Efficient Document Parsing via Parallel Token Prediction

Lei Li[1] Ze Zhao[1] Meng Li[1,2] Zhongwang Lun Yi Yuan[1] Xingjing Lu[1]
Zheng Wei[1†] Jiang Bian[1] Zang Li[1]
[1]Platform and Content Group, Tencent [2]Renmin University of China
arleyli@tencent.com, hemingwei@tencent.com

Abstract

*Document parsing, as a fundamental yet crucial vision task, is being revolutionized by vision-language models (VLMs). However, the autoregressive (AR) decoding inherent to VLMs creates a significant bottleneck, severely limiting parsing speed. In this paper, we propose **Parallel-Token Prediction (PTP)**, a plugable, model-agnostic and simple-yet-effective method that enables VLMs to generate multiple future tokens in parallel with improved sample efficiency. Specifically, we insert some learnable tokens into the input sequence and design corresponding training objectives to equip the model with parallel decoding capabilities for document parsing. Furthermore, to support effective training, we develop a comprehensive data generation pipeline that efficiently produces large-scale, high-quality document parsing training data for VLMs. Extensive experiments on OmniDocBench and olmOCR-bench demonstrate that our method not only significantly improves decoding speed (1.6×-2.2×) but also reduces model hallucinations and exhibits strong generalization abilities.*

1. Introduction

Document parsing, also known as document content extraction [44], aims to transform unstructured or semi-structured documents into structured, machine-readable outputs. This process involves accurately identifying and reconstructing diverse elements including text, images, formulas, and tables while preserving their logical ordering and hierarchical relationships as presented in the original documents. As a cornerstone task in multimodal understanding, document parsing plays a critical role in enabling advanced applications such as Retrieval-Augmented Generation (RAG) [19, 43], document analysis [2, 35], and data management [3, 41], establishing a solid foundation for enabling machines to comprehend the digital world.

Early document parsing methods predominantly adopted pipeline-based approaches [8, 26, 28, 38], which decomposed the task into sequential modules, suffering from error accumulation and limited end-to-end optimization. With recent advances in Vision-Language Models (VLMs), an increasing number of methods have begun leveraging VLMs to revolutionize the document parsing task, either through end-to-end generation [4, 16, 28, 41] or by integrating VLMs into specific pipeline stages [6, 12, 21, 24] for improved multi-element recognition.

However, as a real-world application-oriented task, document parsing demands not only high accuracy but also efficient processing speed, particularly for large-scale deployment scenarios. While VLMs have achieved remarkable improvements in parsing quality, their inherent autoregressive (AR) generation mechanism with next-token prediction (NTP) introduces a singnificant efficiency bottleneck. Recent efforts have explored various optimization strategies to accelerate VLM-based parsing, including output sequence compression [24], visual token reduction [41], and model parameter pruning [34]. Despite these advances, **the sequential generation paradigm remains the inherent bottleneck**, as the autoregressive decoding process leading to substantial latency that grows proportionally with document complexity and content density. Considering that the essence of OCR tasks lies in accurate transcription rather than semantic understanding, we can naturally decompose an image into multiple patches and perform parallel content recognition. This raises a natural question: *Can this parallel recognition capability be inherently embedded within the model itself?* To address this challenge, we propose **Parallel Token Prediction (PTP)**, a novel training and inference framework that breaks the sequential generation bottleneck by enabling models to produce multiple tokens per decoding step. Specifically, we insert some learnable register tokens [9, 13] into training sequences and optimize them to predict future tokens based on their positions. During inference, by appending N special tokens to the input sequence, the model generates N tokens in parallel within each decoding step, achieving theoretical N-fold acceler-

† Corresponding author.

Code is available at https://github.com/flow3rdown/PTP-OCR.

ation. Extensive experiments on the OmniDocBench [25] dataset validate that PTP delivers significant throughput improvements while preserving model accuracy: PTP-1 attains 1.6× throughput over the NTP baseline, and PTP-2 achieves 2.2× acceleration. Furthermore, we generalize PTP to broader vision-language understanding tasks and synergize it with speculative decoding [17, 32], resulting in an impressive 82% acceptance ratio.

To summarize, our key contributions are encapsulated as follows: (1) We propose Parallel Token Prediction (PTP), a model-agnostic, plugable, and highly efficient acceleration method for document parsing. PTP achieves 1.6×-2.2× throughput improvements without compromising accuracy. (2) We construct a high-quality layout-level document parsing dataset through an automated generation framework that integrates multiple types of VLMs for data annotation, coupled with sophisticated filtering and deduplication strategies to ensure data quality. (3) We conduct comprehensive analyses and ablation studies, validating the effectiveness of PTP and further exploring its potential in vision-language understanding (VLU) scenarios.

2. Related Work

Document Parsing Approaches. Document parsing methods can be broadly categorized into two approaches: **(i) Pipeline-based Approaches:** These methods [26, 38] decompose document parsing into sequential modular tasks, including layout analysis, text, formula and table recognition, and reading order detection, etc. Each module employs a specialized model optimized for its specific task. While enabling fine-grained optimization and interpretability, they suffer from error accumulation across stages and exhibits degraded performance in challenging or domain-specific scenarios. **(ii) VLM-based Approaches:** These methods leverage general or domain-specific vision-language models to replace multiple modular components, thereby simplifying the parsing pipeline. Early works [4, 39, 40] introduce end-to-end OCR-free VLMs that directly parse document images, eliminating error propagation, but remain constrained by scalability and efficiency concerns. Recent approaches [12, 18, 24] adopt a hybrid strategies combining layout analysis with VLM-based recognition. While this strategy effectively leverages both the efficiency of pipeline methods and the accuracy of VLMs, two critical limitations persist: (1) autoregressive decoding inherently limits parsing speed, and (2) the scarcity of large-scale, high-quality training data poses challenges for model development.

Efficient Document Parsing. While autoregressive models improve OCR accuracy and robustness, efficiency remains a critical bottleneck. Existing acceleration approaches can be categorized as follows: **(i) Multi-Token Prediction:** Early works [10, 11, 31] employ non-autoregressive (NAR) vision-language models trained with Connectionist Temporal Classification (CTC) loss to achieve multi-token prediction. However, these methods require complex architectural modifications, exhibit limited performance, and are restricted to span-level OCR tasks, failing to scale to paragraph- or document-level parsing. Recent efforts [14, 20] introduce auxiliary MTP heads to enable multi-token prediction in language models, but their applications to document parsing remain unexplored. **(ii) Sequence Compression:** Recent studies reduce computational cost by shortening input or output sequences. [24] designs compact representation languages for formulas and tables to reduce output tokens, thereby improving throughput. [41] proposes DeepEncoder to compress visual representations, reducing input tokens and accelerating prefill stage. [34] prunes redundant vocabulary tokens to decrease model capacity and decoding overhead. While achieving moderate efficiency gains, these methods do not fundamentally address the autoregressive (AR) decoding bottleneck. In this work, we propose Parallel Token Prediction (PTP), which enables parallel decoding in VLMs without sacrificing performance. PTP is model-agnostic and orthogonal to existing architectures and acceleration techniques, delivering substantial improvements in parsing efficiency.

3. Dataset Engine

Current document parsing and OCR datasets mainly focus on span-level or file-level annotations, with a critical shortage of layout-level data. Moreover, existing datasets exhibit limited diversity in document types and difficulty levels, hindering model generalization to real-world scenarios. To address these limitations, we develop a comprehensive and scalable data collection, annotation and cleaning pipeline, as shown in Fig. 1.

3.1. Data Curation

We begin by constructing a diverse document resource pool comprising 200k pages sourced through three channels: open-source datasets, in-house data, and synthetic generated data. We ensure that each document page is valid and contains parsable elements. To maintain diversity and prevent category imbalance, we train a document classification and difficulty assessment model, which can identify document types (e.g. academic papers, technical reports, hand-writings) and difficulty levels, assisting us in controlling the distribution and achieving balanced representation across categories. More details in Supplementary Materials.

3.2. Data Annotation

We employ a layout analysis model [33] to partition each document page into layout-based sub-regions (e.g., text paragraphs, tables, figures) to construct layout-level data. To ensure the quality, we filter out sub-images that are too small, too large, or contain incomplete information due

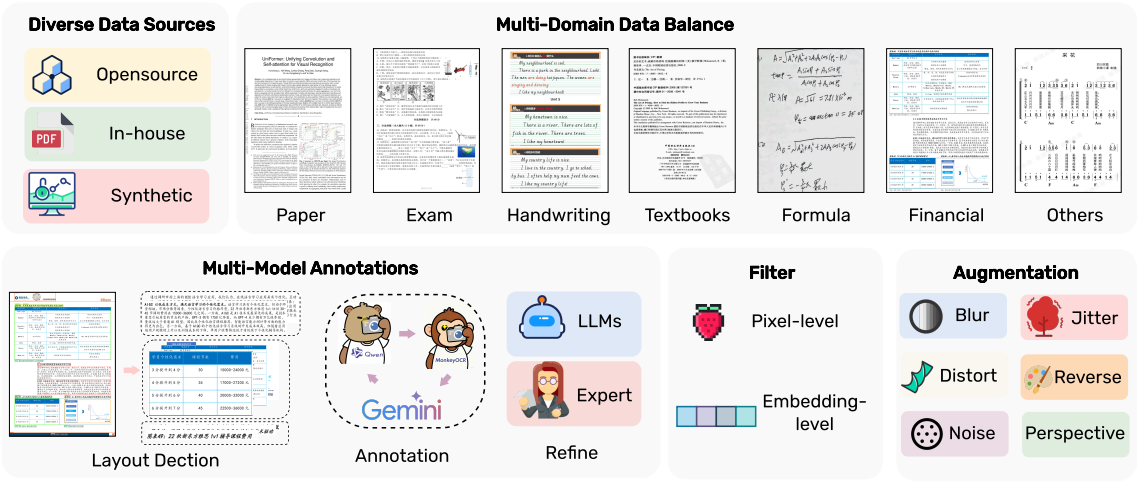

Figure 1. An illustration of the data creation pipeline.

to boundary truncation. Then we develop a multi-model collaborative annotation strategy that leverages three types of models, including a strong frontier VLM [7], an open-source VLM [3], and a specialized model [18]. Annotations from these models are aggregated through majority voting. The consolidated annotations are then refined via LLM-based post-processing to correct formatting errors, followed by selective manual review to ensure quality in cases with low confidence or high inter-model disagreement.

3.3. Filtering and Statistics

To ensure the quality and diversity of the final dataset, we implement a multi-stage filtering pipeline. We first remove corrupted images and samples with abnormal aspect ratios, which typically indicate scanning errors or improper cropping. To reduce redundancy and enhance diversity, we apply two complementary deduplication strategies: (i) Embedding-based similarity: We compute CLIP [29] image embeddings and identify near-duplicates using cosine similarity to capture semantic-level redundancy; (ii) Perceptual hashing: We apply pHash with Hamming distance to detect visually similar images, capturing pixel-level similarity robust to minor transformations. Through this comprehensive filtering pipeline, 10% of the collected data is removed, yielding a final dataset of 1.8M high-quality samples.

4. Method

4.1. Preliminaries

Next-Token Prediction Next-Token Prediction (NTP) is the core objective of autoregressive vision-language models. Given a vision input X_v, a textual query X_q and the answer X_a, NTP can be formulized as follows:

$$P(x_1, \ldots, x_l) = \prod_i^l P(x_{i+1}|X_v, X_q, X_{a,\leq i}) \quad (1)$$

where l is the length of answer X_a. For a model P_θ and dataset $\mathcal{D}$, the training objective is to minimize the cross-entropy loss:

$$\mathcal{L}_{\text{NTP}} = \mathbb{E}_{\mathcal{D}}[-\sum_i^l \log P_\theta(x_{i+1}|X_v, X_q, X_{a,\leq i})] \quad (2)$$

Multi-Token Prediction [14] proposed Multi-Token Prediction (MTP), which generalizes NTP by predicting multiple future tokens at once, as shown in Fig. 2.

$$\mathcal{L}_{\text{MTP}} = \mathbb{E}_{\mathcal{D}}[-\sum_i^l \log P_\theta(x_{i+1:i+n}|X_v, X_q, X_{a,\leq i})] \quad (3)$$

where n is the number of MTP heads.

4.2. Parallel-Token Prediction

Overview. Document parsing is essentially a high-certainty transcription task rather than an open-ended generation task, where the output is uniquely determined by the input image with minimal semantic ambiguity. Consider an image containing the text "West Cowboy": we can either process the entire image holistically or partition it into segments to separately recognize "West" and "Cowboy", both yielding identical results. This observation reveals an inherent parallelizability in document parsing that remains unexploited in previous works. Building upon this insight, we propose Parallel Token Prediction (PTP), which enables

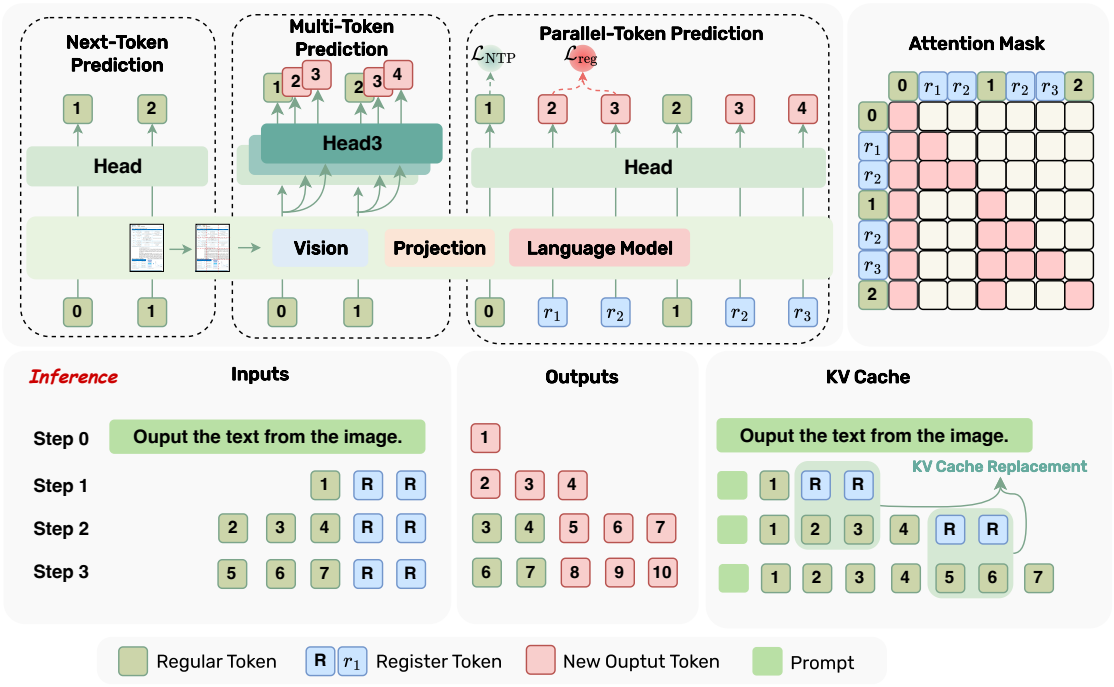

Figure 2. Overview of our method. We show the training architecture and the inference process of our parallel-token prediction method.

models to simultaneously attend to and recognize multiple characters within an image, substantially improving generation efficiency. Specifically, following [9, 13], we introduce a set of learnable continuous tokens, termed *registers*, appended after each token in the training sequence. Each register is trained to predict future tokens based on its relative distance from the preceding context. Through carefully designed training objectives, these registers acquire the capability to perform accurate multi-step-ahead predictions.

Register Tokens. [9] first introduced registers as additional learnable tokens appended to input sequences to store global information and absorb high-norm outlier features. Inspired by this, we repurpose registers for capturing features from distinct regions of the image and predict future tokens in parallel. Notably, all register tokens share the same token ID and learnable embedding, yet through contextual conditioning, they dynamically perform region-specific predictions at different positional offsets.

Training. Given $X_a = (x_1, x_2, \ldots, x_{l-1}, x_l)$ as the answer token sequence to be trained, we insert continuous register tokens after each token (as shown in Fig. 2):

$$\hat{X}_a = (x_1, [r_2, r_3], x_2, [r_3, r_4], \ldots, x_{l-1}, [r_l, r_{l+1}], x_l) \quad (4)$$

where each regular token x_i is augmented with n subsequent continuous register tokens r_j (here $n = 2$). All register tokens share a single learnable embedding but differ in their positional encodings, enabling them to predict future tokens at position-dependent offsets. Specifically, r_{i+1} placed immediately after x_i is trained to predict x_{i+2}, while r_{i+2} predicts x_{i+3} and so on. Accordingly, the shifted training objective corresponding to Eq. 4 becomes:

$$\mathcal{O}_a = (x_2, [x_3, x_4], x_3, [x_4, x_5], \ldots, x_l, [x_l, x_l]) \quad (5)$$

To ensure independent training between regular tokens x_i and register tokens r_i, we modify the causal attention mask to enforce the following constraints: (1) Regular tokens attend only to preceding regular tokens and remain isolated from all register tokens. (2) Register tokens attend to all preceding regular tokens, as well as preceding register tokens within the same group (*i.e.*, register tokens following the same regular token). (3) Register tokens from different groups are mutually isolated and do not interact.

Since our method preserves the original model architecture, we adjust the position IDs of register tokens to enable accurate future token predictoin. Specifically, register token r_i is assigned a position ID equal to its preceding regular token x_{i-1} plus one. Similarly, register token r_{i+1} receives a position ID one greater than r_i. Consequently, the position ID sequence corresponding to Eq. (4) is:

$$\mathcal{P}_a = (1, [2, 3], 2, [3, 4], \ldots, l-1, [l, l+1], l) \quad (6)$$

where we suppose the position id starts from 1.

During training, regular tokens are optimized using the standard NTP loss, while register tokens are optimized with

the following loss:

$$\mathcal{L}_{\text{reg}} = \mathbb{E}_{\mathcal{D}}[-\sum_{i}^{l}\sum_{j}^{n} \log P_\theta(x_{i+j+1}|X_{a,\leq i}, r_{i+j})] \quad (7)$$

Due to our meticulously crafted causal attention mask, regular tokens remain unaffected by register tokens throughout the training process. Finally, the training loss of our PTP approach is defined as:

$$\mathcal{L}_{\text{PTP}} = \alpha * \mathcal{L}_{\text{NTP}} + (1-\alpha) * \mathcal{L}_{\text{reg}} \quad (8)$$

where $\alpha \in (0, 1)$ controls relative weight of each loss term.

4.3. Inference and Analysis

Unlike [13], we do not discard register tokens during inference. Instead, we fully leverage their learned ability to predict future tokens for decoding acceleration. As illustrated in Fig. 2, at each decoding step, we append n additional register tokens after the original input, enabling the model to generate $n+1$ new predictions per step. Subsequently, we can estimate the speedup ratio (SR) as follows:

$$\text{SR} \approx \frac{(1+n) \times L_\theta}{L'_\theta} \quad (9)$$

where L_θ denotes the latency of the model per decode step. L'_θ denotes the latency of a single forward pass processing multiple tokens simultaneously. While this may vary slightly from L_θ due to the hardware, the difference remains negligible when computational resources are sufficient.

Since we only append register tokens at the end of the sequence, our approach fully conforms to the causal LM setting, requiring no modifications to attention masks or positions. The only necessary operation is removing the KV cache corresponding to register tokens after each decoding step. This is because we subsequently perform a forward pass with the tokens predicted by register tokens, which generates more accurate KV cache compared to the speculative register token predictions. Although this approach introduces a slight computational overhead (L_θ vs. L'_θ), it does not impact overall throughput when computational resources are sufficient, since the decoding phase is memory-bound rather than compute-bound. The additional computation is effectively absorbed within memory access latency.

5. Experiments

5.1. Experimental Settings

Datasets & Baselines. We primarily evaluate our method on OmniDocBench [25] and olmOCR-bench [27] document parsing benchmarks, focusing on text recognition and formula recognition performance. OmniDocBench is currently the most widely adopted benchmark for document parsing, designed to assess diverse document understanding in real-world scenarios. It encompasses nine document types, four layout types, and three language types, providing comprehensive coverage of practical document parsing challenges. olmOCR-bench comprises 1,402 PDF documents sourced from various repositories, organized into seven subsets. We mainly compare with three types of methods: pipeline tools [8, 26, 38], general VLMs [1, 3, 7, 45] and specialized VLMs [12, 18, 24, 27, 30].

Implementation Details. Taking into account both performance and effectiveness, we employ the Qwen2.5-VL-3B-Instruct models as our base model and fine-tune it on our constructed dataset. During fine-tuning, we set the max number of register tokens to $n = 3$ and the loss weight to $\alpha = 0.5$. All experiments are conducted on 8 × A100 40GB GPUs for 1 epoch with a learning rate of $2e-5$. All experiments are trained for 1 epoch with a learning rate of $2e-5$. We freeze the vision encoder and aligner parameters, updating only the LLM weights. In all experiments, we denote models trained solely with $\mathcal{L}_{\text{NTP}}$ as $*$-NTP, and models trained with $\mathcal{L}_{\text{PTP}}$ as $*$-PTP-n, where n indicates the number of inserted register tokens during inference.

5.2. Main Results

PTP Enhances Recognition Accuracy. The main performance results of text recognition and formula recognition for all models are shown in Tab. 1, Tab. 2 and Tab. 3, respectively. Firstly, models fine-tuned on our constructed dataset achieve significant performance gains, matching or exceeding many specialized models while using substantially less training data (PTP-0 and NTP). Secondly, when incorporating one register token for parallel inference (PTP-1), the text recognition **performance not only remains intact but further improves**, surpassing other competing methods. This improvement may be attributed to PTP encouraging the model to better leverage surrounding contextual information, thereby reducing hallucinations and producing more accurate predictions. Moreover, although formula recognition involves complex LaTeX syntax reasoning, PTP-1 achieves performance comparable to NTP while significantly accelerating inference.

PTP Improves Throughput. We integrate the PTP implementation into KsanaLLM [36] and evaluate the efficiency of PTP using an H20 (90G) GPU. The results are presented in Fig. 3, we observe that PTP effectively reduces both time per output token (TPOT) and average latency while significant improving decoding throughput. Specifically, **PTP-1 achieves 1.6× speedup over NTP, while PTP-2 attains 2.2× speedup.**

5.3. Analysis

Efficiency Analysis. To comprehensively evaluate the efficiency of our proposed PTP method, we conduct comparative analysis from both training and inference perspectives

Model Type	Models	Slides	Academic Papers	Book	Textbook	Exam Papers	Magazine	News	Notes	Financial Report	Overall(↓)
Pipeline Tools	Marker-1.8.2 [26]	0.1796	0.0412	0.1010	0.2908	0.2958	0.1111	0.2717	0.4656	0.0341	0.1990
	MinerU2-pipeline [38]	0.4244	0.0230	0.2628	0.1224	0.0822	0.3950	0.0736	0.2603	0.0411	0.1872
	PP-StructureV3 [8]	0.0794	0.0236	0.0415	0.1107	0.0945	0.0722	0.0617	0.1236	0.0181	0.0695
General VLMs	GPT-4o [1]	0.1019	0.1203	0.1288	0.1599	0.1939	0.1420	0.6254	0.2611	0.3343	0.2297
	Gemini-2.5 Pro [7]	0.0326	**0.0182**	0.0694	0.1618	0.0937	0.0161	0.1347	0.1169	0.0169	0.0734
	InternVL3-76B [45]	0.0349	0.1052	0.0629	0.0827	0.1007	0.0406	0.5826	**0.0924**	0.0665	0.1298
	Qwen2.5-VL-3B [3]	0.1809	0.1489	0.1895	0.2607	0.3527	0.1599	0.1690	0.2237	0.1893	0.2083
	Qwen2.5-VL-72B [3]	0.0422	0.0801	0.0586	0.1146	0.0681	0.0964	0.2380	0.1232	0.0264	0.0924
Specialized VLMs	Dolphin [12]	0.0957	0.0453	0.0616	0.1333	0.1684	0.0702	0.2388	0.2561	0.0186	0.1209
	olmOCR-7B [27]	0.0497	0.0365	0.0539	0.1204	0.0728	0.0697	0.2916	0.1220	0.0459	0.0957
	MonkeyOCR-pro-1.2B [18]	0.0961	0.0354	0.0530	0.1110	0.0887	0.0494	0.0995	0.1686	0.0198	0.0802
	MonkeyOCR-3B [18]	0.0904	0.0362	0.0489	0.1072	0.0745	0.0475	0.0962	0.1165	0.0196	0.0708
	dots.ocr [30]	**0.0290**	0.0231	0.0433	0.0788	**0.0467**	0.0221	0.0667	0.1116	0.0076	0.0477
	MinerU2.5 [24]	0.0294	0.0235	0.0332	**0.0499**	0.0681	0.0316	0.0540	0.1161	0.0104	0.0462
	Qwen2.5-VL-3B-NTP	0.0812	0.0273	0.0460	0.0835	0.0969	0.0236	0.0366	0.0982	0.0329	0.0585
Ours	Qwen2.5-VL-3B-PTP0	0.0744	0.0327	0.0427	0.0409	0.0797	0.0242	0.0404	0.0950	0.0316	0.0513
	Qwen2.5-VL-3B-PTP1	0.0572	0.0213	**0.0176**	0.0631	0.0779	**0.0079**	**0.0392**	0.0994	**0.0047**	**0.0431**
	Qwen2.5-VL-3B-PTP2	0.0616	0.0351	0.0203	0.0853	0.0992	0.0122	0.0456	0.1627	0.0083	0.0589

Table 1. Document Parsing Performance in Text Edit Distance on OmniDocBench: evaluation using edit distance across 9 PDF page types.

Model	Overall(↑)	AR	OSM	TA	OS	HF	MC	LTT	Base
MinerU2-pipeline [38]	55.6	61.8	13.5	60.9	17.3	**96.6**	59.0	39.1	96.6
GPT-4o [1]	63.2	44.1	37.6	69.1	40.9	94.2	68.9	54.1	96.7
Qwen2.5-VL-72B [3]	64.8	72.2	51.1	67.3	38.6	73.6	68.3	49.1	98.3
MonkeyOCR-pro-3B [18]	68.8	67.7	28.4	74.6	36.1	91.2	76.6	80.1	95.3
olmOCR [27]	71.8	63.9	41.0	72.9	**43.9**	95.1	77.3	81.2	98.9
dots.ocr [30]	73.6	66.3	35.8	**88.3**	40.9	94.1	**82.4**	81.2	**99.5**
MinerU2.5 [24]	**75.2**	**76.6**	**54.6**	84.9	33.7	**96.6**	78.2	83.5	93.7
Qwen2.5-VL-3B-PTP-0	71.0	63.5	38.6	71.3	34.2	95.3	78.3	**87.6**	**99.5**
Qwen2.5-VL-3B-PTP-1	70.6	62.6	38.6	70.4	33.6	96.2	79.4	84.7	99.4

Table 2. Evaluation results on olmOCR-bench grouped by document types, including arXiv Math(AR), Old Scans Math (OSM), Tables (TA), Old Scans (OS), Headers Footers (HF), Multi Column (MC) and Long Tiny Text (LTT). Some results are sourced from the official reports of olmOCR-bench [27] and dots.ocr [30]. The Overall Score (Overall) represents the average across all document types.

Model	CDM(↑)	BLUE(↑)	Norm Edit(↓)
Mathpix [23]	86.60	**66.56**	0.322
Pix2Tex	73.90	46.00	0.337
UniMERNet-B [37]	85.00	60.84	0.238
GPT4o [1]	86.80	45.17	0.282
Qwen2.5-VL-3B [3]	24.11	53.59	0.331
Qwen2.5-VL-3B-NTP	71.65	63.05	**0.226**
Qwen2.5-VL-3B-PTP0	**91.59**	63.38	0.231
Qwen2.5-VL-3B-PTP1	89.63	62.32	0.236
Qwen2.5-VL-3B-PTP2	77.23	57.92	0.284

Table 3. Formula recognition results on OmniDocBench. We report results in terms of both CDM [37], BLEU and Edit Distance.

against NTP and MTP approaches. For fair comparison, we follow the MTP architecture from Mimo [42] and adopt the training strategy from FastMTP [5] to augment Qwen2.5-VL with shared MTP heads and blocks. All models are fine-tuned on identical datasets with same training setting. *(i) Training Efficiency.* The training trajectories in Fig. 4 reveal significant efficiency advantages of PTP over MTP. While both methods exhibit initially high loss values, **PTP demonstrates rapid loss reduction and achieves fast convergence**, whereas MTP requires substantially more training steps to reach comparable performance. Notably, PTP achieves loss levels on par with NTP while substantially outperforming MTP. Additionally, PTP exhibits consistent convergence patterns across different configurations (PTP-1 and PTP-2), while MTP shows notable sensitivity to the number of prediction heads, with MTP-2 exhibiting significantly slower convergence. This maybe attribute to MTP introducing additional head and block parameters, whereas PTP requires only learnable register tokens without architectural modifications, resulting in superior training efficiency and stability. *(ii) Inference Efficiency.* Our PTP method also demonstrates significant advantages during in-

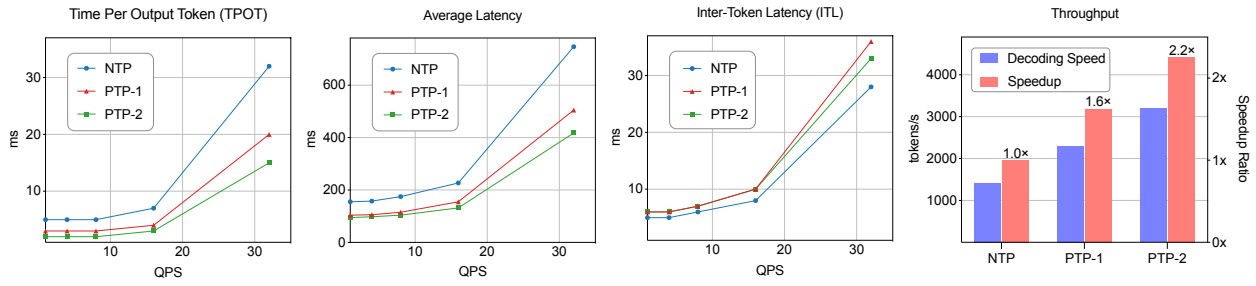

Figure 3. Performance comparison between NTP and PTP, including average TPOT, ITL, and latency under different QPS levels, as well as decoding speed and speedup ratio in synchronous mode, which are measured on OmniDocBench 16,886 images using an H20 GPU.

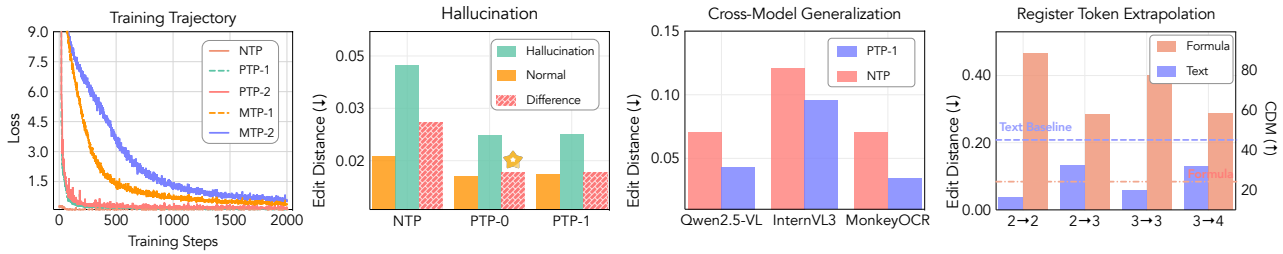

Figure 4. **Left**: Training trajectories of NTP, PTP, and MTP; **Second**: Performance of different methods on normal vs. hallucination-prone data; **Third**: PTP performance across different model architectures; **Right**: Register token extrapolation results.

ference. As illustrated in Fig. 3, PTP achieves substantial decoding acceleration compared to NTP. While MTP employs a self-speculative approach that yields results comparable to NTP (but underperforms PTP-1), it achieves only a 70% acceptance rate, resulting in lower speedup than PTP (the acceptance rate of PTP is 100% theoretically). The superior performance of PTP stems from its architectural advantages: PTP achieves true parallelism across all predicted tokens through register tokens, whereas MTP introduces sequential dependencies between prediction heads, limiting parallelization efficiency. During inference, register tokens in PTP are processed through the complete model architecture identically to regular tokens, ensuring consistent representation quality and robust predictions.

Exploring the Role of PTP. Our PTP method not only significantly improves inference efficiency but also positively impacts model performance on document parsing task. *(i) PTP Enhances NTP Performance.* Similar to observations in multi-token prediction approaches [41, 42], PTP densifies training signals by predicting multiple future tokens simultaneously, improving data efficiency and encouraging the model to develop superior long-term planning strategies. As shown in Tab. 1, PTP-0 trained with $\mathcal{L}_{\text{PTP}}$ consistently outperforms standard NTP trained with $\mathcal{L}_{\text{NTP}}$ across all benchmarks. Notably, both models use identical training data and configurations, with the only difference being the additional regularization objective $\mathcal{L}_{\text{reg}}$ in PTP training. This demonstrates that our parallel prediction framework not only accelerates inference but also improves the model's representational capabilities. *(ii) Hallucination Mitigation.* Hallucination poses a critical challenge for VLMs in OCR tasks, particularly when processing images with blurred, distorted, or linguistically irregular text. Traditional autoregressive generation via NTP is inherently susceptible to error propagation, as models rely heavily on preceding context and may generate erroneous tokens through auto-completion or over-correction mechanisms [15]. In contrast, PTP's parallel generation mechanism enables simultaneous attention to multiple image regions while generating corresponding tokens concurrently, thereby reducing over-dependence on sequential context. To quantify this advantage, we construct a controlled hallucination benchmark by systematically injecting noise into ground-truth annotations and their corresponding images through random word replacement and deletion (details in Supplementary Materials). As illustrated in Fig. 4.Second, PTP exhibits substantially lower hallucination rates compared to NTP. This improvement stems from PTP's ability to leverage global visual information more effectively, making predictions based on direct visual evidence rather than potentially corrupted contextual cues.

Model	Accuracy	Accept Rate (%)
Qwen2.5-VL-3B-NTP	92.21	N/A
Qwen2.5-VL-3B-PTP-1	91.72	100
w/ speculative decoding	92.21	82

Table 4. Results on ScienceQA dataset.

5.4. Generalizability Study

To comprehensively validate the generalizability of our PTP method, we conduct systematic evaluations across three critical dimensions: cross-model generalization, register token extrapolation, and cross-task adaptation.

Cross-Model Generalization. We evaluate PTP across models spanning different architectural paradigms, scales, and domains to assess its model-agnostic nature. As shown in Fig. 4.Third, PTP consistently yields substantial performance gains across all tested configurations, from compact models to large-scale architectures. This demonstrates PTP's flexible compatibility and minimal architectural requirements, making it broadly applicable to diverse vision-language models without extensive customization.

Register Token Extrapolation. We investigate the robustness of PTP when the number of register tokens differs between training and inference phases, as shown in Fig. 4.Right. Specifically, we examine two extrapolation scenarios: training with $n = 2$ tokens while inference with $n = 3$ (PTP-2 $\rightarrow$ PTP-3), and training with $n = 3$ tokens while inferring with $n = 4$ (PTP-3 $\rightarrow$ PTP-4). Although modest performance degradation is observed under these mismatched conditions, the results consistently surpass the vanilla model without register tokens. This degradation can be attributed to distributional shifts in learned token representations and can be effectively mitigated through self-speculative decoding mechanisms (discussed below).

Task Domain Generalization. Beyond document parsing task, we further investigate whether PTP generalizes to Vision-Language Understanding (VLU) tasks that require complex reasoning. We select the ScienceQA [22] benchmark, which is particularly challenging as answers incorporate explicit chain-of-thought (CoT) reasoning, making it particularly suitable for evaluating both accuracy and acceleration efficiency. We train on the official training split and report results on the test set. As shown in Tab. 4, PTP-1 achieves comparable performance to standard NTP while delivering substantial latency reductions. Furthermore, our PTP can be seamlessly integrated with *self-speculative decoding*, which incorporates verification mechanisms without requiring additional draft models (details in Supplementary Materials), achieving performance identical to NTP with an 82% acceptance rate and minimal latency overhead. These findings substantiate the broad applicability of our method across diverse tasks.

Model	Formula			Text Edit (↓)
	CDM(↑)	BLEU(↑)	Edit(↓)	
Qwen2.5-VL-3B	24.11	53.59	0.331	0.208
Qwen2.5-VL-3B-PTP-1	89.63	62.32	0.236	0.043
w/ distinct reg.	89.24	61.78	0.247	0.052
w/ interleaved reg.	88.46	61.45	0.244	0.128
w/o KV Cache	39.43	35.96	0.476	0.070

Table 5. Ablation study on text and formula recognition in OmniDocBench.

5.5. Ablation Study

Shared vs. Distinct Register Embedding. We investigate whether using a single shared learnable embedding across all positions of register outperforms position-specific distinct embeddings. As shown in Tab. 5, the shared embedding configuration achieves marginally better performance, consistent with observations in [13].

Interleaved vs. Continuous Register Tokens. We compare an interleaved insertion strategy based on [13], which inserts a single register token between regular tokens to predict future tokens at specified offsets, against our continuous approach that inserts a fixed number of sequential register tokens for incremental multi-token prediction. Experiments demonstrate clear advantages of our method, particularly for PTP-2. Intuitively, the continuous approach enables PTP-2 to leverage contextual information from PTP-1 during prediction, whereas the interleaved one predicts in isolation without access to intermediate information.

Inference without KV Cache Replacement. As detailed in Sec. 4.3, register tokens serve purely as computational intermediates without carrying semantic content explicitly. Consequently, during inference, we discard all the KV cache entries associated with registers and replace cache states using their corresponding predicted tokens in the next decode step. Notably, this operation introduces only negligible additional computation without impacting inference latency when computation resources are sufficient. Experimental results validate that the cache replacement mechanism is essential for maintaining performance.

6. Conclusion

In this paper, we propose PTP, a parallel token prediction method that enables VLMs to efficiently accelerate document parsing. Our contributions include: (1) a high-quality layout-level document parsing data generation framework, and (2) an architecture-agnostic, plugable and simple-yet-effective framework implementing parallel token prediction with registers injection. Experimental results demonstrate that our approach achieves 1.6×-2.2× decoding speedup while fully preserving parsing accuracy.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 5, 6

[2] Haoli Bai, Zhiguang Liu, Xiaojun Meng, Wentao Li, Shuang Liu, Nian Xie, Rongfu Zheng, Liangwei Wang, Lu Hou, Jiansheng Wei, et al. Wukong-reader: Multi-modal pre-training for fine-grained visual document understanding. *arXiv preprint arXiv:2212.09621*, 2022. 1

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *CoRR*, abs/2502.13923, 2025. 1, 3, 5, 6

[4] Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. Nougat: Neural optical understanding for academic documents. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. 1, 2

[5] Yuxuan Cai, Xiaozhuan Liang, Xinghua Wang, Jin Ma, Haijin Liang, Jinwen Luo, Xinyu Zuo, Lisheng Duan, Yuyang Yin, and Xi Chen. Fastmtp: Accelerating llm inference with enhanced multi-token prediction. *arXiv preprint arXiv:2509.18362*, 2025. 6

[6] Mingxu Chai, Ziyu Shen, Chong Zhang, Yue Zhang, Xiao Wang, Shihan Dou, Jihua Kang, Jiazheng Zhang, and Qi Zhang. Docfusion: A unified framework for document parsing tasks. *arXiv preprint arXiv:2412.12505*, 2024. 1

[7] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 3, 5, 6

[8] Cheng Cui, Ting Sun, Suyin Liang, Tingquan Gao, Zelun Zhang, Jiaxuan Liu, Xueqing Wang, Changda Zhou, Hongen Liu, Manhui Lin, et al. Paddleocr-vl: Boosting multilingual document parsing via a 0.9 b ultra-compact vision-language model. *arXiv preprint arXiv:2510.14528*, 2025. 1, 5, 6

[9] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. 1, 4

[10] Yongkun Du, Zhineng Chen, Caiyan Jia, Xiaoting Yin, Chenxia Li, Yuning Du, and Yu-Gang Jiang. Context perception parallel decoder for scene text recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(6):4668–4683, 2025. 2

[11] Ruchao Fan, Wei Chu, Peng Chang, and Abeer Alwan. A CTC alignment-based non-autoregressive transformer for end-to-end automatic speech recognition. *IEEE ACM Trans. Audio Speech Lang. Process.*, 31:1436–1448, 2023. 2

[12] Hao Feng, Shu Wei, Xiang Fei, Wei Shi, Yingdong Han, Lei Liao, Jinghui Lu, Binghong Wu, Qi Liu, Chunhui Lin, Jingqun Tang, Hao Liu, and Can Huang. Dolphin: Document image parsing via heterogeneous anchor prompting. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 21919–21936. Association for Computational Linguistics, 2025. 1, 2, 5, 6

[13] Anastasios Gerontopoulos, Spyros Gidaris, and Nikos Komodakis. Multi-token prediction needs registers. *CoRR*, abs/2505.10518, 2025. 1, 4, 5, 8

[14] Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. 2, 3

[15] Zhentao He, Can Zhang, Ziheng Wu, Zhenghao Chen, Yufei Zhan, Yifan Li, Zhao Zhang, Xian Wang, and Minghui Qiu. Seeing is believing? mitigating ocr hallucinations in multimodal large language models, 2025. 7

[16] Geewook Kim, Teakgyu Hong, Moonbin Yim, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Donut: Document understanding transformer without ocr. *arXiv preprint arXiv:2111.15664*, 7(15):2, 2021. 1

[17] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR, 2023. 2

[18] Zhang Li, Yuliang Liu, Qiang Liu, Zhiyin Ma, Ziyang Zhang, Shuo Zhang, Zidun Guo, Jiarui Zhang, Xinyu Wang, and Xiang Bai. Monkeyocr: Document parsing with a structure-recognition-relation triplet paradigm. *CoRR*, abs/2506.05218, 2025. 2, 3, 5, 6

[19] Demiao Lin. Revolutionizing retrieval-augmented generation with enhanced pdf structure recognition. *arXiv preprint arXiv:2401.12599*, 2024. 1

[20] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. 2

[21] Yuan Liu, Zhongyin Zhao, Le Tian, Haicheng Wang, Xubing Ye, Yangxiu You, Zilin Yu, Chuhan Wu, Xiao Zhou, Yang Yu, et al. Points-reader: Distillation-free adaptation of vision-language models for document conversion. *arXiv preprint arXiv:2509.01215*, 2025. 1

[22] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 8

[23] Mathpix. Mathpix, 2025. Accessed:2025-09-25. 6

[24] Junbo Niu, Zheng Liu, Zhuangcheng Gu, Bin Wang, Linke Ouyang, Zhiyuan Zhao, Tao Chu, Tianyao He, Fan Wu, Qintong Zhang, et al. Mineru2. 5: A decoupled vision-language model for efficient high-resolution document parsing, 2025. 1, 2, 5, 6

[25] Linke Ouyang, Yuan Qu, Hongbin Zhou, Jiawei Zhu, Rui Zhang, Qunshu Lin, Bin Wang, Zhiyuan Zhao, Man Jiang, Xiaomeng Zhao, Jin Shi, Fan Wu, Pei Chu, Minghao Liu, Zhenxiang Li, Chao Xu, Bo Zhang, Botian Shi, Zhongying Tu, and Conghui He. Omnidocbench: Benchmarking diverse PDF document parsing with comprehensive annotations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 24838–24848. Computer Vision Foundation / IEEE, 2025. 2, 5

[26] Vik Paruchuri. Marker, 2025. Accessed:2025-09-25. 1, 2, 5, 6

[27] Jake Poznanski, Jon Borchardt, Jason Dunkelberger, Regan Huff, Daniel Lin, Aman Rangapur, Christopher Wilhelm, Kyle Lo, and Luca Soldaini. olmocr: Unlocking trillions of tokens in pdfs with vision language models. *CoRR*, abs/2502.18443, 2025. 5, 6

[28] Jake Poznanski, Luca Soldaini, and Kyle Lo. olmocr 2: Unit test rewards for document ocr. *arXiv preprint arXiv:2510.19817*, 2025. 1

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 3

[30] rednote. dots.ocr: Multilingual document layout parsing in a single vision-language model, 2025. Accessed:2025-09-25. 5, 6

[31] Kunyu Shi, Qi Dong, Luis Goncalves, Zhuowen Tu, and Stefano Soatto. Non-autoregressive sequence-to-sequence vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 13603–13612. IEEE, 2024. 2

[32] Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. Blockwise parallel decoding for deep autoregressive models. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2018. 2

[33] Ting Sun, Cheng Cui, Yuning Du, and Yi Liu. Pp-doclayout: A unified document layout detection model to accelerate large-scale data construction. *arXiv preprint arXiv:2503.17213*, 2025. 2

[34] Said Taghadouini, Baptiste Aubertin, and Adrien Cavaillès. Lightonocr-1b: End-to-end and efficient domain-specific vision-language models for ocr, 2025. 1, 2

[35] Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. Unifying vision, text, and layout for universal document processing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19254–19264, 2023. 1

[36] Tencnet. Ksanallm, 2025. 5

[37] Bin Wang, Zhuangcheng Gu, Guang Liang, Chao Xu, Bo Zhang, Botian Shi, and Conghui He. Unimernet: A universal network for real-world mathematical expression recognition. *arXiv preprint arXiv:2404.15254*, 2024. 6

[38] Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, Bo Zhang, Liqun Wei, Zhihao Sui, Wei Li, Botian Shi, Yu Qiao, Dahua Lin, and Conghui He. Mineru: An open-source solution for precise document content extraction, 2024. 1, 2, 5, 6

[39] Haoran Wei, Lingyu Kong, Jinyue Chen, Liang Zhao, Zheng Ge, Jinrong Yang, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. Vary: Scaling up the vision vocabulary for large vision-language model. In *European Conference on Computer Vision*, pages 408–424. Springer, 2024. 2

[40] 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. General OCR theory: Towards OCR-2.0 via a unified end-to-end model. *CoRR*, abs/2409.01704, 2024. 2

[41] Haoran Wei, Yaofeng Sun, and Yukun Li. Deepseek-ocr: Contexts optical compression. *arXiv preprint arXiv:2510.18234*, 2025. 1, 2, 7

[42] LLM Xiaomi, Bingquan Xia, Bowen Shen, Dawei Zhu, Di Zhang, Gang Wang, Hailin Zhang, Huaqiu Liu, Jiebao Xiao, Jinhao Dong, et al. Mimo: Unlocking the reasoning potential of language model - from pretraining to posttraining. *CoRR*, abs/2505.07608, 2025. 6, 7

[43] Junyuan Zhang, Qintong Zhang, Bin Wang, Linke Ouyang, Zichen Wen, Ying Li, Ka-Ho Chow, Conghui He, and Wentao Zhang. Ocr hinders rag: Evaluating the cascading impact of ocr on retrieval-augmented generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17443–17453, 2025. 1

[44] Qintong Zhang, Bin Wang, Victor Shea-Jay Huang, Junyuan Zhang, Zhengren Wang, Hao Liang, Conghui He, and Wentao Zhang. Document parsing unveiled: Techniques, challenges, and prospects for structured information extraction. *arXiv preprint arXiv:2410.21169*, 2024. 1

[45] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *CoRR*, abs/2504.10479, 2025. 5, 6