DARL: Efficient Document-to-Markup Generation via Look-Ahead Diffusion Trajectory Sampling

Wentao Yang[1,2⋆], Yongxin Shi[1⋆], Rui Tang[1⋆], Peirong Zhang[1], Shihang Wu[1], Huiguo He[1], Zheng Huang[3], Dezhi Peng[3], Minghui Liao[3], and Lianwen Jin[1(✉)]

[1] South China University of Technology, Guangzhou, China
eelwjin@scut.edu.cn
[2] HiThink Research, Hangzhou, China
wente_young@foxmail.com
[3] Huawei Technologies Ltd., Shenzhen, China

Abstract. While autoregressive (AR) Multimodal Large Language Models (MLLMs) excel at complex document-to-markup generation, their sequential decoding causes severe latency. Conversely, existing parallel and diffusion-based methods accelerate inference but often struggle to maintain strict structural dependencies, resulting in significantly higher parsing errors. To bridge this gap, we propose DARL (Diffusion AutoRegression with Look-ahead), a novel hybrid decoding framework. DARL introduces a Sliding Diffusion Block (SDB) that maintains a verified AR prefix for strict syntactic correctness while speculatively generating a localized window of future tokens in parallel. To optimize these discrete diffusion trajectories, we introduce Online Monte Carlo Trajectory Generation (OMTG) and Diffusion Trajectory Preference Optimization (DTPO). OMTG dynamically samples candidate paths based on the model's real-time state, effectively mitigating the exposure bias inherent in static heuristics. DTPO integrates immediate and look-ahead rewards to optimize current accuracy and facilitate future correctness, ensuring stable and rapid convergence of the predictive trajectory space. Experiments on OmniDocBench-1.5 and olmOCR-Bench demonstrate that DARL establishes a new Pareto frontier in document parsing. It achieves up to a 2.3× speedup while matching or exceeding the state-of-the-art accuracy of AR baselines. The code and model are available at https://github.com/SCUT-DLVCLab/DARL.

Keywords: Document Parsing · Multimodal Large Language Models · Discrete Diffusion · Preference Optimization

1 Introduction

Mainstream Multimodal Large Language Models (MLLMs) [2,10,31] have revolutionized document parsing, enabling the end-to-end translation of dense visual

⋆ These authors contributed equally to this work.

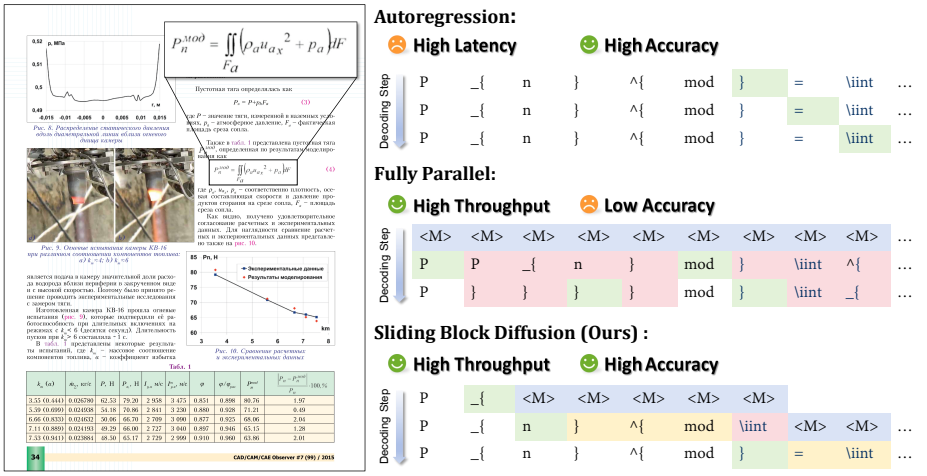

Fig. 1: Qualitative Mechanics of Decoding Paradigms in Structured Parsing. Autoregressive decoding maintains high structural accuracy but suffers from high latency. Fully parallel diffusion maximizes throughput but destroys rigid contextual dependencies (e.g., matching brackets in formulas). Our **DARL** framework introduces a Sliding Diffusion Block that maintains a verified AR prefix while speculatively refining future tokens in parallel, ensuring both absolute syntactic correctness and high speed.

inputs into structured text. This paradigm relies heavily on autoregressive (AR) decoding to handle complex document-to-markup generation tasks, including the conversion of scanned pages into intricate LaTeX formulas or structurally rigid HTML tables. While AR generation effectively captures the rigorous token dependencies required for high-accuracy structural parsing, its sequential next-token prediction [26] mechanism imposes a severe latency bottleneck. In high-density document parsing, where a single visual patch can translate to thousands of formatting tokens, this sequential dependency becomes the primary obstacle to real-time deployment. To mitigate AR latency, various parallel decoding strategies [4,9,14,17] and Diffusion Large Language Models (dLLMs) [3,36] have been proposed. Yet, applying these methods to multimodal document parsing reveals critical structural bottlenecks. Standard speculative decoding relies on relaxed heuristic priors (e.g., n-gram matching) that work well for casual text but fail spectacularly when faced with strict markup constraints, where a single error can silently corrupt an entire layout. Conversely, fully parallel diffusion architectures (as illustrated in Figure 1) attempt to predict all tokens simultaneously. While this maximizes throughput, it destroys the rigorous structural dependencies inherent in code and markup languages. Without a strict directional context, fully diffusion models frequently generate mismatched brackets or corrupted table alignments, yielding fundamentally broken outputs and inferior parsing quality compared to AR baselines. Moreover, these methods struggle to determine when denoising is complete, often relying on fixed refinement steps or confidence thresholds that either under- or over-refine the results.

To bridge this gap between sequential accuracy and parallel efficiency, we propose **DARL** (**D**iffusion **A**uto**R**egression with **L**ook-ahead), a novel hybrid framework tailored for structured visual-text generation. Rather than abandoning AR entirely, DARL introduces a **Sliding Diffusion Block (SDB)** mechanism. As shown in Figure 1, SDB operates as a hybrid engine: it maintains a stable, verified AR prefix to ensure absolute syntactic correctness, while speculatively generating a localized window of future tokens in parallel via discrete denoising. This approach effectively reconciles the structural stability of AR priors with the high throughput of diffusion.

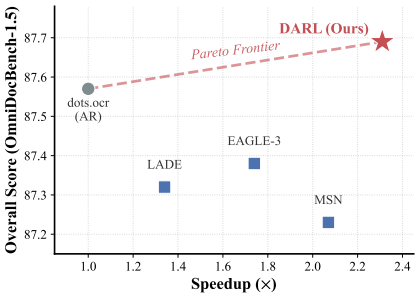

Fig. 2: Pareto Efficiency. DARL establishes a new Pareto frontier on OmniDocBench-1.5, achieving a 2.31× speedup without compromising high accuracy.

Beyond the architecture, deploying discrete diffusion for structural parsing presents a fundamental optimization challenge. Text diffusion must navigate a massive, brittle combinatorial space where local accuracy does not guarantee global layout coherence. To efficiently optimize the diffusion trajectory, we introduce two synergistic training strategies. First, we propose **Online Monte Carlo Trajectory Generation (OMTG)**, which dynamically constructs candidate trajectories based on real-time model states, eliminating the exposure bias inherent in static generation strategies. Second, building upon this dynamic sampling, we introduce **Diffusion Trajectory Preference Optimization (DTPO)**. Framing the decoding window generation as a reinforcement learning problem, DTPO employs both immediate accuracy and look-ahead consistency rewards. By explicitly penalizing generation paths that appear locally plausible but cause structural drift in subsequent windows, DTPO incentivizes the model to predict stable, rapidly converging trajectories.

Extensive experiments on both general MLLMs and specialized document parsing experts demonstrate that DARL effectively translates visual-structural awareness into decoding efficiency. As highlighted in Figure 2, our method completely breaks the traditional latency-accuracy trade-off. It achieves up to a 2.3× wall-time speedup while preserving, and in complex layout categories exceeding, the state-of-the-art (SOTA) recognition performance of AR baselines.

Our primary contributions are summarized as follows:

- We propose DARL, a novel hybrid decoding paradigm that synergizes autoregressive modeling with Sliding Diffusion Blocks (SDB), successfully adapting diffusion-based parallel generation to the rigorous structural demands of complex document parsing.
- We introduce Online Monte Carlo Trajectory Generation (OMTG) to dynamically sample candidate paths based on the model's real-time state, ef-

fectively mitigating the exposure bias inherent in static heuristics during training.
- We design Diffusion Trajectory Preference Optimization (DTPO), which integrates immediate and look-ahead rewards to optimize current accuracy and facilitate future correctness, ensuring stable and rapid convergence of the predictive trajectory space.
- We conduct comprehensive evaluations on document parsing benchmarks, verifying that DARL delivers SOTA Pareto efficiency across diverse MLLM architectures.

2 Related Work

MLLMs for Document Parsing. Multimodal Large Language Models (MLLMs) have revolutionized document parsing by enabling end-to-end mapping from images to structured text [2, 15, 31]. Specialized models [12, 18, 22, 23, 29, 34, 39] have demonstrated state-of-the-art (SOTA) performance in recognizing complex layouts, including tables and formulas [41]. However, these models predominantly rely on the autoregressive (AR) paradigm [10, 26], which suffers from a linear latency bottleneck as the output sequence length increases. In high-density document tasks, this sequential dependency becomes the primary obstacle to real-time deployment.

Parallel and Speculative Decoding. To mitigate the AR latency, various parallel decoding strategies have been proposed. Speculative decoding [4, 17] and Jacobi decoding [27] accelerate inference by verifying multiple candidate tokens in a single forward pass. Lookahead decoding [9] and Consistency LLMs (CLLMs) [14] further reduce the sequential overhead by utilizing fixed-point iterations or Jacobi trajectories. MSN [33] introduces noise during training to unlock parallel capabilities. While effective for general text, these methods often rely on simple n-gram or heuristic-based speculative priors, which struggle to capture the rigorous structural constraints found in LaTeX formulas or Markdown tables.

Diffusion Models for Vision-Language Generation. Discrete denoising diffusion models [1, 11, 13] have emerged as a powerful non-autoregressive alternative for multimodal generation. Recent work explores the synergy between diffusion and autoregressive decoding through hybrid architectures [3, 5, 19, 20, 32], achieving faster-than-AR inference by reconciling causal attention with parallel token generation. In the vision-language domain, LLaDA-V [37], MMADA [35], LaViDa [16], and dVLM-AD [21] explore parallel token generation via iterative denoising for image-to-text tasks. Block-wise diffusion VLMs such as SDAR-VL [6] and DiffusionVL [38] translate pretrained autoregressive models into efficient diffusion architectures. However, these methods often struggle to determine when denoising is complete, requiring fixed refinement steps that may under-refine or over-refine. Our DARL extends this line by introducing a Sliding Diffusion Block with speculative verification, which dynamically determines denoising completion, theoretically guaranteeing $P_{\mathrm{DARL}}(y_{\leq t}|x) \equiv P_{\mathrm{AR}}(y_{\leq t}|x)$ while achieving superior efficiency for structured document parsing.

Preference Optimization in Generation. Reinforcement learning from human feedback (RLHF) and preference optimization (PO) are typically used to align MLLMs with human values or reasoning tasks. Group Relative Policy Optimization (GRPO) [28] has recently shown great potential in mathematical reasoning without a separate reward model. In the context of diffusion, Inpainting-Guided PO [44], diffu-GRPO [43] and LLaDA 1.5 [46] explores aligning diffusion trajectories with specific constraints. Our proposed DTPO draws inspiration from these techniques but shifts the focus to the *decoding trajectory* itself. By incorporating immediate and look-ahead rewards, DTPO incentivizes the model to select trajectories that not only maximize immediate accuracy but also provide a consistent semantic foundation for future generation windows.

3 Preliminaries

3.1 Lookahead Decoding

Lookahead decoding [9] is a parallel inference strategy that accelerates autoregressive generation without requiring an auxiliary draft model. It reformulates decoding as a fixed-point iteration problem, maintaining a lookahead window of size W to simultaneously predict and verify multiple tokens. In each iteration j, candidate tokens within the window are updated in parallel:

$$y_{i+k}^{(j+1)} = \arg\max_{v} p(v \mid \mathbf{y}_{<i}, y_i^{(j)}, \ldots, y_{i+k-1}^{(j)}), \forall k \in \{0, \ldots, W-1\}. \tag{1}$$

By utilizing n-gram matching to identify promising candidate sequences and verifying them in a single forward pass, lookahead decoding can accept multiple tokens per step, effectively breaking the sequential bottleneck of standard autoregressive decoding.

3.2 Diffusion Large Language Models

Diffusion Large Language Models (dLLMs) [3] define a distribution $p_\theta(\mathbf{y}_0)$ through forward masking and reverse denoising. The forward process masks tokens independently with probability $t \in (0, 1)$. The model is trained to predict original tokens at masked positions by minimizing:

$$\mathcal{L}_{\text{dLLM}} = -\mathbb{E}_{t, \mathbf{y}_0, \mathbf{y}_t}\Big[\frac{1}{t}\sum_{i=1}^{L} \mathbf{1}[\mathbf{y}_t^i = \texttt{[M]}] \cdot \log p_\theta(\mathbf{y}_0^i | \mathbf{y}_t)\Big], \tag{2}$$

where $\mathbf{1}[\cdot]$ identifies masked tokens in $\mathbf{y}_t$. This loss serves as a principled upper bound on the negative log-likelihood, enabling efficient parallel generation during the reverse process. During inference, denoising typically terminates after a fixed number of refinement steps or when prediction confidence surpasses a preset threshold.

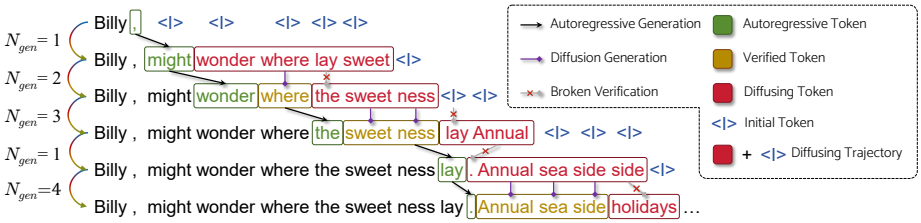

Fig. 3: Illustration of the Sliding Diffusion Block decoding process. In each step, the model performs a hybrid generation: one **Autoregressive Token** (green) is produced via next-token prediction, while a parallel **Diffusing Trajectory** (red) is speculated within the sliding window. The verification mechanism (indicated by purple markers) identifies **Verified Tokens** (gold) by checking the consistency between the current diffusion results and the speculative trajectory generated in the previous step. A **Broken Verification** (grey arrow with cross) occurs at the first point of discrepancy, at which the accepted sequence terminates

4 Method

Autoregressive decoding in MLLMs suffers from inherent latency bottlenecks due to sequential token generation, yet naive parallel decoding methods often compromise generation quality in structured tasks like document parsing. To this end, we propose **DARL** (**D**iffusion **A**uto**R**egression with **L**ook-ahead), a hybrid framework that strategically combines diffusion-based parallelism with autoregressive priors.

Our approach addresses three core challenges: (1) How to parallelize token generation without breaking semantic coherence? We introduce **Sliding Diffusion Blocks (SDB**, §4.1**)** that generate tokens within bounded windows while maintaining cross-window dependencies. (2) How to train diffusion models to match hybrid decoding behavior? We develop **Online Monte Carlo Trajectory Generation (OMTG**, §4.2**)**, which dynamically constructs training trajectories mirroring inference-time distributions. (3) How to optimize diffusion quality beyond standard likelihood objectives? We propose **Diffusion Trajectory Preference Optimization (DTPO**, §4.3**)**, a reinforcement learning method that leverages look-ahead rewards to align diffusion outputs with autoregressive quality .

4.1 Sliding Diffusion Block

Standard autoregressive (AR) models generate the sequence $\mathbf{y}$ token-by-token, modeling $p(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T} p(y_t|\mathbf{y}_{<t}, \mathbf{x})$. Despite its high recognition accuracy, the sequential nature imposes a severe time bottleneck during the parsing of lengthy documents. To accelerate this, we introduce a *Sliding Diffusion Block* mechanism that treats a window of tokens as a discrete diffusion unit.

Let w be the window size (e.g., 16 tokens). At any step t, the model aims to predict a block of future tokens $\mathbf{y}_{t:t+w}$ in parallel. Unlike standard AR which

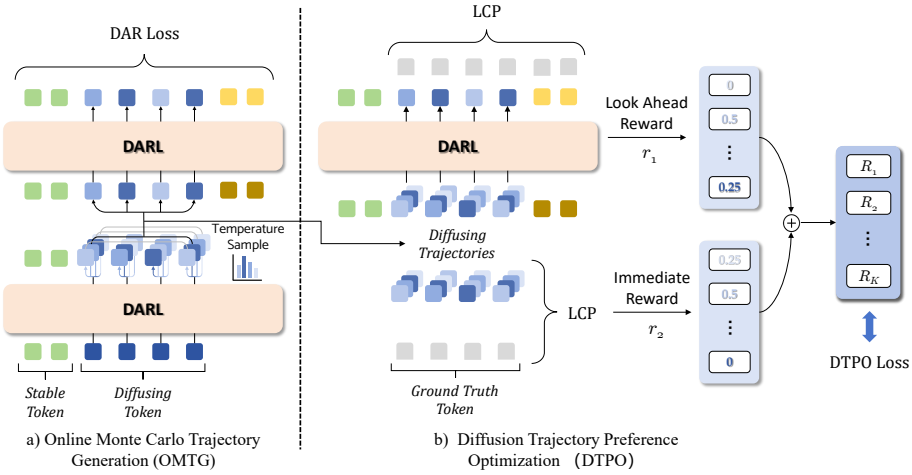

Fig. 4: Overview of the DARL training pipeline. (a) **Online Monte Carlo Trajectory Generation (OMTG)**: The model generates multiple candidate Diffusing Trajectories $\{\hat{\mathbf{h}}^{(1)}, \ldots, \hat{\mathbf{h}}^{(K)}\}$ through multinomial sampling with temperature scaling. These trajectories are then fed back into the model as input to compute the DAR Loss (top-left), training the model to reconstruct ground truth from speculative noise. (b) **Diffusion Trajectory Preference Optimization (DTPO)**: The same sampled trajectories are evaluated to compute the DTPO Loss. An Immediate Reward (r_1) measures the Longest Common Prefix (LCP) between the trajectory and the ground truth, while a Look-ahead Reward (r_2) assesses how well the trajectory facilitates the prediction of the subsequent window.

conditions strictly on ground truth history, our block operates in a hybrid state. We define the input to the model as a concatenation of the *stable context* $\mathbf{y}_{<t}$ and a *speculative window* $\mathbf{z}_{t:t+w}$, where the latter comprises a *partial trajectory* $\mathbf{h}$ (tentatively generated tokens) and *initialization tokens* $\mathbf{m}$ (specialized `<INIT>` masks). Formally, the model approximates the joint probability $p_\theta(\mathbf{y}_{t:t+w}|\mathbf{y}_{<t}, \mathbf{z}_{t:t+w}, \mathbf{x})$, allowing it to refine $\mathbf{h}$ and populate $\mathbf{m}$ simultaneously as a discrete denoising step. As illustrated in Figure 3, the complete inference workflow integrates single-token autoregressive prediction with multi-token diffusion speculation and subsequent consistency-based verification.

4.2 Online Monte Carlo Trajectory Generation (OMTG)

A critical challenge in training non-autoregressive or block-parallel models is the "exposure bias" or distribution shift: during inference, the model must correct its own imperfect predictions, but standard training uses ground-truth forcing. To address this, we propose **Online Monte Carlo Trajectory Generation (OMTG)**, a dynamic data construction strategy implemented directly within the training loop, as illustrated in Figure 4 (a).

Instead of using static masking patterns, OMTG leverages the current model state to generate training trajectories. For a target sequence $\mathbf{y}$ and a window at offset k, we split the window into a *pickup* segment (length l_p) and a *initial* segment (length l_n), such that $l_p + l_n = w$. The lengths l_p and l_n are determined by randomly sampling a split point for each training instance (e.g., $l_p \sim \mathcal{U}(0, w-1)$).

Trajectory Sampling: We freeze the gradient and prompt the model to predict the l_p tokens given the stable context $\mathbf{y}_{<k}$. We use multinomial sampling with temperature scaling to generate K distinct candidate trajectories $\{\hat{\mathbf{h}}^{(1)}, \ldots, \hat{\mathbf{h}}^{(K)}\}$.

Input Construction: For each candidate $\hat{\mathbf{h}}^{(i)}$, we append l_n initialization tokens. The input becomes $[\mathbf{y}_{<k}; \hat{\mathbf{h}}^{(i)}; \texttt{<INIT>}_{\times l_n}]$.

Dynamic Supervision: To inherit the foundational autoregressive (AR) properties of language models, we optimize the model by minimizing the cross-entropy loss across the entire target sequence from the sequence start to the window boundary $(1 : k + w)$:

$$\mathcal{L}_{\text{DAR}} = -\sum_{i=1}^{k+w} \log p_\theta(y_i \mid \hat{\mathbf{y}}_{\text{in}}, \mathbf{x}), \tag{3}$$

where $\hat{\mathbf{y}}_{\text{in}}$ is the composite input comprising stable context, diffusing trajectory, and initialization tokens. By supervising the sequence in its entirety rather than only the speculative window, the model effectively treats the task as a unified next-token prediction objective. This approach allows the model to rectify generation errors through a "self-correction" mechanism while ensuring it retains the robust generative capabilities of standard AR modeling.

4.3 Diffusion Trajectory Preference Optimization (DTPO)

While OMTG aligns the training distribution, standard Cross-Entropy loss ($\mathcal{L}_{\text{DAR}}$) only optimizes for local accuracy and fails to capture long-term semantic consistency. To incentivize the model to select trajectories that are not only locally correct but also facilitate future generation, we introduce **Diffusion Trajectory Preference Optimization (DTPO)**.

DTPO frames the window generation as a Reinforcement Learning problem. We treat the candidate trajectories $\{\hat{\mathbf{h}}^{(i)}\}$ generated by OMTG as actions sampled from the policy π_θ. We employ a Group Relative Policy Optimization (GRPO) approach [28] to update the model, avoiding the need for a separate value network, as illustrated in Figure 4 (b).

Reward Modeling We design a composite reward function $R(\tau)$ consisting of two terms to evaluate a trajectory τ:

1) Immediate Accuracy Reward (r_1): This term measures the accuracy of the generated trajectory against the ground truth. We calculate the length

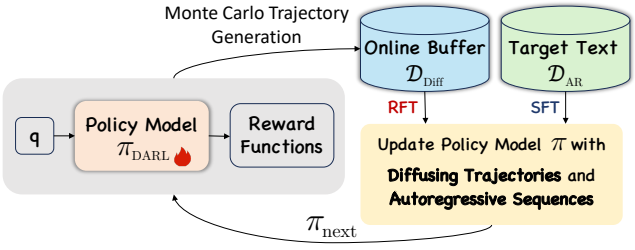

Fig. 5: Joint Optimization Framework of DARL. The training pipeline integrates Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL) simultaneously within a unified objective. To preserve the model's fundamental autoregressive nature without the need for a separate reference model, the DAR loss provides a constant supervised anchor. All Diffusing Trajectories are obtained base on the current model state through the OMTG algorithm to ensurs the supervision remains dynamically aligned with the model's evolution.

of the longest common prefix (LCP) between the generated segment $\hat{\mathbf{h}}$ and the target $\mathbf{y}_{target}$:

$$r_1(\hat{\mathbf{h}}) = \frac{\text{len}(\text{LCP}(\hat{\mathbf{h}}, \mathbf{y}_{target}))}{|\mathbf{y}_{target}|}. \tag{4}$$

2) Look-ahead Consistency Reward (r_2): A high-quality trajectory must not only be locally accurate but also serve as a robust foundation for future generation. To evaluate this, we perform a look-ahead step targeting the subsequent l_n tokens (the *initial* segment for the next iteration).

We construct a look-ahead input $\mathbf{z}_{\text{look}} = [\mathbf{y}_{<t}\,;\,\hat{\mathbf{h}}\,;\,\mathbf{m}_{\text{next}}]$ (where $|\mathbf{m}_{\text{next}}| = l_n$), which is forwarded through the model $\mathcal{M}$ to obtain the predicted future segment $\hat{\mathbf{y}}_{\text{future}} = \mathcal{M}(\mathbf{z}_{\text{look}})[-l_n:]$. The look-ahead reward is then calculated as the LCP ratio between this look-ahead prediction and the corresponding ground truth $\mathbf{y}_{\text{gt\_future}}$:

$$r_2(\hat{\mathbf{h}}) = \frac{\text{len}(\text{LCP}(\hat{\mathbf{y}}_{\text{future}}, \mathbf{y}_{\text{gt\_future}}))}{l_n}. \tag{5}$$

By favoring paths that yield correct look-ahead predictions, DTPO ensures incentivizes the selection of trajectories that significantly reduce the predictive uncertainty of the subsequent window, thereby facilitating rapid convergence and accelerating the inference process in the next sliding iteration.

Policy Optimization The total reward for the i-th trajectory in a group of K samples is $R_i = r_1^{(i)} + r_2^{(i)}$. To reduce variance, we compute the advantage A_i by normalizing rewards within the group:

$$A_i = \frac{R_i - \mu_R}{\sigma_R + \epsilon}, \tag{6}$$

where μ_R and σ_R are the mean and standard deviation of rewards in the current batch. The DTPO loss is formulated as:

$$\mathcal{L}_{\text{DTPO}} = -\frac{1}{K}\sum_{i=1}^{K}\sum_{t} A_i \cdot \log \pi_\theta(y_t^{(i)}|\mathbf{y}_{<t}^{(i)}). \tag{7}$$

This objective increases the probability of tokens belonging to trajectories that yield higher consistent rewards relative to their peers.

5 Experiments

5.1 Experimental Setup

Dataset Construction. To train our DARL, we curated a high-quality document parsing dataset comprising 49K samples across diverse domains, including magazines, newspapers, handwritten notes [40], examination papers, textbooks, PPT slides, financial research reports, and scanned old books. The initial annotations were generated using PaddleOCR-VL [7]. To ensure high-accuracy supervision, we removed all image-related non-textual tokens and performed extensive manual correction to rectify layout and recognition errors, resulting in a clean, text-centric document parsing corpus. Detailed data categories are provided in the supplementary material.

Benchmarks and Metrics. To verify the model's generation performance on multimodal long-form text, we evaluate DARL on two challenging document parsing benchmarks: 1) **OmniDocBench-1.5** [24], a comprehensive benchmark for document parsing that covers various layout elements like complex tables, formulas, and reading order; and 2) **olmOCR-Bench** [25], a rigorous evaluation suite for PDF-to-Markdown conversion that assesses the structural and factual accuracy of long-form document reconstruction through a fine-grained unit-test framework.

For efficiency, we report Wall-time *Speedup* and *Avg. Iterations* (the average number of model forward passes per document). For accuracy, we adopt the original metrics established by each benchmark, including Overall score, Text Edit Distance, and specialized scores for Formulas [30] and Tables [45].

Implementation Details. Our model is initialized from `dots.ocr-3B` [18] given its superior recognition performance. Training was performed on 8 NVIDIA A800 GPUs for 2 epochs with a per-device batch size of 1, lasting approximately 5 days. We utilized the AdamW optimizer with a peak learning rate of 2×10^{-5} and a linear warmup of 500 steps. The weight decay was set to 0.01. During OMTG sampling, we set $K = 4$ candidate trajectories. For DTPO, λ_1 and λ_2 was set to 1 and 0.1, respectively.

For training-free baselines including Jacobi Decoding [27] and LADE [9], we directly utilize their official decoding implementations. For MSN [33] and EAGLE-3 [17], we adapt their frameworks to the `dots.ocr` backbone by training on our curated data following their original configurations to ensure a fair comparison. All benchmarks are conducted with Flash-Attention 2 [8] to ensure an optimized evaluation environment.

Table 1: Performance comparison on OmniDocBench-1.5. We compare DARL with the autoregressive (AR) baseline and state-of-the-art acceleration methods. Efficiency is evaluated using Wall-time Speedup, Average Time (seconds per document), and Average Iterations (number of forward passes). DARL achieves the highest speedup (2.31×) while maintaining or exceeding the recognition accuracy of the AR expert model. * denotes the AR baseline further fine-tuned on our curated dataset for a fair comparison.

Model	Venue	Speedup↑	Avg. Time↓	Avg. Iter.↓	Overall↑	Text↓	Formula^{CDM} ↑	Table↑	Table^{S} ↑	Read Order↓
dots.ocr (AR) [18]	-	1.00×	25.85	1091.5	87.57	0.065	85.44	83.76	87.39	0.075
Jacobi Dec. [27]	ACL'23	1.09×	23.71	1005.0	87.51	0.071	86.30	83.33	87.09	0.076
LADE [9]	ICML'24	1.34×	19.26	733.4	87.32	0.070	85.59	83.37	87.27	0.076
MSN [33]	EMNLP'24	2.07×	12.49	527.2	87.23	0.093	85.17	85.81	88.76	0.079
DiffusionVL [38]	-	1.69×	15.30	683.9	85.92	0.112	84.31	84.65	88.57	0.098
EAGLE-3 [17]	NeurIPS'25	1.74×	14.88	625.0	87.38	0.084	86.14	84.40	88.21	0.089
dots.ocr (AR)*	-	1.04×	24.86	1033.3	87.15	0.095	87.62	83.35	85.69	0.072
DARL	Ours	**2.31×**	**11.19**	**485.6**	**87.69**	0.101	85.43	**87.73**	**89.41**	**0.072**

Table 2: Performance comparison on olmOCR-Bench. We report the wall-time speedup, average decoding time, and iteration counts alongside the parsing accuracy. DARL achieves a state-of-the-art speedup of 2.27× and consistently outperforms both the autoregressive baseline and other parallel decoding methods across most categories. * denotes the AR baseline further fine-tuned on our curated dataset for a fair comparison.

Model	Venue	Speedup↑	Avg. Time↓	Avg. Iter.↓	Overall↑	Base↑	ArXiv Math↑	Old Scans Math↑	Tables↑	Old Scans↑	Headers & Footers↑	Multi Column↑	Long Tiny Text↑
dots.ocr (AR) [18]	-	1.00×	28.65	1216.9	78.9 ± 1.0	99.1	81.3	72.3	89.1	41.6	85.9	82.9	79.2
Jacobi Dec. [27]	ACL'23	1.12×	25.61	1121.0	78.9 ± 1.0	98.7	81.3	72.1	88.9	41.6	84.9	82.5	81.0
LADE [9]	ICML'24	1.38×	20.73	800.6	78.9 ± 1.0	98.8	81.3	72.1	89.0	41.6	84.9	82.5	81.1
MSN [33]	EMNLP'24	1.98×	14.46	614.1	77.9 ± 1.1	97.1	73.5	78.4	79.8	53.8	90.4	79.5	70.6
EAGLE-3 [17]	NeurIPS'25	1.66×	17.24	726.6	77.9 ± 1.1	98.8	81.2	71.0	89.2	39.9	85.0	82.5	75.8
dots.ocr (AR)*	-	1.08×	26.52	1122.3	79.9 ± 1.0	99.4	80.2	71.2	76.7	57.0	97.2	82.5	75.1
DARL	Ours	**2.27×**	**12.61**	**519.3**	**81.7 ± 1.0**	98.0	78.7	**79.3**	88.4	**62.5**	89.2	**82.9**	74.7

5.2 Overall Training Objective

The final training objective combines the supervised diffusion loss and the preference optimization loss:

$$\mathcal{L}_{\text{Total}} = \lambda_1 \mathcal{L}_{\text{DAR}} + \lambda_2 \mathcal{L}_{\text{DTPO}}, \tag{8}$$

where $\mathcal{L}_{\text{DAR}}$ ensures the model learns basic syntax and structure, while $\mathcal{L}_{\text{DTPO}}$ refines the decision boundaries for optimal trajectory planning. This hybrid approach allows DARL to converge significantly faster than pure AR methods while achieving superior generation quality compared to pure diffusion baselines.

5.3 Main Results

Performance on OmniDocBench-1.5. As detailed in Table 1, DARL establishes a new state-of-the-art for efficient document parsing, achieving a 2.31× wall-time speedup while marginally improving parsing accuracy to 87.69. While traditional speculative methods like EAGLE-3 and MSN struggle with the

Table 3: Ablation study of DARL components on OmniDocBench-1.5. We evaluate the impact of the Sliding Diffusion Block (SDB), the immediate reward (r_1), and the look-ahead reward (r_2) on both efficiency and accuracy.

SDB	r_1	r_2	Speedup ↑	Avg. Time↓	Avg. Iter.↓	Overall↑
✗	✗	✗	1.00×	25.85	1091.5	87.57
✓	✗	✗	1.98×	13.03	547.3	87.21
✓	✓	✗	2.14×	12.09	519.7	87.53
✓	✓	✓	**2.31×**	**11.19**	**485.6**	**87.69**

rigid structural dependencies of LaTeX and Markdown, DARL's Sliding Diffusion Block enables joint parallel refinement of 16-token windows. This superiority is most evident in the Table Structure metric (89.41 vs. MSN's 88.76), driven by the Look-ahead Reward (r_2) in DTPO. By penalizing trajectories that are locally plausible but globally inconsistent, r_2 ensures a stable semantic foundation for subsequent windows, reducing the average iteration count to a record of 485.6.

Performance on olmOCR-Bench. As shown in Table 2, DARL achieves a 2.27× speedup with a peak accuracy of 81.7, significantly surpassing both the AR baseline (78.9) and MSN (77.9). Gains are most pronounced in complex elements such as Headers & Footers (89.2) and Multi-column layouts (82.9). Notably, the leap in the Old Scans category from 41.6 to 62.5 reflects the benefit of our meticulous data curation. This highlights that DARL is more than a mechanical acceleration method, as it successfully translates high-quality data into robust parsing expertise that fixed heuristic priors cannot capture.

5.4 Ablation Study

Analysis of Framework Components. Table 3 evaluates the synergy between the Sliding Diffusion Block (SDB) and DTPO. Naive SDB achieves a 1.98× speedup but suffers an accuracy drop (87.21 vs. 87.57) due to error accumulation in parallel decoding. Introducing the immediate reward r_1 recovers the baseline accuracy and boosts speed to 2.14× by prioritizing local accuracy, which reduces re-sampling overhead. The most significant gains arise from the look-ahead reward r_2; by incentivizing "future-friendly" trajectories, r_2 not only maximizes efficiency (2.31×) but also enables DARL to surpass the original AR baseline in accuracy (87.69). This confirms that look-ahead rewards effectively mitigate semantic drift, transforming local speculation into a globally consistent optimization.

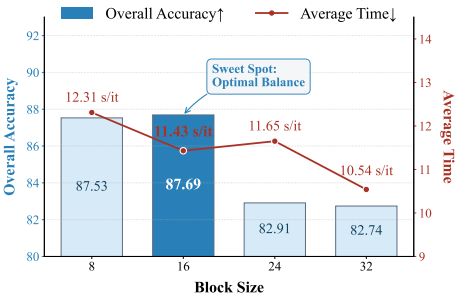

Fig. 6: Effect of Block Size on parsing performance and efficiency. We identify $w = 16$ as the optimal balance, achieving peak accuracy and near-minimal latency.

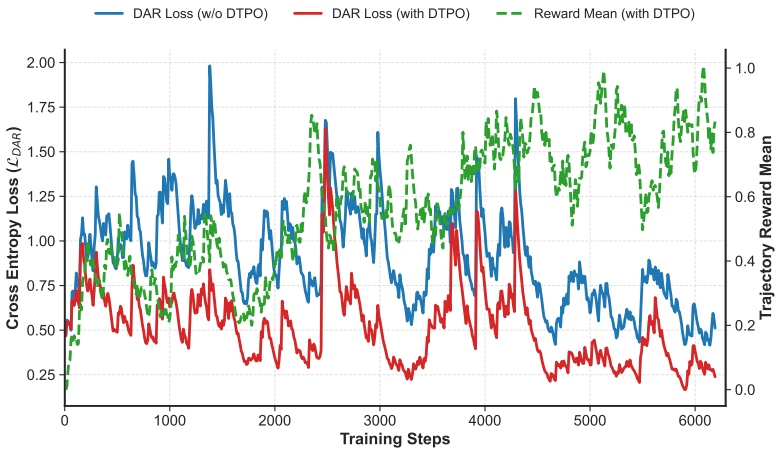

Fig. 7: Training Dynamics of DTPO. Comparison of the Cross-Entropy Loss ($\mathcal{L}_{DAR}$) with and without Diffusion Trajectory Preference Optimization. The integration of DTPO yields a significantly lower and more stable loss convergence, which strongly correlates with the steady increase in the Trajectory Reward Mean.

Training Dynamics and Stability. To further understand the impact of DTPO on the optimization process, we visualize the training cross-entropy loss ($\mathcal{L}_{DAR}$) and the mean trajectory reward in Figure 7. To better observe the impact of DTPO on training stability, we deliberately set a fixed random seed for the training dataloader across different runs. This aligns the data feeding process, a phenomenon corroborated by the synchronous loss spikes observed in both configurations when the models simultaneously encounter inherently challenging document layouts (e.g., dense mathematical formulas or degraded historical scans). However, despite encountering the exact same hard batches, the model equipped with DTPO not only achieves a consistently lower overall loss but also recovers much faster from these spikes. This enhanced stability strongly correlates with the steady increase in the look-ahead trajectory reward, demonstrating that explicitly rewarding the model for foreseeing optimal future trajectories effectively guides the model toward a more robust and rapidly converging generation path.

Impact of Block Size. Figure 6 identifies $w = 16$ as the optimal balance for DARL. While $w = 8$ limits parallelism, $w > 16$ causes a precipitous decline in accuracy as predictive uncertainty grows exponentially with window length, especially in dense segments like formulas. Although $w = 32$ achieves minimal latency per iteration, the resulting trajectory drift from the ground-truth manifold is prohibitive. Thus, $w = 16$ best aligns with the model's semantic receptive field, maximizing acceleration without compromising structural integrity.

Alternatives to LCP Reward. While edit distance is a more generalized metric for measuring sequence similarity, we choose Longest Common Prefix as our reward function because it directly aligns with the continuous token verification mechanism during speculative inference. As shown in Table 4, replacing

Table 4: Comparison of reward functions on OmniDocBench-1.5. ED denotes edit distance, and LCP denotes longest common prefix.

Method	#Params	Speedup ↑	Avg. Time ↓	Avg. Iter. ↓	Overall ↑
dots.ocr (AR)	3B	1.00×	25.85	1091.5	87.57
DARL (ED)	3B	2.13×	12.14	522.4	87.85
DARL (LCP)	3B	2.31×	11.19	485.6	87.69

Table 5: Generalization of DARL across diverse VLLM architectures and scales. The plug-and-play capability of DARL is evaluated across General VLLMs and Expert Models specialized for document parsing.

Type	Model	#Params	Speedup↑	Avg. Time↓	Avg. Iter.↓	Overall↑
General Model	Qwen3-VL	2B	1.00×	30.95	1050.5	82.47
	+DARL		2.33×	13.24	405.2	82.24
	InternVL3.5	2B	1.00×	32.36	1053.8	80.76
	+DARL		2.20×	14.66	421.7	80.19
Expert Model	dots.ocr	3B	1.00×	25.85	1091.5	87.57
	+DARL		2.31×	11.19	485.6	87.69
	Nanonets-OCR2	3B	1.00×	26.56	1159.0	82.76
	+DARL		2.25×	11.80	492.4	82.91

LCP with edit distance yields comparable overall accuracy, with edit distance achieving 87.85 versus LCP's 87.69. However, LCP achieves a higher speedup ratio of 2.31× versus 2.13× and fewer iterations at 485.6 versus 522.4, as it more precisely captures the left-to-right acceptance behavior inherent in our sliding verification process.

5.5 Generalization across VLLMs

To evaluate cross-model compatibility, we apply DARL to diverse backbones, including general-purpose and specialized models. As shown in Table 5, DARL consistently delivers over 2.2× speedup and reduces iterations by ∼60% across all tested architectures. Accuracy fluctuations remain minimal (within ±0.6%), with marginal gains observed in expert models. These results demonstrate that DARL's efficiency gains are robust to different pre-training objectives and model scales, confirming its effectiveness as a general acceleration strategy for VLLM-based document parsing.

5.6 Scaling Law of DARL

To investigate the impact of model scale on DARL's efficiency, we conduct experiments across different parameter sizes using the InternVL3.5 architecture. As shown in Table 6, larger models demonstrate stronger denoising capabilities, resulting in fewer iterations and higher speedup ratios. The 14B model achieves a remarkable 2.78× speedup with only 337.4 average iterations, compared to 2.20× speedup and 421.7 iterations for the 2B variant. This trend suggests that DARL's diffusion-based parallel generation benefits significantly from increased

Table 6: Scaling behavior of DARL across different model sizes of InternVL3.5 on OmniDocBench-1.5.

Method	#Params	Speedup ↑	Avg. Time ↓	Avg. Iter. ↓	Overall ↑
InternVL3.5 (AR)	2B	1.00×	32.36	1053.8	80.76
+DARL	2B	2.20×	14.66	421.7	80.19
InternVL3.5 (AR)	8B	1.00×	42.25	1026.0	86.81
+DARL	8B	2.52×	16.79	376.9	86.28
InternVL3.5 (AR)	14B	1.00×	48.01	1011.6	92.67
+DARL	14B	2.78×	17.27	337.4	92.50

model capacity, as larger models can more accurately predict and refine speculative trajectories in fewer steps. The consistent accuracy across scales further validates the robustness of our sliding verification mechanism.

6 Limitations

Despite its efficiency, DARL has several limitations. First, its performance on highly open-ended tasks (e.g., image captioning) remains unverified, as current rewards based on prefix matching may be too restrictive for high-entropy generation with multiple valid trajectories. Our experiments focus on models up to 14B parameters due to computational constraints. While DARL's architecture-agnostic design suggests it should scale to larger models (e.g., 70B+), empirically validating scaling laws, particularly how the optimal diffusion window size and number of denoising steps vary with model capacity, remains an important open question for future work.

7 Conclusion

In this paper, we introduce **DARL**, a hybrid diffusion-autoregressive framework that achieves 2.3× inference speedup on document parsing tasks while maintaining state-of-the-art accuracy. By introducing Online Monte Carlo Trajectory Generation (OMTG) and Diffusion Trajectory Preference Optimization (DTPO), DARL addresses key challenges in parallel decoding: exposure bias, convergence efficiency, and cross-architecture generalization. Extensive evaluations on OmniDocBench-1.5 and olmOCR-Bench demonstrate DARL's effectiveness in both accelerating inference speed and improving accuracy. While our approach currently focuses on document parsing, the underlying principles of trajectory-guided diffusion and DTPO may extend to other long-form multimodal tasks such as dense image captioning, layout generation [42], and GUI-to-HTML synthesis. Future work will explore scaling to larger MLLMs and integration with tree-structured speculative verification to further reduce latency.

Acknowledgements

This research is supported in part by National Natural Science Foundation of China (Grant No.: 62476093) and the Natural Science Foundation of Guangdong Province (Grant No. 2026A1515012038).

References

1. Austin, J., Johnson, D.D., Ho, J., Tarlow, D., Van Den Berg, R.: Structured denoising diffusion models in discrete state-spaces. Advances in neural information processing systems **34**, 17981–17993 (2021)
2. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., Zhu, K.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)
3. Bie, T., Cao, M., Chen, K., Du, L., Gong, M., Gong, Z., Gu, Y., Hu, J., Huang, Z., Lan, Z., et al.: Llada2.0: Scaling up diffusion language models to 100b. arXiv preprint arXiv:2512.15745 (2025)
4. Cai, T., Li, Y., Geng, Z., Peng, H., Lee, J.D., Chen, D., Dao, T.: Medusa: Simple llm inference acceleration framework with multiple decoding heads. In: Proceedings of the 41st International Conference on Machine Learning. pp. 5209–5235 (2024)
5. Cheng, S., Bian, Y., Liu, D., Jiang, Y., Liu, Y., Zhang, L., Yao, Q., Tian, Z., Wang, W., Guo, Q., et al.: Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation. In: Findings of the Association for Computational Linguistics: ACL 2026. pp. 22058–22075 (2026)
6. Cheng, S., Jiang, Y., Zhou, Z., Liu, D., Wang, T., Zhang, L., Qi, B., Zhou, B.: Sdar-vl: Stable and efficient block-wise diffusion for vision-language understanding. In: Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 28882–28901 (2026)
7. Cui, C., Sun, T., Liang, S., Gao, T., Zhang, Z., Liu, J., Wang, X., Zhou, C., Liu, H., Lin, M., et al.: Paddleocr-vl: Boosting multilingual document parsing via a 0.9 b ultra-compact vision-language model. arXiv preprint arXiv:2510.14528 (2025)
8. Dao, T.: Flashattention-2: Faster attention with better parallelism and work partitioning. arXiv preprint arXiv:2307.08691 (2023)
9. Fu, Y., Bailis, P., Stoica, I., Zhang, H.: Break the sequential dependency of llm inference using lookahead decoding. In: International Conference on Machine Learning. pp. 14060–14079. PMLR (2024)
10. Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al.: The llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024)
11. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
12. Hu, A., Xu, H., Zhang, L., Ye, J., Yan, M., Zhang, J., Jin, Q., Huang, F., Zhou, J.: mplug-docowl2: High-resolution compressing for ocr-free multi-page document understanding. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 5817–5834 (2025)
13. Khanna, S., Kharbanda, S., Li, S., Varma, H., Wang, E., Birnbaum, S., Luo, Z., Miraoui, Y., Palrecha, A., Ermon, S., et al.: Mercury: Ultra-fast language models based on diffusion. arXiv preprint arXiv:2506.17298 **1** (2025)
14. Kou, S., Hu, L., He, Z., Deng, Z., Zhang, H.: Cllms: Consistency large language models. In: International Conference on Machine Learning. pp. 25426–25440. PMLR (2024)

15. Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Zhang, P., Li, Y., Liu, Z., et al.: Llava-onevision: Easy visual task transfer. arXiv preprint arXiv:2408.03326 (2024)
16. Li, S., Kallidromitis, K., Bansal, H., Gokul, A., Kato, Y., Kozuka, K., Kuen, J., Lin, Z., Chang, K.W., Grover, A.: Lavida: A large diffusion language model for multimodal understanding. Advances in Neural Information Processing Systems **38**, 105101–105134 (2026)
17. Li, Y., Wei, F., Zhang, C., Zhang, H.: Eagle-3: Scaling up inference acceleration of large language models via training-time test. In: Advances in Neural Information Processing Systems (2025)
18. Li, Y., Yang, G., Liu, H., Wang, B., Zhang, C.: dots. ocr: Multilingual document layout parsing in a single vision-language model. arXiv preprint arXiv:2512.02498 (2025)
19. Liu, A., He, M., Zeng, S., Zhang, S., Zhang, L., Wu, C., Jia, W., Liu, Y., Zhou, X., Zhou, J.: Wedlm: Reconciling diffusion language models with standard causal attention for fast inference. arXiv preprint arXiv:2512.22737 (2025)
20. Liu, Y., Cao, Y., Li, H., Luo, G., Chen, Z., Wang, W., Liang, X., Qi, B., Wu, L., Tian, C., et al.: Sequential diffusion language models. arXiv preprint arXiv:2509.24007 (2025)
21. Ma, Y., Cao, Y., Ding, W., Zhang, S., Wang, Y., Ivanovic, B., Jiang, M., Pavone, M., Xiao, C.: dvlm-ad: Enhance diffusion vision-language-model for driving via controllable reasoning. arXiv preprint arXiv:2512.04459 (2025)
22. Mandal, S., Talewar, A., Thakuria, S., Ahuja, P., Juvatkar, P.: Nanonets-ocr2: A model for transforming documents into structured markdown with intelligent content recognition and semantic tagging. https://nanonets.com/research/nanonets-ocr-2/ (2025), https://nanonets.com/research/nanonets-ocr-2/
23. Niu, J., Liu, Z., Gu, Z., Wang, B., Ouyang, L., Zhao, Z., Chu, T., He, T., Wu, F., Zhang, Q., Jin, Z., Liang, G., Zhang, R., Zhang, W., Qu, Y., Ren, Z., Sun, Y., Zheng, Y., Ma, D., Tang, Z., Niu, B., Miao, Z., Dong, H., Qian, S., Zhang, J., Chen, J., Wang, F., Zhao, X., Wei, L., Li, W., Wang, S., Xu, R., Cao, Y., Chen, L., Wu, Q., Gu, H., Lu, L., Wang, K., Lin, D., Shen, G., Zhou, X., Zhang, L., Zang, Y., Dong, X., Wang, J., Zhang, B., Bai, L., Chu, P., Li, W., Wu, J., Wu, L., Li, Z., Wang, G., Tu, Z., Xu, C., Chen, K., Qiao, Y., Zhou, B., Lin, D., Zhang, W., He, C.: Mineru2.5: A decoupled vision-language model for efficient high-resolution document parsing (2025), https://arxiv.org/abs/2509.22186
24. Ouyang, L., Qu, Y., Zhou, H., Zhu, J., Zhang, R., Lin, Q., Wang, B., Zhao, Z., Jiang, M., Zhao, X., et al.: Omnidocbench: Benchmarking diverse pdf document parsing with comprehensive annotations. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 24838–24848 (2025)
25. Poznanski, J., Soldaini, L., Lo, K.: olmocr 2: Unit test rewards for document ocr. arXiv preprint arXiv:2510.19817 (2025)
26. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018)
27. Santilli, A., Severino, S., Postolache, E., Maiorca, V., Mancusi, M., Marin, R., Rodolà, E.: Accelerating transformer inference for translation via parallel decoding. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 12336–12355 (2023)
28. Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024)

29. Team, H.V., Lyu, P., Wan, X., Li, G., Peng, S., Wang, W., Wu, L., Shen, H., Zhou, Y., Tang, C., et al.: Hunyuanocr technical report. arXiv preprint arXiv:2511.19575 (2025)
30. Wang, B., Wu, F., Ouyang, L., Gu, Z., Zhang, R., Xia, R., Shi, B., Zhang, B., He, C.: Image over text: Transforming formula recognition evaluation with character detection matching. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 19681–19690 (2025)
31. Wang, W., Gao, Z., Gu, L., Pu, H., Cui, L., Wei, X., Liu, Z., Jing, L., Ye, S., Shao, J., et al.: Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. arXiv preprint arXiv:2508.18265 (2025)
32. Wang, X., Xu, C., Jin, Y., Jin, J., Zhang, H., Deng, Z.: Diffusion llms can do faster-than-ar inference via discrete diffusion forcing. arXiv preprint arXiv:2508.09192 (2025)
33. Wang, Y., Luo, X., Wei, F., Liu, Y., Zhu, Q., Zhang, X., Yang, Q., Xu, D., Che, W.: Make some noise: Unlocking language model parallel inference capability through noisy training. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. pp. 12914–12926 (2024)
34. Wei, H., Sun, Y., Li, Y.: Deepseek-ocr: Contexts optical compression. arXiv preprint arXiv:2510.18234 (2025)
35. Yang, L., Tian, Y., Li, B., Zhang, X., Shen, K., Tong, Y., Wang, M.: Mmada: Multimodal large diffusion language models. arXiv preprint arXiv:2505.15809 (2025)
36. Ye, J., Xie, Z., Zheng, L., Gao, J., Wu, Z., Jiang, X., Li, Z., Kong, L.: Dream 7b: Diffusion large language models. arXiv preprint arXiv:2508.15487 (2025)
37. You, Z., Nie, S., Zhang, X., Hu, J., Zhou, J., Lu, Z., Wen, J.R., Li, C.: Llada-v: Large language diffusion models with visual instruction tuning. arXiv preprint arXiv:2505.16933 (2025)
38. Zeng, L., Yao, J., Liao, B., Tao, H., Liu, W., Wang, X.: Diffusionvl: Translating any autoregressive models into diffusion vision language models. arXiv preprint arXiv:2512.15713 (2025)
39. Zhang, J., Liu, Y., Wu, Z., Pang, G., Ye, Z., Zhong, Y., Ma, J., Wei, T., Xu, H., Chen, W., Wang, Z., Ji, Q., Zhou, F., Zhang, Q., Hu, Y., Liu, J., Li, Z., Zhang, Z., Liu, Q., Bai, X.: Monkeyocr v1.5 technical report: Unlocking robust document parsing for complex patterns (2025), https://arxiv.org/abs/2511.10390
40. Zhang, P., Jiang, J., Liu, Y., Jin, L.: MSDS: A Large-Scale Chinese Signature and Token Digit String Dataset for Handwriting Verification. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 35, pp. 36507–36519 (2022)
41. Zhang, P., Xu, H., Zhang, J., Zheng, X., Xu, G., Zhang, Y., Liu, J., Yang, Z., Zhou, W., Jin, L.: OCRGenBench: A Comprehensive Benchmark for Evaluating OCR Generative Capabilities. arXiv preprint arXiv:2507.15085 (2025)
42. Zhang, P., Zhang, J., Cao, J., Li, H., Jin, L.: Smaller But Better: Unifying Layout Generation with Smaller Large Language Models. International Journal of Computer Vision (IJCV) **133**, 3891–3917 (2025)
43. Zhao, S., Gupta, D., Zheng, Q., Grover, A.: d1: Scaling reasoning in diffusion large language models via reinforcement learning. arXiv preprint arXiv:2504.12216 (2025)
44. Zhao, S., Liu, M., Huang, J., Liu, M., Wang, C., Liu, B., Tian, Y., Pang, G., Bell, S., Grover, A., et al.: Inpainting-guided policy optimization for diffusion large language models. arXiv preprint arXiv:2509.10396 (2025)
45. Zhong, X., ShafieiBavani, E., Jimeno Yepes, A.: Image-based table recognition: data, model, and evaluation. In: European conference on computer vision. pp. 564–580. Springer (2020)

46. Zhu, F., Wang, R., Nie, S., Zhang, X., Wu, C., Zhou, J., Lin, Y., Wen, J.R., Li, C.: Llada 1.5: Variance-reduced preference optimization for large language diffusion models. In: Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 11425–11460 (2026)