	73.3	21.9	-	-	-	-	-
Qwen-VL-MAX (2023)	>72B	94.7	3.9	-	-	-	-	-
Qwen-VL-OCR (2023)	-	95.9	4.1	-	-	-	-	-
Pix2tex (2022)	-	-	-	76.5	41.7	95.9	-	-
Texify (2023)	312M	-	-	88.6	71.7	97.8	-	-
Mathpix	-	-	-	88.9	79.1	98.3	-	-
UniMERNet (2024b)	325M	-	-	**99.0**	89.5	99.7	-	-
MixTex (2024)	85M	-	-	-	-	-	46.2	27.4
StructEqTable (2024)	938M	-	-	-	-	-	90.6	**93.2**
GOT (2024)	580M	96.8	2.2	87.7	67.3	97.8	86.9	81.6
DocFusion(Ours)	289M	**97.4**	**1.8**	98.7	**94.2**	**99.8**	**92.1**	92.5

Table 2: Comparison of DocFusion with other models on three recognition tasks in the document scene. Specifically, DocLaynet(Pfitzmann et al., 2022) was used for OCR, DocGenome(Xia et al., 2024) for TR, and UniMER-1M(Wang et al., 2024b) for MER. More details on the TR experiments can be found in Appendix B.2. *Note:* Nougat is primarily designed for full-page recognition and tends to underperform on isolated tables or mathematical expressions.

Model	Size	DocLayNet			DocLayNet-Scientific			FPS↑	~~NMS~~	~~Conf~~
		Precision↑	Recall↑	F1↑	Precision↑	Recall↑	F1↑			
YOLOv10m (2024a)	16M	90.1	86.9	88.4	94.3	94.5	94.4	93.6	✓	✗
YOLOv11m (2024)	20M	90.5	87.4	88.9	95.1	94.9	95.0	**100.8**	✗	✗
YOLO-DocLayout (2024c)	20M	90.9	87.1	89.0	95.5	94.4	95.0	55.2	✗	✗
DETR (2020)	41M	84.7	87.1	85.8	92.2	92.0	92.1	17.6	✓	✗
DETR-Deformable (2020)	41M	**91.6**	87.1	**89.3**	96.2	95.9	96.0	18.8	✓	✗
DocFusion(Ours)	289M	88.9	**89.3**	89.1	**96.8**	**96.2**	**96.4**	11.4	✓	✓

Table 3: The performance of the models on DLA, where DocLayNet-Scientific refers to the scientific document subset of DocLayNet. **~~NMS~~** indicates that Non-Maximum Suppression is not required, while **~~Conf~~** means no confidence adjustment is needed. *Note:* The results of DETR and YOLO-series models in this table are computed at multiple confidence levels, with the highest F1 score selected as the final result.

4.4 Implementation Details

We conducted our experiments using the PyTorch framework on eight NVIDIA H100 GPUs, with an initial learning rate of 1e-5, a per-GPU batch size of 12, and employing a cosine learning rate scheduler to progressively adjust the model parameters.

4.5 Main Results

4.5.1 MER performance

We use the open-source UniMER-1M (Wang et al., 2024b) as the evaluation dataset to assess the performance on MER. Since DocFusion is specifically designed for processing printed documents, the primary evaluation focuses on the Simple Printed Expression (SPE) and Complex Printed Expression (CPE) subsets of UniMER-1M. As shown in Table 2, DocFusion performs exceptionally well across multiple evaluation metrics, particularly in CSR and ExpRate. Notably, its ExpRate surpasses the second-ranked UniMERNet by 5.2%, demonstrating superior reliability in real-world document parsing. The results presented here merge the performance of both SPE and CPE, with detailed separate results and handwritten expressions provided in B.1.

4.5.2 TR performance

We selected DocGenome (Xia et al., 2024) as the evaluation dataset because it offers a comprehensive collection of 500K scientific documents across various disciplines, covering a wide range of document-oriented tasks. From this dataset, we extracted 3,000 LaTeX-based table samples as the

Train Dataset	OCR		MER		TR		DLA
	BLEU↑	EditDis↓	CDM↑	CSR_{MER}↑	F1↑	CSR_{TR}↑	F1↑
Task-Specific	96.7	2.2	98.5	99.8	91.2	92.7	87.8
OCR+DLA	96.1	2.4	-	-	-	-	**88.9**
OCR+MER+TR	**97.1**	**1.8**	**98.9**	**99.9**	**92.3**	**94.6**	-

Table 4: Ablation experiments on task collaboration, comparison of task performance when using **Task-specific** training, where each task is trained independently, and other joint multi-task strategies.

test set. Using LatexNodes2Text, we extracted the content of each table cell to compute F1 scores. As shown in Table 2, DocFusion excels on this benchmark, achieving F1 scores that surpass those of the second-ranked model by 1.6%, while having less than one-third of its parameter count. ***Note:*** In this work, in order to maintain consistency with MER and explore multi-task collaboration, we also chose Latex as the output format for our TR task. However, in the past, Latex was not mainstream in Table-to-Sequence tasks, so there are fewer models available for comparison. To provide more comprehensive reference information, we have included the F1 scores of other models that output in HTML in the appendix B.2.

4.5.3 OCR performance

As mentioned in 4.1, DocLayNet (Pfitzmann et al., 2022) supports not only DLA but also OCR evaluation. We selected 3,000 English image samples from the dataset as the test set. As shown in Table 2, DocFusion achieves exceptional performance in both BLEU and EditDis. This outstanding result is primarily attributed to DocFusion's joint training on three recognition tasks, which enhances its efficiency and effectiveness in handling complex document structures.

4.5.4 DLA performance

We use the test set from DocLayNet(Pfitzmann et al., 2022) to evaluate the DLA task. In terms of FPS, while DocFusion exhibits a slight disadvantage in processing speed, it offers an out-of-the-box solution that eliminates the need for hyperparameter tuning in practical applications. This enables the model to achieve optimal performance directly, without requiring further adjustments, thereby compensating for its lower speed.

Regarding accuracy, DocFusion generates layout element labels and coordinates by sequentially predicting tokens without relying on confidence scores. Given that the commonly used Average Precision (AP) metric in object detection is based on confidence scores, it is not directly applicable in this evaluation. To ensure a fair comparison with confidence-based models, we adopt an alternative evaluation methodology. Specifically, for these models, we compute Precision, Recall, and F1-score at various thresholds and select the maximum F1-score across all thresholds as the final evaluation metric. As shown in Table 3, DocFusion demonstrates strong performance in the domain of scientific document detection.

4.6 Ablation Study

4.6.1 OCR-Driven Enhancement of DLA

This section explores the impact of OCR on DLA performance. As shown in Table 4, the results in the DLA column from the first and second rows indicate that adding the OCR task improves DLA performance, with an F1 increase of up to 1.3%. This result demonstrates the effectiveness of using textual information in joint training. Compared to independent training that relies only on visual features, OCR significantly enhances the model's robustness. For example, tables and mathematical expressions have distinct visual features, which can often be effectively recognized by DLA models. In contrast, text or titles have less distinctive visual features, making it challenging to predict their labels based on visual information alone. By providing complementary textual information, OCR strengthens the collaboration between visual and semantic features, resulting in better overall performance.

4.6.2 Collaboration of Recognition Tasks

In this section, we explore the collaboration among the recognition tasks OCR, TR, and MER. As shown in Table 4, the experimental results from the first and third rows demonstrate that joint training yields better performance compared to training

Objective Function	OCR		MER		TR		DLA
	BLEU↑	EditDis↓	CDM↑	CSR_{MER}↑	F1↑	CSR_{TR}↑	F1↑
CE	96.5	2.3	97.8	96.5	90.2	89.1	87.9
GK-CEL	**97.4**	**1.8**	**98.7**	**99.8**	**92.1**	**92.5**	**89.1**

Table 5: Ablation analysis of Gaussian-Kernel Cross-Entropy Loss was conducted on the same dataset across four tasks: OCR, MER, TR, and DLA. **CE** represents training with the standard cross-entropy loss, while **GK-CEL** denotes training with Gaussian-Kernel Cross-Entropy Loss.

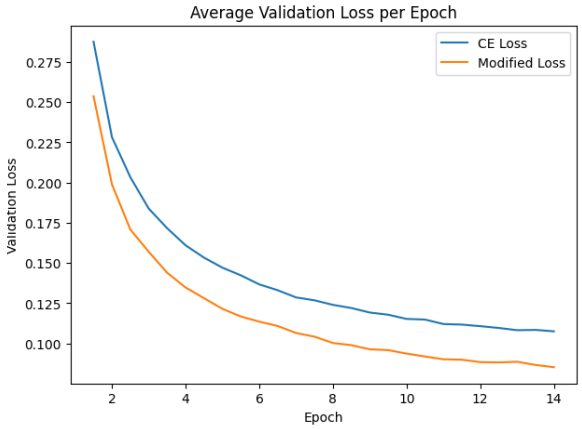


Figure 4: Validation loss curves under identical hyperparameter settings, where the only variation is the choice of the objective function.

each task individually. Specifically, OCR achieves a 0.3% improvement in BLEU score, MER sees increases of 0.4% in CDM and 0.1% in CSR, and TR benefits most significantly, with a 2.1% improvement in F1 score for cell parsing and a 2.0% increase in CSR. This collaboration enables the model to leverage shared information across tasks, enhancing individual task performance and improving overall document parsing capabilities. These results demonstrate that multi-task collaboration effectively enhances performance by leveraging shared information.

4.6.3 Results of improved objective function

In this section, we compared the original cross-entropy and Gaussian-Kernel Cross-Entropy Loss (Gk-CEL) in recognition and detection tasks. As shown in Table 5, the results demonstrate that Gk-CEL led to significant performance gains across both task categories. In recognition tasks, the BLEU score in the OCR task saw an improvement of 1.8%. Additionally, the CDM metric in the MER task increased by 0.9%, while the F1 score in the TR task rose by 2.1%. Notably, for the CSR metric, which assesses LaTeX compilation success, the MER and TR tasks achieved gains of 3.3% and 3.8%, respectively, highlighting enhanced usability and correctness of the LaTeX outputs. For the detection task, the F1 score of the DLA task increased by 0.34%. This improvement can be attributed to Gk-CEL, which alleviates the issue of coordinate token errors dominating the gradient. By addressing this imbalance, the objective function not only enhances the performance of recognition tasks but also improves the accuracy of predicting layout element categories in the detection task itself. These results collectively show that Gk-CEL effectively addresses key challenges in loss minimization, ensuring that tasks such as DLA can operate within a generative framework. It avoids gradient dominance issues while achieving better task balance in a multi-task learning setup.

5 Conclusion

In this work, we introduced DocFusion, the first approach to integrate the four modules of a document parsing pipeline into a unified model by designing Gaussian-Kernel Cross-Entropy Loss tailored to handle diverse data types across tasks. Our method achieved SOTA performance on multiple benchmarks. To enable downstream applications, we re-annotated the widely used DocLayNet dataset and constructed a large-scale formula-to-LaTeX dataset, applying a unified standardization process. Through detailed analysis, we observed that DocFusion, as a lightweight model with fewer parameters, effectively integrates multiple tasks into a single framework, demonstrating both efficiency and versatility in handling complex document parsing challenges. In the future, we aim to extend DocFusion to larger models and further improve dataset standardization to enhance its performance and applicability across broader tasks and domains.

Limitations

While this study primarily focuses on three recognition tasks using standard PDF screenshots, we have enhanced the model's generalization capabilities by incorporating handwritten mathematical expressions. However, the model still has limitations in handling handwritten or other non-standard table formats. For the detection task, although the model demonstrates competitive performance in both accuracy and usability, its processing speed presents challenges for real-time or high-throughput applications. This highlights the need for further optimization in computational efficiency to better meet diverse application demands.

References

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 1(2):3.

Hangbo Bao, Li Dong, and Furu Wei. 2021. Beit: Bert pre-training of image transformers. *Cornell University - arXiv,Cornell University - arXiv*.

Lukas Blecher. 2022. pix2tex - latex ocr. Accessed: 2024-02-29, cited in pages 1, 2, 3, 7, 10, 11.

Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. 2023. Nougat: Neural optical understanding for academic documents. *Preprint*, arXiv:2308.13418.

Michal Bušta, Lukàš Neumann, and Jirí Matas. 2017. Deep textspotter: An end-to-end trainable scene text localization and recognition framework. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2223–2231.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. *Preprint*, arXiv:2005.12872.

Jingye Chen, Bin Li, and Xiangyang Xue. 2021. Scene text telescope: Text-focused scene image super-resolution. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. 2022. Davit: Dual attention vision transformers. *Preprint*, arXiv:2204.03645.

Sergey Filimonov. 2024. Openparse.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Yongshuai Huang, Ning Lu, Dapeng Chen, Yibo Li, Zecheng Xie, Shenggao Zhu, Liangcai Gao, and Wei Peng. 2023. Improving table structure recognition with visual-alignment sequential coordinate modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11134–11143.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking (2022). *arXiv preprint arXiv:2204.08387*.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Structgpt: A general framework for large language model to reason over structured data. *Preprint*, arXiv:2305.09645.

Rahima Khanam and Muhammad Hussain. 2024. Yolov11: An overview of the key architectural enhancements. *Preprint*, arXiv:2410.17725.

Anh Duc Le, Bipin Indurkhya, and Masaki Nakagawa. 2019. Pattern generation strategies for improving recognition of handwritten mathematical expressions. *Pattern Recognition Letters*, 128:255–262.

V.I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Proceedings of the USSR Academy of Sciences,Proceedings of the USSR Academy of Sciences*.

Bohan Li, Ye Yuan, Dingkang Liang, Xiao Liu, Zhilong Ji, Jinfeng Bai, Wenyu Liu, and Xiang Bai. 2022. When counting meets hmer: Counting-aware network for handwritten mathematical expression recognition.

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. 2024. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *Preprint*, arXiv:2407.07895.

Jerry Liu. 2024. Llamaparse.

Yuliang Liu, Biao Yang, Qiang Liu, Zhang Li, Zhiyin Ma, Shuo Zhang, and Xiang Bai. 2024. Textmonkey: An ocr-free large multimodal model for understanding document. *Preprint*, arXiv:2403.04473.

Renqing Luo and Yuhan Xu. 2024. Mixtex: Unambiguous recognition should not rely solely on real data. *Preprint*, arXiv:2406.17148.

Zhiming Mao, Haoli Bai, Lu Hou, Jiansheng Wei, Xin Jiang, Qun Liu, and Kam-Fai Wong. 2024. Visually guided generative text-layout pre-training for document intelligence. *arXiv preprint arXiv:2403.16516*.

Erik G Miller and Paul A Viola. 1998. Ambiguity and constraint in mathematical expression recognition. In *AAAI/IAAI*, pages 784–791.

Lamia Mosbah, Ikram Moalla, Tarek M. Hamdani, Bilel Neji, Taha Beyrouthy, and Adel M. Alimi. 2024. Adocrnet: A deep learning ocr for arabic documents recognition. *IEEE Access*, 12:55620–55631.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*.

Vik Paruchuri. 2023. Texify. Accessed: 2024-02-29, cited in pages 1, 2, 4, 6, 7.

Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S Nassar, and Peter W J Staar. 2022. Doclaynet: A large human-annotated dataset for document-layout segmentation. page 3743–3751.

Shah Rukh Qasim, Hassan Mahmood, and Faisal Shafait. 2019. Rethinking table recognition using graph neural networks. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 142–147. IEEE.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2023. Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking. *Preprint*, arXiv:2110.07367.

R. Smith. 2007. An overview of the tesseract ocr engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 629–633.

Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. 2024a. Yolov10: Real-time end-to-end object detection. *Preprint*, arXiv:2405.14458.

Bin Wang, Zhuangcheng Gu, Guang Liang, Chao Xu, Bo Zhang, Botian Shi, and Conghui He. 2024b. Unimernet: A universal network for real-world mathematical expression recognition. *Preprint*, arXiv:2404.15254.

Bin Wang, Fan Wu, Linke Ouyang, Zhuangcheng Gu, Rui Zhang, Renqiu Xia, Bo Zhang, and Conghui He. 2024c. Cdm: A reliable metric for fair and accurate formula recognition evaluation. *Preprint*, arXiv:2409.03643.

Haoran Wei, Chenglong Liu, Jinyue Chen, Jia Wang, Lingyu Kong, Yanming Xu, Zheng Ge, Liang Zhao, Jianjian Sun, Yuang Peng, Chunrui Han, and Xiangyu Zhang. 2024. General ocr theory: Towards ocr-2.0 via a unified end-to-end model. *Preprint*, arXiv:2409.01704.

Christoph Wick and Frank Puppe. 2018. Fully convolutional neural networks for page segmentation of historical document images. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*.

Renqiu Xia, Song Mao, Xiangchao Yan, Hongbin Zhou, Bo Zhang, Haoyang Peng, Jiahao Pi, Daocheng Fu, Wenjie Wu, Hancheng Ye, et al. 2024. Docgenome: An open large-scale scientific document benchmark for training and testing multi-modal large language models. *arXiv preprint arXiv:2406.11633*.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. 2023. Florence-2: Advancing a unified representation for a variety of vision tasks (2023). *URL https://arxiv. org/abs/2311.06242*.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. 2022. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. *Preprint*, arXiv:2012.14740.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang, Qin Jin, Liang He, Xin Alex Lin, and Fei Huang. 2023. Ureader: Universal ocr-free visually-situated language understanding with multimodal large language model. *Preprint*, arXiv:2310.05126.

Zhichang Yu. 2024. Deepdoc.

Ye Yuan, Xiao Liu, Wondimu Dikubab, Hui Liu, Zhilong Ji, Zhongqin Wu, and Xiang Bai. 2022. Syntax-aware network for handwritten mathematical expression recognition. *arXiv preprint arXiv:2203.01601*.

Hang Zhang, Yeyun Gong, Yelong Shen, Jiancheng Lv, Nan Duan, and Weizhu Chen. 2022. Adversarial retriever-ranker for dense text retrieval. *Preprint*, arXiv:2110.03611.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2024a. A survey of large language models. *Preprint*, arXiv:2303.18223.

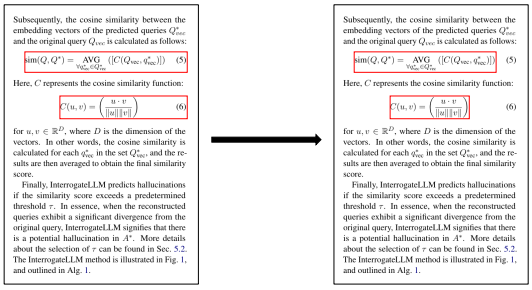
Xiaomeng Zhao, Kaiwen Liu, and Bin Wang. 2024b. Deepdoc.

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

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. 2020. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint*.

A Details of Datasets

A.1 DLA Dataset Reconstruction



Subsequently, the cosine similarity between the embedding vectors of the predicted queries Q^*_{vec} and the original query Q_{vec} is calculated as follows:

$$\text{sim}(Q, Q^*) = \underset{\forall q^*_{\text{vec}} \in Q^*_{\text{vec}}}{\text{AVG}} \left(\{ C(Q_{\text{vec}}, q^*_{\text{vec}}) \} \right) \quad (5)$$

Here, C represents the cosine similarity function:

$$C(\mathbf{u}, \mathbf{v}) = \left(\frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} \right) \quad (6)$$

for $\mathbf{u}, \mathbf{v} \in \mathbb{R}^D$, where D is the dimension of the vectors. In other words, the cosine similarity is calculated for each q^*_{vec} in the set Q^*_{vec}, and the results are then averaged to obtain the final similarity score.

Finally, InterrogateLLM predicts hallucinations if the similarity score exceeds a predetermined threshold τ. In essence, when the reconstructed queries exhibit a significant divergence from the original query, InterrogateLLM signifies that there is a potential hallucination in A^*. More details about the selection of τ can be found in Sec. 5.2. The InterrogateLLM method is illustrated in Fig. 1, and outlined in Alg. 1.

Subsequently, the cosine similarity between the embedding vectors of the predicted queries Q^*_{vec} and the original query Q_{vec} is calculated as follows:

$$\text{sim}(Q, Q^*) = \underset{\forall q^*_{\text{vec}} \in Q^*_{\text{vec}}}{\text{AVG}} \left(\{ C(Q_{\text{vec}}, q^*_{\text{vec}}) \} \right) \quad (5)$$

Here, C represents the cosine similarity function:

$$C(\mathbf{u}, \mathbf{v}) = \left(\frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} \right) \quad (6)$$

for $\mathbf{u}, \mathbf{v} \in \mathbb{R}^D$, where D is the dimension of the vectors. In other words, the cosine similarity is calculated for each q^*_{vec} in the set Q^*_{vec}, and the results are then averaged to obtain the final similarity score.

Finally, InterrogateLLM predicts hallucinations if the similarity score exceeds a predetermined threshold τ. In essence, when the reconstructed queries exhibit a significant divergence from the original query, InterrogateLLM signifies that there is a potential hallucination in A^*. More details about the selection of τ can be found in Sec. 5.2. The InterrogateLLM method is illustrated in Fig. 1, and outlined in Alg. 1.

Figure 5: The corresponding numbers were removed from the annotated data for mathematical expression detection.

In DocLaynet and other similar datasets, the annotation of mathematical formulas has certain limitations, as show in figure 5, the content of math expression and numbering are typically annotated within the same bounding box. This annotation approach introduces noise in subsequent Mathematical Expression Recognition (MER) tasks.

To address this issue, we extracted formulas from arXiv LaTeX source files using regular expressions and assigned unique colors and bounding boxes to each element. Then, we employed a fuzzy matching algorithm to ensure annotation accuracy and eliminate overlaps. Finally, we trained a lightweight detection model and, combined with manual verification, re-annotated pages containing formulas. These improvements significantly enhance the dataset's applicability to subsequent MER tasks.

A.2 MER and TR Dataset

MER Dataset. The UniMER-1M (Wang et al., 2024b) has significantly advanced MER research but contains many redundant spaces in LaTeX code. Although some spaces are syntactically necessary, most are unnecessary, increasing output length and computational overhead. To address this, we constructed a new dataset by extracting content from LaTeX files, normalizing style variations and verifying accuracy through re-rendering. Models trained on our dataset produce LaTeX code that is approximately 34.2% shorter for complex expressions and 37.5% shorter for simple expressions on the UniMER-1M test set, demonstrating improved efficiency.

TR Dataset. In the TR task of DocFusion, we adopted LaTeX as the output format for two main reasons: (1) to ensure consistency with the MER task's output format, enabling better multi-task collaboration; and (2) because LaTeX facilitates both the extraction of cell content and the restoration of the original table layout. Existing LaTeX-based TR datasets either lack sufficient scale or fail to separate tables from captions, conflicting with our DLA task. To overcome these limitations, we constructed a high-quality TR dataset with 100K samples by following a similar approach to the MER dataset.

A.3 Latex-based data standardization

Issue	Original	Standardized
Bracket	`\{`	`\lbrace`
Subsup	`a^1_2`	`a_2^1`
Prime	a'	`a^{\prime}`
Fraction	`\over`	`\frac`
Space	`\tabular{l c}`	`\tabular{lc}`

Table 6: Examples of LaTeX standardization for various symbols and expressions.

We chose to standardize the output format as LaTeX for two recognition tasks involving non-plain-text elements. For MER, converting to LaTeX was essential as it provides a precise representation of mathematical formulas. For TR, in addition to ensuring format consistency, converting to LaTeX also allows for the restoration of the original content through compilation, and enables the extraction of cell elements using tools such as LatexNodes2Text, thus enhancing processing flexibility.

We used regular expressions to extract relevant content from the LaTeX source files of research papers. However, due to variations in author writing styles, the same formula or table may appear in multiple forms, increasing the complexity of training. As show in table 6 , we analyzed these different representations, standardized them to eliminate ambiguities and ensured consistency. To verify the accuracy of the standardized LaTeX code, we re-rendered it into images, creating a high-quality dataset that aligns with the actual input-output content.

Model	size	SPE			CPE			HWE		
		CDM↑	ExpRate↑	CSR↑	CDM↑	ExpRate↑	CSR↑	CDM↑	ExpRate↑	CSR↑
Pix2tex (2022)	-	92.1	59.0	99.8	45.2	7.2	88.1	24.7	8.1	16.3
Texify (2023)	312M	98.7	89.8	99.8	69.8	35.6	94.3	49.9	21.3	25.8
GOT (2024)	580M	95.0	82.7	98.6	73.3	36.4	96.4	31.2	17.7	10.2
UniMERNet (2024b)	325M	**99.7**	95.6	**99.9**	**97.6**	77.4	99.2	**94.7**	65.3	98.1
DocFusion(Ours)	289M	**99.7**	**97.3**	**99.9**	96.9	**88.1**	**99.5**	94.1	**72.1**	**99.3**

Table 7: Supplementary details of **MER**. **SPE** refers to simple printed mathematical expressions, **CPE** refers to complex printed mathematical expressions, and **HWE** refers to handwritten mathematical expressions.

B Other supplementary experiments

B.1 Details of MER Performance

we provide a detailed presentation of the main experimental results for MER, showing the performance of the relevant models on simple, complex, and non-standard handwritten mathematical expressions. For specifics, please refer to Table 7.

B.2 Other Table-to-Sequence Method

Methods	F1	CSR
surya	37.4	-
ppstructure_table	78.1	-
Deepdoctection	53.7	-
RapidTable	87.9	-
MixTex	46.2	27.4
GOT	86.9	81.6
StructEqTable	90.6	93.2
DocFusion	92.1	92.5

Table 8: Due to differences in the method of extracting cell contents, the fairness of the experiment cannot be guaranteed, therefore, it is provided for reference only.

This study aims to explore multi-task collaboration, and therefore, the TR task also adopts Latex as the output format to maintain consistency with MER. However, Latex has not been the mainstream approach for TR tasks in recent times, resulting in a limited number of TR models available for comparison in the main experiment. To address this limitation, we incorporated other methods based on HTML as the output format. **However, due to differences in sequence extraction methods, ensuring a fair comparison is challenging. Therefore, we have included the supplementary experimental results in the appendix for reference.**

C Other optimization methods

The challenge of this experiment lies in effectively optimizing continuous coordinate-type data within a discrete generative framework. Since there are inherent errors in coordinate annotations, these errors are further amplified when training the generative framework using cross-entropy loss, especially when the framework performs multiple tasks, which exacerbates the issue. To address this problem, in addition to the Gaussian-Kernel Cross-Entropy Loss introduced in the main text, we employed several other optimization strategies, including the basic adjustments of data ratios or loss weights, as well as using soft-argmax to continuously map discrete coordinate tokens.

C.1 Hyperparameters Adjustment Strategies

The root cause of the training difficulty lies in the fact that the discrete coordinate tokens do not effectively dominate the loss during training, leading to poor gradient propagation and inefficient parameter updates. To address this, one possible solution is to adjust the data ratios or the loss weights across different task types. However, while this approach can improve training stability to some extent, it is overly engineering-driven and does not fundamentally solve the underlying issue of inadequate gradient flow caused by the discrete nature of the coordinate tokens.

C.2 Soft-argmax Strategies

The core issue lies in the fact that while multi-task frameworks need to be discrete, coordinates are inherently continuous. A natural solution to this problem is to "smooth" the coordinate loss, effectively making it continuous. This approach offers an intuitive way to handle the challenge, and we primarily use the soft-argmax technique to obtain the position coordinates while maintaining the gradient flow, followed by the computation of the loss via Mean Squared Error (MSE).

```python
import torch
import torch.nn as nn
import torch.nn.functional as F

class CustomLoss(nn.Module):
    def soft_argmax_with_auto_beta(self,logits, initial_beta=1e+48, min_beta=1e-1, decay_factor=0.1):
        beta = initial_beta

        while beta >= min_beta:
            softmax_probs = F.softmax(logits * beta, dim=-1)
            indices = torch.arange(logits.size(-1), device=logits.device).float()
            soft_argmax_result = torch.sum(softmax_probs * indices, dim=-1)
            if not torch.isnan(soft_argmax_result).any():
                return soft_argmax_result
            beta *= decay_factor
            print(f"NaN detected. Reducing beta to {beta}")
        print("Warning: Could not compute valid result. Returning NaN.")
        return torch.full_like(logits[..., 0], float('nan'))

    def cood_loss(self,logits,labels):
        coord_start, coord_end = 50269, 51268
        b,s,v = logits.size()
        assert (b, s) == labels.shape, f"Shape mismatch: logits has shape ({b}, {s}, {v}), but labels has shape {labels.shape}"
        preds = self.soft_argmax_with_auto_beta(logits)
        print(preds)
        coord_mask = (labels >= coord_start) & (labels <= coord_end) & (preds >= coord_start) & (preds <= coord_end)
        token_loss = F.cross_entropy(logits.view(-1,v),labels.view(-1),reduction='none')
        token_loss = token_loss.view(b,s)*(~coord_mask)
        mse_loss = F.mse_loss(preds.view(-1).float(),labels.view(-1).float(),reduction='none')

        token_loss_max = token_loss.view(-1)[token_loss.view(-1).argmax(-1)]
        masked_token_loss = token_loss.clone()
        masked_token_loss[masked_token_loss == 0] = float('inf')
        token_loss_min = masked_token_loss.view(-1)[masked_token_loss.view(-1).argmin()]

        mse_loss = ((mse_loss/1e+6)*(token_loss_max-token_loss_min)+token_loss_min).view(b,s)*(coord_mask)

        loss = token_loss+mse_loss
        mean_loss = torch.mean(loss)

        return mean_loss

    def forward(self,logits,labels):
        return self.cood_loss(logits,labels)
```

Figure 6: Soft-argmax Loss

However, the difficulty arises during multi-task training: after calculating the MSE, we need to ensure that it remains within the same range as other cross-entropy (CE) losses. The challenge here is to maintain balance and prevent the MSE loss from overwhelming the CE losses. Moreover, if the hyperparameters of the soft-argmax are not set appropriately, it can easily lead to gradient explosion during training, further complicating the optimization process.

Although this method aims to address the issue at its core by making the coordinate loss continuous, it still relies heavily on the correct setting of hyperparameters. Furthermore, it presents generalization issues when applied to different tasks or datasets. In comparison, the Gaussian-Kernel Cross-Entropy Loss (GK-CEL) offers a more robust solution, as it reduces the dependency on hyperparameters while improving generalization performance.

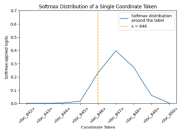


Figure 3: The Softmax distribution of logits for a target token and its neighboring tokens after the loss has stabilized.

embeddings $\mathbf{V} \in \mathbb{R}^{N_v \times D_v}$. These embeddings are transformed for compatibility with task-specific prompt embeddings $\mathbf{T}\text{prompt} \in \mathbb{R}^{N_t \times D}$. The combined input $\mathbf{X} = [\mathbf{V}; \mathbf{T}\text{prompt}]$ is then passed to the Transformer decoder to generate predictions. By integrating Dual Attention, coordinate quantization, and optimizing its architecture, DocFusion efficiently handles complex document parsing tasks with high precision and computational efficiency.

3.2 Challenges and Motivations

While representing object detection as text regression enables joint training of layout analysis and page element recognition under a unified cross-entropy-based framework, it inherently forces continuous coordinates into discrete token spaces. This mismatch creates several challenges, especially in fine-tuning small coordinate adjustments, where the model struggles to produce accurate gradients, reducing training stability. As shown in Figure 3, small unavoidable deviations in coordinate labels smooth out the softmax distribution, preventing the target token's probability from forming a sharp peak. This makes it harder for the model to escape local optima and limits its learning capacity. Additionally, traditional cross-entropy loss, which is designed for discrete classification tasks, does not handle continuous changes well, further increasing inaccuracies during training.

In multi-task settings, these issues become even more challenging. The conflict between discrete loss functions and continuous coordinate optimization can skew gradients, causing one task to dominate at the cost of others. This imbalance reduces performance in other tasks and harms the model's ability to predict coordinates accurately, limiting its overall effectiveness in complex document parsing tasks. Solving these problems is critical to improving both localization accuracy and training stability across tasks.

3.3 Objective function

To address these challenges, we propose an improved objective function that applies a one-dimensional convolution over the probability distribution, refining the model's sensitivity to small coordinate changes while preserving the discrete treatment of cross-entropy. This approach helps alleviate the mismatch between discrete tokens and continuous coordinates, improves gradient quality, and prevents the coordinate prediction task from dominating the optimization process. In doing so, it enhances localization accuracy, supports stable multi-task training, and achieves better alignment with the desired properties identified in the motivating considerations.

Let the model's output logits be denoted as $\mathbf{Z} \in \mathbb{R}^{B \times L \times V}$, where B is the batch size, L is the sequence length, and V is the vocabulary size. The target labels are denoted as $\mathbf{T} \in \mathbb{N}^{B \times L}$. The range of indices corresponding to coordinate tokens is defined as $[s, e]$, representing their positions in the vocabulary.

The standard softmax probability distribution is first computed as:

$$\mathbf{P} = \text{softmax}(\mathbf{Z}) \tag{1}$$

A mask is then applied to zero out probabilities outside the range $[s, e]$, creating a modified probability tensor $\mathbf{P}'$:

$$\mathbf{P}'_{ijk} = \begin{cases} \mathbf{P}_{ijk}, & \text{if } k \in [s, e] \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Next, a one-dimensional convolution kernel $\mathbf{K} \in \mathbb{R}^{1 \times 1 \times k}$ is constructed based on a Gaussian distribution, where k is the kernel size (an odd integer greater than 1), and σ is the standard deviation. The kernel weights are computed as:

$$\mathbf{K}_i = \exp\left(-\frac{(i - \frac{k-1}{2})^2}{2\sigma^2}\right) \tag{3}$$

The kernel is then applied to $\mathbf{P}'$ via one-dimensional convolution along the vocabulary dimension:

$$\mathbf{C} = \text{conv1d}(\mathbf{P}', \mathbf{K}, \text{padding} = \frac{k-1}{2}) \tag{4}$$

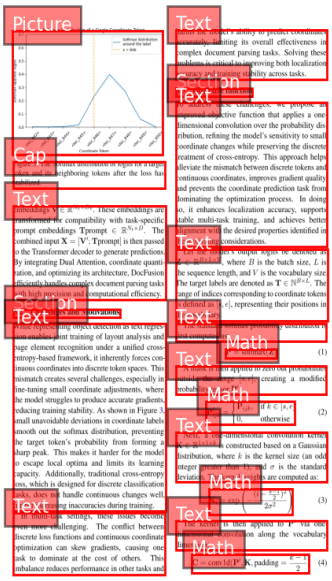


Figure 7: DLA Effect Presentation

Input image:

$$\|\mathcal{S}_{\zeta+i\rho_{q_1}}(f)\|^{p'}_{L^{(q_1,p')}(Y,d\mu_0)}$$
$$= C\int_{\mathbb{R}}\left(\int_K \left|\widetilde{f}(\lambda+i\rho_{q_1},k)\right|^{q_1}\frac{|\lambda+i\rho_{q_1}|^{q_1}}{|\lambda+i\rho_{q_1}+i2\rho|^{q_1}}\,dk\right)^{\frac{p'}{q_1}}(1+|\lambda-\zeta|)^{n-1}d\lambda.$$

Rendered Output Effect:

$$\|\mathcal{S}_{\zeta+i\rho_{q_1}}(f)\|^{p'}_{L^{(q_1,p')}(Y,d\mu_0)}$$
$$= C\int_{\mathbb{R}}\left(\int_K \left|\widetilde{f}(\lambda+i\rho_{q_1},k)\right|^{q_1}\frac{|\lambda+i\rho_{q_1}|^{q_1}}{|\lambda+i\rho_{q_1}+i2\rho|^{q_1}}\,dk\right)^{\frac{p'}{q_1}}(1+|\lambda-\zeta|)^{n-1}d\lambda.$$

Figure 8: MER Effect Presentation

Input image:

Model	Size	DocLayNet			DocLayNet-Scientific			FPS↑
		Precision↑	Recall↑	F1↑	Precision↑	Recall↑	F1↑	
DETR (2020)	41M	87.1	**91.6**	**89.3**	95.9	**96.2**	96.0	3.7
DocLayout-YOLO (2024c)	20M	86.7	91.1	88.9	94.4	95.5	95.0	**85.2**
DocFusion	289M	**88.0**	88.4	88.2	**96.8**	**96.2**	**96.4**	7.5

Rendered Output Effect:

Model	Size	DocLayNet			DocLayNet-Scientific			FPS↑
		Precision↑	Recall↑	F1↑	Precision↑	Recall↑	F1↑	
DETR (2020)	41M	87.1	**91.6**	**89.3**	95.9	**96.2**	96.0	3.7
DocLayout-YOLO (2024c)	20M	86.7	91.1	88.9	94.4	95.5	95.0	**85.2**
DocFusion	289M	**88.0**	88.4	88.2	**96.8**	**96.2**	**96.4**	7.5

Figure 9: TR Effect Presentation