DocPO: Advancing Document Policy Optimization via Tailored Step-Aware Rewards

Anonymous ACL submission

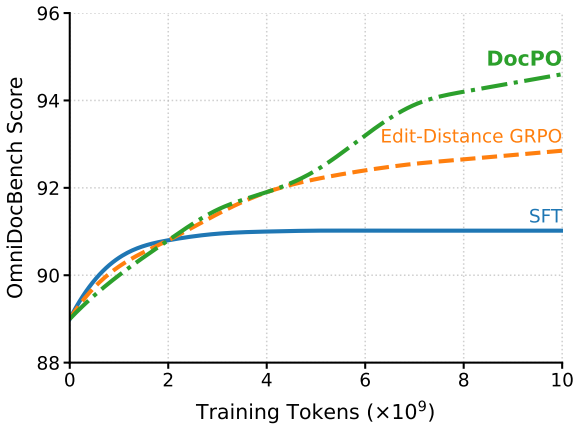

Figure 1: We propose *DocPO*, which leverages domain-specific preference optimization to surpass SFT and Edit-Distance GRPO on OmniDocBench.

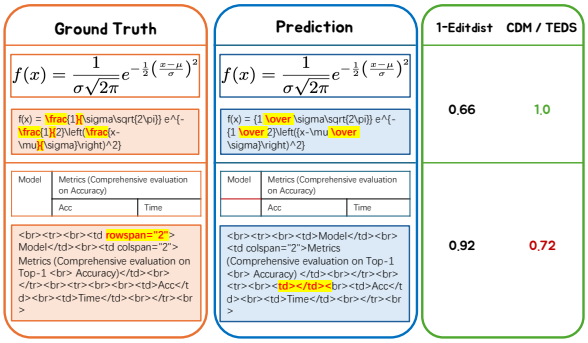

Figure 2: **Limitations of Edit Distance as a reward.** *False Negative (Top)*: Visually correct formulas are penalized for synonymous LaTeX variations. *False Positive (Bottom)*: Broken tables are rewarded for high lexical overlap. This misalignment motivates the need for a structure-aware objective.

Abstract

Despite its success in Large Language Models, Reinforcement Learning (RL) remains underutilized in document because generic rewards fail to effectively evaluate various document elements such as formulas and tables. Existing metrics (e.g., Edit Distance) often fail to capture the semantic validity of structures like LaTeX formulas or nested tables, while training dedicated Reward Models requires expensive human annotation. To bridge this gap, we introduce **DocPO**, a novel policy optimization framework featuring **Tailored Step-Aware Rewards**. Unlike generic approaches, DocPO constructs domain-specific reward functions without preference data: it integrates LLM-based semantic verification with syntactic constraints for formulas, utilizes structure-weighted TEDS for tables, and employs continuous distance metrics for text to mitigate sparsity. Additionally, we propose Step-Aware Annealing to dynamically modulate reward discriminability for distinguishing hard samples. Experiments show DocPO boosts parsing precision across elements, establishing an annotation-free paradigm for document understanding.

1 Introduction

Document parsing serves as a foundational task in both Vision Language Models (VLMs) and Document AI (Cui et al., 2021). Its core objective is to precisely decode heterogeneous elements arranged in 2D document layouts, such as text, formula, and table, into sequential 1D sequences. Crucially, this task transcends vanilla Optical Character Recognition (OCR) by necessitating a deep understanding of element-specific characteristics. Specifically, accurately resolving details such as *varied formula expressions* and *tables with nested cells* places stringent demands on models' fine-grained perception and sequence generation capabilities.

While VLMs have made remarkable progress in this field, current mainstream approaches still predominantly rely on the Supervised Fine-Tuning (SFT) paradigm (Kim et al., 2021; Blecher et al., 2023). Although SFT has achieved significant success driven by massive training data, it is inherently constrained by the Teacher Forcing training mode, which inevitably leads to Exposure Bias and overfitting (Zhang et al., 2025). In response to

Model	Methods	Metric & Strategy	Reward Type	Granularity
Monkey v1.5	RM	BT Model	Continuous	Table
INFINITY Parser	RLVR	Edit Distance	Continuous	Page
HunyuanOCR	RLVR	Edit Distance	Continuous	Page
olm-OCR2	RLVR	Ensemble	Binary	Page
DocPO (ours)	RLVR	Task-specific	Continuous	Text / Formula / Table

Table 1: Comparison of existing paradigms and our method. This motivates our design of a structure-respecting reward mechanism without costly human annotation.

these challenges, researchers have begun exploring Reinforcement Learning (RL), seeking improvements via sequence-level optimization. Yet, in stark contrast to its success in standard Large Language Models (LLMs) (Hurst et al., 2024; Kumar et al., 2024; Guo et al., 2025), RL's gains in OCR tasks remain relatively scarce. In our view, this discrepancy stems primarily from the misalignment between generic RL methods and the intrinsic characteristics of document elements.

Existing RL methods applied to document parsing (Zhang et al., 2025; Poznanski et al., 2025; Team et al., 2025; Wang et al., 2025) can be categorized into two main streams, both of which face specific limitations. The first stream relies on edit distance (Team et al., 2025; Wang et al., 2025), using character-level metrics as rewards. However, this rigid character-level matching often misaligns with actual parsing quality. As detailed in Figure 2, this metric suffers from a dual failure: it generates false negatives for formulas by penalizing valid synonymous LaTeX variations (Top), while simultaneously yielding false positives for tables by rewarding high lexical overlap despite structural corruption (Bottom). This creates a severe gap between the reward signal and human visual perception. The second stream involves Human-annotated Reward Models (Zhang et al., 2025). However, constructing high-quality preference datasets is prohibitively expensive and hard to scale. Therefore, the pivotal question becomes: *How can we design an efficient reward that aligns with the rendered visual quality without relying on expensive annotation?*

We introduce **DocPO**, a domain-aware reward framework. As demonstrated in Figure 1, it delivers plug-and-play improvements for all document elements. Distinct from existing approaches, we **formulate our reward based on the intrinsic properties of document elements**, directly mapping domain priors to continuous feedback signals. We summarize the characteristics and limitations of these approaches in Table 1. Specifically, we design tailored rewards for heterogeneous elements: an LLM-based evaluative reward for formulas (Section 3.1), a structure-weighted TEDS for tables (Section 3.2), and a continuous reward for text (Section 3.3). Crucially, our results demonstrate that DocPO surpasses generic rewards like NED, offering a scalable solution for document scenarios without the need for large-scale preference data.

We list our main contributions as follows:

- **Preference-Free RL**: We introduce a plug-and-play, domain-aware reward framework that operates independently of human preference data. By leveraging intrinsic rule-based rewards, our method enables scalable performance without preference annotation.
- **Tailored Reward Formulation**: We design fine-grained rewards for heterogeneous subtasks, such as tables and formulas. These mechanisms bridge the gap between rigid character-level metrics and visual perception.
- **Fine-Grained Element Benchmark**: We construct a new benchmark for fine-grained element parsing that addresses existing limitations on complex structures, enabling a holistic assessment of capabilities.

2 Related Work

2.1 Document Parsing Paradigms

Existing methodologies in document parsing can be broadly categorized into three paradigms: Pipelines, End-to-End VLMs, and Modular VLMs.

Pipelines: Traditional pipeline approaches, such as PaddleOCR (Cui et al., 2025), MinerU (Wang et al., 2024), and MonkeyOCR (Li et al., 2025), typically involve a sequential workflow: detecting layout regions, extracting content via OCR, and linearizing the results based on spatial layout (Wang

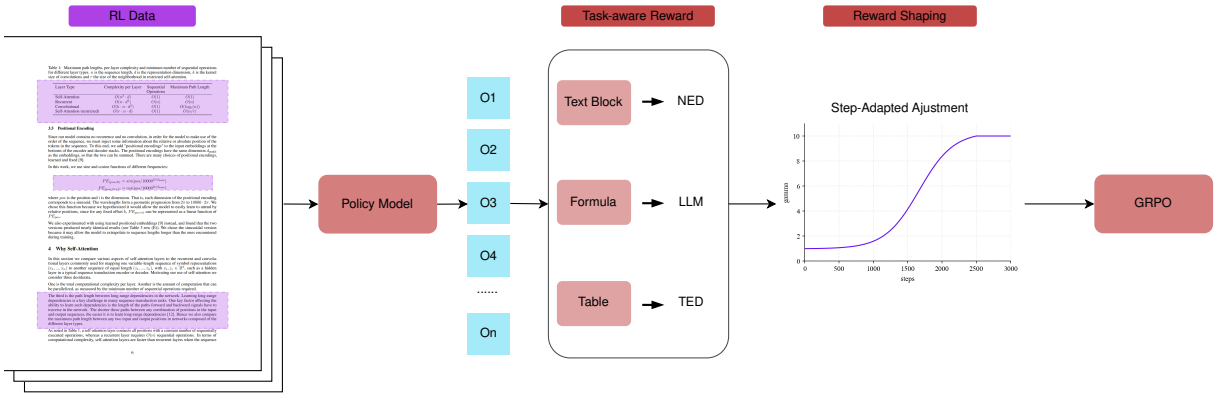

Figure 3: Pipeline of DocPO.

et al., 2021; Ha et al., 1995). The primary advantages of this paradigm are its efficiency and flexibility, often achieving the best trade-off between performance and precision in specific scenarios.

End-to-End VLMs: In contrast, End-to-End VLMs employ a single model to directly transcribe all document content into a linear sequence, as exemplified by Nougat (Blecher et al., 2023), Kosmos-2.5 (Lv et al., 2023), Qwen-VL (Bai et al., 2025), olmOCR (Poznanski et al., 2025), DeepseekOCR (Wei et al., 2025), and HunyuanOCR (Team et al., 2025). Characterized by its simplicity, this approach requires only one single module to complete the task. It demonstrates superior adaptability across diverse scenarios, achieving robust generalization (e.g., camera-captured images). However, due to the inherent limitations of autoregressive architectures, End-to-End VLMs often struggle with inference latency when processing long documents or dense text.

Modular VLMs: To address the limitations of the aforementioned paradigms, recent works like Dolphin (Feng et al., 2025) and MinerU2.5 (Niu et al., 2025) have introduced the Modular paradigm. This approach utilizes distinct functional modules within a single model framework, typically adopting a *crop-then-parse* strategy. By enabling element-level parallel decoding, it achieves a balance between usability and efficiency.

2.2 Reinforcement Learning in Document

Despite the architectural diversity of the aforementioned paradigms, their training predominantly relies on SFT. Consequently, RL for document parsing remains in its nascent stages. We compare the few existing methodologies in Table 1, with specific limitations detailed below.

Learned Reward Models: Targeting complex table recognition, MonkeyOCR v1.5 (Zhang et al., 2025) introduces a solution that eliminates the reliance on human-annotated HTML ground truth. It adopts a *render-and-compare* strategy, where generated HTML is rendered into an image and evaluated against the original document via a specifically trained Reward Model (RM). While this mechanism prioritizes structural visual consistency over absolute character matching, it depends heavily on data-intensive RM training and is narrowly confined to table optimization, overlooking other critical modalities such as text and formulas.

Edit Distance Rewards: Within End-to-End VLMs, HunyuanOCR (Team et al., 2025) integrates the RLVR framework with the GRPO algorithm to circumvent the overhead of a large Critic model. Its reward design relies on Normalized Edit Distance (NED), enforced by strict formatting constraints where syntactic errors yield zero reward. However, as noted in the introduction, a significant discrepancy persists between edit distance and human visual perception. For instance, a missing LaTeX symbol may cause negligible edit distance but severe rendering failure.

Binary Rule-Based Rewards: Diverging from continuous metrics, olm-OCR2 (Poznanski et al., 2025) employs binary unit tests as rewards. By leveraging massive synthetic data combined with a sparse binary reward signal, it has demonstrated the efficacy of this method on Qwen2.5-VL-7b. However, its robustness and generalizability to complex, real-world distributions remain to be verified.

3 Reward Design

As illustrated in Figure 3, we first detail tailored rewards for formula, table, and text, followed by the step-aware annealing to enhance stability.

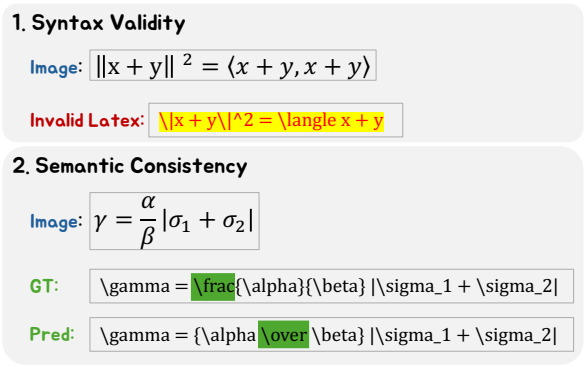

Figure 4: Examples of challenges in evaluating LaTeX generation. The *top case* demonstrates a **Syntax Validity** failure where the generated sequence is incomplete. The *bottom case* highlights the need for **Semantic Consistency**, where the model predicts a valid alternative syntax (`\over`) that differs lexically from the ground truth (`\frac`) but remains mathematically correct.

3.1 Reward Design for Formula Recognition

As illustrated in Figure 4, LaTeX formula recognition faces a dual challenge regarding representation diversity and generation validity. While standard metrics struggle with the one-to-many mapping of equivalent strings, RL exploration is prone to producing syntactically invalid sequences. To resolve this, we propose a hierarchical reward mechanism that prioritizes syntax validity before jointly evaluating semantic and structural precision.

Syntax Validity Gating: We first implement a hard syntax filter to eliminate invalid noise. Let y be the generated formula. We define a validity indicator $v_{\text{syn}} \in \{0, 1\}$, where $v_{\text{syn}} = 1$ if y can be successfully compiled by a LaTeX engine, and 0 otherwise. Invalid samples are immediately assigned a zero reward, ensuring the model focuses optimization solely on executable expressions.

Semantic and Structural Reward: For syntactically valid formulas, we compute a hybrid score:

- **Semantic Consistency** (r_{sem}): To handle format variations (e.g., `\frac{\alpha}{\beta}` vs. `\alpha\over\beta`), we employ an LLM as a semantic judge. The LLM performs a binary classification to determine if the rendered image of y is **mathematically equivalent** to the ground truth y^*, outputting $r_{\text{sem}} \in \{0, 1\}$.
- **Structural Consistency** (r_{struct}): We complement the semantic score with Normalized Edit Distance (NED) to capture subtle typos overlooked by LLMs, defining $r_{\text{struct}} = 1 - \text{NED}(y, y^*)$ to provide dense gradients.

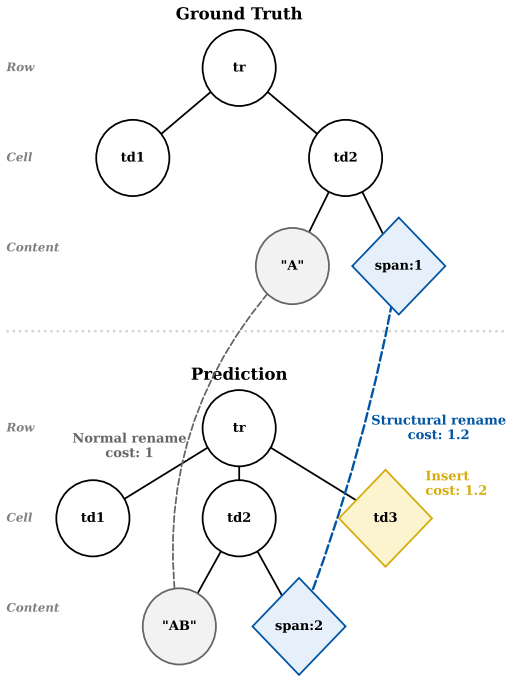

Figure 5: Illustration of the weighted tree edit distance-based similarity calculation. Distinct costs are assigned to operations like structural renaming and insertion to jointly evaluate structural and content consistency.

Unified Reward Formulation: The final reward R_{formula} is formulated as a gated weighted sum:

$$R_{\text{formula}} = v_{\text{syn}} \cdot (\alpha \cdot r_{\text{sem}} + \beta \cdot r_{\text{struct}}) \quad (1)$$

where α and β are hyperparameters balancing semantic correctness and character-level accuracy. Empirically, we set $\alpha > \beta$ (e.g., $\alpha = 0.8, \beta = 0.2$) to prioritize mathematical meaning while maintaining structural supervision.

3.2 Reward Design for Table Recognition

For table recognition, we leverage the *Tree Edit Distance-based Similarity* (TEDS) based on the APTED algorithm (Pawlik and Augsten, 2016, 2015) to evaluate the topological similarity of HTML trees.

In contrast to standard TEDS, which typically assigns uniform costs to all edit operations, we introduce a weighted cost scheme to enforce structural integrity, as illustrated in Figure 5. Specifically, we assign a higher penalty (e.g., 1.2) to structural discrepancies, such as span attribute modifications and node insertions, relative to simple content errors (1.0). This weighting strategy incentivizes the model to prioritize topological accuracy, ensuring that layout correctness is not compromised for the sake of minor textual matching.

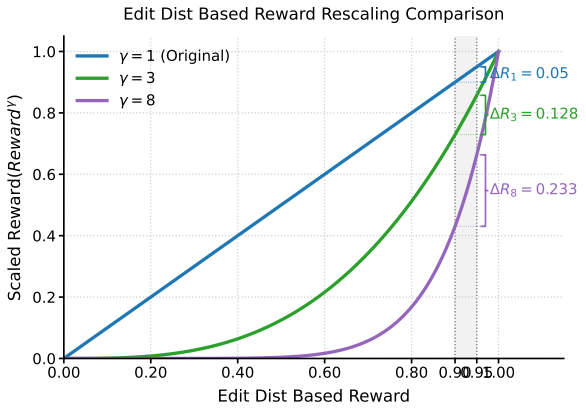

Figure 6: Comparison of reward rescaling functions with curvature factors (γ). Non-linear scaling (e.g., $\gamma = 8$) amplifies the reward difference (ΔR) in the high-score region, providing sharper optimization signals for fine-grained improvements.

We define the reward R_{table} as the complement of the weighted tree edit distance-based similarity:

$$R_{\text{table}} = \text{TEDS}_{\text{weighted}}(y, y^*) \tag{2}$$

where y is the predicted HTML sequence, y^* is the ground truth, and $\text{TEDS}_{\text{weighted}} \in [0, 1]$ is the normalized tree edit distance-based similarity using adjusted weights.

3.3 Reward Design for Text Recognition

For text recognition tasks, we adopt the *Normalized Edit Distance* (NED), defined as a lightweight metric in the range $[0, 1]$ where smaller values indicate higher similarity to the ground truth.

This continuous nature offers two advantages. First, it provides fine-grained gradient guidance, enabling the model to distinguish between minor typos (e.g., 0.1) and severe hallucinations (e.g., 0.9), preventing optimization stagnation. Second, it enhances noise robustness. Unlike binary labels, NED mitigates the impact of incorrect labels through soft error quantification.

Formally, to align with the objective of reward maximization, we define the text recognition reward R_{text} as the complement of the distance:

$$R_{\text{text}} = 1 - \text{NED}(y, y^*) \tag{3}$$

where y is the predicted text, y^* is the ground truth, and $\text{NED} \in [0, 1]$ denotes the normalized edit distance, ensuring rewards scale with similarity.

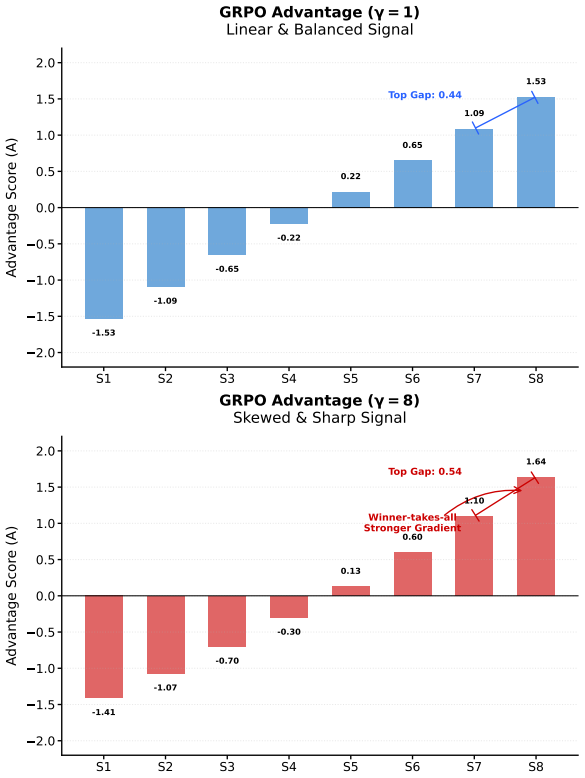

Figure 7: Comparison of advantage distributions for $\gamma = 1$ (linear) and $\gamma = 8$ (skewed). The higher γ value sharpens the signal, creating a "winner-takes-all" effect that amplifies gradients for top-performing samples.

3.4 Step-Aware Annealing

To address the challenge of distinguishing hard samples during training, we propose a Step-Aware Annealing Mechanism. This mechanism dynamically modulates the reward disparity between samples as training progresses. In this section, we detail the non-linear reward shaping and the adaptive scheduling of γ.

Non-Linear Reward Shaping: At the core of our method lies a dynamic scaling reward function, which adjusts the magnitude of sample differentiation. We define the reward as a power-law transformation of the base metric (see Figure 6):

$$\text{Reward}(\tau) = M^{\gamma} \tag{4}$$

where $M \in \{R_{\text{formula}}, R_{\text{table}}, R_{\text{text}}\}$ represents the base reward. γ is the curvature factor; a higher γ amplifies the reward difference in high-score regions, providing sharper optimization signals.

Crucially, γ represents the dynamic scaling factor, which controls the amplification of reward disparity at different training steps. Its value is adaptively updated throughout the training process.

Adaptive Scheduling of γ: To balance training stability with the sharpened discrimination power illustrated in Figure 7, γ is not fixed but adaptively updated based on runtime statistics. The update rule is defined as:

$$\gamma = \gamma_{\text{init}} + \Delta_\gamma \cdot \left[1 - \exp\left(-\frac{s}{\tau_{\text{adaptive}}}\right)\right] \quad (5)$$

Here, γ_{init} sets the baseline amplification, Δ_γ controls the maximum adjustment range, and s denotes the current training step.

To ensure training stability, the update rate is modulated by an adaptive time scale τ_{adaptive}, which incorporates the Coefficient of Variation (CV) to prevent reward oscillation:

$$\tau_{\text{adaptive}} = \begin{cases} \tau, & \text{if } s < s_{window} \\ \dfrac{\tau}{1 + \text{CV}_{window}}, & \text{if } s \geq s_{window} \end{cases} \quad (6)$$

where s_{window} is the backtracking window size (set to 3). The term $\text{CV}_{window} = \frac{\sigma}{\mu + \epsilon}$ measures the dispersion of reward values within the window.

This piecewise design ensures robustness: in the early phase ($s < s_{window}$), a fixed τ prevents abnormal scaling due to limited data; in later phases, τ is normalized by reward volatility, ensuring smooth transitions even when performance fluctuates.

4 Comparative Analysis

We first elaborate on the *RL Data Strategy* for high-value sample filtering, followed by the detailed *Experimental Settings*. Subsequently, we report the *Main Results* on both public and self-constructed benchmarks. We conclude with an *Ablation Study* to validate the effectiveness of key components, such as task-specific and step-aware rewards.

4.1 RL Data Strategy

During the RL training phase, sample efficiency is critical. Samples typically exhibit varying degrees of learning utility: fully correct samples (which have already achieved optimality) provide negligible gradient signals, whereas fully incorrect samples may initially seem useless but can gain value if they show signs of improvement. To maximize training efficiency, we introduce a dynamic data filtering mechanism that retains only samples with demonstrable *improvement potential*.

Specifically, we track the performance history of each sample i and retain it only if the variance in its score indicates learning progress. The filtering criterion is formally defined as:

$$\mathcal{D} = \left\{ i \mid \max_t \text{Score}_i(t) - \min_t \text{Score}_i(t) > 0 \right\} \quad (7)$$

where t represents the training step. This strategy is grounded in the following rationale:

- **Fully correct samples** are discarded because their scores remain constant (max equals min), offering no new optimization information.
- **Improvable hard samples** (initially incorrect) are retained if their scores rise during training (e.g., from 0.1 to 0.3), as this positive difference signals that the model is beginning to learn the pattern.
- **Partially correct samples** inherently exhibit score variance and are retained to provide fine-grained gradient guidance.

By filtering for positive score differences, this design prioritizes 'hard but learnable' samples. It ensures capacity is not wasted on trivial cases or intractable noise, establishing a data-efficient paradigm for fine-grained document parsing tasks.

4.2 Experimental Settings

In this subsection, we detail the experimental configuration, covering the construction of hierarchical datasets for both SFT and RL stages. We also specify the baseline model architecture, training hyperparameters, and the evaluation protocols used to benchmark performance.

Datasets: We constructed two distinct datasets to enhance parsing performance across different granularities. For the initial SFT stage, we utilize 490k full-page document samples with coarse-grained Mathpix annotations. For the subsequent RL fine-grained optimization, we curated a high-precision dataset comprising 600k element patches. This RL set spans three structural categories: (1) *RL-Tables* (206k samples), combining 86k manually annotated high-quality entries with filtered synthetic data; (2) *RL-Formulas* (196k samples), sourced from open datasets and LaTeX rendering; and (3) *RL-Text Blocks* (210k samples), incorporating open-source data and hard-case examples.

Baseline Model: The base model adopts Qwen2.5-VL-3B, with the SFT parameters set as follows: maximum sequence length of 12k, global batch size of 512, constant learning rate of 3e-5, and training epochs of 1.

Table 2: Performance Comparison on OmniDocBench and DocElemHard

Task Type	Method	OmniDocBench	DocElemHard
Text (Edit-Distance) ↓	Baseline (w/o RL)	0.0358	0.091
	Edit-Dist Reward	0.0238	0.033
	Edit-Dist + Step-Aware Annealing	**0.0125**	**0.022**
Formula (CDM) ↑	Baseline (w/o RL)	92.61	85.58
	Edit-Dist Reward	92.86	85.18
	LLM Semantic Score	93.93	86.30
	LLM Semantic + Step-Aware Annealing	**94.38**	**86.55**
Table (TEDS) ↑	Baseline (w/o RL)	89.30	83.20
	Edit-Dist Reward	92.05	90.01
	APTED Reward	92.11	90.21
	APTED + Step-Aware Annealing	**93.09**	**90.60**

Note: The symbols ↑ and ↓ indicate that higher and lower scores are better, respectively. **Bold** denotes the best performance achieved in each task category.

RL Settings: RL is performed on the baseline model with the following training parameters: input sequence length of 4k, output sequence length of 8k, global batch size of 256, constant learning rate of 1e-6, rollout number of 8, no KL divergence constraint (prioritizing structural alignment performance over policy conservatism), and the training stops when the test set performance stabilizes.

Evaluation: Evaluation is performed on the OmniDocBench dataset (Ouyang et al., 2025), which contains 1,355 pages, and our self-constructed DocElemHard benchmark, comprising a total of 9,400 images. We employ three normalized metrics (ranging from 0 to 1, where higher is better) to assess specific parsing modalities: Normalized Edit Distance (NED) for text, Character Detection Matching (CDM) for formulas, and Tree Edit Distance-based Similarity (TEDS) for tables. To provide a unified performance assessment, we calculate an overall metric defined as:

$$\text{Overall} = \frac{(1 - \text{ED}) \times 100 + \text{TEDS} + \text{CDM}}{3} \tag{8}$$

4.3 Ablation Study

We conduct an ablation study to validate our framework across three dimensions: First, we compare RL against the SFT baseline to demonstrate exploration benefits. Second, we contrast generic Edit Distance with Task-Aware rewards to highlight domain-specific signals. Finally, we show Step-Aware rewards outperform Step-Constant baselines by providing denser supervision.

Impact of RL Training (RL vs. SFT): Table 2 presents a quantitative comparison between the SFT baseline and our RL framework. The results demonstrate that introducing RL consistently boosts performance across all task types, breaking the performance ceiling of supervised learning. As shown in the *Baseline (w/o RL)* rows, the SFT model achieves a text error rate of 0.0358 and a formula score of 92.61 on OmniDocBench. However, simply applying a basic RL method (Edit-Dist Reward) immediately improves these metrics to 0.0238 and 92.86, respectively. A similar trend is observed in the Table task, where the RL-based approach surpasses the baseline by a significant margin (e.g., improving from 89.30 to 92.05 on OmniDocBench). This confirms that the exploration capability of RL effectively refines the model's policy beyond the limits of static supervised data.

Effectiveness of Task-Specific Signals (Edit Distance vs. Task-Aware): We further investigate the impact of reward design by comparing general-purpose metrics (Edit-Dist Reward) with domain-specific signals (Task-Aware Reward). While Edit Distance provides a reasonable proxy for surface-level quality, it lacks the nuance required for complex structures. In the Formula task, the LLM Semantic Score (93.93) outperforms the Edit-Dist Reward (92.86) on OmniDocBench, indicating that semantic-level feedback guides the model better than character-level matching. Similarly, for the Table task, the APTED Reward—which calculates tree-edit distance to capture structural accuracy—achieves superior results compared to the standard edit distance (92.11 vs. 92.05). These findings suggest that aligning the reward function with the intrinsic properties of the data (e.g., mathematical semantics or tabular tree structures) is crucial for high-precision generation.

Table 3: End-to-End Evaluation on OmniDocBench.

Model Type	Model	Post-proc.	Size	OmniDocBench			
				overall↑	**text↓**	**formula↑**	**table↑**
General VLMs	Gemni-2.5-pro	No	-	88.03	0.075	85.92	85.71
	Qwen3-VL-235B	No	235B	89.15	0.069	88.14	86.21
Specialized VLMs (End2End)	Deepseek-OCR	Yes	3B	87.01	0.073	83.37	84.97
	dots.ocr	Yes	3B	88.41	0.048	83.22	86.78
	HunyuanOCR	Yes	1B	**94.10**	0.042	**94.73**	**91.81**
Specialized VLMs (Modular)	MonkeyOCR-pro	Yes	3B	88.85	0.075	87.50	86.78
	MinerU2.5	Yes	1.2B	90.67	0.047	88.46	88.22
	PaddleOCR-VL	Yes	0.9B	92.86	**0.035**	91.22	90.89
	Ours	No	3B	92.13	0.038	88.90	91.31

Table 4: Element Evaluation on DocElemHard and OmniDocBench.

Model Type	Model	Post-proc.	Size	DocElemHard				OmniDocBench			
				overall↑	text↓	formula↑	table↑	overall↑	text↓	formula↑	table↑
General VLMs	Qwen2.5-VL-72B	No	72B	85.03	0.034	78.21	80.27	89.95	0.0424	87.47	86.64
	Qwen2.5-VL-3B	No	3B	81.72	0.096	76.86	77.90	88.05	0.0792	87.27	84.85
Specialized VLMs	dots.ocr	Yes	3B	87.50	0.037	85.39	80.8	89.6	0.034	90.4	81.9
	HunyuanOCR	Yes	1B	90.11	0.024	88.94	83.8	**95.9**	0.0285	**94.8**	**95.7**
	PaddleOCR-VL	Yes	0.9B	91.50	0.033	**90.90**	86.9	94.87	0.0142	94.1	91.95
	Ours	No	3B	**91.65**	**0.022**	86.55	**90.6**	95.38	**0.0125**	94.38	93.01

Superiority of Dense Supervision (Step-Aware vs. Step-Constant): Finally, we analyze the granularity of feedback by comparing Step-Constant rewards (sparse feedback at the end of generation) against our proposed Step-Aware mechanism. The results indicate that dense, step-wise supervision yields the best performance across all categories. For the Text task, the Step-Aware Reward drastically reduces the error rate to **0.0125**, representing a nearly **47% relative reduction** compared to the generic reward (0.0238). In the Table task, combining task-aware metrics with step-aware feedback leads to state-of-the-art results, reaching **93.09** on OmniDocBench and **90.60** on DocElemHard. This validates that providing intermediate feedback during the decoding process helps the model correct errors earlier and converge to a more optimal policy than sparse, sentence-level rewards.

4.4 Evaluation Results on OmniDocBench and DocElemHard

Superior Performance without Post-Processing: Tables 3 and 4 compare our model against leading General and Specialized VLMs. *Crucially, unlike competitors (e.g., HunyuanOCR) relying on complex post-processing, our results derive directly from raw outputs.* Despite this simpler setup, our 3B model demonstrates exceptional efficiency. In Table 3, we achieve an overall score of **92.13**, outperforming massive models like **Qwen3-VL-235B** (89.15) and Gemini-2.5-pro (88.03), validating the effectiveness of our granular reward modeling.

State-of-the-Art on Complex Layouts: The advantages of our approach are most pronounced in fine-grained element evaluation on challenging datasets. As shown in Table 4, our model achieves the lowest text edit distance on OmniDocBench (**0.0125**), surpassing all baselines including post-processed ones. Furthermore, on the challenging *DocElemHard* benchmark, we secure the best **Overall** performance (**91.65**), outperforming specialized engines like PaddleOCR-VL (91.50) and HunyuanOCR (90.11). This superiority is evident in the demanding **Table** recognition task, where we score **90.6**, surpassing the nearest competitor (PaddleOCR-VL) by a substantial margin (+3.7). This demonstrates that our end-to-end method generalizes effectively to complex layouts where rule-based post-processing often struggles.

4.5 Conclusion

In this work, we introduced **DocPO**, a reinforcement learning framework utilizing **Tailored Step-Aware Rewards** to align optimization with the structural complexity of document parsing. By integrating semantic verification and continuous feedback, our method optimizes formulas, tables, and text without relying on expensive human preference data. Experiments on OmniDocBench and DocElemHard demonstrate that DocPO significantly outperforms generic baselines, establishing a scalable paradigm for high-precision document understanding and paving the way for future research into advanced document parsing.

Limitations

Despite the promising results, DocPO presents certain limitations. First, the RL training phase incurs higher computational overhead compared to SFT, primarily due to the generation of multiple rollouts and the calculation of complex structural rewards (e.g., TEDS), although inference latency remains unaffected. Second, the framework relies on proxy rewards—such as LLM judges and rule-based metrics—which, while effective, are not infallible and carry a risk of reward hacking if the proxies fail to capture subtle semantic nuances. Finally, our current scope is restricted to text, formulas, and tables; extending this step-aware optimization to broader modalities like charts and geometric diagrams remains a direction for future research.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

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

Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, and 1 others. 2025. Paddleocr 3.0 technical report. *arXiv preprint arXiv:2507.05595*.

Lei Cui, Yiheng Xu, Tengchao Lv, and Furu Wei. 2021. Document ai: Benchmarks, models and applications. *arXiv preprint arXiv:2111.08609*.

Hao Feng, Shu Wei, Xiang Fei, Wei Shi, Yingdong Han, Lei Liao, Jinghui Lu, Binghong Wu, Qi Liu, Chunhui Lin, and 1 others. 2025. Dolphin: Document image parsing via heterogeneous anchor prompting. *arXiv preprint arXiv:2505.14059*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Jaekyu Ha, Robert M Haralick, and Ihsin T Phillips. 1995. Recursive xy cut using bounding boxes of connected components. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 2, pages 952–955. IEEE.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

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

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, and 1 others. 2024. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*.

Zhang Li, Yuliang Liu, Qiang Liu, Zhiyin Ma, Ziyang Zhang, Shuo Zhang, Zidun Guo, Jiarui Zhang, Xinyu Wang, and Xiang Bai. 2025. Monkeyocr: Document parsing with a structure-recognition-relation triplet paradigm. *arXiv preprint arXiv:2506.05218*.

Tengchao Lv, Yupan Huang, Jingye Chen, Yuzhong Zhao, Yilin Jia, Lei Cui, Shuming Ma, Yaoyao Chang, Shaohan Huang, Wenhui Wang, and 1 others. 2023. Kosmos-2.5: A multimodal literate model. *arXiv preprint arXiv:2309.11419*.

Junbo Niu, Zheng Liu, Zhuangcheng Gu, Bin Wang, Linke Ouyang, Zhiyuan Zhao, Tao Chu, Tianyao He, Fan Wu, Qintong Zhang, and 1 others. 2025. Mineru2. 5: A decoupled vision-language model for efficient high-resolution document parsing. *arXiv preprint arXiv:2509.22186*.

Linke Ouyang, Yuan Qu, Hongbin Zhou, Jiawei Zhu, Rui Zhang, Qunshu Lin, Bin Wang, Zhiyuan Zhao, Man Jiang, Xiaomeng Zhao, and 1 others. 2025. OmniDocBench: Benchmarking diverse pdf document parsing with comprehensive annotations. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24838–24848.

Mateusz Pawlik and Nikolaus Augsten. 2015. Efficient computation of the tree edit distance. *ACM Transactions on Database Systems (TODS)*, 40(1):1–40.

Mateusz Pawlik and Nikolaus Augsten. 2016. Tree edit distance: Robust and memory-efficient. *Information Systems*, 56:157–173.

Jake Poznanski, Aman Rangapur, Jon Borchardt, Jason Dunkelberger, Regan Huff, Daniel Lin, Christopher Wilhelm, Kyle Lo, and Luca Soldaini. 2025. olmocr: Unlocking trillions of tokens in pdfs with vision language models. *arXiv preprint arXiv:2502.18443*.

Hunyuan Vision Team, Pengyuan Lyu, Xingyu Wan, Gengluo Li, Shangpin Peng, Weinong Wang, Liang Wu, Huawen Shen, Yu Zhou, Canhui Tang, and 1 others. 2025. Hunyuanocr technical report. *arXiv preprint arXiv:2511.19575*.

Baode Wang, Biao Wu, Weizhen Li, Meng Fang, Zuming Huang, Jun Huang, Haozhe Wang, Yanjie Liang, Ling Chen, Wei Chu, and 1 others. 2025. Infinity parser: Layout aware reinforcement learning for scanned document parsing. *arXiv preprint arXiv:2506.03197*.

Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, and 1 others. 2024. Mineru: An open-source solution for precise document content extraction. *arXiv preprint arXiv:2409.18839*.

Zilong Wang, Yiheng Xu, Lei Cui, Jingbo Shang, and Furu Wei. 2021. Layoutreader: Pre-training of text and layout for reading order detection. *arXiv preprint arXiv:2108.11591*.

Haoran Wei, Yaofeng Sun, and Yukun Li. 2025. Deepseek-ocr: Contexts optical compression. *arXiv preprint arXiv:2510.18234*.

Jiarui Zhang, Yuliang Liu, Zijun Wu, Guosheng Pang, Zhili Ye, Yupei Zhong, Junteng Ma, Tao Wei, Haiyang Xu, Weikai Chen, and 1 others. 2025. Monkeyocr v1. 5 technical report: Unlocking robust document parsing for complex patterns. *arXiv preprint arXiv:2511.10390*.

This appendix provides supplementary implementation details to facilitate reproducibility and offers a deeper statistical analysis of the evaluation benchmarks. We first present the specific prompts used for multi-modal parsing and semantic verification. Subsequently, we detail the composition of the *DocElemHard* dataset, highlighting its structural complexity compared to existing benchmarks.

A Prompts for Document Element Parsing

To ensure consistent output formats across different modalities, we employ specific instructions for text, formula, and table generation. The exact prompts used during the inference stage are detailed below:

[Text Parsing]

`Parse the text block without using any $$...$$.`

[Formula Parsing]

`Parse the formula with latex format.`

[Table Parsing]

`Please convert this cropped image directly into html format of table.`

B Prompt for Formula Semantic Verification

Beyond standard generation, assessing mathematical accuracy requires distinguishing between stylistic variations and semantic errors. We utilize the following prompt to instruct the verifier to check for semantic equivalence between the predicted formula and the ground truth:

`Please determine whether [Formula 1] and [Formula 2] are semantically consistent. Ignore variations in representation (e.g., spacing or LaTeX command synonyms) and focus solely on semantic equivalence.`

`The evaluation must be strict, including identical variable names. If [Formula 2] contains abnormal repetitions, it should be deemed Inconsistent.`

[Input Format]

`[Formula 1] = """{gt}"""`

`[Formula 2] = """{pred}"""`

[Output]

`Respond only with` **`Consistent`** `or` **`Inconsistent`**`.`

C Dataset Statistics and Comparisons

Finally, we provide a statistical breakdown of the *DocElemHard* dataset. As shown in Table 5, the dataset covers three core element categories. More importantly, Table 6 highlights the structural complexity of our dataset compared to OmniDocBench, specifically the significantly higher proportion of tables containing embedded equations (61.0%) and spanning cells (71.5%).

Table 5: Overall distribution of document elements in the proposed *DocElemHard* dataset ($N = 9,579$).

Element Category	Instance Count
Text Block	8,101
Formula	480
Table	998

D Training Reward

Visualized reward trajectories for text, formula, and table converge rapidly, validating our granular reward modeling.

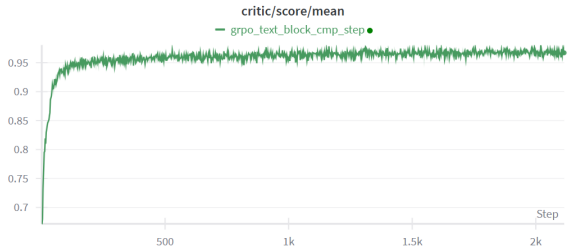

Figure 8: Text Reward during DocPO training.

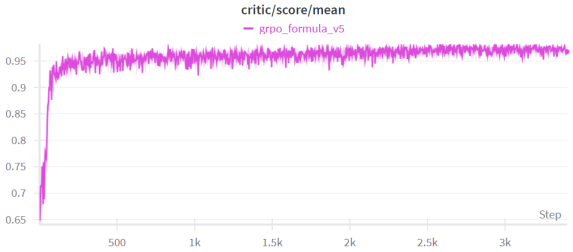

Figure 9: Text Reward during DocPO training.

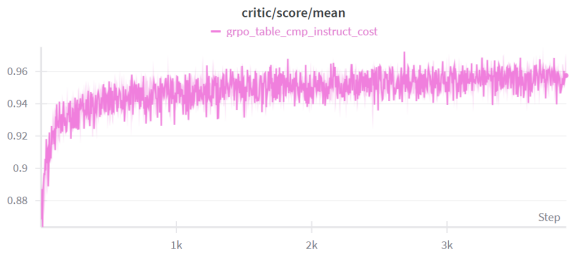

Figure 10: Table Reward during DocPO training.

Table 6: Structural and linguistic comparison between $DocElemHard_{\text{table}}$ and $OmniDocBench_{\text{table}}$.

Dimension	**Attribute**	$DocElemHard_{\textbf{table}}$ **(Ours)**	$OmniDocBench_{\textbf{table}}$
Scale	Total Samples	998	512
Language	English	998 (100%)	196 (38.2%)
	Chinese (Simp.)	-	295 (57.7%)
	Mixed	-	21 (4.1%)
Equation	With Embedded Equation	**609 (61.0%)**	87 (17.0%)
	Text-only Content	389 (39.0%)	424 (83.0%)
Complexity	With Spanning Cells (Span)	**714 (71.5%)**	158 (30.9%)
	Regular Layout	284 (28.5%)	353 (69.1%)