

SP-DocReader: Difference-Aware Self-Play for Precise Document OCR

Wenjie Liao¹, Xiaohui Song¹, Liangjie Zhao², and Haonan Lu¹

¹Guangdong OPPO Mobile Telecommunications Corp., Ltd
jie3040@akane.waseda.jp, songxiaohui@oppo.com, luhaonan@oppo.com

²Adelaide University
zhaoliangjie55@gmail.com

Abstract

Accurate page transcription remains difficult for vision language models under limited input and training budgets. We present SP-DocReader, a self-play framework for optical character recognition (OCR) that targets residual errors after supervised fine-tuning. Reading Discrepancy Masking aligns reference and generated model tokens through a longest common subsequence, then scores unmatched positions with their full conditioning prefixes. Focused Fidelity Loss adds direct negative log-likelihood supervision at unmatched ground-truth positions. Only the OCR module is trained, while the backbone remains frozen. We derive the combined gradient to distinguish relative score optimization from direct supervision. Compared with SFT-2, SP-DR-3 reduces Vary-600K character error rate on both backbones. On Qwen3-VL-4B, it reduces character error rate by approximately 54 percent and improves DocVQA Average Normalized Levenshtein Similarity (ANLS) by 3.7 points. These results show the value of focusing self-play training on the discrepancies that remain after supervised fine-tuning.

1. Introduction

Page-level transcription must reproduce text across a document, including small characters and numbers. DocFormer and LayoutLM represent text and spatial information (Appalaraju et al., 2021; Xu et al., 2020). Document images and infographics combine dense text with varied layouts (Mathew et al., 2021, 2022). Recent vision language models (VLMs) support document understanding and multimodal reasoning (Bai et al., 2025; Chen et al., 2024a; Luo et al., 2024), but page transcription differs from producing a short answer to a document question.

Higher resolution can improve small-text reading while increasing visual tokens, as in PaliGemma (Beyer et al., 2024). InternLM-XComposer-2.5 uses image tiles (Zhang et al., 2024), while LLaVA-OneVision combines AnyRes processing with pooling to control token budgets (Li et al., 2024). ScreenAI studies optical character recognition (OCR) text as an additional input (Baechler et al., 2024), and LATIN-Prompt preserves layout through spaces and line breaks (Wang et al., 2023a). DocVLM compresses OCR text and layout into queries for a frozen VLM (Nacson et al., 2025). We study reduced-resolution page transcription, distinguishing source resolution from the visual tokens passed to the language model.

Our study starts from a reader adapted by supervised fine-tuning (SFT) on annotated pages. Full-sequence SFT supervises every target position without selecting positions from discrepancies in a generated reading. SP-DocReader uses these discrepancies to focus self-play training on the tokens that still need correction.

SP-DocReader builds on DocVLM’s modular reader (Nacson et al., 2025) and follows SPIN’s iterative pairing of references with previous-model responses (Chen et al., 2024c). Reading Discrepancy Masking

(RDM) aligns the model-token sequences through a longest common subsequence and selects unmatched positions in both readings. The contrastive loss compares score changes against the previous model at these positions while retaining each sequence’s full conditioning prefixes. Focused Fidelity Loss (FFL) adds direct negative log-likelihood supervision at unmatched ground-truth positions. Only the OCR module is updated, with the visual encoder and language backbone frozen.

Compared with SFT-2, SP-DocReader reduces transcription error on both backbones and improves most evaluated question-answering scores. On Qwen3-VL-4B, the gains extend from in-distribution reading to zero-shot document benchmarks.

We make three main contributions.

- We formulate a discrepancy-based self-play objective for page transcription that combines masked relative scoring with direct supervision at unmatched ground-truth positions.
- We derive the combined gradient to explain the roles of relative and direct supervision, and clarify why selecting loss positions still requires full-prefix computation.
- We evaluate SP-DocReader on bilingual transcription and zero-shot document benchmarks, and analyze its components, resolution robustness and training cost.

2. Related Work

Efficient Document Reading with VLMs. Document readers combine image features, recognized text and layout in different ways. ScreenAI uses OCR text as an additional input (Baechler et al., 2024), DocLLM models text and boxes without an image encoder (Wang et al., 2024b), and Hi-VT5 summarizes pages for multi-page question answering (Tito et al., 2023). DocVLM uses a DocFormerV2-based OCR encoder (Nacson et al., 2025; Appalaraju et al., 2024) and compresses its outputs into learnable queries for a frozen VLM (Nacson et al., 2025). Building on this reader, we introduce discrepancy-focused self-play training for page transcription.

Specialized OCR Systems for Document Recognition. PP-OCR integrates text detection, direction classification and recognition (Du et al., 2020). Nougat produces markup from scientific document images (Blecher et al., 2023), while GOT-OCR2.0 supports plain and formatted OCR (Wei et al., 2024). MinerU combines layout analysis and specialized recognition for document parsing (Wang et al., 2024a). DeepSeek-OCR studies optical compression of text into visual representations (Wei et al., 2025). These systems emphasize recognition and parsing designs. We focus on discrepancy-based training of a modular reader.

Supervised and Self-Play Fine-Tuning. For document recognition, we use paired images and transcriptions for SFT. Given input images I drawn from a distribution $q(\cdot)$ and ground-truth transcriptions y following $p_{data}(\cdot | I)$, SFT minimizes the expected negative log-likelihood,

$$\mathcal{L}_{SFT}(\theta) = -\mathbb{E}_{I \sim q(\cdot), y \sim p_{data}(\cdot | I)} [\log p_{\theta}(y | I)]. \quad (1)$$

By autoregressive factorization, SFT sums token losses across the reference reading. SP-DocReader uses the model’s generated reading to select positions for self-play training.

Direct Preference Optimization (DPO) trains on preferred and dispreferred responses without fitting a separate reward model (Rafailov et al., 2023). SPIN pairs human demonstrations with responses from the previous model and optimizes a full-response relative score (Chen et al., 2024c). SPACE

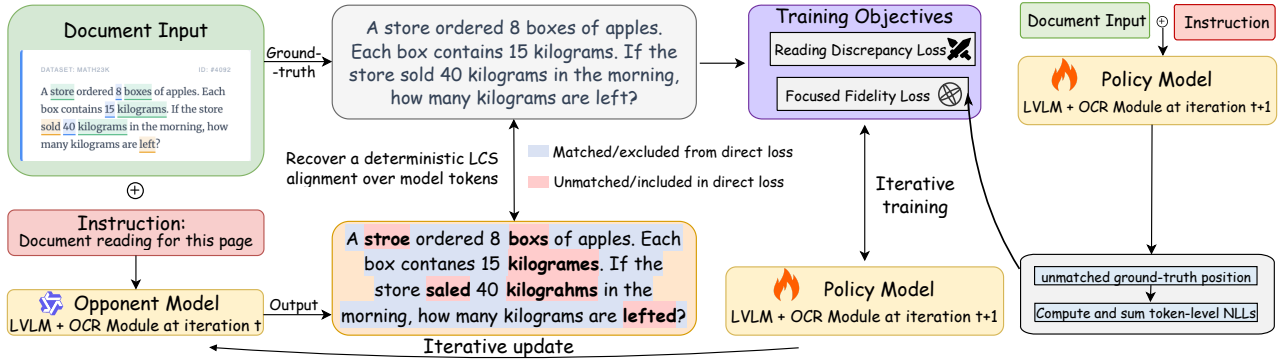


Figure 1: Overview of the SP-DocReader training framework. A frozen opponent generates a synthetic reading of the document, and Reading Discrepancy Masking aligns it with the ground truth to isolate the errors. The main player then optimizes a dual objective that combines the Reading Discrepancy Loss for difference-aware correction with the Focused Fidelity Loss for direct supervision on the hard tokens, after which the opponent is refreshed.

uses separate classification terms for real and generated responses, with full-response scores relative to an opponent (Wang et al., 2025b). SPPO frames preference alignment as a constant-sum game (Wu et al., 2024), while DNO improves a policy iteratively using on-policy responses and preference feedback (Rosset et al., 2024). SP-DocReader instead uses the known page transcription to locate unmatched model-token positions in reference and generated readings. It compares the two readings at those positions and directly supervises unmatched reference tokens, while retaining their original conditioning prefixes. Appendix A reviews further work on visual-language alignment, adversarial learning and iterative self-improvement.

3. Methodology

SP-DocReader adapts the modular reader of DocVLM (Nacson et al., 2025) for page transcription. Reading Discrepancy Masking (RDM) selects positions for the Reading Discrepancy Loss, our contrastive objective. Focused Fidelity Loss (FFL) adds direct supervision at unmatched ground-truth positions. Both losses score tokens with their original prefixes.

3.1. Problem Formulation and Document Reading

We model page transcription as conditional sequence generation. The input is (I, x) , where I is a document image and x is a natural language instruction. The reference sequence $y = (y_1, \dots, y_T)$ contains tokens from a fixed vocabulary $\mathcal{V}$. The reference y transcribes I , whereas a generated reading may not cover the page.

We follow the modular reader design of DocVLM (Nacson et al., 2025). The visual encoder maps I to features $\Phi(I)$, while a fixed external OCR engine extracts text and bounding boxes. The OCR branch encodes this text and layout and uses learnable queries to produce compact representations $Q_{\text{ocr}, \theta}(I)$. The parameters θ belong to the OCR encoder, projection layer and learnable queries. The visual encoder

and language backbone remain frozen. The reference likelihood factorizes autoregressively as

$$P_\theta(y \mid I, x) = \prod_{k=1}^T P_\theta(y_k \mid y_{<k}, \Phi(I), Q_{\text{ocr},\theta}(I), x), \quad (2)$$

where $y_{<k}$ is the prefix preceding position k . Standard supervised fine-tuning maximizes the ground-truth likelihood under (2). Self-play uses the unmatched positions defined below to select direct loss terms. Matched tokens remain in the conditioning prefixes.

3.2. Iterative Self-Play Training

Self-play starts from the OCR parameters θ_0 obtained by supervised fine-tuning on the training set $\mathcal{D}$. At round t , the model P_{θ_t} is held fixed and serves as the opponent. It generates a reading $\hat{y} = \text{Gen}_{\theta_t}(I, x)$, where Gen denotes the decoding procedure. The current model P_θ , called the main player, is updated using pairs of y and $\hat{y}$. Their discrepancies determine the loss positions. Appendix C gives the decoding configuration.

Reading Discrepancy Masking (RDM). RDM aligns the generated reading $\hat{y}$ with the ground truth y to identify unmatched positions. It computes the Longest Common Subsequence (LCS) directly over the model token sequences y and $\hat{y}$. Tokens in a subsequence retain their order but need not be adjacent. Let $\mathcal{S}_{\text{common}}$ be the set of ordered token subsequences shared by these sequences. The alignment operator selects a longest common subsequence,

$$\psi(y, \hat{y}) \in \arg \max_{s \in \mathcal{S}_{\text{common}}} |s|. \quad (3)$$

We recover its matched positions from the sequence starts using dynamic-programming lengths for all suffix pairs. Equal tokens are matched and both indices advance. Otherwise, we advance the index that preserves the larger suffix LCS length, advancing only the synthetic index on ties. Appendix C gives the recurrence in (15) and details EOS handling. Let $\mathcal{E}_y$ and $\mathcal{E}_{\hat{y}}$ be the unmatched positions in the original ground-truth and synthetic sequences. Selecting these positions in increasing order gives

$$y' = (y_k)_{k \in \mathcal{E}_y}, \quad \hat{y}' = (\hat{y}_k)_{k \in \mathcal{E}_{\hat{y}}}. \quad (4)$$

The end-of-sequence (EOS) token follows the same rule. These sets select the tokens to score without removing matched tokens from their conditioning prefixes. For either full sequence $z \in \{y, \hat{y}\}$ and its position set $\mathcal{E}_z$, we define the masked log-likelihood score

$$S_\theta(z, \mathcal{E}_z \mid I, x) = \sum_{k \in \mathcal{E}_z} \log P_\theta(z_k \mid z_{<k}, I, x). \quad (5)$$

Conditioning on (I, x) abbreviates the multimodal inputs in (2). Ground-truth tokens use the full prefix $y_{<k}$, whereas synthetic tokens use their own full prefix $\hat{y}_{<k}$. The current model and fixed opponent score the same tokens at the same positions with the same prefixes, but each uses its own OCR representations. We sum the selected log probabilities without length normalization. The shortened sequences y' and $\hat{y}'$ are not fed back as new sequences for scoring, and $\exp(S_\theta)$ is not treated as a normalized probability over these shortened sequences.

Contrastive Score Optimization. We adapt SPIN’s contrastive score formulation (Chen et al., 2024c) to compare reference and generated readings at their unmatched positions. The comparison is motivated by the expectation gap used in Integral Probability Metrics (IPMs) (Müller, 1997). Let $\mathcal{F}_t$ be a class of scoring functions. Each function assigns a scalar score to a full reading at selected positions with its original context. We leave the instruction x implicit in $f \in \mathcal{F}_t$. The objective favors a higher score for the ground truth than for the generated reading,

$$\max_{f \in \mathcal{F}_t} \mathbb{E}_{\mathcal{D}} [f(I, y, \mathcal{E}_y) - f(I, \hat{y}, \mathcal{E}_{\hat{y}})], \quad (6)$$

where $\mathbb{E}_{\mathcal{D}}$ averages over (I, x, y) in the training set, with $\hat{y} = \text{Gen}_{\theta_t}(I, x)$ fixed within the round. We use a monotonically decreasing convex loss ℓ to penalize a small score gap,

$$\min_{f \in \mathcal{F}_t} \mathbb{E}_{\mathcal{D}} [\ell(f(I, y, \mathcal{E}_y) - f(I, \hat{y}, \mathcal{E}_{\hat{y}}))]. \quad (7)$$

To motivate the comparison with the opponent, consider a fixed complete-sequence scorer $g(I, y)$, with x implicit. For fixed (I, x) and $\beta > 0$, assume the reference distribution is normalized over complete sequences and the normalizing constant below is finite. Over distributions supported by the reference, the well-defined objective $\mathbb{E}_{y \sim P}[g(I, y)] - \beta D_{\text{KL}}(P \| P_{\theta_t})$ with a Kullback-Leibler penalty (Kullback and Leibler, 1951) has the following complete-sequence optimum, as used in DPO (Rafailov et al., 2023)

$$P^*(y | I, x) \propto P_{\theta_t}(y | I, x) \exp\left(\frac{1}{\beta} g(I, y)\right). \quad (8)$$

If this optimum is realizable as P_{θ} , then $g(I, y) = \beta \log(P_{\theta}(y | I, x)/P_{\theta_t}(y | I, x))$ up to an input-dependent constant. This complete-sequence relation motivates our opponent-relative score. RDM applies the tokenwise comparison at selected original positions. For the current model θ and fixed opponent θ_t , define

$$\Delta_t(z) = S_{\theta}(z, \mathcal{E}_z | I, x) - S_{\theta_t}(z, \mathcal{E}_z | I, x). \quad (9)$$

Both terms use the same full sequence z and position set $\mathcal{E}_z$. Setting $f(I, z, \mathcal{E}_z) = \beta \Delta_t(z)$ and using the logistic surrogate $\ell(u) = \log(1 + e^{-u})$ in (7) yields the Reading Discrepancy Loss

$$\begin{aligned} \mathcal{L}_{\text{RDM}}(\theta, \theta_t) = \\ \mathbb{E}_{\mathcal{D}} \left[\log(1 + \exp(\beta[\Delta_t(\hat{y}) - \Delta_t(y)])) \right]. \end{aligned} \quad (10)$$

The loss favors a larger score increase over the opponent for the ground-truth discrepancies than for the synthetic discrepancies.

Focused Fidelity Loss (FFL). The relative objective in (10) can decrease by lowering the synthetic score without increasing the ground-truth score. FFL therefore adds direct supervision at the unmatched ground-truth positions. Under teacher forcing, each target token is scored using its ground-truth prefix. FFL sums the negative log-likelihood over the same position set $\mathcal{E}_y$,

$$\begin{aligned} \mathcal{L}_{\text{FFL}}(\theta) = \\ - \mathbb{E}_{\mathcal{D}} \left[\sum_{k \in \mathcal{E}_y} \log P_{\theta}(y_k | y_{<k}, I, x) \right]. \end{aligned} \quad (11)$$

Algorithm 1: Iterative Self-Play Training for SP-DocReader

Require: Training set $\mathcal{D} = \{(I_i, x_i, y_i)\}_{i=1}^N$, SFT-initialized OCR parameters θ_0 , rounds $R \in \mathbb{Z}_{>0}$, loss coefficients $\beta > 0$ and $\lambda > 0$.

- 1: **for** $t = 0$ **to** $R - 1$ **do**
- 2: Fix opponent θ_t for this round
- 3: Initialize training records $\mathcal{B}_t \leftarrow \emptyset$
- 4: **for** $i = 1$ **to** N **do**
- 5: $\hat{y}_i \leftarrow \text{Gen}_{\theta_t}(I_i, x_i)$
- 6: Recover $\mathcal{E}_{y_i}, \mathcal{E}_{\hat{y}_i}$ from the LCS alignment in (3) with the fixed tie rule
- 7: Add $(I_i, x_i, y_i, \hat{y}_i, \mathcal{E}_{y_i}, \mathcal{E}_{\hat{y}_i})$ to $\mathcal{B}_t$
- 8: **end for**
- 9: $\theta_{t+1} \leftarrow \text{Optimize}(\theta_t, \mathcal{B}_t; \beta, \lambda)$ (Eq. (13))
- 10: **end for**
- 11: **Return** OCR parameters θ_R

The round’s readings and masks remain fixed during optimization. We suppress this round dependence in $\mathcal{L}_{FFL}(\theta)$. Both losses retain the original prefixes, and FFL adds no direct loss at matched positions.

Training Objective and Opponent Update. The training objective adds FFL to the masked contrastive loss,

$$\mathcal{L}_{total}(\theta, \theta_t) = \mathcal{L}_{RDM}(\theta, \theta_t) + \lambda \mathcal{L}_{FFL}(\theta), \quad (12)$$

with $\lambda > 0$ controlling the weight of direct supervision. Pairs with only one empty mask are retained. Pairs with both masks empty contribute no loss gradient, but batch averaging still uses the original number of samples, as specified in Appendix C, (16). The round’s training records $\mathcal{B}_t$ retain all samples, including their full sequences and masks. We write the update as

$$\theta_{t+1} \leftarrow \text{Optimize}(\theta_t, \mathcal{B}_t; \beta, \lambda), \quad (13)$$

where Optimize initializes the trainable OCR parameters at θ_t and applies AdamW to (12) under a finite training budget. The opponent and training records remain fixed during this update. The notation does not assume an exact global minimizer. The updated model with OCR parameters θ_{t+1} becomes the next round’s opponent and generates new readings and masks. Algorithm 1 summarizes this procedure.

3.3. Optimization Analysis

The two losses play complementary roles in each update. The Reading Discrepancy Loss compares how much the current model improves on the opponent’s scores for reference and generated readings. When the reference has a small relative advantage, the comparison gives more weight to raising its score and lowering the generated score. FFL adds direct pressure to raise the reference score, so correction does not depend only on penalizing the generated reading. Equation (5) scores unmatched tokens using their original prefixes, with matched tokens still supplying context. The loss therefore targets reading disagreements without treating the extracted fragments as new sequences. Appendix B derives these contributions in (14).

Table 1: Reading accuracy and zero-shot generalization. Vary-600K is in-distribution while DocBank, IIT-CDIP, DocVQA and InfographicVQA are zero-shot. SFT-2 is trained for two supervised epochs and SP-DR-3 follows three self-play iterations. Parentheses show SP-DR-3’s change over SFT-2, with improvements in red and regressions in blue. CER is reported as a percentage. Full metrics and iteration trajectories appear in Appendix E.

Models	Method	Vary-600K		DocBank		IIT-CDIP		VQA (ANLS $\uparrow$)	
		CER $\downarrow$	NED $\uparrow$	CER $\downarrow$	NED $\uparrow$	CER $\downarrow$	NED $\uparrow$	DocVQA	InfoVQA
InternVL 3.5-4B	Base (low-res)	4.85	0.938	13.44	0.812	19.56	0.735	72.79	50.52
	SFT-2	2.12	0.971	11.60	0.848	17.83	0.778	73.01	48.15
	SP-DR-3	1.17(-0.95)	0.988(+0.017)	9.73(-1.87)	0.884(+0.036)	15.35(-2.48)	0.823(+0.045)	74.27(+1.26)	48.06(-0.09)
Qwen3 VL-4B	Base (low-res)	5.22	0.931	12.82	0.834	18.45	0.756	78.54	52.21
	SFT-2	2.40	0.967	10.95	0.868	16.71	0.802	81.22	55.34
	SP-DR-3	1.11(-1.29)	0.987(+0.020)	9.05(-1.90)	0.898(+0.030)	14.42(-2.29)	0.847(+0.045)	84.89(+3.67)	58.94(+3.60)

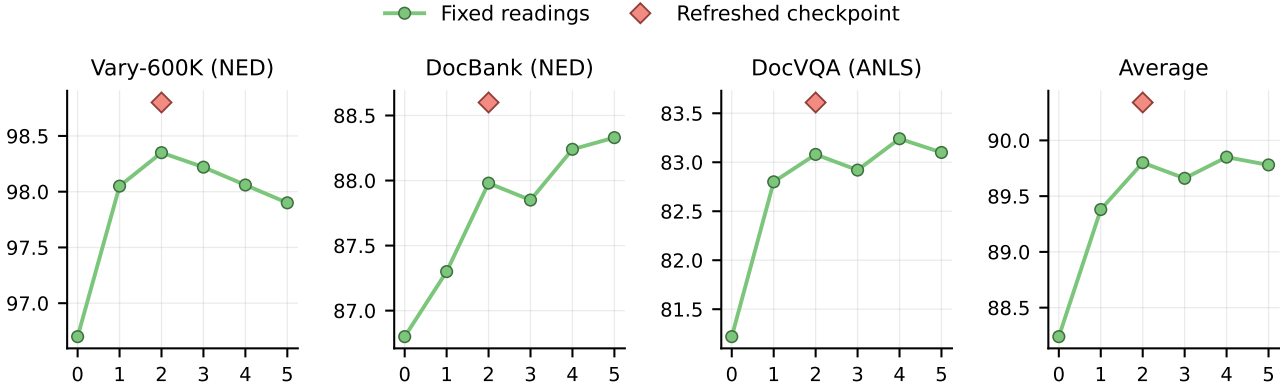


Figure 2: Fixed and refreshed self-play training on Qwen3-VL-4B. Green circles show the fixed-reading branch from its starting checkpoint through five local epochs. The red diamond represents a single checkpoint after two local epochs in the refreshed second round, which updates the opponent, generated readings, and discrepancy masks. The panels report Vary-600K NED, DocBank NED, DocVQA ANLS, and their mean, with NED scaled to 0–100 for the mean. The refreshed checkpoint scores higher than the fifth fixed-reading epoch in every panel.

4. Experiments

4.1. Experimental Setup

Base Models and Architecture. We integrate SP-DocReader with two vision language models, InternVL-3.5-4B (Wang et al., 2025a) and Qwen3-VL-4B (Bai et al., 2025). Following DocVLM (Nacson et al., 2025), we attach a specialized OCR modality to each backbone. EasyOCR (Kittinaradorn, 2020) supplies recognized text and bounding boxes and is not updated during training. DocVLM’s OCR encoder is derived from the T5-based DocFormerV2 model (Appalaraju et al., 2024; Raffel et al., 2020). We adapt this encoder’s input representation using the layout and text interleaving approach of LayTextLLM (Lu et al., 2025). Appendix C specifies this adaptation. For the main SP-DocReader model, the visual encoder and language backbone remain frozen, and only the OCR encoder, projection layer and learnable queries are updated.

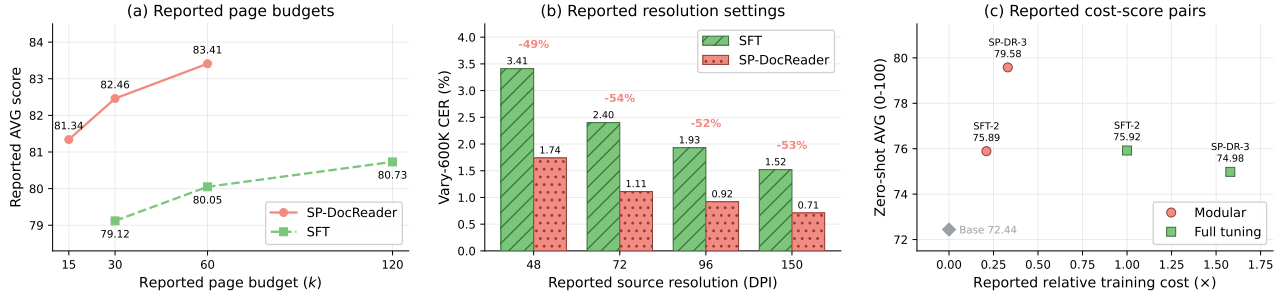


Figure 3: Reported Qwen3-VL-4B comparisons across training-page budgets, source-image resolutions and relative training costs. Panel (a) averages Vary-600K, DocBank and IIT-CDIP NED with DocVQA and InfographicVQA ANLS, weighting each dataset equally and scaling NED by 100. Panel (b) compares CER at four resolutions. Panel (c) uses the mean of DocBank, IIT-CDIP NED, DocVQA and InfographicVQA ANLS.

Datasets and Training Protocol. We draw training pages from Vary-600K (Wei et al., 2023). We use a balanced subset of 30,000 English and 30,000 Chinese pages for the supervised and self-play stages. Training uses two supervised epochs followed by self-play with generated readings. We use SFT-2 to denote the checkpoint after two supervised epochs and SP-DR- r to denote the checkpoint after r self-play rounds initialized from SFT-2. A self-play round is distinct from an epoch, and SP-DR-3 is the default model reported in the main results. Optimization settings and experimental protocols appear in Appendix C.

Evaluation Benchmarks and Metrics. We measure reading accuracy on the Vary-600K validation set and test zero-shot generalization on DocBank (Li et al., 2020) and IIT-CDIP (Lewis et al., 2006), which cover academic pages and scanned documents, respectively. We also evaluate question answering on DocVQA (Mathew et al., 2021) and InfographicVQA (Mathew et al., 2022). Additional evaluations use OmniDocBench (Ouyang et al., 2024), OCRBench v2 (Fu et al., 2024) and CC-OCR (Yang et al., 2024). The experiments use reduced-resolution inputs, with PDF pages rendered at 72 DPI and raster-image preprocessing specified in Appendix C. We report Character Error Rate (CER) as a percentage and retain the higher-is-better edit score labeled NED in the tables. Appendix C defines their computation from character edit distance (Levenshtein, 1966). Question answering uses Average Normalized Levenshtein Similarity (ANLS) (Biten et al., 2019) on a 0 to 100 scale. Appendix D describes the datasets and data audit, reference checks and benchmark scoring. Here, zero-shot refers to evaluation without target-benchmark adaptation during the supervised and self-play stages described in this paper. It does not imply that the pretrained backbones or OCR components have never encountered these datasets or related documents.

4.2. Main Results

Recognition and Generalization. Table 1 compares SP-DR-3 with SFT-2 on transcription and question answering. Vary-600K CER falls from 2.40 to 1.11 on Qwen3-VL-4B and from 2.12 to 1.17 on InternVL-3.5-4B. On Qwen3-VL-4B, zero-shot CER falls by 1.90 percentage points on DocBank and 2.29 on IIT-CDIP, while ANLS rises by 3.67 points on DocVQA and 3.60 on InfographicVQA. On InternVL-3.5-4B, transcription improves across all three reading benchmarks, while InfographicVQA ANLS changes slightly from 48.15 to 48.06. The transcription gains thus extend from the training domain to both zero-shot reading datasets. DocVQA also improves on both backbones, showing that better page reading

Table 2: Results on three additional document parsing benchmarks. Lower is better for the OmniDocBench edit distance and higher is better for the other two.

Models	Method	OmniDoc Edit↓	OCRBench v2↑	CC-OCR ↑
InternVL 3.5-4B	Base (low-res)	0.439	40.1	55.8
	SFT-2	0.366	46.5	62.7
	SP-DR-3	0.312(-0.054)	52.0(+5.5)	67.9(+5.2)
Qwen3 VL-4B	Base (low-res)	0.418	42.6	58.3
	SFT-2	0.347	49.1	65.4
	SP-DR-3	0.291(-0.056)	55.3(+6.2)	71.0(+5.6)

accompanies stronger question answering in this benchmark. Appendix E reports the full supervised and self-play trajectories and specifies the metric sets used in the reported averages.

Results on Additional Benchmarks. Table 2 reports aggregate scores on OmniDocBench, OCRBench v2 and CC-OCR. SP-DR-3 improves all three aggregate scores over SFT-2 on both backbones. On Qwen3-VL-4B, OmniDocBench edit distance decreases from 0.347 to 0.291, while OCRBench v2 rises from 49.1 to 55.3 and CC-OCR from 65.4 to 71.0. These results extend the gains beyond the five primary evaluation datasets.

Fixed versus refreshed training records. Figure 2 compares continued training on fixed readings with a refreshed second-round checkpoint on Qwen3-VL-4B. The refreshed checkpoint reaches an average near 90.3 after two local epochs, above the fixed-reading branch’s average near 89.8 after five local epochs. Each plotted metric also favors the refreshed checkpoint over the fifth fixed-reading epoch. Refreshing updates the opponent, generated readings and LCS masks together, giving the next round new errors to learn from. Appendix C describes the separate refresh control with matched update counts.

4.3. Ablation Study

Component Analysis. Table 3 compares the full model with two ablation rows on Qwen3-VL-4B. Without the contrastive Reading Discrepancy Loss, the model still applies FFL to unmatched reference tokens, but Vary-600K CER rises from 1.11 to 1.88. Without FFL, the model retains the reference-versus-generated comparison, but DocVQA ANLS falls from 84.89 to 82.96. Both reduced variants improve on SFT-2 across all five columns, and combining the losses gives the best score in every column. Appendix C details the component settings and additional controls.

Data efficiency. Figure 3(a) compares Qwen3-VL-4B results at labeled page budgets of 15k, 30k and 60k for SP-DocReader and 30k, 60k and 120k for SFT. AVG gives equal weight to Vary-600K, DocBank and IIT-CDIP NED, scaled by 100, and DocVQA and InfographicVQA ANLS. It combines in-distribution reading with zero-shot performance, unlike the zero-shot-only AVG in panel (c). At 30k training pages, SP-DocReader reaches an AVG of 82.46, exceeding the 80.73 obtained by SFT with 120k pages. The comparison highlights the value of self-play updates when labeled pages are limited, with Vary-600K included in the overall score. Appendix C details how distinct reference pages are counted.

Table 3: Component analysis on Qwen3-VL-4B. CER is a percentage and the VQA columns report ANLS.

Method	Vary CER↓	DocBank CER↓	IIT-CDIP CER↓	DocVQA↑	InfoVQA↑
SFT-2	2.40	10.95	16.71	81.22	55.34
w/o RDM	1.88	10.31	15.36	83.47	57.81
w/o FFL	1.57	9.88	15.04	82.96	57.29
SP-DR (full)	1.11	9.05	14.42	84.89	58.94

Table 4: Architecture ablation on Qwen3-VL-4B.

Method	Vary CER↓	DocBank CER↓	IIT-CDIP CER↓	DocVQA↑	Train. Params
OCR-free	1.04	11.93	17.61	79.62	3.3B
Full-parameter	0.98	11.28	16.87	80.41	3.8B
Modular (ours)	1.11	9.05	14.42	84.89	0.5B

Architecture ablations. Table 4 compares adaptation variants on Qwen3-VL-4B. Modular tuning has higher Vary-600K CER than OCR-free and full-parameter tuning, but achieves the lowest CER on DocBank and IIT-CDIP and the highest DocVQA ANLS. The contrast is clearest against full tuning: its Vary-600K CER is 0.98, compared with 1.11 for modular tuning, yet its DocBank and IIT-CDIP CER are 11.28 and 16.87, compared with 9.05 and 14.42. Thus, the best in-domain result does not yield the best zero-shot reading scores in this comparison. Modular tuning achieves those scores with 0.5B trainable parameters, compared with 3.3B for OCR-free tuning and 3.8B for full-parameter tuning. Appendix F provides their training trajectories, while Appendix C specifies the architecture configurations.

Resolution robustness. Figure 3(b) compares Vary-600K CER from 48 to 150 DPI. SP-DocReader has lower CER than SFT at all four tested resolutions. At 48 DPI, CER is 3.41 for SFT and 1.74 for SP-DocReader. At 150 DPI, it is 1.52 and 0.71, respectively. The advantage persists across the tested resolution range. Appendix C details the branch inputs, token budgets and resolution protocol.

4.4. Cost Analysis

Figure 3(c) uses the four-task zero-shot AVG from Table 7. From SFT-2 to the third self-play round, it rises from 75.89 to 79.58 for modular tuning, while full tuning changes from 75.92 to 74.98. The base score is 72.44 from Table 6. At the third round, modular tuning reaches an AVG of 79.58 at a relative training cost of 0.33, compared with 74.98 at 1.58 for full tuning. Costs are normalized by the complete joint-tuning SFT-2 adaptation run. The reported inference overhead is about ten percent relative to the backbone without OCR. Appendix C details the cumulative training-cost accounting and end-to-end inference protocol, including external OCR.

4.5. Qualitative Results

Figure 4 compares readings of a constitutional document and a mathematics page at low resolution. In the constitutional excerpt, SFT-2 misreads the repeal year and changes “State legislatures” to “State legislators.” SP-DocReader restores the date and the original wording. On the mathematics page, SFT-2 introduces errors into a dimension formula and reads the reference to Theorem 3.14 as Theorem 3.11. SP-DocReader recovers the theorem number and transcribes the formula more accurately. The lemma excerpt further shows the importance of preserving symbols alongside ordinary text. Together,

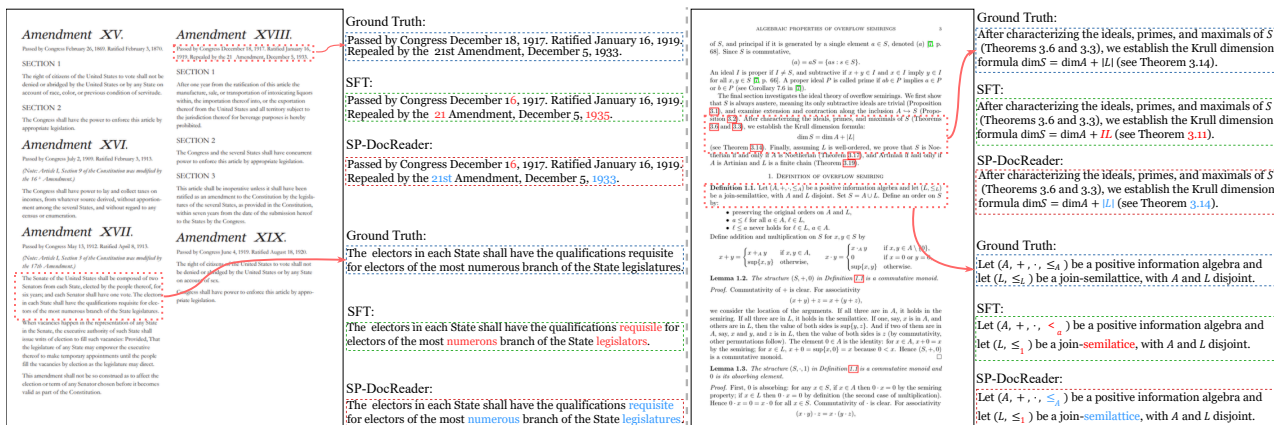


Figure 4: Qualitative comparisons on two low-resolution documents. Blue, green and red dashed boxes contain the reference, supervised and SP-DocReader readings, respectively. SP-DocReader corrects the discrepancies.

the examples illustrate how discrepancy-focused training improves recognition of both prose and mathematical notation.

5. Conclusion

We introduced SP-DocReader, a self-play framework that trains a modular document reader on differences between generated readings and reference text. Reading Discrepancy Masking selects unmatched model-token positions while preserving their original context, and Focused Fidelity Loss directly supervises unmatched reference tokens. The OCR branch is updated while the visual encoder and language backbone remain frozen.

Compared with two-epoch supervised fine-tuning, three self-play rounds reduce Vary-600K CER from 2.40 to 1.11 on Qwen3-VL-4B and from 2.12 to 1.17 on InternVL-3.5-4B. On Qwen3-VL-4B, DocBank and IIT-CDIP CER decrease by 1.90 and 2.29 percentage points, while DocVQA and InfographicVQA ANLS increase by 3.67 and 3.60 points. The full objective also outperforms either reduced-loss variant in the component analysis. At a 30k-page training budget, SP-DocReader reaches a five-task AVG of 82.46, compared with 80.73 for SFT using 120k pages. Together, these results show that focusing self-play on reading discrepancies improves transcription across the evaluated document datasets and supports strong performance with fewer reference pages in the tested budget comparison.

Limitations

SP-DocReader focuses on single-page transcription rather than multi-page reading or structured extraction. Its fixed generation budget may limit coverage of longer pages. Our experiments use English and Chinese training pages and evaluate two vision language backbones. The results therefore do not establish generalization to other languages or model families.

Ethical Considerations

Document transcription can expose personal or confidential information, so its use should respect data access conditions and protect sensitive content. Model outputs may contain recognition errors and

should be checked against the source documents before use in consequential decisions. Any reuse or release of data and model components should comply with their applicable licenses and permissions.

References

- Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R. Manmatha. 2021. [DocFormer: End-to-end transformer for document understanding](#). In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 993–1003.
- Srikar Appalaraju, Peng Tang, Qi Dong, Nishant Sankaran, Yichu Zhou, and R. Manmatha. 2024. [DocFormerv2: Local features for document understanding](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 709–718.
- Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. [Wasserstein generative adversarial networks](#). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR.
- Gilles Baechler, Srinivas Sunkara, Maria Wang, Fedir Zubach, Hassan Mansoor, Vincent Etter, Victor Cărbune, Jason Lin, Jindong Chen, and Abhanshu Sharma. 2024. [ScreenAI: A vision-language model for UI and infographics understanding](#). *arXiv preprint arXiv:2402.04615*.
- Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. [Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond](#). *arXiv preprint arXiv:2308.12966*.
- Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025. [Qwen3-VL technical report](#). *arXiv preprint arXiv:2511.21631*.
- Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, and 16 others. 2024. [PaliGemma: A versatile 3B VLM for transfer](#). *arXiv preprint arXiv:2407.07726*.
- Ali Furkan Biten, Rubèn Tito, Andres Mafla, Lluís Gomez, Marçal Rusiñol, Ernest Valveny, C. V. Jawahar, and Dimosthenis Karatzas. 2019. [Scene text visual question answering](#). In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4291–4301.
- Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. 2023. Nougat: Neural optical understanding for academic documents. *arXiv preprint arXiv:2308.13418*.
- Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yimin Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, and 21 others. 2024a. [Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling](#). *arXiv preprint arXiv:2412.05271*.
- Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, Ji Ma, Jiaqi Wang, Xiaoyi Dong, Hang Yan, Hewei Guo, Conghui He, Botian Shi, Zhenjiang Jin, Chao Xu, and 16 others. 2024b. [How far are we to GPT-4V? closing the gap to commercial multimodal models with open-source suites](#). *arXiv preprint arXiv:2404.16821*.

- Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024c. [Self-play fine-tuning converts weak language models to strong language models](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 6621–6642. PMLR.
- Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2024a. How abilities in large language models are affected by supervised fine-tuning data composition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 177–198.
- Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. 2024b. [RLHF workflow: From reward modeling to online RLHF](#). *arXiv preprint arXiv:2405.07863*.
- Yuning Du, Chenxia Li, Ruoyu Guo, Xiaoting Yin, Weiwei Liu, Jun Zhou, Yifan Bai, Zilin Yu, Yehua Yang, Qingqing Dang, and Haoshuang Wang. 2020. [PP-OCR: A practical ultra lightweight OCR system](#). *arXiv preprint arXiv:2009.09941*.
- Ling Fu, Biao Yang, Zhebin Kuang, Jiajun Song, Yuzhe Li, Linghao Zhu, Qidi Luo, Xinyu Wang, Hao Lu, Mingxin Huang, Zhang Li, Guozhi Tang, Bin Shan, Chunhui Lin, Qi Liu, Binghong Wu, Hao Feng, Hao Liu, Can Huang, and 5 others. 2024. [OCRBench v2: An improved benchmark for evaluating large multimodal models on visual text localization and reasoning](#). *arXiv preprint arXiv:2501.00321*.
- Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. [Generative adversarial nets](#). In *Advances in Neural Information Processing Systems*, volume 27.
- Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. 2022. Accelerate: Training and inference at scale made simple, efficient and adaptable. <https://github.com/huggingface/accelerate>.
- Rakpong Kittinaradorn. 2020. EasyOCR. <https://github.com/JaidedAI/EasyOCR>. Software. Accessed September 13, 2026.
- S. Kullback and R. A. Leibler. 1951. [On information and sufficiency](#). *The Annals of Mathematical Statistics*, 22(1):79–86.
- V. I. Levenshtein. 1966. [Binary codes capable of correcting deletions, insertions, and reversals](#). *Soviet Physics Doklady*, 10(8):707–710.
- D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman, and J. Heard. 2006. [Building a test collection for complex document information processing](#). In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 665–666.
- Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2024. [LLaVA-OneVision: Easy visual task transfer](#). *arXiv preprint arXiv:2408.03326*.
- Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. 2020. [DocBank: A benchmark dataset for document layout analysis](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 949–960. International Committee on Computational Linguistics.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization*

Branches Out, pages 74–81.

- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. [Visual instruction tuning](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916.
- Jinghui Lu, Haiyang Yu, Yanjie Wang, Yongjie Ye, Jingqun Tang, Ziwei Yang, Binghong Wu, Qi Liu, Hao Feng, Han Wang, Hao Liu, and Can Huang. 2025. [A bounding box is worth one token - interleaving layout and text in a large language model for document understanding](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7252–7273. Association for Computational Linguistics.
- Gen Luo, Xue Yang, Wenhan Dou, Zhaokai Wang, Jiawen Liu, Jifeng Dai, Yu Qiao, and Xizhou Zhu. 2024. [Mono-intervl: Pushing the boundaries of monolithic multimodal large language models with endogenous visual pre-training](#). *arXiv preprint arXiv:2410.08202*.
- Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*.
- Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. 2022. [InfographicVQA](#). In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706.
- Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. 2021. [DocVQA: A dataset for VQA on document images](#). In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2200–2209.
- Alfred Müller. 1997. [Integral probability metrics and their generating classes of functions](#). *Advances in Applied Probability*, 29(2):429–443.
- Mor Shpigel Nacson, Aviad Aberdam, Roy Ganz, Elad Ben Avraham, Alona Golts, Yair Kittenplon, Shai Mazor, and Ron Litman. 2025. [DocVLM: Make your VLM an efficient reader](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 29005–29015.
- Linke Ouyang, Yuan Qu, Hongbin Zhou, Jiawei Zhu, Rui Zhang, Qunshu Lin, Bin Wang, Zhiyuan Zhao, Man Jiang, Xiaomeng Zhao, Jin Shi, Fan Wu, Pei Chu, Minghao Liu, Zhenxiang Li, Chao Xu, Bo Zhang, Botian Shi, Zhongying Tu, and Conghui He. 2024. [OmniDocBench: Benchmarking diverse PDF document parsing with comprehensive annotations](#). *arXiv preprint arXiv:2412.07626*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.
- Matt Post. 2018. [A call for clarity in reporting BLEU scores](#). In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. [Learning transferable visual models from natural language supervision](#). In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea

- Finn. 2023. [Direct preference optimization: Your language model is secretly a reward model](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Corby Rosset, Ching-An Cheng, Arindam Mitra, Michael Santacrose, Ahmed Awadallah, and Tengyang Xie. 2024. [Direct Nash optimization: Teaching language models to self-improve with general preferences](#). *arXiv preprint arXiv:2404.03715*.
- Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. 2023. [Hierarchical multimodal transformers for multipage DocVQA](#). *Pattern Recognition*, 144:109834.
- Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, Bo Zhang, Liqun Wei, Zhihao Sui, Wei Li, Botian Shi, Yu Qiao, Dahua Lin, and Conghui He. 2024a. [MinerU: An open-source solution for precise document content extraction](#). *arXiv preprint arXiv:2409.18839*.
- Dongsheng Wang, Natraj Raman, Mathieu Sibue, Zhiqiang Ma, Petr Babkin, Simerjot Kaur, Yulong Pei, Armineh Nourbakhsh, and Xiaomo Liu. 2024b. [DocLLM: A layout-aware generative language model for multimodal document understanding](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8529–8548. Association for Computational Linguistics.
- Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, and 56 others. 2025a. [InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency](#). *arXiv preprint arXiv:2508.18265*.
- Wenjin Wang, Yunhao Li, Yixin Ou, and Yin Zhang. 2023a. [Layout and task aware instruction prompt for zero-shot document image question answering](#). *arXiv preprint arXiv:2306.00526*.
- Yibo Wang, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Lijun Zhang. 2025b. [SPACE: Noise contrastive estimation stabilizes self-play fine-tuning for large language models](#). *arXiv preprint arXiv:2512.07175*.
- Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023b. [How far can camels go? exploring the state of instruction tuning on open resources](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 74764–74786.
- Haoran Wei, Lingyu Kong, Jinyue Chen, Liang Zhao, Zheng Ge, Jinrong Yang, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. 2023. [Vary: Scaling up the vision vocabulary for large vision-language models](#). *arXiv preprint arXiv:2312.06109*.
- Haoran Wei, Chenglong Liu, Jinyue Chen, Jia Wang, Lingyu Kong, Yanming Xu, Zheng Ge, Liang Zhao, Jianjian Sun, Yuang Peng, Chunrui Han, and Xiangyu Zhang. 2024. [General OCR theory: Towards OCR-2.0 via a unified end-to-end model](#). *arXiv preprint arXiv:2409.01704*.
- Haoran Wei, Yaofeng Sun, and Yukun Li. 2025. Deepseek-ocr: Contexts optical compression. *arXiv preprint arXiv:2510.18234*.

- Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. 2024. [Self-play preference optimization for language model alignment](#). *Preprint*, arXiv:2405.00675.
- Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. [LayoutLM: Pre-training of text and layout for document image understanding](#). In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200.
- Zhibo Yang, Jun Tang, Zhaohai Li, Pengfei Wang, Jianqiang Wan, Humen Zhong, Xuejing Liu, Mingkun Yang, Peng Wang, Yuliang Liu, LianWen Jin, Xiang Bai, Shuai Bai, and Junyang Lin. 2024. [CC-OCR: A comprehensive and challenging OCR benchmark for evaluating large multimodal models in literacy](#). *arXiv preprint arXiv:2412.02210*.
- Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Yuhao Dan, Chenlin Zhao, Guohai Xu, Chenliang Li, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. 2023. [mPLUG-DocOwl: Modularized multimodal large language model for document understanding](#). *arXiv preprint arXiv:2307.02499*.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. [STaR: Bootstrapping reasoning with reasoning](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 15476–15488.
- Pan Zhang, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Rui Qian, Lin Chen, Qipeng Guo, Haodong Duan, Bin Wang, Linke Ouyang, Songyang Zhang, Wenwei Zhang, Yining Li, Yang Gao, Peng Sun, Xinyue Zhang, Wei Li, Jingwen Li, Wenhai Wang, and 8 others. 2024. [InternLM-XComposer-2.5: A versatile large vision language model supporting long-contextual input and output](#). *arXiv preprint arXiv:2407.03320*.

A. Additional Related Work

Fine-grained Alignment in Vision Language Models. CLIP learns visual representations by matching images with their paired text (Radford et al., 2021). LLaVA connects a vision encoder to a language model and uses visual instruction tuning for multimodal interaction (Liu et al., 2023). For text-rich images, Qwen-VL increases the input resolution during training and uses a fixed set of visual queries (Bai et al., 2023), whereas InternVL 1.5 divides images into tiles for dynamic high-resolution processing (Chen et al., 2024b). mPLUG-DocOwl uses a visual abstractor and document-oriented instruction tuning to connect visual features with a language model (Ye et al., 2023).

Adversarial Learning and Iterative Self-Improvement. Generative adversarial networks train a generator and discriminator through a minimax objective (Goodfellow et al., 2014). Integral probability metrics compare distributions through expectations over a specified function class (Müller, 1997). The Wasserstein generative adversarial network (WGAN) uses a Lipschitz-constrained critic to approximate the Wasserstein distance (Arjovsky et al., 2017).

STaR repeatedly generates rationales, filters them by answer correctness, and fine-tunes on the retained solutions (Zelikman et al., 2022). The reinforcement learning from human feedback (RLHF) workflow of Dong et al. (2024b) trains a proxy reward model on public preference data and uses it to label response pairs for online policy updates.

Studies of instruction tuning examine how supervised data affects different capabilities (Dong et al., 2024a; Wang et al., 2023b). Continual fine-tuning studies report forgetting when language models learn successive tasks (Luo et al., 2023).

B. Supporting Derivations

Gradient of the combined objective. Fix an input (I, x) , a reading pair $(y, \hat{y})$, the opponent θ_t and both position sets. For the current model, write

$$\begin{aligned} s_y &= S_\theta(y, \mathcal{E}_y \mid I, x), \\ s_{\hat{y}} &= S_\theta(\hat{y}, \mathcal{E}_{\hat{y}} \mid I, x), \\ u &= \beta[\Delta_t(y) - \Delta_t(\hat{y})]. \end{aligned}$$

The per-pair contribution to (12) is

$$\mathcal{L}_{\text{pair}}(\theta) = \log(1 + e^{-u}) - \lambda s_y.$$

Define $\sigma(v) = (1 + e^{-v})^{-1}$. The logistic term has derivative $-\sigma(-u)$ with respect to u . Since the opponent and masks are fixed,

$$\nabla_\theta u = \beta(\nabla_\theta s_y - \nabla_\theta s_{\hat{y}}).$$

Combining the two terms gives

$$\begin{aligned} \nabla_\theta \mathcal{L}_{\text{pair}} &= -[\beta\sigma(-u) + \lambda]\nabla_\theta s_y \\ &\quad + \beta\sigma(-u)\nabla_\theta s_{\hat{y}}. \end{aligned} \tag{14}$$

Both score gradients retain the full original prefixes and sum over the selected positions. Averaging these contributions follows (16), including pairs with empty masks. This identity describes the existing objective.

Scope of the variance interpretation. Fix the current parameters θ and opponent θ_t and consider variation across the round’s training examples. Each example retains its fixed reading pair and masks. In (14), both score gradients and the coefficient $\beta\sigma(-u)$ can vary across examples. The variance of the combined gradient depends on the variances and covariance of the two weighted contributions, with FFL included in the ground-truth contribution. The number of selected positions alone does not determine these quantities. We make no independence or equal-variance assumption about token contributions and derive no variance bound relative to full-sequence training.

Loss masking and computation. When vocabulary logits are computed at every position before applying the loss masks, those dense projection operations have already been performed. Reducing projection work through masking requires an implementation that skips the excluded projections. Selected losses can depend on preceding hidden states through contextual computation, so a matched position need not have zero gradient in the internal computation. The frozen backbone weights are not optimized, but its derivatives with respect to the OCR representations are still needed to update the OCR branch.

Scope of cost comparisons. A complete adaptation-cost comparison must include input preprocessing, supervised initialization, generation, LCS alignment, both current-model and opponent scoring, and optimization over all required rounds. A single step or round does not cover this total, and shared operations must not be counted twice. Runtime and memory claims require measurements with a stated scope.

C. Implementation Details

OCR architecture. We interleave projected bounding-box embeddings with their OCR text embeddings at the input of the DocFormerV2-derived encoder. This replaces the source design’s additive fusion of text and spatial features rather than adding a complete LayTextLLM model. We retain DocVLM’s query compression design without the source DocFormerV2 visual branch or decoder. The input embedding layers belong to the trainable OCR encoder. The output projection maps query features to the frozen VLM. Before training, we check input compatibility, encoded length and the trainable parameter count.

We build SP-DocReader on the Accelerate library for distributed training (Gugger et al., 2022). All runs use two NVIDIA H20 GPUs with 96GB each. We set a batch of 2 per device, which gives a global batch of 4. The projection layer and the learnable queries use a learning rate of 10^{-4} , and the OCR encoder uses 5×10^{-5} . These two learning rates match those used in the initial OCR alignment stage of DocVLM (Nacson et al., 2025). We optimize with AdamW under a cosine schedule, and we warm up over the first ten percent of the steps. The weight λ on the Focused Fidelity Loss is set to 0.5 by default.

The reported training uses two supervised epochs followed by up to five self-play rounds, with the third round as the default model unless stated otherwise. The Vary-600K training subset contains 30,000 English and 30,000 Chinese pages.

Supervision-budget accounting. We count distinct reference pages used in project-specific initialization, supervised training and all self-play rounds. Other task-specific training stages also enter this count. Repeated use of the same pages does not create new reference pages. We record references used for development and model selection, their union with the training pages, and the held-out evaluation data separately from the provenance of common pretrained components. Training-reference counts do not measure human annotation time or cost.

Data-efficiency comparison. For Qwen3-VL-4B, we use training subsets of 15k, 30k, 60k and 120k

pages, nested across budgets and balanced between English and Chinese. Each budget starts from the same pre-task initialization and completes its own two-epoch supervised stage. The 15k, 30k and 60k variants then use only their respective training pages for three self-play rounds. A smaller-budget variant does not inherit a checkpoint adapted on a larger training set. Any project-specific initialization training must remain within the stated budget or be included in the reported training union.

We compare OCR-only modular SFT and self-play under the same input and scoring protocol. The primary comparison uses the 30k SP-DR-3 and 120k SFT-2 checkpoints, with development data recorded separately. We use fixed stage endpoints and the same held-out evaluation sets, while recording cumulative computation separately from reference-page counts.

Reading generation. We greedily generate one reading per page at each round’s start and reuse it within the round. Decoding stops at EOS or 1,024 new tokens, excluding the prompt, with no forced EOS, resampling, candidate filtering, or repetition constraints.

Length protocol. We use a 1,024-new-token response budget for self-play generation and evaluation. This budget excludes the prompt and does not cap the multimodal inputs or the teacher-forced reference. We use one decoding call per task request, without continuation or stitching multiple readings. SFT and self-play scoring retain the complete reference and its original prefixes. Each response sequence together with its conditioning input must fit the model and runtime limits. We check sequence lengths before running and do not silently truncate or remove samples.

Coverage evaluation. We score each capped page transcription against the complete frozen reference, without shortening the reference to the output length or excluding pages because generation reaches the cap. We record reference lengths with each backbone’s tokenizer, output lengths and termination reasons. Emitting EOS does not guarantee complete transcription, and reaching the cap alone does not prove an omission. VQA and other tasks retain their task-specific references and evaluation rules.

Transcription scoring. We fix each reference’s reading order and serialization. We apply Unicode NFC normalization and collapse whitespace in both references and predictions, while preserving case, punctuation and content order. CER is the total character edit count divided by the total reference characters, multiplied by 100. For each page, we compute normalized edit similarity as $1 - d(r, p) / \max(|r|, |p|)$, where r and p are the normalized reference and prediction and d is character edit distance. We average this similarity over pages, assigning 1 when both strings are empty.

Auxiliary metrics. We use a fixed SacreBLEU zh tokenizer (Post, 2018) for word error rate (WER), Token-F1, ROUGE-L (Lin, 2004) and BLEU-4 (Papineni et al., 2002). Its units include Chinese characters and non-Chinese words or punctuation, rather than model tokens. WER uses corpus edit counts. Token-F1 and ROUGE-L F1 are averaged over pages, and BLEU-4 uses corpus statistics. This evaluation tokenization does not change the model-token LCS or EOS rules used in training.

Branch inputs. We provide the same reduced-resolution page to the visual preprocessing pipeline and EasyOCR. We render PDFs at 72 DPI when their physical page dimensions are available. For raster-only pages, we preserve aspect ratio and reduce the longer edge to at most 842 pixels, without upsampling smaller sources or assigning an assumed DPI. EasyOCR uses a fixed configuration with a canvas size of 2,560 and a magnification ratio of 1.0. Both branches receive the same page content, without a higher-resolution OCR input.

Visual and OCR budgets. We use one full-page 448×448 tile without an extra thumbnail for standard InternVL3.5. For Qwen3-VL, we use an image-pixel budget from 64×32^2 to 256×32^2 , using its native processor. The OCR branch produces 64 query representations per page. We record processed image

sizes, tile counts and the actual visual positions passed to the language model. The query count does not limit the OCR encoder’s input text.

OCR input checks. We retain the complete extracted OCR text and boxes, record their encoded length and verify that they fit the OCR encoder before running. Inputs are checked against the encoder capacity without silent truncation or removal. Fixed external OCR outputs can be reused across matched comparisons, but trainable OCR representations must be recomputed. OCR preprocessing and encoding remain part of the system cost.

Resolution check. We evaluate fixed SFT-2 and SP-DR-3 checkpoints under four source-image settings, without retraining at each setting. For PDFs with known page dimensions, we use 48, 72, 96 and 150 DPI. For raster-only pages, we use long-edge caps of 561, 842, 1,123 and 1,754 pixels without upsampling. Both branches receive the image for that setting, and OCR outputs are recomputed and shared between the two methods. Visual preprocessing rules, budget limits, query counts and decoding settings remain fixed, with actual token counts recorded. This evaluates the combined response of both input branches to downsampling, rather than isolating the visual encoder.

Component comparisons. We initialize all component variants from the same modular SFT-2 checkpoint and retain the LCS masks. The full model uses $\mathcal{L}_{\text{RDM}} + 0.5\mathcal{L}_{\text{FFL}}$. Removing the contrastive term leaves $0.5\mathcal{L}_{\text{FFL}}$, whereas removing FFL leaves $\mathcal{L}_{\text{RDM}}$. Each variant generates readings with its own previous-round model and refreshes them between rounds. We use two epochs per round and report the third round under the same data and update budget.

Refresh comparison. For Qwen3-VL-4B, we branch from a common checkpoint after two supervised epochs and two epochs in the first self-play round. Both branches continue from identical model weights. The fixed branch retains the first round’s opponent, generated readings and masks. The refreshed branch uses the common checkpoint as its new opponent, regenerates the readings and recomputes the masks and opponent scores.

Both branches use the same 60k training pages for two further epochs, with matched batch order, optimizer settings and update counts. We compare their endpoints after four cumulative self-play epochs and report the shared supervised stage separately. Computation includes generation, alignment and scoring, so equal update counts do not imply equal cost. This test concerns the joint refresh of the opponent and training records, not regenerated readings alone.

Minimal controls. We use two additional controls on Qwen3-VL-4B from the same SFT-2 checkpoint. Continued SFT uses three two-epoch blocks with full-reference negative log-likelihood. The masking control scores every response position in the contrastive term while retaining discrepancy-based FFL and its coefficient. It uses three two-epoch rounds and refreshes its own readings between rounds. We also compare first-round checkpoints, where the self-play variants share generated readings and FFL masks. Later rounds compare complete training procedures because their readings and masks may diverge.

For each of three training seeds, the branches share that seed’s SFT-2 checkpoint. Data splits, stage endpoints and the optimizer reset policy are fixed across comparisons. We use the frozen Vary-600K and DocVQA evaluation sets and a fixed 5k-page DocBank test subset, keeping this subset separate from full-test results. We retain per-seed scores and measure actual update counts and cumulative computation.

Architecture comparisons. We compare language-model tuning without OCR, joint language-model and OCR tuning, and OCR-only modular tuning. The visual encoder and its connector remain frozen in

all three variants. Each variant completes its own two-epoch SFT stage before self-play, using identical pretrained weights for shared modules. The OCR-equipped variants share the OCR initialization and fixed external OCR outputs. This comparison concerns complete adaptation strategies rather than changing trainable modules only after a shared SFT checkpoint.

Training-cost measurement. We record elapsed time and allocated GPU-hours from local input preparation and model loading to the saved checkpoint. The ledger includes external OCR, supervised training, each round’s generation, alignment, opponent scoring, optimization and required I/O. We record stage and cumulative costs separately, count shared preprocessing once per standalone adaptation run, and retain a separate ledger of actual project-wide computation. Evaluation and startup costs are recorded separately.

Cost comparison. We record costs for OCR-only and joint language-model and OCR tuning on Qwen3-VL-4B. Both retain the frozen visual pathway. We compare their SFT-2 and SP-DR-3 checkpoints under the same data and input protocol. For normalized costs, we use the complete joint-tuning SFT-2 adaptation run as the denominator. Each quality score must use the metric protocol and checkpoint corresponding to its cost.

Inference measurement. We perform a batch-one comparison between the complete reader and its backbone without OCR on one H20 GPU. The same fixed English and Chinese pages use matched image and decoding settings. End-to-end latency includes page processing, external OCR, encoding and generation after model loading, with startup recorded separately. We record output lengths, response failures, latency distributions, throughput and peak memory. A separate cached-OCR measurement does not replace the online result.

Deterministic token alignment. We compute LCS lengths for all suffix pairs before recovering matches. Let $n = |y|$ and $m = |\hat{y}|$, and let $D_{i,j}$ be the LCS length of the suffixes starting at positions i and j . We set $D_{n+1,j} = D_{i,m+1} = 0$ and compute

$$D_{i,j} = \begin{cases} 1 + D_{i+1,j+1}, & y_i = \hat{y}_j, \\ \max(D_{i+1,j}, D_{i,j+1}), & y_i \neq \hat{y}_j. \end{cases} \quad (15)$$

Starting at $(i, j) = (1, 1)$, we match equal tokens at their original positions and increment both indices. At unequal tokens, we increment i only when $D_{i+1,j} > D_{i,j+1}$. Otherwise, we increment j , so ties skip the synthetic token while retaining the ground-truth position. Each step preserves an optimal LCS for the remaining suffixes. Recovery stops when either sequence ends. All unmatched original indices, including trailing positions, form $\mathcal{E}_y$ and $\mathcal{E}_{\hat{y}}$. Scoring retains each sequence’s full original prefix. Matched EOS tokens are masked out. If $n < 1,024$ and the synthetic reading lacks EOS, the ground-truth EOS contributes to RDM and FFL. We add no separate EOS loss.

Empty masks and batch averaging. An empty position set has score zero in (5). Pairs with one empty mask are retained. FFL is zero whenever $\mathcal{E}_y$ is empty. With both masks empty, (10) gives the constant $\log 2$ and (11) gives zero, so neither term contributes a gradient. For a batch of B original samples, let r_b and h_b be the per-sample RDM and FFL losses. We use

$$\mathcal{L}_{\text{batch}} = \frac{1}{B} \sum_{b=1}^B (r_b + \lambda h_b). \quad (16)$$

The denominator remains B , including samples with both masks empty.

Table 5: Full reading accuracy on the Vary-600K validation set across all supervised epochs and self-play rounds. Each cell shows the change over the previous row, and SP-DR-1 is measured against SFT-2. Improvements are in red and regressions in blue. AVG is the mean of NED, Token-F1, ROUGE-L, and BLEU-4 on a 0 to 100 scale. SP-DR-3 is the model reported in the main text.

Models	Method	CER(%)↓	WER(%)↓	NED↑	Token-F1↑	ROUGE-L↑	BLEU-4↑	AVG↑
InternVL 3.5-4B	Base	4.85	9.12	0.938	0.941	87.29	80.44	88.91
	SFT-1	2.31(-2.54)	4.60(-4.52)	0.969(+0.031)	0.972(+0.031)	92.87(+5.58)	88.32(+7.88)	93.82(+4.91)
	SFT-2	2.12(-0.19)	4.00(-0.60)	0.971(+0.002)	0.975(+0.003)	93.39(+0.52)	89.04(+0.72)	94.26(+0.44)
	SFT-3	2.34(+0.22)	4.37(+0.37)	0.968(-0.003)	0.976(+0.001)	92.93(-0.46)	88.60(-0.44)	93.98(-0.28)
	SP-DR-1	1.77(-0.35)	3.49(-0.51)	0.978(+0.007)	0.982(+0.007)	95.28(+1.89)	91.43(+2.39)	95.68(+1.42)
	SP-DR-2	1.35(-0.42)	2.83(-0.66)	0.984(+0.006)	0.989(+0.007)	96.77(+1.49)	93.01(+1.58)	96.77(+1.09)
	SP-DR-3	1.17(-0.18)	2.35(-0.48)	0.988(+0.004)	0.991(+0.002)	97.90(+1.13)	94.52(+1.51)	97.58(+0.81)
	SP-DR-4	1.13(-0.04)	2.24(-0.11)	0.989(+0.001)	0.992(+0.001)	98.06(+0.16)	94.79(+0.27)	97.74(+0.16)
	SP-DR-5	1.11(-0.02)	2.20(-0.04)	0.990(+0.001)	0.993(+0.001)	98.13(+0.07)	94.88(+0.09)	97.83(+0.09)
Qwen3 VL-4B	Base	5.22	9.89	0.931	0.935	86.43	79.11	88.04
	SFT-1	2.50(-2.72)	5.17(-4.72)	0.964(+0.033)	0.968(+0.033)	91.99(+5.56)	87.24(+8.13)	93.11(+5.07)
	SFT-2	2.40(-0.10)	4.94(-0.23)	0.967(+0.003)	0.971(+0.003)	92.63(+0.64)	88.05(+0.81)	93.62(+0.51)
	SFT-3	2.66(+0.26)	5.33(+0.39)	0.962(-0.005)	0.967(-0.004)	91.88(-0.75)	87.43(-0.62)	93.05(-0.57)
	SP-DR-1	1.82(-0.58)	4.06(-0.88)	0.9835(+0.0165)	0.978(+0.007)	94.47(+1.84)	90.50(+2.45)	95.28(+1.66)
	SP-DR-2	1.45(-0.37)	3.30(-0.76)	0.988(+0.0045)	0.987(+0.009)	96.29(+1.82)	92.63(+2.13)	96.61(+1.33)
	SP-DR-3	1.11(-0.34)	2.39(-0.91)	0.987(-0.001)	0.992(+0.005)	97.74(+1.45)	94.49(+1.86)	97.53(+0.93)
	SP-DR-4	1.09(-0.02)	2.15(-0.24)	0.989(+0.002)	0.994(+0.002)	98.02(+0.28)	94.78(+0.29)	97.78(+0.24)
	SP-DR-5	1.06(-0.03)	2.18(+0.03)	0.990(+0.001)	0.994(+0.000)	98.10(+0.08)	94.86(+0.08)	97.84(+0.06)

D. Benchmarks and Data Sources

- **Vary-600K** is a released subset of the pretraining data used in Vary (Wei et al., 2023).¹ It contains approximately 600,000 PDF page images paired with extracted text. We use it for the supervised stage and for in-distribution evaluation.
- **DocBank** (Li et al., 2020) contains 500,000 pages from arXiv, where each token carries a semantic label and a box. We use its test set for zero-shot academic recognition. Full-page transcription evaluation also requires a defined reading order and text serialization.
- **IIT-CDIP** (Lewis et al., 2006) contains millions of scanned documents made public through tobacco-industry litigation. We use it to test robustness to scanning noise, scored by CER and NED. Reference verification is described below.
- **DocVQA** (Mathew et al., 2021) poses questions about individual document images. Answering can require reading text and interpreting its layout. We report ANLS.
- **InfographicVQA** (Mathew et al., 2022) targets infographics that mix dense text with charts and icons. It probes reading under varied and non-linear layouts.
- **OmniDocBench** (Ouyang et al., 2024) evaluates parsing across diverse PDFs with rich annotations. Table 2 reports Text Edit Distance.
- **OCRBench v2** (Fu et al., 2024) measures how well a model locates text and reasons over it across many subtasks. It reports separate results for its English and Chinese subsets.
- **CC-OCR** (Yang et al., 2024) spans multi-scene reading, multilingual reading, parsing, and key information extraction. It uses task-specific metrics. Our cross-track summary is described below.

Data audit. We fix the dataset revisions, split manifests, reference files and evaluation settings. We record both question and image counts for VQA tasks and distinguish public corpus sizes from the examples actually evaluated. We keep target-benchmark data out of training and model selection and

¹<https://github.com/Ucas-HaoranWei/Vary-tiny-600k>

Table 6: Full zero-shot trajectories on DocBank, IIT-CDIP, DocVQA, and InfographicVQA. Each cell shows the change over the previous row, and SP-DR-1 is measured against SFT-2. Improvements are in red and regressions in blue. AVG is the mean of the two NED scores and the two ANLS scores on a 0 to 100 scale. SP-DR-3 is the model reported in the main text.

Models	Method	DocBank		IIT-CDIP		VQA (ANLS $\uparrow$)		AVG $\uparrow$
		CER(%) $\downarrow$	NED $\uparrow$	CER(%) $\downarrow$	NED $\uparrow$	DocVQA	InfoVQA	
InternVL 3.5-4B	Base	13.44	0.812	19.56	0.735	72.79	50.52	69.50
	SFT-1	12.40(-1.04)	0.835(+0.023)	18.39(-1.17)	0.762(+0.027)	72.71(-0.08)	49.20(-1.32)	70.40(+0.90)
	SFT-2	11.60(-0.80)	0.848(+0.013)	17.83(-0.56)	0.778(+0.016)	73.01(+0.30)	48.15(-1.05)	70.94(+0.54)
	SFT-3	12.17(+0.57)	0.829(-0.019)	18.44(+0.61)	0.762(-0.016)	72.52(-0.49)	48.24(+0.09)	69.97(-0.97)
	SP-DR-1	10.55(-1.05)	0.866(+0.018)	16.70(-1.13)	0.802(+0.024)	73.48(+0.47)	48.40(+0.25)	72.17(+1.23)
	SP-DR-2	10.08(-0.47)	0.876(+0.010)	15.96(-0.74)	0.813(+0.011)	73.92(+0.44)	48.22(-0.18)	72.76(+0.59)
	SP-DR-3	9.73(-0.35)	0.884(+0.008)	15.35(-0.61)	0.823(+0.010)	74.27(+0.35)	48.06(-0.16)	73.26(+0.50)
	SP-DR-4	9.58(-0.15)	0.888(+0.004)	15.08(-0.27)	0.827(+0.004)	74.41(+0.14)	48.02(-0.04)	73.48(+0.23)
	SP-DR-5	9.55(-0.03)	0.887(-0.001)	15.06(-0.02)	0.826(-0.001)	74.39(-0.02)	47.98(-0.04)	73.42(-0.06)
Qwen3 VL-4B	Base	12.82	0.834	18.45	0.756	78.54	52.21	72.44
	SFT-1	11.58(-1.24)	0.857(+0.023)	17.34(-1.11)	0.788(+0.032)	80.05(+1.51)	54.12(+1.91)	74.67(+2.23)
	SFT-2	10.95(-0.63)	0.868(+0.011)	16.71(-0.63)	0.802(+0.014)	81.22(+1.17)	55.34(+1.22)	75.89(+1.22)
	SFT-3	11.42(+0.47)	0.852(-0.016)	16.25(-0.46)	0.803(+0.001)	80.45(-0.77)	54.52(-0.82)	75.12(-0.77)
	SP-DR-1	10.02(-0.93)	0.8798(+0.0118)	15.92(-0.79)	0.814(+0.012)	83.08(+1.86)	56.42(+1.08)	77.22(+1.33)
	SP-DR-2	9.44(-0.58)	0.886(+0.0062)	15.04(-0.88)	0.829(+0.015)	83.61(+0.53)	57.65(+1.23)	78.19(+0.97)
	SP-DR-3	9.05(-0.39)	0.898(+0.012)	14.42(-0.62)	0.847(+0.018)	84.89(+1.28)	58.94(+1.29)	79.58(+1.39)
	SP-DR-4	8.95(-0.10)	0.904(+0.006)	14.01(-0.41)	0.845(-0.002)	84.83(-0.06)	59.51(+0.57)	79.81(+0.23)
	SP-DR-5	8.98(+0.03)	0.900(-0.004)	14.00(-0.01)	0.845(0.000)	84.85(+0.02)	59.88(+0.37)	79.81(0.00)

use no task demonstrations at inference. Development and final evaluation pages are separated from training at the source-document level wherever provenance permits.

Reference verification. We verify the provenance and accuracy of the IIT-CDIP evaluation transcriptions before using them as references for CER and NED.

Benchmark scoring. For DocVQA and InfographicVQA, we compute ANLS by taking the best thresholded edit similarity over each question’s allowed answers, then averaging over questions on a 0 to 100 scale. The scorer lowercases answers and requires normalized edit distance below 0.5. For OmniDocBench, OCRBench v2 and CC-OCR, we fix the dataset and evaluator revisions and retain their task-specific processing. We use Text Edit Distance for OmniDocBench, an equal-language mean for OCRBench v2, and an equal-track mean for CC-OCR, with the underlying language and track scores retained.

E. Full Iteration Trajectories

Tables 5 and 6 give the supervised and self-play trajectories. Improvements are not monotonic across all metrics. For example, Qwen3-VL-4B’s Vary-600K NED changes from 0.988 at SP-DR-2 to 0.987 at SP-DR-3. The main comparison uses the third-round endpoint rather than selecting a different round for each metric.

Aggregate score scope. The averages summarize different sets of metrics. Table 5 combines NED, Token-F1, ROUGE-L and BLEU-4. Tables 6 and 7 combine the two zero-shot NED scores and two ANLS scores. Figure 3(c) uses this four-task mean for the base, SFT-2 and third-round checkpoints. Figure 2 combines Vary-600K NED, DocBank NED and DocVQA ANLS. Each mean uses a 0 to 100 scale. Figure 3(a) uses the equal-weight mean of Vary-600K, DocBank and IIT-CDIP NED, each multiplied by 100, and DocVQA and InfographicVQA ANLS. This five-task mean includes in-distribution performance

Table 7: Modality-specific analysis on Qwen3-VL-4B. Full tuning updates the backbone and the OCR module, while the modular design freezes the backbone. The Vary-600K column measures in-distribution reading, and AVG averages the four zero-shot scores on a 0 to 100 scale. Each cell shows the change over the previous row, with improvements in red and regressions in blue.

Config	Method	Vary CER(%)↓	DocBank NED↑	IIT-CDIP NED↑	DocVQA↑	InfoVQA↑	AVG↑
Full tuning	SFT-2	2.25	0.872	0.805	81.00	55.00	75.92
	Iter-1	1.55(-0.70)	0.878(+0.006)	0.812(+0.007)	81.50(+0.50)	55.30(+0.30)	76.45(+0.53)
	Iter-2	1.18(-0.37)	0.871(-0.007)	0.804(-0.008)	81.10(-0.40)	54.60(-0.70)	75.80(-0.65)
	Iter-3	0.98(-0.20)	0.862(-0.009)	0.795(-0.009)	80.41(-0.69)	53.80(-0.80)	74.98(-0.82)
	Iter-4	0.95(-0.03)	0.851(-0.011)	0.786(-0.009)	79.80(-0.61)	53.10(-0.70)	74.15(-0.83)
	Iter-5	0.94(-0.01)	0.843(-0.008)	0.779(-0.007)	79.50(-0.30)	52.70(-0.40)	73.60(-0.55)
Modular (ours)	SFT-2	2.40	0.868	0.802	81.22	55.34	75.89
	Iter-1	1.82(-0.58)	0.8798(+0.0118)	0.814(+0.012)	83.08(+1.86)	56.42(+1.08)	77.22(+1.33)
	Iter-2	1.45(-0.37)	0.886(+0.0062)	0.829(+0.015)	83.61(+0.53)	57.65(+1.23)	78.19(+0.97)
	Iter-3	1.11(-0.34)	0.898(+0.012)	0.847(+0.018)	84.89(+1.28)	58.94(+1.29)	79.58(+1.39)
	Iter-4	1.09(-0.02)	0.904(+0.006)	0.845(-0.002)	84.83(-0.06)	59.51(+0.57)	79.81(+0.23)
	Iter-5	1.06(-0.03)	0.900(-0.004)	0.845(0.000)	84.85(+0.02)	59.88(+0.37)	79.81(0.00)

on Vary-600K, whereas panel (c) includes only the four zero-shot tasks. The two averages therefore measure different evaluation scopes and should not be compared directly.

Reported precision. AVG values and their changes are displayed to two decimal places. A change may differ by 0.01 from subtracting the displayed averages because of rounding.

F. Modality-Specific Analysis

Table 7 compares full-tuning and modular trajectories. Across the reported iterations, modular tuning improves the four-task zero-shot AVG from 75.89 to 79.58, while full tuning changes from 75.92 to 74.98. The two trajectories show the advantage of the modular configuration on these evaluated tasks. The aggregate scores use the metric set specified in Appendix E.

G. Case Study

The selected Vary-600K cases compare supervised readings with third-round InternVL-3.5-4B outputs. Tables 8 and 9 show the supplied references and corresponding model readings. Red shading marks differences from the displayed references, including punctuation and spelling. These examples show corrected words and remaining errors in the displayed readings.

Table 8: Vary-600K Example 1, a Quicksort page, with the supplied reference text and reported readings. The reference does not cover the full page shown in the thumbnail. Red shading marks differences from the displayed reference without changing the readings.

 Vary 600K Example 1	Ground Truth:"<lb><lb>Our recursive calls to Quicksort use this index (splitPoint) to reduce the size of the problem in the general case.<lb>Quicksort(first, splitPoint - 1) sorts the left "half" of the array. Quicksort(splitPoint + 1, last) sorts the right "half" of the array. (The "halves" are not necessarily the same size.) splitVal is already in its correct position in data[splitPoint].<lb>What is the base case? When the segment being examined has only one item, we do not need to go on. That is represented in the algorithm by the missing else statement. If there is only one value in the segment being sorted, it is already in its place.<lb>We must find a way to get all elements that are equal to or less than splitVal on one side of splitVal and all elements that are greater than splitVal on the other side. We do this by moving a pair of the indexes from the ends toward the middle of the array, looking for items that are on the wrong side of the split value. When we find pairs that are on the wrong side, we swap them and continue working our way into the middle of the array.<lb> <pre>\begin{tabular}{ c } \hline \\ \end{tabular}<lb>"</pre>
SFT-2	Document Reading:"<lb><lb>Our recurslve calls to Quicksort use this index (splitPoint) to reduce the size of the problem in the genral case.<lb> Quicksort(first, splitPoint - 1) sorts the left "half" of the array. Quicksort (splitPoint + 1, last) sorts the right "half" of the arry . (The "halves" are not necessarily the same size.) splitval is already in its correct position in data[splitPoint].<lb> What is the base case? When the segment being examned has only one item, we do not need to go on. That is represented in the algorithm by the missing else statement. If there is only one value in the segment being sorted, it is already in its place.<lb> We must find a way to get all elements that are equal to or less than splitVal on one side of splitVal and all elements that are greater than splitVal on the other side. we do this by moving a pair of the indices from the ends toward the middle of the array, looking for items that are on the wrong side of the split value. When we find pairs that are on the wrong side, we swap them and continue working our way into the middle of the array.<lb> <pre>\begin{tabular}{ c } \hline \\ \end{tabular}<lb> "</pre>
SP-DocReader iteration 3	Document Reading:"<lb><lb>Our recursive calls to Quicksort use this index (splitPoint) to reduce the size of the problem in the general case.<lb> Quicksort(first, splitPoint - 1) sorts the left "half" of the array. Quicksort (splitPoint + 1, last) sorts the right "half" of the array. (The "halves" are not necessarily the same size.) splitVal is already in its correct position in data[splitPoint].<lb> What is the base case? When the segment being examined has only one item, we do not need to go on. That is represented in the algorithm by the missing else statement. If there is only one value in the segment being sorted, it is already in its place.<lb> We must find a way to get all elements that are equal to or less than splitVal on one side of splitVal and all elements that are greater than splitVal on the other side. We do this by moving a pair of the indexes from the ends toward the middle of the array, looking for items that are on the wrong side of the split value. When we find pairs that are on the wrong side, we swap them and continue working our way into the middle of the array.<lb> <pre>\begin{tabular}{ c } \hline \\ \end{tabular}<lb> "</pre>

[illegible]

Table 9 (continued)

SFT-2	Document Reading:"<lb>Summa Theologica Saint Thomas Aquinas<lb> will reflect honor on the whole state: as when he brings to affect what the whole state is striving for.<lb> Reply to Objection 4: The chief act of virtue is the inward choice, and a virtue may have this without outward fortune: so that even a poor man may be magnificent. But goods of fortune are requisite as instruments to the external acts of virtue: and in this way a poor man cannot accomplish the outward act of magnificence in things that are great simply. Perhaps, however, he may be able to do so in things that are great by comparison to some particular work; which, though little in itself, can nevertheless be done magnificently in proportion to its genus: for little and great are relative terms, as the Philosopher says (De Predic . Cap. Ad aliqid .).<lb> Whether magnificence is a part of fortitude?<lb> Objection 1: It seems that magnificence is not a part of fortitude. For magnificence agrees in matter with liberality, as stated above (A[3]). But liberality is a part, not of fortitude, but of justice. Therefore magnificence is not a part of fortitude.<lb> Objection 2: Further, fortitude is about fear and daring . But magnificence seems to have nothing to do with fear, but only with expenditure, which is a kind of action. Therefore magnificence seems to pertain to justice, which is about actions, rather than to fortitude.<lb> Objection 3: Further, the Philosopher says (Ethic. iv, 2) that "the magnificent man is like the man of science." Now science has more in common with prudence than with fortitude. Therefore magnificence should not be reckoned a part of fortitude.<lb> On the contrary, Tully (De Invent. Rhet. ii) and Marcrobius (De Somn. Scp . i) and Andronicus reckon magnificence to be a part of fortitude.<lb> I answer that, Magnificence, in so far as it is a special virtue, cannot be reckond a subjective part of fortitude, since it does not agree with this virtue in the point of matter: but it is reckoned a part thereof, as being anexed to it as secondary to principal virtue.<lb> In order for a virtue to be annexed to a principal virtue, two things are necessary, as stated above (Q[80]). The one is that the secondary virtue agree with the principal, and the other is that in some respect it be exceeded thereby. Now magnificence agrees with fortitude in the point that as fortitude tends to something arduous and difficult, so also does magnificence: wherefore seemingly it is seated, like fortitude, in the irascible . Yet magnificence falls short of fortitude, in that the arduous thing to which fortitude tends derives its difficulty from a danger that threatens the person, whereas the arduous thing to which magnificence tends, derives its difficulty from the dispossession of one's property, which is of much less acount than danger to one's person. Wherefore magnificence is accounted a part of fortitude.<lb> Reply to Objection 1: Justice regards operations in themselves, as viewed under the aspect of something due: but liberality and magnificence regard sumptary operations as related to the passions of the soul, albit in different ways. For liberality regards expenditure in reference to the love and desire of money, which are passions of the concupiscble faculty, and do not hinder the<lb> 258I <lb>"
-------	--

Table 9 (continued)

SP-DocReader iteration 3	Document Reading:"<lb>Summa Theologica Saint Thomas Aquinas<lb> will reflect honor on the whole state: as when he brings to effect what the whole state is striving for.<lb> Reply to Objection 4: The chief act of virtue is the inward choice, and a virtue may have this without outward fortune: so that even a poor man may be magnificent. But goods of fortune are requisite as instruments to the external acts of virtue: and in this way a poor man cannot accomplish the outward act of magnificence in things that are great simply. Perhaps, however, he may be able to do so in things that are great by comparison to some particular work; which, though little in itself, can nevertheless be done magnificently in proportion to its genus: for little and great are relative terms, as the Philosopher says (De Praedlc . Cap. Ad aliquid .).<lb> Whether magnificence is a part of fortitude?<lb> Objection 1: It seems that magnificence is not a part of fortitude. For magnificence agrees in matter with liberality, as stated above (A[3]). But liberality is a part, not of fortitude, but of justice. Therefore magnificence is not a part of fortitude.<lb> Objection 2: Further, fortitude is about fear and daring . But magnificence seems to have nothing to do with fear, but only with expenditure, which is a kind of action. Therefore magnificence seems to pertain to justice, which is about actions, rather than to fortitude.<lb> Objection 3: Further, the Philosopher says (Ethic. iv, 2) that "the magnificent man is like the man of science." Now science has more in common with prudence than with fortitude. Therefore magnificence should not be reckoned a part of fortitude.<lb> On the contrary, Tully (De Invent. Rhet. ii) and Macrobius (De Somn. Scip. 1) and Andronicus reckon magnificence to be a part of fortitude.<lb> I answer that, Magnificence, in so far as it is a special virtue, cannot be reckoned a subjective part of fortitude, since it does not agree with this virtue in the point of matter: but it is reckoned a part thereof, as being annexed to it as secondary to principal virtue.<lb> In order for a virtue to be annexed to a principal virtue, two things are necessary, as stated above (Q[80]). The one is that the secondary virtue agree with the principal, and the other is that in some respect it be exceeded thereby. Now magnificence agrees with fortitude in the point that as fortitude tends to something arduous and difficult, so also does magnificence: wherefore seemingly it is seated, like fortitude, in the irascible. Yet magnificence falls short of fortitude, in that the arduous thing to which fortitude tends derives its difficulty from a danger that threatens the person, whereas the arduous thing to which magnificence tends, derives its difficulty from the dispossession of one's property, which is of much less account than danger to one's person. Wherefore magnificence is accounted a part of fortitude.<lb> Reply to Objection 1: Justice regards operations in themselves, as viewed under the aspect of something due: but liberality and magnificence regard sumptuary operations as related to the passions of the soul, albet in different ways. For liberality regards expenditure in reference to the love and desire of money, which are passions of the concupiscible faculty, and do not hinder the<lb> 2581<lb>"
--------------------------	--