

MinerU-Diffusion: Rethinking Document OCR as Inverse Rendering via Diffusion Decoding

Hejun Dong^{1*}, Junbo Niu^{2*}, Bin Wang^{1✉‡}, Weijun Zeng², Wentao Zhang^{2✉}, Conghui He^{1✉}

¹Shanghai Artificial Intelligence Laboratory, OpenDataLab, ²Peking University

Optical character recognition (OCR) has evolved from line-level transcription to structured document parsing, requiring models to recover long-form sequences containing layout, tables, and formulas. Despite recent advances in vision-language models, most existing systems rely on autoregressive decoding, which introduces sequential latency and amplifies error propagation in long documents. In this work, we revisit document OCR from an inverse rendering perspective, arguing that left-to-right causal generation is an artifact of serialization rather than an intrinsic property of the task. Motivated by this insight, we propose **MinerU-Diffusion**, a unified diffusion-based framework that replaces autoregressive sequential decoding with parallel diffusion denoising under visual conditioning. MinerU-Diffusion employs a block-wise diffusion decoder and an uncertainty-driven curriculum learning strategy to enable stable training and efficient long-sequence inference. Extensive experiments demonstrate that MinerU-Diffusion consistently improves robustness while achieving up to 3.2× faster decoding compared to autoregressive baselines. Evaluations on the proposed Semantic Shuffle benchmark further confirm its reduced dependence on linguistic priors and stronger visual OCR capability.

* Equal contribution ✉ Corresponding author ‡ Project leader:

Correspondence: Conghui He, heconghui@pjlab.org.cn

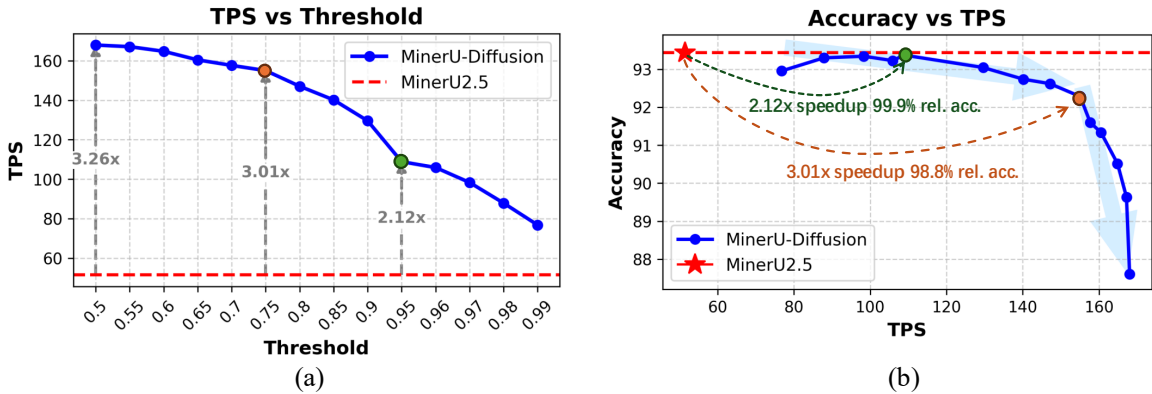
Code: <https://github.com/opendatalab/MinerU-Diffusion>

Model: <https://huggingface.co/opendatalab/MinerU-Diffusion-V1-0320-2.5B>

Date: March 25, 2026

Contents

1	Introduction	3
2	Related Works	4
3	Method	5
3.1	Problem Formulation: Inverse Rendering via Diffusion	5
3.2	MinerU-Diffusion: Unified Diffusion Architecture for OCR	6
3.3	Two-Stage Curriculum Learning with Uncertainty-Driven Refinement	8
3.3.1	Stage I: Diversity-Driven Foundational Learning	8
3.3.2	Stage II: Uncertainty-Driven Boundary Refinement	8
4	Experiments	9
4.1	Experimental Setups	9
4.1.1	Data	9
4.1.2	Models and Optimization	9
4.1.3	Evaluation	10
4.2	Full-Document Parsing Task Results	10
4.3	Element-Specific Parsing Task Results	11
4.3.1	Table Recognition	11
4.3.2	Formula Recognition	11
4.4	Ablation Study	11
4.4.1	Confidence Threshold vs. Decoding Parallelism	11
4.4.2	Decoding Parallelism vs Accuracy	12
4.4.3	Decoding Strategy Comparisons	13
4.4.4	Full-Attn vs Block-Attn	13
4.4.5	Two-Stage Curriculum Learning	14
4.5	Semantic Shuffle Analysis	14
5	Conclusion	15
A	Training Details	20
A.1	Stage-0: Modality Alignment	20
A.2	Stage-1: Large-Scale OCR Adaptation	20
A.3	Stage-2: Hard-Case Specialization	21
A.4	Training Dynamics	21
A.5	Data Augmentation	21
A.6	Practical Notes	21
B	Prompt Templates	22
B.1	Layout Detection	23
B.2	Text Recognition	23
B.3	Formula Recognition	23
B.4	Table Recognition	23
C	Full-Attn vs Block-Attn	24
C.1	Why Full-Attn Is Inferior	24
C.2	Generation Quality under Length Mismatch	24
D	Qualitative Examples	27
D.1	Complete Recognition Examples	27
D.2	Diffusion Decoding Examples	27



MinerU-Diffusion

Original Image

Figure 4: Performance and efficiency scaling of MEM1 (trained on 2-objective QA) with the number of objectives in multi-objective tasks. MEM1 outperforms the other models and baselines while having an almost constant scaling in memory usage. Note that at 16-objective, the context of baseline models does not increase anymore since their model performance has degraded (some collapsed).

Generated Text

Text Recognition: Figure 4: Performance and efficiency scaling of MEM1 (trained on 2-objective QA) with the number of objectives in multi-objective tasks. **MEM1 outperforms the other models and baselines** while having an almost constant scaling in memory usage. Note that at 16-objective, the context of baseline models does not increase anymore since their model performance has degraded (some collapsed).

54/81 masks decoded

Diffusion

54/81 • Remaining: 27 • Step: 4 (+22)

(c)

Figure 1: **(a)** The confidence threshold controls decoding parallelism in MinerU-Diffusion. Compared to MinerU2.5, our method achieves up to 3.26× speedup. **(b)** MinerU-Diffusion maintains a strong accuracy–efficiency trade-off, achieving 2.12× speedup with 99.9% and 3.01× speedup with 98.8% relative accuracy. **(c)** Diffusion decoding progressively reconstructs structured text from masked tokens under visual conditioning: black tokens are confirmed, red tokens are being updated, and yellow tokens remain masked, enabling parallel generation with global consistency, in contrast to autoregressive left-to-right decoding.

1 Introduction

In recent years, Vision-Language Models (VLMs) [1, 6, 44, 59, 11, 15, 56, 55, 28] have become the dominant paradigm for document Optical Character Recognition (OCR) [29, 17, 16, 47, 7]. These models encode textual images into visual representations and generate structured text through left-to-right autoregressive decoding [41], achieving strong performance across benchmarks [32, 9, 42]. Despite architectural unification and scaling, the decoding process remains strictly sequential. This design introduces efficiency and reliability bottlenecks when parsing long documents and complex layouts, particularly in highly structured scenarios such as tables and formulas [32].

From the perspective of task formulation, a high-quality OCR system should primarily depend on authentic visual evidence to perform character-level recognition, rather than relying on semantic completion from a language model. However, autoregressive formulations implicitly cast OCR as language-conditioned reconstruction, where textual outputs are generated under strong linguistic

priors. When visual signals are weak or semantic constraints are disrupted, models tend to over-rely on these priors, leading to semantic hallucinations and cumulative errors. Experiments on the Semantic Shuffle benchmark confirm that disrupting semantic structure causes substantial performance degradation in AR-based OCR systems. This fragility stems not merely from data or training strategy, but from the causal factorization inherent in autoregressive decoding.

In contrast, Diffusion Language Models (DLMs) [14, 10, 37, 27, 52], based on discrete diffusion processes, provide a modeling paradigm better aligned with the structural characteristics of document OCR tasks. Masked diffusion models assume conditional independence among tokens given partially observed sequences and visual inputs [53], which is reasonable in OCR scenarios where the mapping between image content and text is largely deterministic with limited semantic ambiguity. This property allows models to exploit the local consistency of visual signals and perform parallel decoding of long textual segments while maintaining global coherence and accuracy. Compared with autoregressive generation for open-ended text tasks [19], diffusion decoding naturally matches the deterministic nature of document OCR [32]. Moreover, DLMs support parallel multi-token updates, significantly improving inference efficiency for long-document parsing. Although current diffusion-based VLMs [5, 53, 20, 50] still encounter issues such as instability in long sequences, repetition, and hallucination in high-resolution settings, these limitations can be progressively alleviated through improved model design and training strategies, making masked diffusion a promising and principled alternative for accurate and efficient document OCR modeling.

Motivated by these observations, we formulate document OCR explicitly as an inverse rendering problem under visual conditioning, shifting from autoregressive causal decoding to diffusion-based decoding. We propose MinerU-Diffusion, a unified diffusion-based parsing framework tailored for document OCR. Centered on block-wise diffusion decoding [2, 4, 46] and coupled with an uncertainty-driven curriculum learning strategy, MinerU-Diffusion enables global parallel reconstruction for document OCR. While maintaining high recognition accuracy, MinerU-Diffusion significantly improves long-sequence inference efficiency and effectively mitigates semantic hallucination and cumulative error propagation observed in autoregressive decoding. Extensive experiments demonstrate that MinerU-Diffusion achieves performance on par with state-of-the-art approaches across multiple challenging document parsing benchmarks and semantic perturbation settings, while attaining a more favorable balance among recognition accuracy, robustness, and decoding efficiency. More examples are provided in [Appendix D](#).

2 Related Works

Vision-Language Models for Document OCR. Driven by large-scale pre-training, document OCR has evolved from traditional modular pipelines [7, 22, 33, 43] toward end-to-end Vision-Language Models (VLMs) that generate structured text directly from pixels [17, 16, 34, 25, 6, 1, 11, 47]. Representative systems such as MinerU2.5 [29] and PaddleOCR-VL [7] formulate document OCR as sequence generation and rely on autoregressive (AR) decoders to produce text token by token. While this unified paradigm simplifies traditional pipelines and improves cross-domain generalization, it inherits structural limitations from causal left-to-right decoding. Inference latency scales linearly with output length, limiting efficiency in long-document scenarios. Moreover, the strong coupling between generation order and linguistic context encourages reliance on language priors, which may compromise robustness when visual evidence is ambiguous or semantic structure is disrupted. These limitations motivate alternative decoding paradigms that enable global dependency modeling and reduce dependence on unidirectional factorization.

Masked Diffusion Language Models. Diffusion Language Models (DLMs) [14, 10, 37, 27, 52] offer a non-autoregressive generative framework based on discrete diffusion processes. In masked diffusion, tokens in a clean sequence x_0 are progressively replaced by mask tokens [MASK] under a continuous

corruption schedule $t \in [0, 1]$, yielding a noised sequence x_t . The forward process is defined as:

$$q(x_t | x_0) = \prod_{i=1}^n \text{Cat}\left(x_t^i; (1-t)\delta_{x_0^i} + t\delta_{[\text{MASK}]}\right). \tag{1}$$

The corresponding training objective can be derived from maximum likelihood estimation [36, 35, 57, 31], resulting in an evidence lower bound (ELBO) on $\log p_\theta(x)$:

$$\mathcal{J}_{\text{full}}(x_0, Q, \theta) = \int_0^1 \frac{1}{t|x_0|} \mathbb{E}_{q(x_t|x_0)} \left[\sum_{i:x_t^i=[\text{MASK}]} \log p_\theta(x_0^i | x_t, Q) \right] dt, \tag{2}$$

where Q denotes the prompt and $|x_0|$ is the sequence length.

To enhance scalability, block diffusion models [2, 3, 48, 46] introduce block-wise attention mechanisms that balance the optimization stability of autoregressive (AR) training with the parallel sampling efficiency of diffusion-based generation. Their structured attention patterns naturally enable KV-cache reuse, alleviating the inference latency commonly observed in full-attention DLMS [12, 21, 23, 2, 54, 49].

Masked diffusion models are structurally well aligned with the characteristics of document OCR tasks. In document OCR, the target text typically exhibits a near-deterministic mapping to the textual content present in the image, with limited semantic ambiguity. Under this setting, the conditional independence assumption underlying masked diffusion—that each token can be predicted independently given the input and partially observed sequence, as illustrated in Figure 2—becomes considerably more reasonable than in open-ended language generation [24, 58, 27]. This alignment allows the model to decode long-range text spans in parallel without sacrificing consistency.

Therefore, diffusion-based decoding is not merely an efficiency-oriented alternative to autoregressive methods. Instead, it constitutes a modeling paradigm that is inherently better matched to the structural properties of OCR, offering both theoretical justification and practical advantages for large-scale text recognition.

3 Method

3.1 Problem Formulation: Inverse Rendering via Diffusion

We model document OCR [32] as the inverse rendering of a unified structured token sequence:

$$y = \left(y^{(1)}, \dots, y^{(L)}\right) \in \mathcal{V}^L, \tag{3}$$

where \mathcal{V} is a shared vocabulary encompassing text symbols, layout markers, table delimiters, and mathematical operators. This unified representation enables the encoding of heterogeneous document elements—such as paragraphs, tables, formulas, and reading order—within a single sequential interface.

Although serialized as a one-dimensional sequence, y corresponds to an underlying two-dimensional document structure. The statistical dependencies between tokens arise primarily from spatial arrangement, layout regularities, and formatting constraints, rather than from an intrinsic causal generation order. Therefore, the serialization order should be viewed as an implementation artifact introduced for representation convenience, rather than a fundamental property of the document generation process. In this sense, OCR output is more naturally modeled as a spatially coupled discrete random field, rather than a strictly directional sequence.

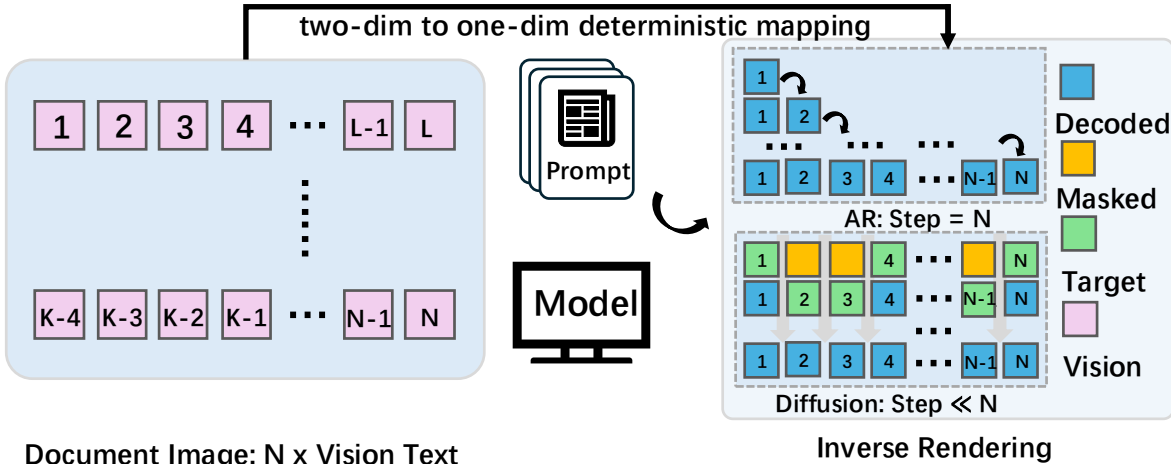


Figure 2: Overview of the document OCR inverse rendering process via different decoding methods.. The model maps a 2D document image to a 1D token sequence for decoding through autoregressive and diffusion-based methods.

Document OCR can be framed as posterior inference over a latent structured token sequence, where the input document x serves as partial and noisy evidence, constraining both token identities and spatial positioning. Traditional OCR systems typically parameterize the posterior through autoregressive decompositions [29, 7, 47, 16], which impose a fixed causal order and limit the ability to model document structure globally. In contrast, as illustrated in Figure 2, diffusion-based decoding methods [2, 27] introduce a discrete diffusion process that avoids a fixed causal ordering, enabling global iterative refinement under visual conditioning, which naturally aligns with the structural properties of OCR tasks. The conditional independence assumption inherent in masked diffusion models [27]—i.e., each token can be independently predicted given the input and partially observed sequence—becomes particularly reasonable in OCR, where the target text has a one-to-one correspondence with the text in the image. Through multiple denoising iterations, diffusion models jointly update all tokens across the entire sequence, circumventing the single-pass update limitation of autoregressive decoding, and providing a more structurally aligned approximation to the posterior distribution $p(y | x)$.

3.2 MinerU-Diffusion: Unified Diffusion Architecture for OCR

A straightforward implementation of discrete diffusion for OCR is to apply a full-attention dLM [27, 52, 54] over the entire token sequence at each denoising step. However, when scaling to long structured documents, such a design suffers from fundamental structural and computational limitations [49].

Full self-attention incurs quadratic complexity $O(L^2)$ with respect to sequence length L , making it computationally expensive for long structured documents with thousands of tokens. Moreover, full-attention diffusion operates globally, which introduces positional instability as early denoising errors can propagate across the sequence. Unlike autoregressive decoding [1], global diffusion lacks structural anchoring and is prone to long-range drift. Additionally, document structures exhibit strong locality, with high intra-region consistency and weak long-range dependencies. Full-attention unnecessarily couples independent regions, conflicting with the spatially constrained posterior structure of document OCR.

These observations suggest that purely full-attention diffusion is not structurally aligned with OCR. To address these limitations, we introduce MinerU-Diffusion, a block-attention [2, 4] dVLM that incorporates structural locality into posterior refinement, as illustrated in Figure 3. The output

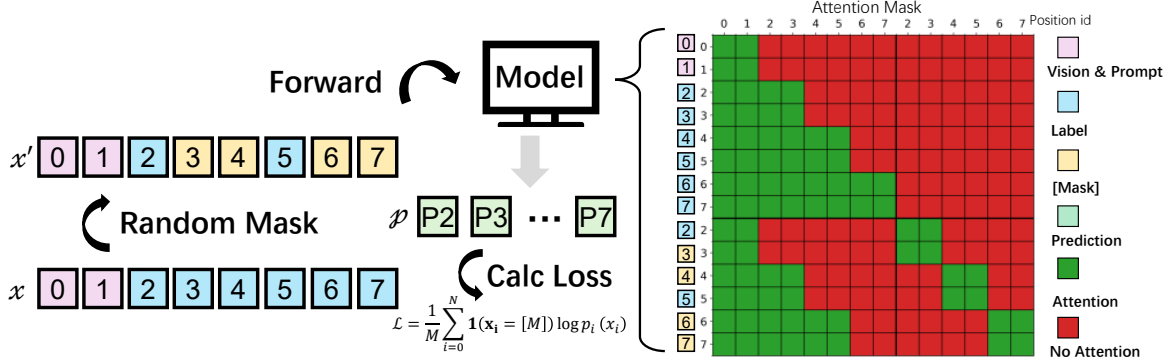


Figure 3: Training of MinerU-Diffusion. **Left:** the target token sequence is randomly masked to form a partially observed input, and the model predicts only the masked positions under visual and prompt conditioning. **Right:** the structured block-attention mask used during training, where tokens attend bidirectionally within each block and causally to all preceding blocks, enabling parallel diffusion refinement within blocks while preserving coarse autoregressive structure across blocks.

sequence is partitioned into B contiguous blocks:

$$y = (y^{(1)}, \dots, y^{(B)}), \quad y^{(b)} \in \mathcal{V}^{L'}, \quad L = BL'. \quad (4)$$

Rather than modeling the entire sequence as a monolithic denoising problem, we factorize the conditional posterior:

$$p_\theta(y | x) = \prod_{b=1}^B p_\theta(y^{(b)} | y^{(<b)}, x), \quad (5)$$

where $y^{(<b)}$ denotes all preceding blocks. Within each block, diffusion operates locally:

$$p_\theta(y_{t-1}^{(b)} | y_t^{(b)}, y^{(<b)}, x). \quad (6)$$

This hybrid factorization introduces coarse-grained autoregressive structure across blocks and parallel diffusion refinement within blocks. Block boundaries serve as structural anchors, preventing long-range alignment drift, while preserving parallel efficiency inside each block.

At each denoising step, a structured attention mask is applied. Tokens can attend fully to tokens within the same block, causally to tokens in preceding blocks, and not to tokens in future blocks, as shown in Figure 3. Formally, the attention mask \mathcal{M}_{ij} is defined as:

$$\mathcal{M}_{ij} = \begin{cases} 1, & \text{if } b(i) = b(j), \\ 1, & \text{if } b(j) < b(i), \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

where $b(i)$ denotes the block index of token i . This structured masking reduces unnecessary global coupling, stabilizes positional alignment, and ensures that decoding errors remain locally bounded.

MinerU-Diffusion conditions the diffusion process on native-scale visual features [29, 30, 44, 8], ensuring that posterior refinement remains grounded in visual evidence. Compared to full-attention diffusion, block-attention reduces complexity from $O(L^2)$ to $O(BL^2)$. The causal structure across blocks enables efficient KV-caching during inference, while maintaining parallel decoding within each block. Overall, MinerU-Diffusion provides a structurally grounded and computationally scalable diffusion architecture tailored for document OCR.

3.3 Two-Stage Curriculum Learning with Uncertainty-Driven Refinement

To fully leverage large-scale heterogeneous data and alleviate the performance bottleneck caused by noisy labels, we propose a two-stage curriculum learning framework to train the MinerU-Diffusion. Compared with autoregressive (AR) models that generate tokens in a fixed order, diffusion models can decode tokens in any order, introducing more intricate inter-token dependencies that make training less stable and more sensitive to noise—often requiring larger datasets and more carefully tuned training strategies [37, 26]. Furthermore, random masking modeling typically achieves lower data utilization efficiency compared to AR models at the same data scale. This is because AR models make predictions based on the complete prefix information at each step, while random masking disperses the supervisory signal, reducing the density of effective conditional information. To address this, we divide the dataset \mathcal{D} into two subsets: \mathcal{D}_{base} , which is easier to train, and \mathcal{D}_{hard} , which is more challenging. We first train on \mathcal{D}_{base} to establish the foundational structure understanding, then fine-tune on \mathcal{D}_{hard} to improve the model’s robustness to noisy labels and boundary precision. This framework effectively resolves the optimization complexity and low data utilization efficiency faced by diffusion models in any-order modeling.

3.3.1 Stage I: Diversity-Driven Foundational Learning

In the first stage, our goal is to establish robust foundational representations and general parsing abilities across multiple document understanding tasks. To this end, we construct a large-scale, diverse, and balanced dataset \mathcal{D}_{base} through data curation and automated annotation refinement.

This dataset satisfies:

$$\mathcal{D}_{base} \sim p_{div}(x), \tag{8}$$

where $p_{div}(x)$ denotes a high-entropy data distribution covering diverse layouts, languages, document types, and visual styles.

This stage emphasizes broad visual-semantic alignment, stable feature learning, and robust cross-domain generalization. Although \mathcal{D}_{base} contains moderate annotation noise, its large scale and diversity enable effective representation learning. From an optimization perspective, training on \mathcal{D}_{base} yields a relatively smooth loss landscape, facilitating stable convergence.

3.3.2 Stage II: Uncertainty-Driven Boundary Refinement

After Stage I convergence, the model acquires strong general capabilities. However, performance is constrained by noisy supervision and limited exposure to complex edge cases. To overcome this limitation, we introduce an uncertainty-driven curriculum refinement stage.

Hard Case Mining via Inference Consistency. For each unlabeled or weakly labeled sample x , we perform T stochastic inference passes:

$$\{\hat{y}^{(t)}\}_{t=1}^T = \{f_{\theta}(x; \zeta_t)\}_{t=1}^T, \tag{9}$$

where ζ_t represents stochastic factors such as sampling temperature or dropout.

We define a task-specific consistency metric $S(\cdot, \cdot)$: (1) PageIoU for layout, (2) CDM for formula, (3) TEDS for tables.

The mean consistency score is computed as:

$$C(x) = \frac{2}{T(T-1)} \sum_{i < j} S(\hat{y}^{(i)}, \hat{y}^{(j)}). \tag{10}$$

Low values of $C(x)$ indicate high prediction uncertainty.

We select hard samples as:

$$\mathcal{D}_{hard} = \{x \mid C(x) < \tau\}, \quad (11)$$

where τ is a task-dependent threshold.

Expert Refinement and Dataset Construction. Samples in \mathcal{D}_{hard} are processed through an AI-assisted human annotation pipeline to produce high-precision labels:

$$\tilde{\mathcal{D}}_{hard} = \{x_i, \tilde{y}_i\}_{i=1}^M. \quad (12)$$

The final fine-tuning dataset is constructed as:

$$\mathcal{D}_{SFT} = \tilde{\mathcal{D}}_{hard} \cup \alpha \mathcal{D}_{rand}, \quad (13)$$

where \mathcal{D}_{rand} is a randomly sampled subset of \mathcal{D}_{base} , and α controls the regularization ratio.

The Stage II optimization objective is defined as:

$$\mathcal{L}_{hard}(\theta) = \mathbb{E}_{(x,y) \sim \mathcal{D}_{SFT}} [w(x) \ell(f_{\theta}(x), y)], \quad (14)$$

where the sample weight is:

$$w(x) = 1 + \beta(1 - C(x)), \quad (15)$$

with β controlling the emphasis on hard samples.

This adaptive weighting further encourages the model to focus on decision-boundary regions. The progressive curriculum mitigates optimization instability and performance ceilings caused by diffusion’s any-order modeling by organizing data from broad to difficult, enabling MinerU-Diffusion to overcome annotation noise and long-tail complexity for superior real-world document OCR performance.

4 Experiments

In this section, we present a comprehensive quantitative evaluation of MinerU-Diffusion to demonstrate its effectiveness in document OCR tasks. More ablation studies and qualitative examples are provided in [Appendix C](#) and [Appendix D](#).

4.1 Experimental Setups

4.1.1 Data

All meta training data are derived from the MinerU2.5 dataset [29], with a total volume of approximately 7.5M samples. The dataset primarily focuses on Chinese and English document parsing tasks. Therefore, no dedicated evaluation was conducted for low-resource languages.

4.1.2 Models and Optimization

Our experiments adopt a block-wise attention dVLM architecture. Specifically, we employ the SDAR-1.7B-Chat-b32 [4] with a block size of 32. The remaining components follow the MinerU2.5 architecture, except that M-RoPE [44, 38] is removed. We first fine-tune the MinerU-Diffusion on the LLaVA-NeXT dataset [19] for visual question answering (VQA) tasks. Based on this initialization, we further conduct specialized training for document optical character recognition (OCR). Additional optimization details are provided in [Appendix A](#).

MinerU-Diffusion: Rethinking Document OCR as Inverse Rendering via Diffusion Decoding

Type	Methods	Params.	Overall ↑	Text ↓	Formula ↑	Table _{TEDS} ↑	Table _{TEDS-S} ↑	Reading Order ↓	GT Layout
Pipeline	Mineru2-pipeline [43]	-	75.51	0.209	76.55	70.90	79.11	0.225	×
	PP-StructureV3 [7]	-	86.73	0.073	85.79	81.68	89.48	0.073	×
AR	Qwen2.5-VL-72B [40]	72B	87.02	0.094	88.27	82.15	86.22	0.102	×
	Gemini-2.5 Pro [6]	-	88.03	0.075	85.82	85.71	90.29	0.097	×
	MinerU2-VLM [43]	0.9B	85.56	0.078	80.95	83.54	87.66	0.086	×
	Nanonets-OCR-s [25]	3B	85.59	0.093	85.90	80.14	85.57	0.108	×
	MonkeyOCR-pro-1.2B [17]	1.9B	86.96	0.084	85.02	84.24	89.02	0.130	×
	MonkeyOCR-pro-3B [17]	3.7B	88.85	0.075	87.25	86.78	90.63	0.128	×
	dots.ocr [16]	3B	88.41	0.048	83.22	86.78	90.62	0.053	×
	MinerU2.5 [29]	1.2B	90.67	0.047	88.46	88.22	92.38	0.044	×
	PaddleOCR-VL [7]	0.9B	92.56	0.035	91.43	89.76	93.52	0.043	×
	dLM	MinerU-Diffusion	2.5B	88.94	0.061	86.41	86.50	90.29	0.059
AR	MinerU2.5 [29]	1.2B	93.44	0.025	91.98	90.84	95.10	-	✓
	PaddleOCR-VL [7]	0.9B	93.91	0.021	92.13	91.70	95.42	-	✓
dLM	MinerU-Diffusion	2.5B	93.37	0.028	91.92	91.00	94.86	-	✓

Table 1: Comprehensive evaluation of document parsing on OmniDocBench v1.5. ↑ denotes higher is better, ↓ denotes lower is better.

4.1.3 Evaluation

All experiments are conducted with a block size of 32 and a dynamic decoding strategy. The decoding threshold is set to $T = 0.95$, with $\text{top-}k = 0$, temperature = 1.0, and $\text{top-}p = 1.0$.

For full document parsing and layout analysis, we evaluate our models on OmniDocBench v1.5 [32]. Table recognition is assessed using CC-OCR [51] and OCRBench v2 [9], while formula recognition is evaluated on UniMER-Test [42]. The inference prompts for these tasks are summarized in Appendix B.

Unless otherwise stated, all OmniDocBench results use the same inference setting as above and follow the latest evaluation protocol on 1,355 pages with hybrid matching. On OmniDocBench, text is evaluated by edit distance (↓), formulas by CDM (↑), and tables by TEDS / TEDS-S (↑). The Overall score is computed from the three core parsing metrics:

$$\text{Overall} = \frac{(1 - \text{Text}^{\text{Edit}}) \times 100 + \text{Formula}^{\text{CDM}} + \text{Table}^{\text{TEDS}}}{3}.$$

Therefore, Reading Order and Table_{TEDS-S} are auxiliary metrics and are not included in Overall. Reading Order is only reported under the *w/o GT Layout* setting, where the model must jointly predict layout and content from the full page. In contrast, under *w/ GT Layout*, oracle layout regions are provided, so the evaluation mainly isolates recognition quality after removing layout detection errors.

4.2 Full-Document Parsing Task Results

As shown in Table 1, under the fully automatic setting without GT Layout, MinerU-Diffusion achieves an Overall score of 88.94, outperforming most AR-based models and demonstrating strong end-to-end parsing capability without relying on oracle layout information. When evaluated with GT Layout, MinerU-Diffusion further improves to 93.37 Overall, reaching a score that is close to top-tier AR-based systems and indicating high competitiveness in overall parsing performance. Meanwhile, the sizable gap between the two settings suggests that layout understanding remains a key bottleneck: MinerU-Diffusion can leverage accurate layout signals effectively, and its remaining weaknesses are primarily attributed to layout analysis, leaving clear room for improvement in layout prediction to further close the gap in fully automatic parsing.

To complement the aggregate results in Table 1, Table 2 further breaks down the text edit distance over the nine OmniDocBench page types. Lower values indicate better page-level parsing quality.

MinerU-Diffusion: Rethinking Document OCR as Inverse Rendering via Diffusion Decoding

Type	Methods	Slides	Academic Papers	Book	Textbook	Exam Papers	Magazine	Newspaper	Notes	Financial Report	GT Layout
Pipeline	MinerU2-pipeline [43]	0.4244	0.0230	0.2628	0.1224	0.0822	0.3950	0.0736	0.2603	0.0411	×
	PP-StructureV3 [7]	0.0794	0.0236	0.0415	0.1107	0.0945	0.0722	0.0617	0.1236	0.0181	×
AR	Qwen2.5-VL-72B [40]	0.0422	0.0801	0.0586	0.1146	0.0681	0.0964	0.2380	0.1232	0.0264	×
	Gemini-2.5 Pro [6]	0.0326	0.0182	0.0694	0.1618	0.0937	0.0161	0.1347	0.1169	0.0169	×
	MinerU2-VLM [43]	0.0745	0.0104	0.0357	0.1276	0.0698	0.0652	0.1831	0.0803	0.0236	×
	Nanonets-OCR-s [25]	0.0551	0.0578	0.0606	0.0931	0.0834	0.0917	0.1965	0.1606	0.0395	×
	MonkeyOCR-pro-1.2B [17]	0.0961	0.0354	0.0530	0.1110	0.0887	0.0494	0.0995	0.1686	0.0198	×
	MonkeyOCR-pro-3B [17]	0.0879	0.0459	0.0517	0.1067	0.0726	0.0482	0.0937	0.1141	0.0211	×
	dots.ocr [16]	0.0290	0.0231	0.0433	0.0788	0.0467	0.0221	0.0667	0.1116	0.0076	×
	MinerU2.5 [29]	0.0304	0.0213	0.0322	0.0590	0.0501	0.0388	0.0570	0.1036	0.0071	×
	PaddleOCR-VL [7]	0.0260	0.0168	0.0320	0.0691	0.0447	0.0194	0.0357	0.0593	0.0079	×
dLM	MinerU-Diffusion	0.0440	0.0244	0.0444	0.0677	0.0580	0.0474	0.0863	0.1618	0.0104	×
AR	MinerU2.5 [29]	0.0164	0.0184	0.0144	0.0259	0.0273	0.0126	0.0288	0.0559	0.0045	✓
	PaddleOCR-VL [7]	0.0173	0.0224	0.0145	0.0349	0.0292	0.0129	0.0311	0.0455	0.0177	✓
dLM	MinerU-Diffusion	0.0253	0.0226	0.0306	0.0316	0.0281	0.0181	0.0336	0.0528	0.0053	✓

Table 2: Breakdown of document parsing on OmniDocBench v1.5 by page type, reported in text edit distance (↓).

4.3 Element-Specific Parsing Task Results

4.3.1 Table Recognition

We evaluate MinerU-Diffusion’s table recognition performance on the table subsets of CC-OCR [51] and OCRBench v2 [9], as shown in Table 3. MinerU-Diffusion is competitive with other AR-based models on both table subsets. It achieves 81.18/88.66 (TEDS/TEDS-S) on OCRBench v2 and 73.77/82.06 on CC-OCR, matching or surpassing several strong AR baselines in the same evaluation setting and remaining close to the top group. In particular, the strong TEDS-S scores indicate that MinerU-Diffusion can reliably preserve table structure during diffusive decoding, demonstrating solid end-to-end table parsing capability and overall competitiveness among AR approaches.

4.3.2 Formula Recognition

We evaluate formula recognition on UniMER-Test [42], with results summarized in Table 3. MinerU-Diffusion achieves 91.6/91.6/92.0/96.8 on CPE/HWE/SCE/SPE, consistently demonstrating strong performance across complex, handwritten, and printed settings. Compared with other autoregressive (AR) baselines, MinerU-Diffusion is clearly competitive, substantially outperforming general-purpose models such as GPT-4o [13], and reaching similar ranges to strong AR systems on multiple subsets. The remaining gap to the best specialized pipeline (e.g., MinerU2.5 [29]) is most visible on harder categories like CPE, suggesting future improvements should focus on more precise symbol-level modeling and structure-aware decoding for complex printed expressions.

4.4 Ablation Study

4.4.1 Confidence Threshold vs. Decoding Parallelism

We systematically analyze the impact of dynamic confidence thresholds on decoding parallelism and inference efficiency. As shown in Figure 4, as the threshold increases from 0.5 to 0.99, both Tokens Per Forward (TPF) and Throughput Per Second (TPS) decrease monotonically: lower thresholds relax confirmation constraints, enabling greater parallel decoding and higher throughput, whereas thresholds above 0.95 cause decoding to approach near token-by-token generation, leading to a marked efficiency drop. The threshold thus serves as a controllable system-level knob that enables a continuous and predictable trade-off between efficiency and conservativeness. Compared with MinerU2.5 at

Type	Method	CC-OCR		OCRBench v2		UniMER-Test			
		TEDS↑	TEDS-S↑	TEDS↑	TEDS-S↑	CPE↑	HWE↑	SCE↑	SPE↑
AR	InternVL3.5-241B [45]	62.87	69.52	79.50	85.81	91.7	93.2	95.1	97.8
	Qwen2.5-VL-72B [40]	81.22	86.48	81.33	86.58	88.9	91.8	95.5	96.2
	GPT-4o [13]	66.98	79.04	70.51	79.55	82.7	85.9	87.8	96.7
	dots.ocr [16]	75.42	81.65	82.04	86.27	86.8	90.5	94.7	97.5
	MinerU2.5 [29]	79.76	85.16	87.13	90.62	96.6	94.4	96.4	98.4
dLM	MinerU-Diffusion	73.77	82.06	81.18	88.66	91.6	91.6	92.0	96.8

Table 3: Comprehensive recognition results on CC-OCR [51], OCRBench v2 [9], and UniMER-Test [42]. ↑ denotes higher is better.

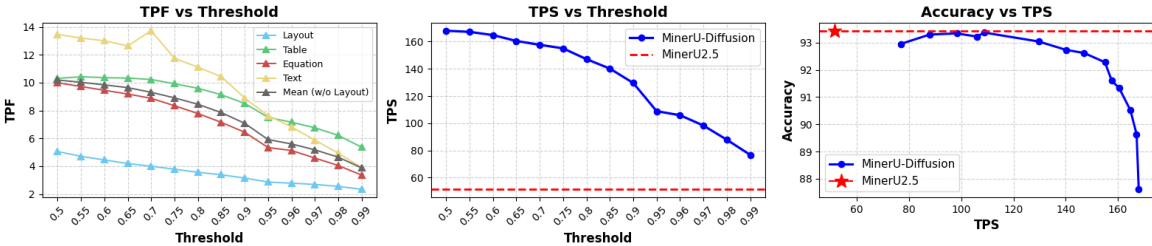


Figure 4: Threshold sensitivity analysis of TPF, TPS, and accuracy. TPF denotes tokens per forward, and TPS refers to throughput measured on an NVIDIA H200 GPU with a batch size of 1.

matched accuracy (fixed at ~52 TPS), our method achieves 108.9 TPS at 93%+ accuracy (thr = 0.95), corresponding to a 2.1× speedup, and reaches 164.8 TPS while maintaining over 90% accuracy (thr = 0.6), yielding a peak acceleration of approximately 3.2×, thereby unlocking substantial parallelism without sacrificing quality.

4.4.2 Decoding Parallelism vs Accuracy

This experiment analyzes the impact of decoding parallelism on model accuracy by varying the dynamic confidence threshold (0.5–0.99) under the *w/ GT Layout* setting. As shown in Figure 5, as the threshold increases, performance first improves, then saturates, and slightly declines: the Overall score rises from 87.61 (thr=0.5) to 93.37 (thr=0.95), indicating that moderately reducing parallelism mitigates low-confidence error propagation and enhances accuracy. However, further increasing the threshold to 0.99 leads to a slight drop, suggesting that overly conservative decoding weakens the benefits of parallel information interaction. At the task level, higher thresholds consistently improve structural consistency, benefiting text, table, and formula recognition (e.g., Table TEDS improves from 81.10 to 90.99), as reduced parallelism alleviates structural error accumulation. Combined with the throughput analysis in Figure 4, these results reveal a clear accuracy–efficiency trade-off: lower thresholds favor speed, while higher thresholds improve stability at the cost of efficiency. Notably, at thr=0.95, MinerU-Diffusion achieves an Overall score of 93.37, comparable to MinerU2.5 while retaining superior decoding efficiency, making it an effective balance point between accuracy and efficiency.

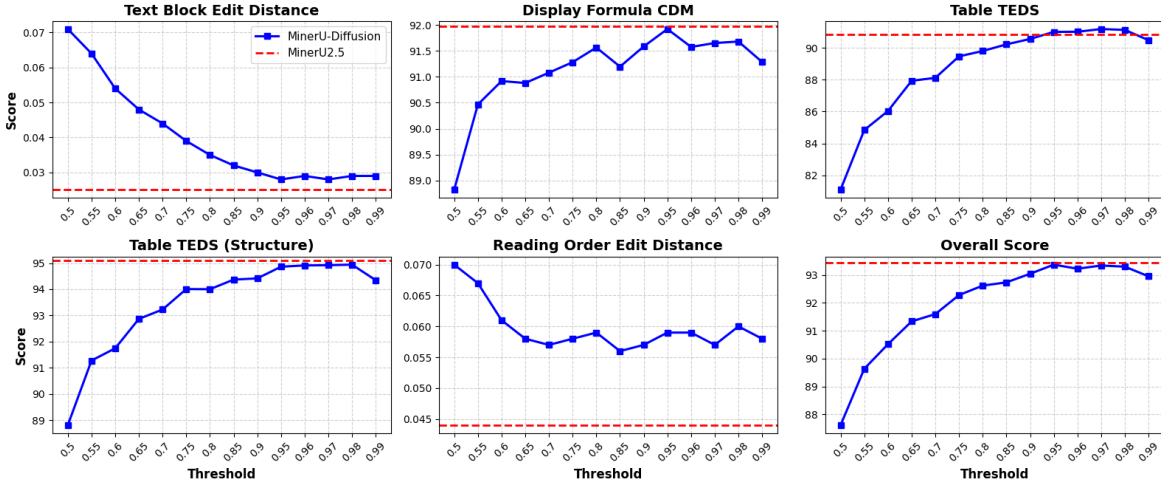


Figure 5: Visualizing the accuracy-throughput trade-off of different models across different OCR tasks under the *w/ GT Layout* setting.

Type	Method	Inference		Score (w/ GT Layout)					Score (w/o GT Layout)	
		TPF	TPS	Overall↑	Text↓	Formula↑	Table _{TEDS} ↑	Table _{TEDS-S} ↑	Overall↑	Reading Order↓
dLM	Static (step=6)	5.33	91.56	88.31	0.064	85.52	85.81	9120	82.73	0.077
	Dynamic ($\tau = 0.97$)	5.18	98.32	93.34	0.028	91.65	91.17	94.92	89.04	0.057
	Static (step=32)	1.00	21.86	93.02	0.030	91.10	90.95	94.73	89.76	0.060
AR	MinerU2.5 [29]	1.00	51.46	93.44	0.025	91.98	90.84	95.10	90.67	0.044
	PaddleOCR-VL [7]	1.00	40.77	93.91	0.021	92.13	91.70	95.42	92.79	0.043

Table 4: Comparison of different inference strategies and model types in terms of efficiency (TPF, TPS) and document parsing performance.

4.4.3 Decoding Strategy Comparisons

We compare static decoding with dynamic scheduling on OmniDocBench v1.5 [32], as summarized in Table 4. Increasing the total decoding steps from 6 to 32 improves overall performance from 88.31 to 93.02, indicating that decoding fewer tokens per step reduces error accumulation, albeit at the cost of requiring more rounds of inference and thus lower throughput (21.86 TPS). In contrast, dynamic scheduling outperforms static step=6 in both overall and sub-task metrics (93.34 vs. 88.31) while achieving higher throughput (98.32 vs. 91.56 TPS), indicating that confidence-based adaptive token selection effectively reduces premature low-confidence commitments and balances efficiency with accuracy. As further shown in Table 4, compared with strong AR baselines such as MinerU2.5 [29] and PaddleOCR-VL [7], our approach achieves competitive parsing performance while maintaining high throughput, highlighting the potential of parallel generation strategies for complex document understanding.

4.4.4 Full-Attn vs Block-Attn

To analyze the impact of attention mechanisms on efficiency and stability, we conducted ablation experiments on Full-Attn and Block-Attn Diffusion using LLaDA-MoE-7B-A1B-Instruct [58], trained on an OCR task. The main difference lies in the amount of attention computed per forward step. Full-Attn

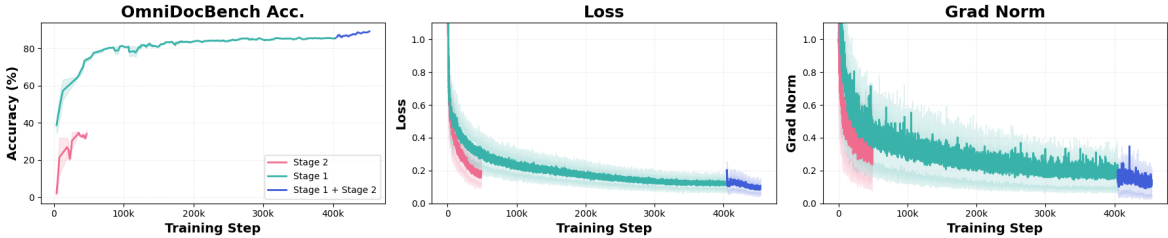


Figure 6: Comparison of training dynamics across different curriculum strategies. The two-stage framework achieves smoother optimization and higher final accuracy compared to single-stage base-lines.

Setting	Score (w/ GT Layout)					Score (w/o GT Layout)	
	Overall↑	Text↓	Formula↑	Table _{TEDS} ↑	Table _{TEDS-S} ↑	Overall↑	Reading Order↓
Stage 1	92.89	0.032	91.58	90.28	94.93	86.13	0.056
Stage 2	89.33	0.038	88.50	83.29	88.93	35.71	0.472
Stage 1 + Stage 2	93.37	0.028	91.92	91.00	94.86	88.94	0.059

Table 5: Comparison of different training stages under GT and non-GT layout settings.

computes roughly $L \times L$ attention, leading to quadratic growth in memory and computational costs with longer sequences, which slows inference and can induce repetitive decoding in OCR. In contrast, Block-Attn computes roughly $B \times L$ attention per forward step, using a “self-regressive across blocks, diffusion within blocks” design, achieving near-linear scalability while maintaining bidirectional context modeling. Experimental results show Block-Attn reduces memory and latency significantly and better mitigates repetition in OCR generation, yielding stronger scalability and stability. A more detailed discussion of these ablation results is provided in [Appendix C](#).

4.4.5 Two-Stage Curriculum Learning

To validate the proposed two-stage curriculum learning framework, we conduct systematic ablations on OmniDocBench v1.5, comparing Stage 1 only, Stage 2 only, and the full two-stage strategy. As shown in [Figure 6](#), training dynamics (Accuracy, Loss, and Grad Norm) reveal that Stage 2 alone exhibits rapid early loss reduction but low accuracy and significant gradient oscillation, indicating high-variance optimization caused by emphasizing high-uncertainty samples without stable representation initialization. In contrast, Stage 1 achieves smooth and stable convergence with accuracy improving to around 85%, yet performance saturates due to limited boundary refinement. The full two-stage strategy integrates both advantages: Stage 1 establishes low-variance global representations under diverse data, while Stage 2 performs uncertainty-driven boundary refinement, enabling continued improvement beyond the Stage 1 plateau and achieving over 89% accuracy. As shown in [Table 5](#), under the challenging *w/o gt layout* setting, Stage 2 alone largely fails (35.712), whereas the full framework reaches 88.937, confirming that the progressive curriculum effectively mitigates optimization instability and performance ceilings induced by diffusion’s any-order modeling.

4.5 Semantic Shuffle Analysis

Most modern VLM-based OCR systems [\[29, 7, 16, 47\]](#) follow a “vision encoder + autoregressive (AR) language decoder” pipeline: the image is converted into visual tokens, and an LM-style decoder [\[39\]](#) generates the transcription token by token. Although this paradigm achieves strong OCR scores, it is often unclear whether these gains come from faithful visual reading or from the decoder’s ability to

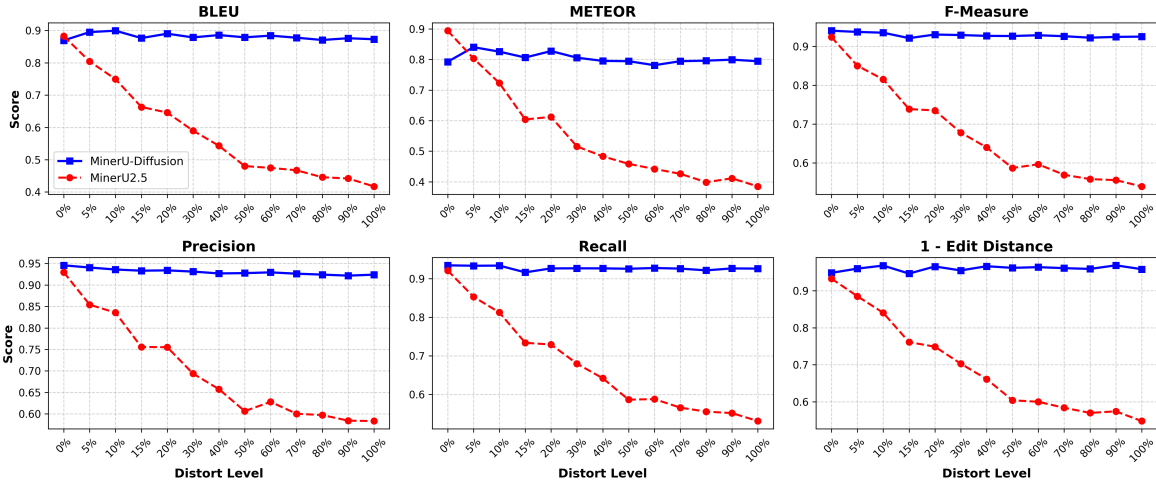


Figure 7: Semantic Shuffle benchmark results across distortion levels.

“fill in” plausible text using linguistic priors and global context.

To better isolate true visual decoding ability, and to validate our hypothesis, we construct **Semantic Shuffle**, a benchmark that removes semantic coherence while keeping the visual presentation comparable. Starting from 112 English document images in the FOX dataset [18], we shuffle a controlled proportion of words and re-render the resulting text into new document images with consistent formatting.

As shown in Figure 7, experiments reveal a clear contrast between decoding paradigms. As the distortion level increases, the performance of AR decoders degrades sharply and consistently, indicating a heavy reliance on linguistic plausibility. In contrast, a diffusion-style decoder remains robust, maintaining nearly constant performance across distortion levels. These results suggest that AR decoders tend to reconstruct text using global language regularities when semantics are disrupted, whereas diffusion-based decoding aligns more directly with the visual signal under semantically invalid inputs.

5 Conclusion

In this paper, we propose MinerU-Diffusion, a 2.5B-parameter diffusion-based framework for document OCR, replacing autoregressive decoding with block-level parallel diffusion decoding and confidence-guided scheduling to improve efficiency and scalability. A two-stage curriculum learning strategy stabilizes training and enhances boundary precision and robustness. Experiments across document, table, formula, and Semantic Shuffle benchmarks show competitive performance against AR baselines and stronger resilience to disrupted semantics, highlighting diffusion-based parallel decoding as a promising alternative for document ocr. We hope our work inspires future research on efficient and reliable diffusion-based decoding for document OCR.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [2] Marianne Arriola, Aaron Gokaslan, Justin T Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. *arXiv preprint arXiv:2503.09573*, 2025.
- [3] Shuang Cheng, Yihan Bian, Dawei Liu, Yuhua Jiang, Yihao Liu, Linfeng Zhang, Wenghai Wang, Qipeng Guo, Kai Chen, Biqing Qi*, and Bowen Zhou. Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation, 2025. URL <https://github.com/JetAstra/SDAR>.
- [4] Shuang Cheng, Yihan Bian, Dawei Liu, Linfeng Zhang, Qian Yao, Zhongbo Tian, Wenhai Wang, Qipeng Guo, Kai Chen, Biqing Qi, et al. Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation. *arXiv preprint arXiv:2510.06303*, 2025.
- [5] Shuang Cheng, Yuhua Jiang, Zineng Zhou, Dawei Liu, Wang Tao, Linfeng Zhang, Biqing Qi, and Bowen Zhou. Sdar-vl: Stable and efficient block-wise diffusion for vision-language understanding. *arXiv preprint arXiv:2512.14068*, 2025.
- [6] 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.
- [7] Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiakuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, et al. Paddleocr 3.0 technical report. *arXiv preprint arXiv:2507.05595*, 2025.
- [8] Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim M Alabdulmohsin, et al. Patch n’pack: Navit, a vision transformer for any aspect ratio and resolution. *Advances in Neural Information Processing Systems*, 36: 2252–2274, 2023.
- [9] Ling Fu, Zhebin Kuang, Jiajun Song, Mingxin Huang, Biao Yang, Yuzhe Li, Linghao Zhu, Qidi Luo, Xinyu Wang, Hao Lu, et al. Ocrbench v2: An improved benchmark for evaluating large multimodal models on visual text localization and reasoning. *arXiv preprint arXiv:2501.00321*, 2024.
- [10] Google DeepMind. Gemini diffusion. <https://blog.google/technology/google-deepmind/gemini-diffusion/>, 2025. Accessed: 2025-05-20.
- [11] Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, et al. Seed1. 5-vl technical report. *arXiv preprint arXiv:2505.07062*, 2025.
- [12] Zhanqiu Hu, Jian Meng, Yash Akhauri, Mohamed S Abdelfattah, Jae-sun Seo, Zhiru Zhang, and Udit Gupta. Accelerating diffusion language model inference via efficient kv caching and guided diffusion. *arXiv preprint arXiv:2505.21467*, 2025.
- [13] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.
- [14] Inception Labs, Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, et al. Mercury: Ultra-fast language models based on diffusion. *arXiv preprint arXiv:2506.17298*, 2025.
- [15] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.
- [16] Yumeng Li, Guang Yang, Hao Liu, Bowen Wang, and Colin Zhang. dots. ocr: Multilingual document layout parsing in a single vision-language model. *arXiv preprint arXiv:2512.02498*, 2025.
- [17] 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. *arXiv preprint arXiv:2506.05218*, 2025.

- [18] Chenglong Liu, Haoran Wei, Jinyue Chen, Lingyu Kong, Zheng Ge, Zining Zhu, Liang Zhao, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. Focus anywhere for fine-grained multi-page document understanding. *arXiv preprint arXiv:2405.14295*, 2024.
- [19] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.
- [20] Yudong Liu, Spencer Hallyburton, Jiwoo Kim, Yueqian Lin, Yiming Li, Qinsi Wang, Hui Ye, Jingwei Sun, Miroslav Pajic, Yiran Chen, et al. Llavida: A large language vision driving assistant for explicit reasoning and enhanced trajectory planning. *arXiv preprint arXiv:2512.18211*, 2025.
- [21] Zhiyuan Liu, Yicun Yang, Yaojie Zhang, Junjie Chen, Chang Zou, Qingyuan Wei, Shaobo Wang, and Linfeng Zhang. dllm-cache: Accelerating diffusion large language models with adaptive caching. *arXiv preprint arXiv:2506.06295*, 2025.
- [22] Nikolaos Livathinos, Christoph Auer, Maksym Lysak, Ahmed Nassar, Michele Dolfi, Panos Vagenas, Cesar Berrospi Ramis, Matteo Omenetti, Kasper Dinkla, Yusik Kim, et al. Docling: An efficient open-source toolkit for ai-driven document conversion. *arXiv preprint arXiv:2501.17887*, 2025.
- [23] Xinyin Ma, Runpeng Yu, Gongfan Fang, and Xinchao Wang. dkv-cache: The cache for diffusion language models. *arXiv preprint arXiv:2505.15781*, 2025.
- [24] Sean Man, Roy Ganz, Roi Ronen, Shahar Tsiper, Shai Mazor, and Niv Nayman. Dodo: Discrete ocr diffusion models. *arXiv preprint arXiv:2602.16872*, 2026.
- [25] Souvik Mandalm. Nanonets-ocr-s. <https://nanonets.com/research/nanonets-ocr-s/>, 2025. Accessed:2025-09-25.
- [26] Jinjie Ni, Qian Liu, Longxu Dou, Chao Du, Zili Wang, Hang Yan, Tianyu Pang, and Michael Qizhe Shieh. Diffusion language models are super data learners. *arXiv preprint arXiv:2511.03276*, 2025.
- [27] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.
- [28] Junbo Niu, Yifei Li, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, et al. Ovo-bench: How far is your video-llms from real-world online video understanding? In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18902–18913, 2025.
- [29] 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. *arXiv preprint arXiv:2509.22186*, 2025.
- [30] Junbo Niu, Yuanhong Zheng, Ziyang Miao, Hejun Dong, Chunjiang Ge, Hao Liang, Ma Lu, Bohan Zeng, Qiahao Zheng, Conghui He, et al. Native visual understanding: Resolving resolution dilemmas in vision-language models. *arXiv preprint arXiv:2506.12776*, 2025.
- [31] Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*, 2024.
- [32] Linke Ouyang, Yuan Qu, Hongbin Zhou, Jiawei Zhu, Rui Zhang, Qunshu Lin, Bin Wang, Zhiyuan Zhao, Man Jiang, Xiaomeng Zhao, et al. Omnidocbench: Benchmarking diverse pdf document parsing with comprehensive annotations. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24838–24848, 2025.
- [33] Vik Paruchuri. Marker. <https://github.com/datalab-to/marker>, 2025. Accessed:2025-09-25.
- [34] Jake Poznanski, Aman Rangapur, Jon Borchardt, Jason Dunkelberger, Regan Huff, Daniel Lin, Christopher Wilhelm, Kyle Lo, and Luca Soldaini. olmocr: Unlocking trillions of tokens in pdfs with vision language models. *arXiv preprint arXiv:2502.18443*, 2025.

- [35] Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.
- [36] Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37:103131–103167, 2024.
- [37] Yuxuan Song, Zheng Zhang, Cheng Luo, Pengyang Gao, Fan Xia, Hao Luo, Zheng Li, Yuehang Yang, Hongli Yu, Xingwei Qu, et al. Seed diffusion: A large-scale diffusion language model with high-speed inference. *arXiv preprint arXiv:2508.02193*, 2025.
- [38] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- [39] Qwen Team. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.
- [40] Qwen Team. Qwen2.5-vl, January 2025. URL <https://qwenlm.github.io/blog/qwen2.5-vl/>.
- [41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [42] 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.
- [43] Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, et al. Mineru: An open-source solution for precise document content extraction. *arXiv preprint arXiv:2409.18839*, 2024.
- [44] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.
- [45] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internv13. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.
- [46] Yinjie Wang, Ling Yang, Bowen Li, Ye Tian, Ke Shen, and Mengdi Wang. Revolutionizing reinforcement learning framework for diffusion large language models. *arXiv preprint arXiv:2509.06949*, 2025.
- [47] Haoran Wei, Yaofeng Sun, and Yukun Li. Deepseek-ocr: Contexts optical compression. *arXiv preprint arXiv:2510.18234*, 2025.
- [48] Chengyue Wu, Hao Zhang, Shuchen Xue, Shizhe Diao, Yonggan Fu, Zhijian Liu, Pavlo Molchanov, Ping Luo, Song Han, and Enze Xie. Fast-dllm v2: Efficient block-diffusion llm. *arXiv preprint arXiv:2509.26328*, 2025.
- [49] Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. *arXiv preprint arXiv:2505.22618*, 2025.
- [50] Ling Yang, Ye Tian, Bowen Li, Xincheng Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada: Multi-modal large diffusion language models. *arXiv preprint arXiv:2505.15809*, 2025.
- [51] Zhibo Yang, Jun Tang, Zhaohai Li, Pengfei Wang, Jianqiang Wan, Humen Zhong, Xuejing Liu, Mingkun Yang, Peng Wang, Shuai Bai, et al. Cc-ocr: A comprehensive and challenging ocr benchmark for evaluating large multimodal models in literacy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21744–21754, 2025.
- [52] Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b, 2025. URL <https://hkunlp.github.io/blog/2025/dream>.
- [53] Zebin You, Shen Nie, Xiaolu Zhang, Jun Hu, Jun Zhou, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. Llada-v: Large language diffusion models with visual instruction tuning. *arXiv preprint arXiv:2505.16933*, 2025.
- [54] Runpeng Yu, Xinyin Ma, and Xinchao Wang. Dimple: Discrete diffusion multimodal large language model with parallel decoding. *arXiv preprint arXiv:2505.16990*, 2025.

- [55] Bohan Zeng, Kaixin Zhu, Daili Hua, Bozhou Li, Chengzhuo Tong, Yuran Wang, Xinyi Huang, Yifan Dai, Zixiang Zhang, Yifan Yang, et al. Research on world models is not merely injecting world knowledge into specific tasks. *arXiv preprint arXiv:2602.01630*, 2026.
- [56] Pan Zhang, Xiaoyi Dong, Yuhang Cao, Yuhang Zang, Rui Qian, Xilin Wei, Lin Chen, Yifei Li, Junbo Niu, Shuangrui Ding, et al. Internlm-xcomposer2. 5-omnilive: A comprehensive multimodal system for long-term streaming video and audio interactions. *arXiv preprint arXiv:2412.09596*, 2024.
- [57] Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. *arXiv preprint arXiv:2409.02908*, 2024.
- [58] Fengqi Zhu, Zebin You, Yipeng Xing, Zenan Huang, Lin Liu, Yihong Zhuang, Guoshan Lu, Kangyu Wang, Xudong Wang, Lanning Wei, Hongrui Guo, Jiaqi Hu, Wentao Ye, Tiejuan Chen, Chenchen Li, Chengfu Tang, Haibo Feng, Jun Hu, Jun Zhou, Xiaolu Zhang, Zhenzhong Lan, Junbo Zhao, Da Zheng, Chongxuan Li, Jianguo Li, and Ji-Rong Wen. Llada-moe: A sparse moe diffusion language model. *arXiv preprint arXiv:2509.24389*, 2025.
- [59] 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. *arXiv preprint arXiv:2504.10479*, 2025.

Appendix

A Training Details

MinerU-Diffusion uses a vision encoder initialized from Qwen2-VL-7B, a diffusion decoder initialized from SDAR-1.7B-Chat-b32, and an abstractor initialized from random parameters, and is trained in three phases summarized in Table 6: Stage-0 initializes multimodal alignment on generic image-text data, Stage-1 performs large-scale OCR adaptation on the easier subset $\mathcal{D}_{\text{base}}$, and Stage-2 further fine-tunes the model on the harder subset $\mathcal{D}_{\text{hard}}$.

		Stage-0		Stage-1	Stage-2
		a	b		
<i>Vision</i>	Max Resolution	2048 × 28 × 28	4096 × 28 × 28	2048 × 28 × 28	2048 × 28 × 28
	#Tokens per Image	4 ~ 2048	4 ~ 4096	4 ~ 2048	4 ~ 2048
<i>Data</i>	Dataset	LLaVA-Pretrain	LLaVA-NeXT-Data	Layout&OCR	Layout&OCR
	#Samples	550K	739K	6.9M	630K
<i>Model</i>	Trainable	MLP Adaptor	All	All	All
	Sequence Length	4096	8192	12288	16384
	Data Augmentation	No	No	Yes	Yes
<i>Training</i>	Batch Size	256	64	256	256
	LR	1×10^{-3}	4×10^{-5}	4×10^{-5}	2×10^{-5}
	Warmup Ratio	0.03	0.03	0.03	0.1
	Epoch	1	1	9	4

Table 6: Training setup and hyperparameters in three training stages.

A.1 Stage-0: Modality Alignment

Stage-0 provides the initialization used before document-specific supervision is introduced. In Stage-0a, LLaVA-Pretrain is used to optimize only the MLP adaptor, while the remaining modules are frozen. This step aligns visual features with the language model input space and stabilizes subsequent OCR training. In Stage-0b, training is continued on LLaVA-NeXT-Data with all parameters unfrozen, improving multimodal instruction following and adaptation to longer visual contexts.

As shown in Table 6, this phase uses shorter sequences and no document augmentation. Its primary role is to provide a stable initialization for later large-scale OCR optimization.

A.2 Stage-1: Large-Scale OCR Adaptation

Stage-1 constitutes the main OCR training phase and corresponds to the easier split $\mathcal{D}_{\text{base}}$ in Section 3.3. Training is conducted on the Layout&OCR mixture, which jointly covers page-level layout detection and element-level recognition for text, formulas, and tables. Layout samples use full-page images with relative coordinates, whereas recognition samples use cropped regions to preserve local detail without incurring the full-page token cost.

The objective of this phase is to establish reliable OCR performance on standard document samples before emphasizing more difficult cases. Because masked diffusion is trained from partially observed targets rather than a fixed left-to-right prefix, optimization is more sensitive to label noise and heterogeneous data quality. Stage-1 therefore relies on scale, task diversity, and document augmentation to

improve training stability. All parameters are updated in this phase. The sequence length is 12288, the batch size is 256, the learning rate is 4×10^{-5} , and training lasts for 9 epochs.

A.3 Stage-2: Hard-Case Specialization

Stage-2 starts from the Stage-1 checkpoint and shifts the sampling distribution toward the harder subset $\mathcal{D}_{\text{hard}}$. This subset contains samples on which the model remains comparatively unstable, including crowded layouts, ambiguous crop boundaries, structurally complex tables, and noisy supervision. Hard samples are mixed with a smaller replay portion drawn from the broader Stage-1 distribution so that baseline OCR coverage is preserved while failure cases are further reduced.

This phase is intended to improve performance on the samples that dominate the residual error after Stage-1. Accordingly, the sequence length is increased to 16384 and the learning rate is reduced to 2×10^{-5} , while the batch size is kept at 256 and the warmup ratio is increased to 0.1. All parameters remain trainable, and fine-tuning lasts for 4 epochs.

A.4 Training Dynamics

As shown in [Figure 8](#), the loss and gradient norm remain stable throughout Stage 0a, Stage 0b, Stage 1, and Stage 2. Stage 0a and Stage 0b converge rapidly during multimodal initialization, Stage 1 exhibits smooth long-horizon optimization on $\mathcal{D}_{\text{base}}$, and Stage 2 remains stable after introducing hard-case refinement. These dynamics are consistent with the curriculum design: a stable base solution is first established on broad OCR data, after which hard-case refinement improves performance while preserving optimization stability.

A.5 Data Augmentation

Document-specific augmentation is enabled in both Stage-1 and Stage-2. Following the same augmentation taxonomy used in MinerU2.5, we inject four types of perturbations during OCR training: geometric changes, background disturbances, color shifts, and image degradation. In practice, this includes scaling, grid distortion, and rotation; background effects such as texture overlays, weather-like artifacts, watermarking, scanlines, and shadows; photometric changes such as brightness/contrast adjustment, illumination variation, and RGB shift; and degradation operators including PSF blur, motion-like blur, Gaussian blur, and light morphological corruption such as erosion or dilation.

These transformations target common OCR noise. We mainly simulate uneven lighting, compression artifacts, scan traces, and cluttered backgrounds. As in MinerU2.5, spatial augmentation is not applied to layout detection samples, since it can alter the geometric supervision itself. Augmentation strength and sampling probability are instead tuned by task type, so that the model gains robustness without drifting too far from valid OCR targets.

A.6 Practical Notes

For Stage-1 and Stage-2, the visual token budget is limited to 2048 tokens per image, and inputs are resized only when this budget is exceeded. Page-level layout detection and crop-level recognition are trained separately within the same unified model. This design aligns training with downstream usage: global structure is learned from full pages, whereas precise transcription and structure recovery are learned from native-content regions.

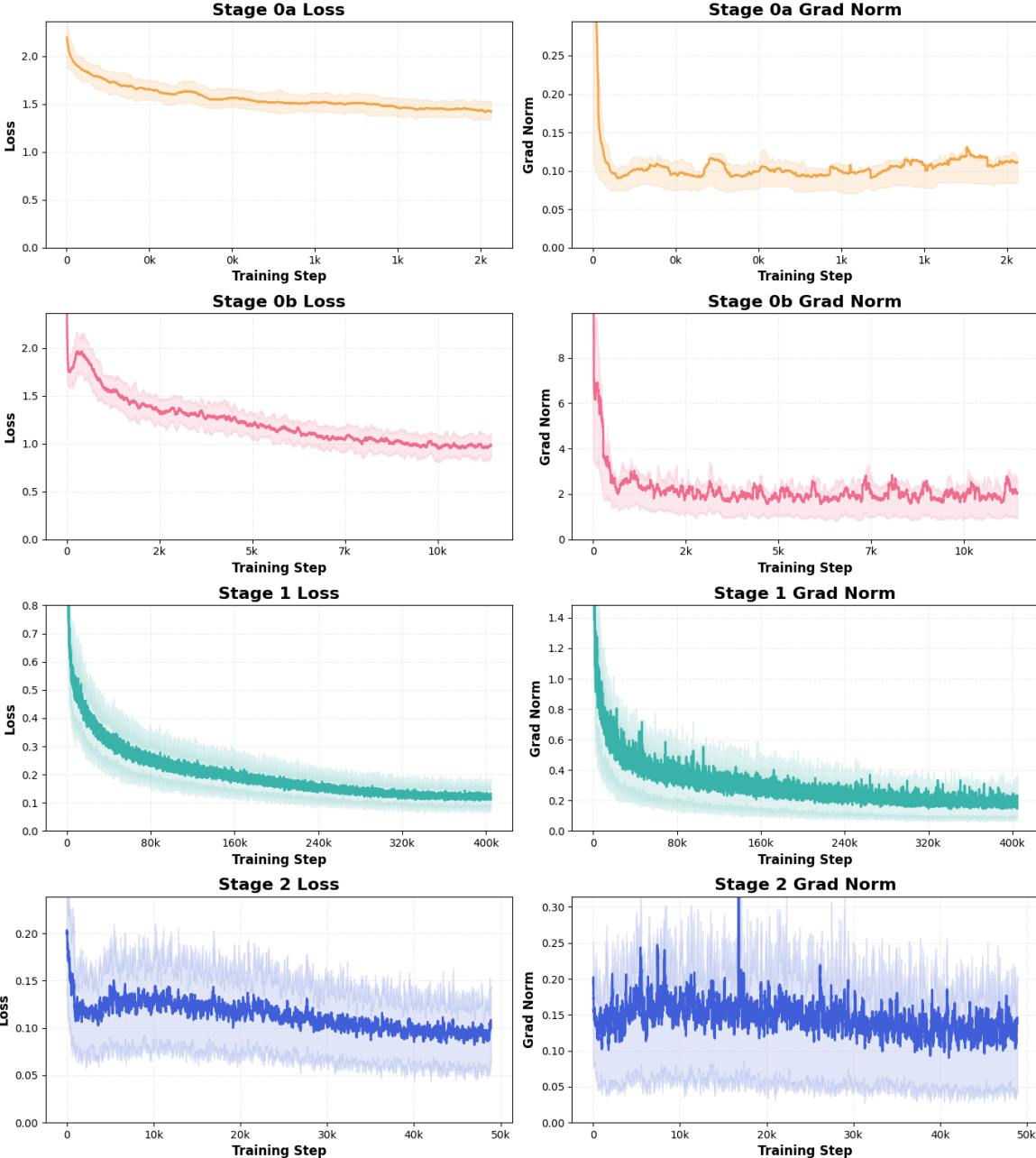


Figure 8: Training dynamics across the four stages in our training recipe: Stage 0a, Stage 0b, Stage 1, and Stage 2. For each stage, we report both the loss and the gradient norm.

B Prompt Templates

This section summarizes the prompts used by MinerU-Diffusion at inference time. All OCR tasks share the same system prompt:

```
System: You are a helpful assistant.
```

B.1 Layout Detection

Layout detection uses the following prompt call:

```
System: You are a helpful assistant.  
User: Layout Detection:
```

The input page image is resized to 1036×1036 before decoding. The returned sequence lists document elements in reading order. Each entry serializes the relative box coordinates, the semantic category, and the orientation tag, so that the page layout can be reconstructed directly from the decoded string. A representative prediction is shown below:

```
<|box_start|>130 071 368 102<|box_end|><|ref_start|>title<|ref_end|><|rotate_up|>  
<|box_start|>275 068 320 072<|box_end|><|ref_start|>text<|ref_end|><|rotate_up|>  
<|box_start|>015 300 496 524<|box_end|><|ref_start|>image<|ref_end|><|rotate_up|>
```

B.2 Text Recognition

Text recognition uses the following prompt call:

```
System: You are a helpful assistant.  
User: Text Recognition:
```

The image is processed at its original resolution whenever possible, with the visual token budget constrained to 4–2048 tokens. If this limit is exceeded, the image is resized accordingly. The model returns a plain text transcription without additional structural markers. A representative prediction is shown below:

```
We believe that LLMs can be trained to identify proof issues without reference solutions.
```

B.3 Formula Recognition

Formula recognition uses the following prompt call:

```
System: You are a helpful assistant.  
User: Formula Recognition:
```

The image remains at native resolution unless the visual token count exceeds the range of 4–2048 tokens, in which case resizing is applied. The decoded result is a LaTeX string representing the target expression. A representative prediction is shown below:

```
\[  
\begin{aligned}  
\frac{d\mathbf{u}}{dt} &= \mathbf{P}^{-1}\mathbf{A}\mathbf{P}\mathbf{u} \ \\  
\alpha &= \mathrm{tr}(\mathbf{A}) = \lambda_1 + \lambda_2 \ \\  
\beta &= |\mathbf{A}| = \lambda_1\lambda_2  
\end{aligned}  
\]
```

B.4 Table Recognition

Table recognition uses the following prompt call:

```
System: You are a helpful assistant.  
User: Table Recognition:
```

The original image resolution is preserved when the visual token count remains within 4–2048; otherwise, the image is proportionally rescaled. The model outputs the table in OTSL (Open Table Structure Language), which can be parsed into a structured table representation. A representative prediction is shown below:

```
<fcel>Team<fcel>Main Score<fcel>SROCC<fcel>PLCC<n1>
<fcel>TB-VQA(ours)<fcel>0.8576<fcel>0.8493<fcel>0.8659<n1>
<fcel>2nd<fcel>0.8396<fcel>0.8408<fcel>0.8383<n1>
<fcel>3rd<fcel>0.8289<fcel>0.8261<fcel>0.8317<n1>
<fcel>4th<fcel>0.8199<fcel>0.8163<fcel>0.8236<n1>
<fcel>5th<fcel>0.7994<fcel>0.7962<fcel>0.8026<n1>
```

C Full-Attn vs Block-Attn

This section expands Section 4.4.4 and explains why Block-Attn is consistently preferable to Full-Attn for document OCR diffusion decoding.

C.1 Why Full-Attn Is Inferior

We summarize the main disadvantages of Full-Attn in three aspects:

- **Lower speed in both training and inference.** With output length L and block size B , Full-Attn computes global pairwise attention over the entire target sequence, leading to quadratic attention cost per denoising step, whereas Block-Attn reduces this cost to a near-linear form in practice. During training, the effective context updated at each step also differs substantially. Following our implementation, Full-Attn processes roughly L^2/B token interactions per step, whereas Block-Attn only needs about $2L$, which makes Block-Attn noticeably faster to train under the same block size.
- **Fixed-length mismatch wastes computation.** Full-Attn must predefine a global decoding length L , which is hard to match across documents with very different output lengths. When L is much larger than the actual target length, many decoding positions are allocated but never needed. When L is too small, the preset budget is exhausted before the target is completed. In both cases, the mismatch leads to inefficient use of computation.
- **Fixed-length mismatch also harms generation quality.** Empirically, Full-Attn works best when the preset length L is close to the actual target length N . When $L < N$, the model may stop early or become unstable near the tail. When $L \gtrsim N$, generation is relatively stable. When $L \gg N$, the model often fails to stop cleanly and keeps producing repeated rows or empty cells.

C.2 Generation Quality under Length Mismatch

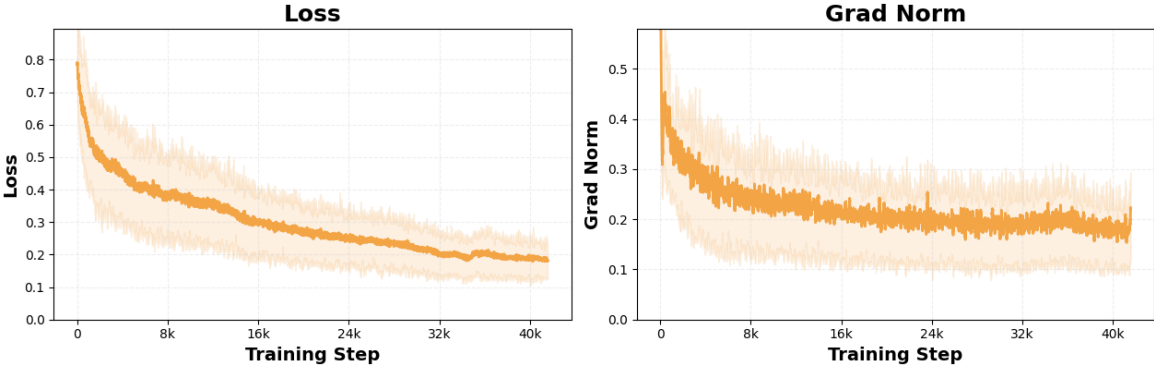


Figure 9: Training curves on a table-only subset.

We next focus on the generation-quality issue caused by fixed-length mismatch in Full-Attn. As a reference, Figure 9 shows the training curves of Full-Attn on a table-only subset; its training loss exhibits a trend consistent with Block-Attn.

We then provide three qualitative cases beyond the summary discussion in Section 4.4.4. We select three tables with different lengths, namely a short table, a medium-length table, and a long table. For each case, we compare Full-Attn with preset lengths $L = 256, 512,$ and $1024,$ together with Block-Attn at $L = 1024.$

For readability, green text denotes normal content and red text denotes wrong content. The notation “...” denotes omitted intermediate content, “...<repeat for x times>” denotes repeated content, and “<missing>” marks missing tail content.

For the short table in Figure 10, all tested Full-Attn lengths are already over-provisioned: $L = 256$ and $L = 512$ produce repeated empty rows, while $L = 1024$ repeats much more severely. For the medium-length table in Figure 11, both $L = 256$ and $L = 512$ generate normally, whereas $L = 1024$ introduces repeated rows after the valid content. For the long table in Figure 12, $L = 256$ and $L = 512$ are both truncated, while $L = 1024$ reaches the tail but still repeats the final record.

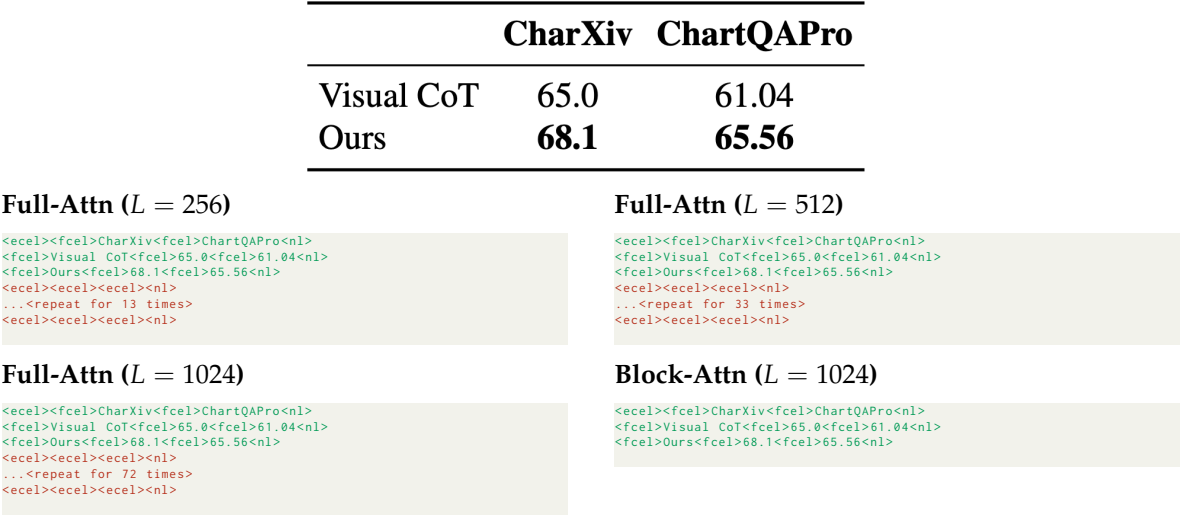


Figure 10: A very short table where over-provisioned Full-Attn repeats empty rows.

MinerU-Diffusion: Rethinking Document OCR as Inverse Rendering via Diffusion Decoding

Cluster ID	Cluster Size	Tool Definition (examples)	Summary Tools
0	148	draw.vertical_guide_line draw.horizontal_guide_line mark.threshold	Adding Auxiliary Lines
1	128	highlight_subfigure crop_subfigure isolate_subplot	Subfigure Cropping
2	101	highlight_series_mask highlight_mp_curve highlight_data_points	Masking Data with Legend
3	64	magnify_region_around_x crop_and_magnify_region crop_region	Region Magnification
Others	27	find_peaks_in_data highlight_max_point crop_beta_vs_time_panel highlight_intersection_point draw_spectral_type_guides	Useless tools

Full-Attn ($L = 256$)

```
<fcel>Cluster ID<fcel>Cluster Size...<n1>
<fcel>0<fcel>148...Adding Auxiliary Lines<n1>
...
<fcel>3<fcel>64...Region Magnification<n1>
<fcel>Others<fcel>27...Useless tools<n1>
```

Full-Attn ($L = 512$)

```
<fcel>Cluster ID<fcel>Cluster Size...<n1>
<fcel>0<fcel>148...Adding Auxiliary Lines<n1>
...
<fcel>3<fcel>64...Region Magnification<n1>
<fcel>Others<fcel>27...Useless tools<n1>
```

Full-Attn ($L = 1024$)

```
<fcel>Cluster ID<fcel>Cluster Size...<n1>
<fcel>0<fcel>148<fcel>draw_vertical_guide_line...<n1>
...
<fcel>Others<fcel>27...Useless tools<n1>
<u><u><u><u><u>
...<repeat for 47 times>
<u><u><u><u><u>
```

Block-Attn ($L = 1024$)

```
<fcel>Cluster ID<fcel>Cluster Size...<n1>
<fcel>0<fcel>148...Adding Auxiliary Lines<n1>
...
<fcel>Others<fcel>27...Useless tools<n1>
```

Figure 11: A medium-length table where matched Full-Attn remains stable but over-provisioned Full-Attn repeats.

Method	L2 (m) ↓				Collision (%) ↓			
	1s	2s	3s	Avg.	1s	2s	3s	Avg.
NMP (Zeng et al., 2019)	-	-	2.31	-	-	-	1.92	-
SA-NMP (Zeng et al., 2019)	-	-	2.05	-	-	-	1.59	-
FF (Hu et al., 2021)	0.55	1.20	2.54	1.43	0.06	0.17	1.07	0.43
EO (Khurana et al., 2022)	0.67	1.36	2.78	1.60	0.04	0.09	0.88	0.33
ST-P3 (Hu et al., 2022b)	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71
UniAD (Hu et al., 2022c)	0.48	0.96	1.65	1.03	0.05	0.17	0.71	0.31
VAD (Jiang et al., 2023)	0.41	0.70	1.05	0.72	0.07	0.17	0.41	0.22
BEV-Planner (Li et al., 2024b)	0.30	0.52	0.83	0.55	0.10‡	0.37‡	1.30‡	0.59‡
LAW (perception-free)	0.26	0.57	1.01	0.61	0.14	0.21	0.54	0.30
LAW (perception-based)	0.24	0.46	0.76	0.49	0.08	0.10	0.39	0.19

Full-Attn ($L = 256$)

```
<fcel>Method<fcel>L2 (m) ...<n1>
<fcel>NMP (Zeng et al., 2019)...<n1>
<fcel>SA-NMP (Zeng et al., 2019)...<n1>
<fcel>FF (Hu et al., 2021)<fcel>0.55...<n1>
<missing>
```

Full-Attn ($L = 512$)

```
<fcel>Method<fcel>L2 (m) ...<n1>
...
<fcel>ST-P3 (Hu et al., 2022b)...<n1>
<fcel>UniAD (Hu et al., 2022c)...<n1>
<missing>
```

Full-Attn ($L = 1024$)

```
<fcel>Method<fcel>L2 (m) ...<n1>
...
<fcel>LAW (perception-free)...<n1>
<fcel>LAW (perception-based)...<n1>
<fcel>LAW (perception-based)...<n1>
<fcel>LAW total...<n1>
```

Block-Attn ($L = 1024$)

```
<fcel>Method<fcel>1s<fcel>L2 (m) ...<n1>
...
<fcel>VAD (Jiang et al., 2023)...<n1>
<fcel>BEV-Planner Li et al., 2024b)...<n1>
<fcel>LAW (perception-free)...<n1>
<fcel>LAW (perception-based)...<n1>
```

Figure 12: A long table where short Full-Attn truncates and long Full-Attn repeats tail rows.

D Qualitative Examples

This section presents qualitative examples illustrating the behavior of MinerU-Diffusion on representative document pages and diffusion decoding tasks. This section is structured as follows: [Figures 13](#) and [14](#) show complete recognition examples from OmniDocBench, covering the full pipeline from the input page to layout prediction and rendered parsed output. [Figures 15](#) to [18](#) visualize diffusion decoding for layout, text, table, and formula generation.

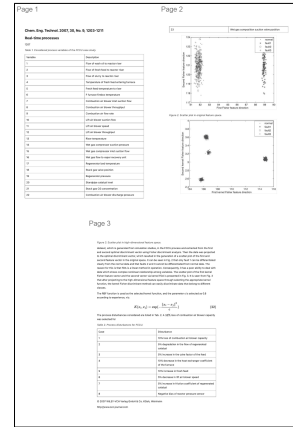
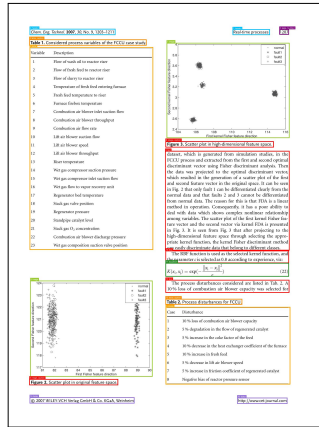
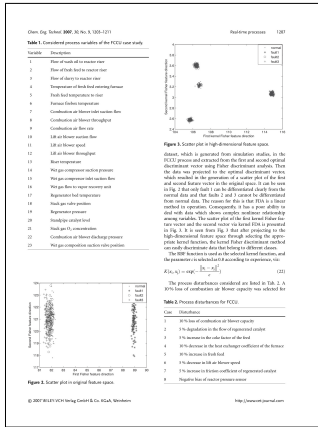
D.1 Complete Recognition Examples

Examples demonstrating end-to-end parsing performance on diverse page types are provided in [Figures 13](#) and [14](#), including academic literature, presentation slides, colorful report pages, newspapers, formula-dense exam solutions, and illustrated instructional pages. These examples show that MinerU-Diffusion can preserve complex academic elements such as tables, figures, captions, and equations, remain robust to watermark interference, maintain reading order on dense multi-column pages, and recover mixed text-image layouts with fine-grained blocks. Each case is shown in a consistent three-column format with the original page image, the predicted layout, and the rendered parsed output.

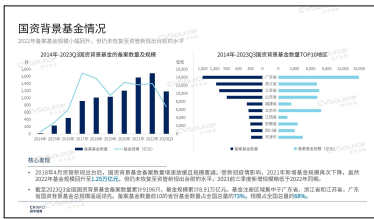
D.2 Diffusion Decoding Examples

The diffusion decoding process on four representative tasks is illustrated in [Figures 15](#) to [18](#). These examples show how the model progressively refines page structure, text content, table organization, and formula symbols during decoding. Each figure groups two samples from the same category to make the visual patterns directly comparable, with the raw cropped input shown above and the corresponding diffusion decoding visualization shown below.

Case 1. Academic literature



Case 2. Watermarked PPT page



Case 3. Colorful report page

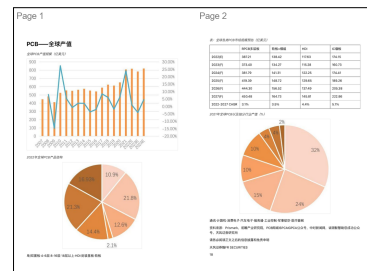
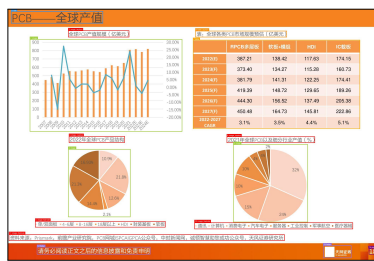
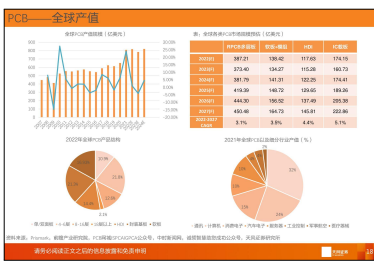
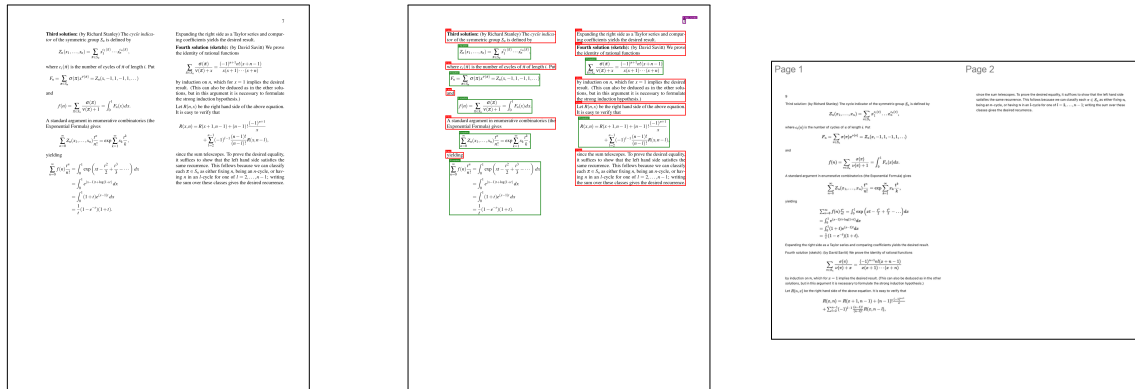


Figure 13: Complete recognition examples. Left: original page image. Middle: predicted layout. Right: rendered parsed output.

Case 4. Newspaper page



Case 5. Formula-dense exam solution



Case 6. Illustrated recipe page

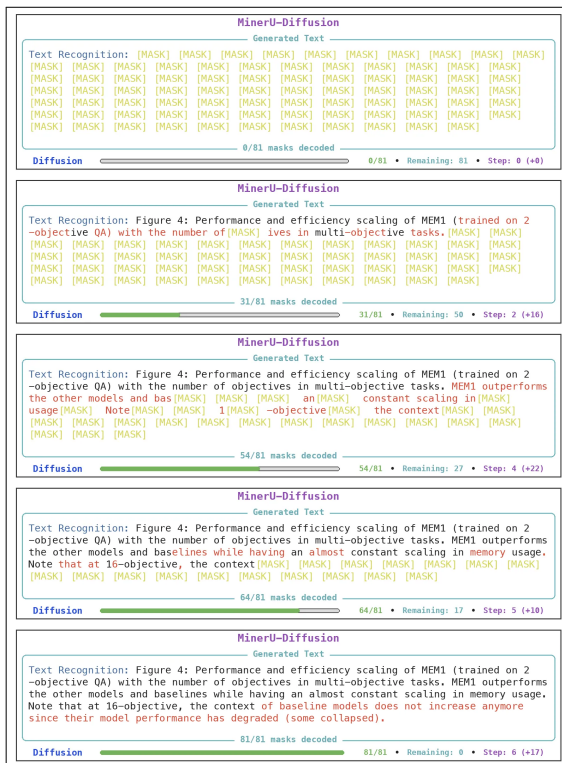


Figure 14: Additional complete recognition examples in the same format.

Text Decoding

Example 1

Figure 4: Performance and efficiency scaling of MEM1 (trained on 2-objective QA) with the number of objectives in multi-objective tasks. MEM1 outperforms the other models and baselines while having an almost constant scaling in memory usage. Note that at 16-objective, the context of baseline models does not increase anymore since their model performance has degraded (some collapsed).



Example 2

Autoregressive language models (ARLMs) dominate open-ended text generation, but DLMs offer complementary benefits such as parallel decoding and iterative refinement. Prior work has explored AR-to-diffusion conversion to bootstrap ARLM training artifacts into



Figure 16: Diffusion decoding examples for text recognition.

Table Decoding

Example 1

CharXiv ChartQPro		
Visual CoT	65.0	61.04
Ours	68.1	65.56

Example 2

Logistic parameter	9-year-olds		16-year-olds		t(40)	p	Cohen's d
	M	SD	M	SD			
Maximum asymptote, proportion	.843	.135	.877	.082	0.951	.347	0.302
Crossover, in ms	759	.87	694	42	2.877	.006	0.840
Slope, as change in proportion per ms	.001	.0002	.002	.0002	2.635	.012	2.078

Figure 17: Diffusion decoding examples for table recognition.

