PERCEIVING BEFORE COGNIZING: AGENTIC VISUAL ENHANCEMENT FOR ROBUST DOCUMENT PARSING

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern document parsing models perform well on clean images but remain vulnerable to real-world visual degradations, motivating input enhancement to improve robustness across parsers. Yet document-dependent enhancement effects and potentially erroneous tool choices call for adaptive policies that can reassess intermediate results and explore alternative paths. We propose *Perceiving Before Cognizing*, which defines visual enhancement before parsing as an independently learnable interactive decision-making task, using parsing utility to guide optimization and improve the accessibility of task-relevant visual evidence. We introduce the *AgenticDoc* framework to support the reuse of this enhancement capability across parsers. To support policy learning for this task, we introduce *Action-to-Token On-Policy Distillation* (A2T-OPD), which bridges the gap between action-level teacher supervision and token-level generative policy learning. AgenticDoc improves all 13 evaluated parsers on Wild-OmniDocBench by 7.15 on average, including an average gain of 7.44 across nine parsers not used for training feedback. Overall performance on OmniDocBench v1.6 documents is preserved, with an average score change of +0.41. On PureDocBench, average gains across ten parsers reach 2.48 across the three scenarios.

1 INTRODUCTION

Document parsing models continue to set new benchmark records(Ouyang et al., 2024), yet benchmarks such as Wild-OmniDocBench (Li et al., 2026a) show that a substantial robustness gap persists under real-world capture conditions: even when document content and semantic structure remain unchanged, visual variations can still cause parsing errors. Existing studies improve parsers' generalization across degradation scenarios primarily from the training-data side, through methods such as realistic scene synthesis, document-aware training, and targeted data optimization (Bin et al., 2026; Zhang et al., 2026). Complementary to this direction, we focus on an opportunity for active intervention at inference time: before a document is passed to a downstream parsing model, can its visual presentation be improved so that the information already present becomes easier to recognize?

This question does not imply that every document should undergo enhancement. The benefit of an enhancement operation depends on the current image state, while the composition and execution order of operations also affect the final result. Unnecessary processing may even damage legible content and degrade parsing performance. Moreover, complex real-world degradations may extend beyond those encountered during training, making initial assessments and tool selection more prone to error. Document enhancement therefore requires both adaptive decisions about subsequent operations and stopping, informed by observed outcomes, and the flexibility to explore alternatives: reassessing intermediate results and revisiting earlier image states to pursue different enhancement paths rather than continuing along a fixed sequence.

We adopt the principle of *Perceiving Before Cognizing*: learning to adaptively enhance task-relevant visual evidence before passing the document to a frozen parser. To this end, we introduce *AgenticDoc*, a task-driven agent framework for visual enhancement in document parsing. It makes input-side perceptual enhancement measurable under an explicit parsing objective and allows the resulting capability to be shared across different parsing models. To support training and evaluation, we implement an interactive visual enhancement environment with controllable degradations and access to previously generated image states. During training, the resulting parsing performance provides

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

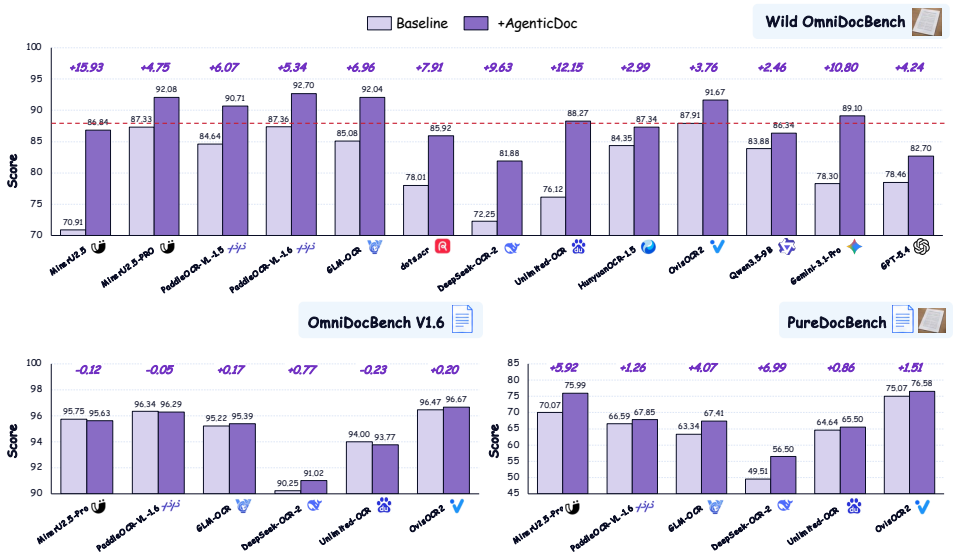

Figure 1: AgenticDoc improves diverse document parsers on Wild-OmniDocBench while preserving performance on OmniDocBench v1.6. Seven parsers surpass the best unenhanced baseline (red dashed line) when equipped with AgenticDoc. On PureDocBench, all displayed parsers also improve in average score across its clean, digitally degraded, and real degraded subsets.

task-level feedback for autonomous policy exploration. We further use the discrete action space formed by the enhancement tools to explore multi-step enhancement trajectories online, and construct an action-level teacher distribution at each state from the parsing returns of subsequent paths. To translate this supervision into updates of a generative agent, we propose *Action-to-Token On-Policy Distillation* (A2T-OPD), which aligns the action-level teacher distribution with token-level decisions at states encountered during actual interaction.

Our main contributions are:

- **A framework and interaction environment for visual enhancement.** We propose AgenticDoc, which organizes visual enhancement guided by parsing utility as an independently learnable capability reusable across parsers. Building on this framework, we implement an interactive enhancement environment in which an agent executes enhancement operations under controlled degradation conditions, revisits historical image states to explore alternative paths, and explores and learns policies from parsing feedback, providing a practical basis for training and evaluating this capability.

- **A2T-OPD for supervising generative decisions in the semantic action space.** We propose A2T-OPD, which uses a semantic action trie to decompose action-level teacher distributions into conditional token targets. It aligns these targets with student token predictions along on-policy trajectories, transferring search-derived action preferences to both tool selection and parameter decisions.

- **Robustness gains across diverse document parsers.** AgenticDoc improves all 13 evaluated parsers by 7.15 points on average on Wild-OmniDocBench, with an average gain of 7.44 points across the nine unseen parsers in training, while preserving overall performance on clean OmniDocBench v1.6 documents (+0.41 points on average). On PureDocBench, average gains across ten parsers reach 1.93, 2.27, and 3.24 points on the clean, digitally degraded, and real-world degraded subsets, respectively, demonstrating benefits across diverse document conditions.

2 Related Work

2.1 Document Parsing Benchmarks

OmniDocBench Ouyang et al. (2024) has become a widely recognized benchmark for document parsing, on which VLM-based methods have continuously advanced the state of the art. Wild-OmniDocBench Li et al. (2026a) draws attention to the remaining failures in real-world scenarios by photographing documents from OmniDocBench under diverse capture conditions. More recently, PureDocBench Li & collaborators (2026) provides three subsets derived from the same source documents: clean, digitally degraded, and real-world degraded inputs, enabling controlled comparison across degradation settings.

2.2 Document Parsing Models

Representative document parsers include two-stage approaches such as the PaddleOCR-VL series Cui et al. (2025; 2026) and MinerU Wang et al. (2024); Niu et al. (2025) series, as well as end-to-end approaches such as dots.ocr Li et al. (2025) and the DeepSeek-OCR series Wei et al. (2025; 2026). Across these approaches, scaling training data and broadening its coverage play a central role in improving parsing performance. Recent iterations further emphasize data quality and hard-example refinement Bin et al. (2026); Zhang et al. (2026). These efforts improve parsers through training on larger and more diverse corpora. We pursue a complementary direction during inference: enhancing the input image before parsing with the parser frozen. AgenticDoc learns this input-side enhancement as a reusable policy that can benefit various parsers without retraining them.

2.3 Reinforcement Learning and Distillation for Generative Policies

Reinforcement learning and knowledge distillation provide complementary approaches to optimizing generative policies. Proximal Policy Optimization (PPO) (Schulman et al., 2017) provides a widely adopted framework for policy optimization using sampled returns. In LLMs, reinforcement learning has been applied to alignment with human preferences (Ouyang et al., 2022), mathematical reasoning (Shao et al., 2024), and multi-turn tool use (Jin et al., 2025). On-policy distillation (OPD) provides teacher supervision on student-generated sequences, mitigating the distribution mismatch between training and inference (Agarwal et al., 2024). SOD applies OPD to tool-using agents, adaptively weighting supervision across interaction steps to account for student–teacher divergence (Zhong et al., 2026). Our work investigates how action-level supervision obtained through search and environment evaluation can support on-policy distillation of generative policies.

3 Methods

3.1 AgenticDoc

3.1.1 Task formulation.

We formulate document image enhancement as an interactive decision process. At time t, the agent observes the current image I_t, the image–action history H_t, and the action set $\mathcal{A}_t$. The history retains the input, previously generated images, and the calls that produced them. The agent samples

$$s_t = (I_t, H_t, \mathcal{A}_t), \qquad a_t \sim \pi_\theta(\cdot \mid s_t). \tag{1}$$

The action set contains enhancement calls and `parse`. An enhancement call a_t specifies a configured tool and the image to which it is applied. Let $i_t \in \{0, \ldots, t\}$ index the selected image among the current and retained image states. Executing the call produces

$$I_{t+1} = f_{a_t}(I_{i_t}). \tag{2}$$

The resulting image becomes the current observation and is added to the history together with the executed call. Selecting $i_t = t$ continues the current enhancement path, whereas selecting $i_t < t$ revisits an earlier image to explore an alternative path. The `parse` action terminates the interaction and returns the current image I_T for downstream parsing.

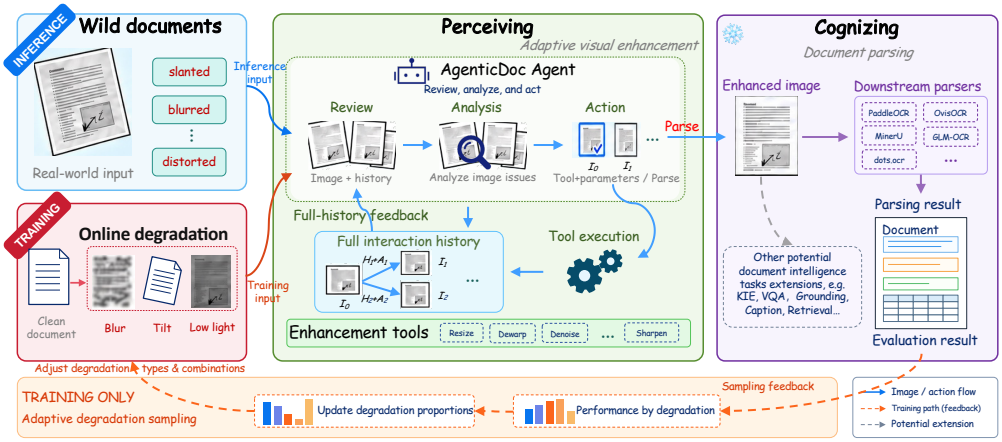

Figure 2: Overview of AgenticDoc framework. AgenticDoc decouples adaptive visual enhancement (*Perceiving*) from document parsing (*Cognizing*). Guided by image and action history, the agent selects any historical image, and either applies an enhancement tool or submits the selected image for parsing. All interactions are added to the history, forming an iterative enhancement loop. During training, clean documents are degraded online, and parsing feedback adaptively adjusts the sampling probabilities of degradation types and combinations. Orange dashed arrows denote this training-only feedback loop, while gray dashed arrows indicate potential applications beyond document parsing.

3.1.2 Online degradation–enhancement environment.

To our knowledge, AgenticDoc is the first framework in document parsing to integrate online degradation with multi-step agentic enhancement. The environment can reproduce degradation patterns on which the current policy underperforms. We represent it as $\mathcal{E} = (\mathcal{G}, \mathcal{F}, \mathcal{V}_\mathcal{T})$, where $\mathcal{G}$ is a controllable online degradation generator, $\mathcal{F}$ is a set of executable enhancement operators, and $\mathcal{V}_\mathcal{T}$ is a task-metric-based verifier.

Degradation operators. We integrate eight categories of degradation operators to simulate diverse visual degradations encountered in real-world document capture and processing, including blur, noise, geometric distortion, etc. Operators can be composed to produce compound degradations. Through preliminary calibration experiments, we assign their parameter configurations to three severity levels based on the resulting drop in parsing performance, enabling controlled variation in degradation difficulty. The taxonomy and calibration procedure are detailed in Appendix C.1.

Enhancement tools. We integrate tools for geometric correction, illumination and contrast adjustment, denoising, sharpening, and resizing. These tools provide complementary ways to improve the visual evidence available to downstream parsers, but are not assigned to degradation types in a one-to-one mapping. Their usefulness depends on the current image state and the effects of preceding operations; the agent therefore selects and composes tools based on the observed images. The enhancement operators are described in Appendix C.4.

Task-dependent visual-quality proxy and verifier. We use downstream parsing scores as a proxy for document image quality. For image I and task $\mathcal{T}$, we define task-dependent visual-quality $Q_\mathcal{T}$:

$$Q_\mathcal{T}(I) = \mathbb{E}_{m\sim\rho_\mathcal{T}}\left[S_\mathcal{T}(m, I)\right], \tag{3}$$

where $S_\mathcal{T}(m, I)$ is parser m's task score and $\rho_\mathcal{T}$ is a fixed distribution over the frozen parsers. For a content-preserving action a, its task-level quality gain is

$$\Delta Q_\mathcal{T}(I, a) = Q_\mathcal{T}(f_a(I)) - Q_\mathcal{T}(I). \tag{4}$$

The verifier computes these quantities from hidden target annotations and the fixed parsers; a positive gain therefore indicates improved parsing utility, not ground-truth perceptual quality. Verifier selection and isolation details are provided in Appendix A.2.

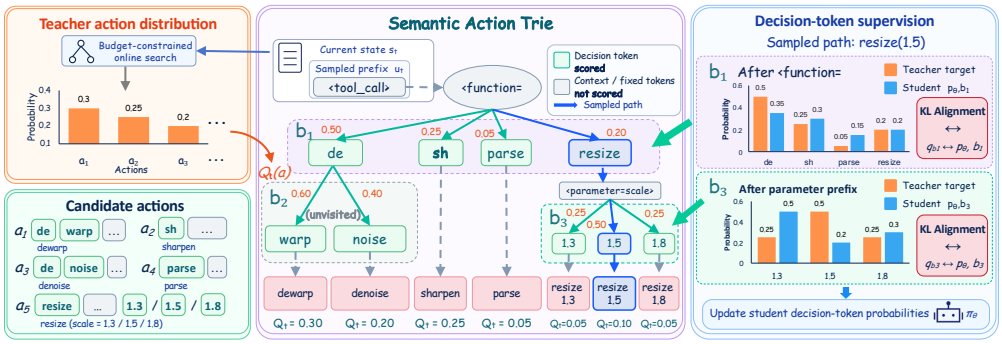

Figure 3: Illustration of A2T-OPD pipeline, with tokenization shown schematically. Semantic Action Trie converts action probabilities from budget-constrained search into conditional token targets by aggregating and normalizing probability mass at decision positions. Along the student's sampled trajectory, KL alignment supervises distributions over all legal child tokens at each visited decision position. For the sampled action `resize(1.5)`, distillation is applied at b_1 and b_3. Shared and fixed tokens remain part of the conditioning context without receiving direct distillation supervision.

3.1.3 AGENT–ENVIRONMENT INTERACTION.

In AgenticDoc, given an input document image, the agent inspects the available visual evidence, analyzes image quality issues, selects an enhancement tool and a current or historical image on which to execute it. The environment retains each result so that the agent can continue processing it or revisit an earlier state to try a different enhancement path. This interaction repeats until the agent selects `parse`, after which the final image is passed to a downstream parser. The same interaction loop therefore supports both training interactions and the processing of real-world inputs.

During training, the degradation generator supplies controlled variations of clean documents, and the verifier evaluates the resulting images using ground-truth annotations to provide task feedback for policy learning. Recorded performance also guides dynamic degradation sampling toward the policy's current weaknesses while retaining broader coverage; details are provided in Appendix A.2.

3.2 DATA CONSTRUCTION

Data sources. We retain clean documents that two OCR models both parse correctly and split them at the source-document level to prevent visual variants of the same content from crossing data splits. Training-time scoring parsers and held-out evaluation parsers are disjoint; selection and isolation details are given in Appendix D.

SFT warm-start data. From approximately 1,000 base images, we generate degraded SFT examples and assign plausible action sequences using prior knowledge of tool behavior. These sequences warm-start degradation recognition and tool-call formatting; they are neither fixed degradation–action mappings nor optimal trajectories.

RL data. RL reuses the same base images with additional compound degradations and provides no restoration sequences or optimal action labels. The policy explores tool choice, order, and stopping online; OCR ground truth is used only to compute rewards and A2T-OPD supervision at visited states. Agent outputs follow the standard ReAct Yao et al. (2023) structure and the standard Qwen function-call format. Further construction details are provided in Appendix D.

3.3 A2T-OPD: ACTION-TO-TOKEN ON-POLICY DISTILLATION

Building on AgenticDoc's interactive environment and discrete enhancement tool space, we conduct budget-constrained online search over multi-step enhancement trajectories rooted at student-visited states. Candidate actions are evaluated by their downstream parsing returns over subsequent steps, yielding an action-level teacher distribution Q_t. Each candidate specifies the parameter values re-

quired for execution, and its probability reflects the relative preference inferred from search. The search procedure and construction of Q_t are detailed in Appendix A.3.

We seek to exploit this distribution through on-policy distillation (OPD) (Agarwal et al., 2024), using relative preferences among candidate actions to provide additional supervision at student-visited states and complement policy optimization based solely on returns from executed actions. However, this action-level distribution does not directly specify the conditional token targets required by the generative student. Let $\mathcal{A}_t$ denote the set of complete executable actions, $\mathcal{V}$ the student vocabulary, and $c_{t,j}$ the current generation context. The teacher provides $Q_t \in \Delta(\mathcal{A}_t)$, whereas the student predicts $P_\theta(\cdot \mid c_{t,j}) \in \Delta(\mathcal{V})$. These distributions assign probability to different events, so direct KL alignment is not well-defined without establishing a common event correspondence:

$$\underbrace{D_{\mathrm{KL}}\left(\underbrace{Q_t(\cdot)}_{\text{complete actions } a\in\mathcal{A}_t} \,\middle\|\, \underbrace{P_\theta(\cdot \mid c_{t,j})}_{\text{next tokens } v\in\mathcal{V}}\right)}_{\text{Direct alignment lacks a common event space}}. \tag{5}$$

The mismatch extends beyond dimensionality: a complete action spans multiple tokens, and a token may be shared by multiple actions. Consequently, action probabilities and token probabilities do not admit a direct one-to-one correspondence. Applying token-level OPD to search-derived supervision therefore requires converting the distribution over complete actions into conditional token targets appropriate to each generation context. To this end, we propose ***A**ction-to-**T**oken **O**n-**P**olicy **D**istillation* (A2T-OPD), which uses a *Semantic Action Trie* to establish the correspondence between complete actions and local generation decisions, as illustrated in Figure 3. The trie decomposes teacher action probabilities into conditional token targets that can be aligned with the student's local distributions under the same contexts.

3.3.1 Semantic Action Trie

To convert the teacher distribution over complete actions into conditional token distributions at generation positions, we use a trie (Fredkin, 1960). At tool-call step t, let s_t denote the current state, u_t the prefix generated by the student. We append the canonical call $\mathrm{canon}(a)$ of each candidate action $a \in \mathcal{A}_t$ to u_t, tokenize the resulting sequence, and merge shared token prefixes to construct a *Semantic Action Trie*. Each candidate action specifies the parameter values required for execution. We define a *decision node* b as a token prediction position with at least two distinct candidate tokens, denote its candidate token set by $\mathcal{T}_b$, and collect all decision nodes in $\mathcal{B}_t$.

For any decision node b, let $\mathcal{A}_b$ denote the candidate actions compatible with the preceding call prefix, and let $\mathcal{A}_{b,v} \subseteq \mathcal{A}_b$ contain those whose token at the current position is v. The candidate tokens partition $\mathcal{A}_b$ into disjoint subsets. We therefore construct a conditional token target by aggregating the teacher probability mass within each subset and normalizing over the current node:

$$Q_b(v) = \frac{\sum_{a\in\mathcal{A}_{b,v}} Q_t(a)}{\sum_{a\in\mathcal{A}_b} Q_t(a)}, \qquad v \in \mathcal{T}_b. \tag{6}$$

This target is defined at nodes with positive total teacher probability mass and satisfies $\sum_{v\in\mathcal{T}_b} Q_b(v) = 1$. It expresses the teacher's relative preferences over candidate tokens conditional on the preceding generation prefix. Thus, comparative information over complete actions becomes conditional supervision at individual decision positions, without requiring the teacher to provide next-token probabilities directly. This decomposition preserves the teacher probability of each complete action; construction details and a formal proof are provided in Appendix A.4.

3.3.2 Distribution alignment at decision-token positions

Along trajectories sampled from the student policy, we match every valid decision position to the trie. We denote the resulting set at interaction step t by $\mathcal{B}_t^{\mathrm{vis}} \subseteq \mathcal{B}_t$ and apply distillation at these positions. For $b \in \mathcal{B}_t^{\mathrm{vis}}$, let c_b denote the complete context containing s_t, u_t, and all prefix tool tokens. We normalize the student's predictions under this context over the same candidate set $\mathcal{T}_b$:

$$P_{\theta,b}(v) = \frac{\exp z_\theta(v \mid c_b)}{\sum_{w\in\mathcal{T}_b} \exp z_\theta(w \mid c_b)}, \qquad v \in \mathcal{T}_b, \tag{7}$$

where $z_\theta(v \mid c_b)$ is the student's next-token logit. The teacher target Q_b and student distribution $P_{\theta,b}$ are now defined over the same candidate tokens under the same context, enabling alignment through KL divergence. All candidate tokens at each visited position participate in the alignment, rather than only the token sampled by the student. Averaging the alignment losses over all valid visited decision positions yields $\mathcal{L}_{\text{A2T-OPD}}$(detailed in A.5). Teacher targets and sampled contexts are held fixed, while gradients through the student logits update the model parameters.

Finally, we jointly optimize A2T-OPD and critic-free PPO:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{A2T-OPD}}\mathcal{L}_{\text{A2T-OPD}} + \lambda_{\text{PPO}}\mathcal{L}_{\text{PPO}}. \tag{8}$$

A2T-OPD transfers search-derived action preferences to decision tokens, while critic-free PPO uses task returns and invalid-call penalties to optimize interaction behavior and encourage correctly formatted tool calls with valid actions.

4 Experiments

4.1 Experimental Setup

Datasets and Evaluation. For each downstream parser, we compare performance on original images and images processed by AgenticDoc on three public benchmarks: Wild-OmniDocBench (Li et al., 2026a), PureDocBench (Li & collaborators, 2026) and OmniDocBench v1.6 (Ouyang et al., 2024), covering degraded and clean document images. Model parameters, inference settings, and scoring procedures are kept fixed within each comparison. All three benchmarks adopt the OmniDocBench metrics and evaluation framework; we follow their respective evaluation protocols.

Implementation Details. AgenticDoc is initialized from Qwen3.5-2B (Qwen Team, 2026). During the RL process, we sample 4 trajectories per input image, with at most 3 enhancement budgets per trajectory. Training feedback is provided by four frozen downstream models: PaddleOCR-VL-1.6 (Zhang et al., 2026), MinerU2.5-Pro (Bin et al., 2026), OvisOCR2 (Lu et al., 2026) and Unlimited-OCR (Yin et al., 2026), covering mainstream modular expert VLM and End-to-End expert VLM approaches. The task scores of these models are used to construct rewards and search-derived teacher supervision during training. Additional details are provided in the appendix A.5.

Table 1: Document parsing result of different models on Wild-OmniDocBench and OmniDocBench v1.6, before (Original) and after (+AD) applying AgenticDoc. Δ denotes the score change. † marks models that provided training feedback, and $*$ marks baseline results evaluated by us.

Datasets ($\rightarrow$) **Baselines** ($\downarrow$) / **Overall Scores** ($\rightarrow$)	**Params**	**Wild-OmniDocBench**			**OmniDocBench v1.6**		
		Original	+AD	Δ	Original	+AD	Δ
Pipeline Expert VLMs							
MinerU2.5 Niu et al. (2025)	1.2B	70.91	86.84	**+15.93**	93.04	92.54	**-0.50**
MinerU2.5-Pro† Bin et al. (2026)	1.2B	87.33	92.08	**+4.75**	95.75	95.63	**-0.12**
PaddleOCR-VL-1.5 Cui et al. (2026)	0.9B	84.64	90.71	**+6.07**	94.93	94.94	**+0.01**
PaddleOCR-VL-1.6† Zhang et al. (2026)	0.9B	87.36	92.70	**+5.34**	96.34	96.29	**-0.05**
GLM-OCR Duan et al. (2026)	0.9B	85.08	92.04	**+6.96**	95.22	95.39	**+0.17**
End-to-End Expert VLMs							
dots.ocr Li et al. (2025)	2.9B	78.01	85.92	**+7.91**	90.77	90.49	**-0.28**
DeepSeek-OCR-2 Wei et al. (2026)	3B-A0.5B	72.25*	81.88	**+9.63**	90.25	91.02	**+0.77**
Unlimited-OCR† Yin et al. (2026)	3B-A0.5B	76.12	88.27	**+12.15**	94.00	93.77	**-0.23**
HunyuanOCR-1.5 Li et al. (2026b)	1B	84.35*	87.34	**+2.99**	94.74	94.74	**0.00**
OvisOCR2† Lu et al. (2026)	0.8B	87.91	91.67	**+3.76**	96.47	96.67	**+0.20**
General VLMs							
Qwen3.5-9B Qwen Team (2026)	9B	83.88*	86.34	**+2.46**	88.25*	88.92	**+0.67**
Gemini-3.1-Pro Google (2026)	-	78.30*	89.10	**+10.80**	91.59*	91.86	**+0.27**
GPT-5.4 OpenAI (2026)	-	78.46*	82.70	**+4.24**	86.20*	90.67	**+4.47**

4.2 Main Results

4.2.1 Performance across image conditions and unseen model transfer.

Table 1 reports results on real-world document photographs and clean digital documents with largely corresponding content. On Wild-OmniDocBench, AgenticDoc improves all 13 downstream models by **7.15** on average, spanning pipeline expert models, end-to-end expert models, and general VLMs. After excluding the four models used to provide training feedback, **all nine remaining models improve 7.44 on average**, demonstrating the transferability of AgenticDoc. These gains are accompanied by largely preserved performance on clean documents: the average score change across all 13 models on OmniDocBench v1.6 is **+0.41**. Table 2 further evaluates the same document content under clean, digitally degraded, and real degraded conditions in PureDocBench. Averaged across all 10 models, the gains on these three subsets are **1.93**, **2.27**, and **3.24**, respectively. These results indicate that the benefits of AgenticDoc extend beyond degraded inputs to clean digital documents that remain challenging to parse. On Wild-OmniDocBench, OmniDocBench v1.6, and PureDocBench, AgenticDoc averages 1.67, 0.85, and 1.18 enhancement actions per image, with mean processing times of 1.52, 1.13, and 1.61 s, respectively.

4.2.2 Improvements with limited training data.

AgenticDoc is trained using only 1,000 source document images augmented through online degradation, with frozen downstream parsers. For comparison, PaddleOCR-VL-1.6 Zhang et al. (2026) continues pre-training from PaddleOCR-VL-1.5 Cui et al. (2026) on 16.8M samples, while MinerU2.5-Pro Bin et al. (2026) builds on MinerU2.5's Stage 0 checkpoint and expands the document parsing pre-training dataset from 6.9M to 65.5M samples. As shown in Table 1, PaddleOCR-VL-1.5 with AgenticDoc achieves 90.71 on Wild-OmniDocBench, exceeding original PaddleOCR-VL-1.6 by **3.35**. Similarly, MinerU2.5 with AgenticDoc reaches 86.84, only **0.49** below original MinerU2.5-Pro. These results show that an enhancement policy learned from a small collection of source documents can enable existing parsers to approach or surpass the performance of counterparts optimized with millions of training samples under real-world degradation. AgenticDoc also provides gains of 5.34 and 4.75 for PaddleOCR-VL-1.6 and MinerU2.5-Pro, respectively, demonstrating that visual enhancement remains complementary to improvements in the underlying parsing models.

Table 2: Document parsing result of different models on three scenarios of PureDocBench, before (Original) and after (+AD) applying AgenticDoc. Δ denotes the score change. † marks models that provided training feedback, and $*$ marks baseline results evaluated by us.

Datasets (→)	**Clean**			**Digital Degraded**			**Real Degraded**		
Baselines (↓) / **Scores** (→)	Original	+AD	Δ	Original	+AD	Δ	Original	+AD	Δ
MinerU2.5-Pro†	75.87	80.91	**+5.04**	71.77	77.63	**+5.86**	62.56	69.44	**+6.88**
PaddleOCR-VL-1.6†	71.99*	73.81	**+1.82**	66.37*	66.62	**+0.25**	61.41*	63.13	**+1.72**
GLM-OCR	68.65	71.76	**+3.11**	63.06	66.37	**+3.31**	58.31	64.10	**+5.79**
DeepSeek-OCR-2	55.53	61.74	**+6.21**	49.41	56.80	**+7.39**	43.60	50.96	**+7.36**
Unlimited-OCR†	72.70*	72.57	**-0.13**	66.89*	66.43	**-0.46**	54.34*	57.50	**+3.16**
HunyuanOCR-1.5	73.27*	73.06	**-0.21**	68.75*	69.31	**+0.56**	59.61*	61.02	**+1.41**
OvisOCR2†	81.55	82.55	**+1.00**	77.09	78.48	**+1.39**	66.56	68.70	**+2.14**
Qwen3.5-9B	73.87	74.09	**+0.22**	73.34	73.56	**+0.22**	65.45	67.07	**+1.62**
Gemini-3.1-Pro	70.04	73.48	**+3.44**	69.28	73.85	**+4.57**	71.98	71.57	**-0.41**
GPT-5.4	67.91*	66.74	**-1.17**	65.26*	64.84	**-0.42**	53.07*	55.78	**+2.71**

4.3 Ablation Studies

4.3.1 Policy Backends within AgenticDoc

We fix AgenticDoc's operators, interaction budget, and execution environment, and vary only the policy backend. We compare no enhancement, random selection, four fixed operators, GPT-5.4, and our task-trained agent. All enhanced images are evaluated by the same four frozen parsers.

As shown in Table 3, random selection scores 83.91, 0.77 below *Do Nothing*, while the best fixed operator, Dewarp, provides only a 1.61-point gain. GPT-5.4 improves the average by 4.11, demonstrating the compatibility of AgenticDoc to directly use a general-purpose VLM as its policy. Our

Table 3: Ablation of AgenticDoc policy-backends on Wild-OmniDocBench. Entries are Overall scores (%); Avg is the unweighted mean of the four parsers. Bold indicates the best.

Evaluation Parsers (→)	**Wild-OmniDocBench**				
Policy Backends (↓) / **Scores** (→)	PaddleOCRVL-1.6	MinerU2.5-Pro	Unlimited-OCR	OvisOCR2	Avg
No Enhancement					
Do Nothing	87.36	87.33	76.12	87.91	84.68
Non-Adaptive Policies					
Random Choose	86.16 (-1.20)	86.78 (-0.55)	76.64 (+0.52)	86.04 (-1.87)	83.91 (-0.77)
Fixed Action (Dewarp)	88.17 (+0.81)	89.16 (+1.83)	79.20 (+3.08)	88.63 (+0.72)	86.29 (+1.61)
Fixed Action (Sharpen)	86.10 (-1.26)	86.65 (-0.68)	75.90 (-0.22)	86.86 (-1.05)	83.88 (-0.80)
Fixed Action (Resize)	85.54 (-1.82)	86.48 (-0.85)	76.36 (+0.24)	87.32 (-0.59)	83.92 (-0.76)
Fixed Action (Tone)	87.39 (+0.03)	87.62 (+0.29)	76.44 (+0.32)	87.82 (-0.09)	84.82 (+0.14)
Model-Based Policies					
GPT-5.4	90.92 (+3.56)	90.76 (+3.43)	83.63 (+7.51)	89.86 (+1.95)	88.79 (+4.11)
AgenticDoc (Ours)	**92.70 (+5.34)**	**92.08 (+4.75)**	**88.27 (+12.15)**	**91.67 (+3.76)**	**91.18 (+6.50)**

trained agent reaches **91.18**, improving over *Do Nothing* and GPT-5.4 by **6.50** and **2.39**, respectively. Since random selection hurts and fixed operators provide inconsistent gains, the improvement comes from conditionally selecting and stopping enhancement rather than applying a universal operation.

4.3.2 Training methods

Starting from the same SFT checkpoint, we compare five settings: SFT baseline, additional SFT on the highest-return paths identified by search from the joint PPO and A2T-OPD process (Search-SFT), PPO alone, A2T-OPD alone, and joint PPO and A2T-OPD training. All continued-training settings are trained on identical data and configurations for the same number of optimization steps.

Table 4 shows that our full method achieves the highest scores across all four downstream parsers, increasing the average score from 87.46 at SFT initialization to **91.18 (+3.72)**. Removing A2T-OPD or PPO reduces the average score by 2.28 and 0.68, respectively, supporting the complementary roles of action-preference distillation and reward-based policy optimization. The full method also outperforms Search-SFT on all four parsers by **0.86** on average, demonstrating the benefit of our training strategy over additional supervised learning on the highest-return paths identified by search.

Table 4: Ablation of training methods on Wild-OmniDocBench. Entries are Overall scores (%); Avg is the unweighted mean of the four parsers. Bold indicates the best.

Evaluation Parsers (→)	**Wild-OmniDocBench**				
Training Methods (↓) / **Scores** (→)	Paddle-VL-1.6	MinerU2.5-Pro	Unlimited-OCR	OvisOCR2	Avg
SFT baseline	88.91	88.94	84.11	87.89	87.46
PPO	90.71 (+1.80)	90.32 (+1.38)	84.66 (+0.55)	89.89 (+2.00)	88.90 (+1.44)
A2T-OPD	92.07 (+3.16)	91.82 (+2.88)	87.40 (+3.29)	90.70 (+2.81)	90.50 (+3.04)
Search-SFT	91.50 (+2.59)	91.59 (+2.65)	87.49 (+3.38)	90.68 (+2.79)	90.32 (+2.86)
PPO + A2T-OPD (Ours)	**92.70 (+3.79)**	**92.08 (+3.14)**	**88.27 (+4.16)**	**91.67 (+3.78)**	**91.18 (+3.72)**

5 Conclusion

We introduced *AgenticDoc*, a task-driven visual enhancement framework for document parsing in real-world conditions. Its interactive environment supports online degradation and retains the full interaction history. The agent review the history to select enhancement tools and parameters and decide when to stop, with the option to revisit earlier image states for trial-and-error exploration. We further introduced A2T-OPD, which converts search-derived action distributions into conditional token supervision along student-generated trajectories at decision token positions. Trained with online degradation, AgenticDoc improves diverse parsers under real-world degradation while preserving performance in clean scenarios. These findings support the principle of *Perceiving Before Cognizing* in document parsing: an agentic visual enhancement policy can improve robustness across diverse downstream parsers without retraining them.

Ethics statement

All document data used in this work were collected from publicly available open-source datasets and used in accordance with their respective licenses and terms of use. The visual assets used to instantiate degradation effects were collected internally and are used only to simulate visual degradation during data construction. This study does not involve human-subject experiments. To the best of our knowledge, the data, degradation assets, methodology, and intended research use do not introduce material privacy, safety, or other ethical risks.

Reproducibility statement

We provide the task formulation, action-level teacher construction, Semantic Action Trie, optimization objectives, degradation taxonomy, enhancement operators, and data-construction procedure in the main text and appendices. Search uses $\delta = 0.2$, $\tau = 0.01$, and a reserved probability of $\varepsilon_t(a) = 10^{-4}$ for each candidate action. SFT uses a batch size of 64 for 64 optimization steps with a learning rate of 10^{-4}, while freezing the Qwen3.5 visual encoder and aligner. The same modules remain frozen during RL. Training is conducted on eight NVIDIA A100 80GB GPUs.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

Wang Bin, He Tianyao, Ouyang Linke, Wu Fan, Zhao Zhiyuan, Chu Tao, Qu Yuan, Jin Zhenjiang, Zeng Weijun, Miao Ziyang, Xu Bangrui, Niu Junbo, et al. Mineru2.5-pro: Pushing the limits of data-centric document parsing at scale, 2026. URL https://arxiv.org/abs/2604.04771.

Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, et al. PaddleOCR 3.0 Technical Report. *arXiv preprint arXiv:2507.05595*, 2025.

Cheng Cui, Ting Sun, Suyin Liang, Tingquan Gao, Zelun Zhang, Jiaxuan Liu, Xueqing Wang, Changda Zhou, Hongen Liu, Manhui Lin, Yue Zhang, Yubo Zhang, Yi Liu, Dianhai Yu, and Yanjun Ma. Paddleocr-vl-1.5: Towards a multi-task 0.9b vlm for robust in-the-wild document parsing, 2026. URL https://arxiv.org/abs/2601.21957.

Shuaiqi Duan, Yadong Xue, Weihan Wang, Zhe Su, Huan Liu, Sheng Yang, Guobing Gan, Guo Wang, Zihan Wang, Shengdong Yan, Dexin Jin, Yuxuan Zhang, Guohong Wen, Yanfeng Wang, Yutao Zhang, Xiaohan Zhang, Wenyi Hong, Yukuo Cen, Da Yin, Bin Chen, Wenmeng Yu, Xiaotao Gu, and Jie Tang. Glm-ocr technical report, 2026. URL https://arxiv.org/abs/2603.10910.

Edward Fredkin. Trie memory. *Communications of the ACM*, 3(9):490–499, 1960.

Google. Gemini 3.1 Pro: A smarter model for your most complex tasks. https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/, 2026.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Gengluo Li, Pengyuan Lyu, Chengquan Zhang, Huawen Shen, Liang Wu, Xingyu Wan, Gangyan Zeng, Han Hu, Can Ma, and Yu Zhou. Towards real-world document parsing via realistic scene synthesis and document-aware training, 2026a. URL https://arxiv.org/abs/2603.23885.

Gengluo Li, Xingyu Wan, Shangpin Peng, Weinong Wang, Hao Feng, Yongkun Du, Binghong Wu, Zheng Ruan, Zhiqiong Lu, Liang Wu, Pengyuan Lyu, Huawen Shen, Zibin Lin, Shijing Hu, Jieneng Yang, Hongbing Wen, Guanghua Yu, Hong Liu, Bochao Wang, Can Ma, Han Hu, Chengquan Zhang, and Yu Zhou. HunyuanOCR-1.5: Making lightweight OCR VLMs faster and better. *arXiv preprint arXiv:2607.04884*, 2026b.

Yumeng Li, Guang Yang, Hao Liu, Bowen Wang, and Colin Zhang. dots.ocr: Multilingual document layout parsing in a single vision-language model, 2025. URL https://arxiv.org/abs/2512.02498.

Zhiheng Li and collaborators. How far is document parsing from solved? puredocbench: A source-traceable benchmark across clean, degraded, and real-world settings. https://github.com/zhihengli-casia/puredocbench, 2026. Dataset and benchmark release.

Shiyin Lu, Yinglun Li, Yu Xia, Yuhui Chen, An-Yang Ji, Jun-Peng Jiang, Qing-Guo Chen, Jianshan Zhao, En Lin, Haijun Li, Cheng Qin, Zhao Xu, and Weihua Luo. Ovisocr2 technical report. *arXiv preprint arXiv:2607.13639*, 2026.

Junbo Niu, Zheng Liu, Zhuangcheng Gu, Bin Wang, Linke Ouyang, Zhiyuan Zhao, Tao Chu, Tianyao He, Fan Wu, Qintong Zhang, Zhenjiang Jin, et al. Mineru2.5: A decoupled vision-language model for efficient high-resolution document parsing, 2025. URL https://arxiv.org/abs/2509.22186.

OpenAI. Introducing GPT-5.4. https://openai.com/index/introducing-gpt-5-4/, 2026.

Linke Ouyang, Yuan Qu, Hongbin Zhou, Jiawei Zhu, Rui Zhang, Qunshu Lin, Bin Wang, Zhiyuan Zhao, Man Jiang, Xiaomeng Zhao, Jin Shi, Fan Wu, Pei Chu, Minghao Liu, Zhenxiang Li, Chao Xu, Bo Zhang, Botian Shi, Zhongying Tu, and Conghui He. Omnidocbench: Benchmarking diverse pdf document parsing with comprehensive annotations, 2024. URL https://arxiv.org/abs/2412.07626.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, Bo Zhang, Liqun Wei, Zhihao Sui, Wei Li, Botian Shi, Yu Qiao, Dahua Lin, and Conghui He. MinerU: An Open-Source Solution for Precise Document Content Extraction, 2024. URL https://arxiv.org/abs/2409.18839.

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

Haoran Wei, Yaofeng Sun, and Yukun Li. Deepseek-ocr 2: Visual causal flow. *arXiv preprint arXiv:2601.20552*, 2026.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629.

Youyang Yin, Huanhuan Liu, YY, Qunyi Xie, Chaorun Liu, Shiqi Yang, Shaohua Wang, Zhanlong Liu, Hao Zou, Jinyue Chen, Shu Wei, Jingjing Wu, Mingxin Huang, Zhen Wu, Guibin Wang, Tengyu Du, and Lei Jia. Unlimited ocr works, 2026. URL https://arxiv.org/abs/2606.23050.

Zelun Zhang, Hongen Liu, Suyin Liang, Yubo Zhang, Yiqing Xiang, Jiaxuan Liu, Ting Sun, Manhui Lin, Yue Zhang, Changda Zhou, Tingquan Gao, Cheng Cui, Yi Liu, Dianhai Yu, and Yanjun Ma. Paddleocr-vl-1.6: Expanding the frontier of document parsing with under-optimized region refinement and progressive post-training, 2026. URL https://arxiv.org/abs/2606.03264.

Qiyong Zhong, Mao Zheng, Mingyang Song, Xin Lin, Jie Sun, Houcheng Jiang, Xiang Wang, and Junfeng Fang. Sod: Step-wise on-policy distillation for small language model agents, 2026. URL https://arxiv.org/abs/2605.07725.

A Appendix

A.1 AI Use Statement

In this work, we used generative AI tools to provide feedback on research methodology and experimental design, assist with implementing and editing method and experiment code, translate manuscript text, and generate image-description-centered text annotations for SFT seed trajectories using GPT models. We also used generative AI tools to search for, identify, and summarize related literature; format references; and audit the manuscript for logical consistency and the wording and consistency of reported results.

Except for the text annotations of the SFT seed data described above, we did not use generative AI tools to generate document images or degraded images for training or evaluation. We also did not use them to develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for or write mathematical proofs, propose or refine research hypotheses, clean or reformat data, interpret or present experimental results, or conduct qualitative or thematic data analysis. The degraded images used for training were generated by the programmatic degradation operators defined in this work rather than by generative AI.

The authors manually reviewed all AI-assisted outputs and made the final scientific decisions. AI-assisted code was tested by the authors, and identified errors were corrected. GPT-generated SFT annotations were reviewed by the authors; literature-search and citation suggestions were checked against the original sources; and AI-assisted text was reviewed and revised. We take full responsibility for the final content of this work, including text, claims, and other artifacts produced with the assistance of generative AI.

A.2 Adaptive Environment and Verifier Details

Verifier selection and isolation. An independent validation set provides model-reliability evidence $\mathcal{E}_{\mathcal{T}}$ for selecting the parsers used by the task proxy and determining their weights $\rho_{\mathcal{T}}$. Selection is based on clean OmniDocBench performance, after which both the parser set and its weights are held fixed. Neither test data nor scores on degraded inputs or candidate enhancement actions are used for selection or reweighting. The verifier $\mathcal{V}_{\mathcal{T}}$ computes task scores from the target annotation y and the frozen parsers; annotations and scores remain environment-side and are not exposed to the agent.

Weakness reproduction and adaptive sampling. The environment associates task scores with recorded degradation configurations to identify patterns that still yield poor parsing performance after enhancement by the current policy. Scores on the corresponding clean images serve as references, reducing confounding from document-content difficulty. For an identified pattern, the environment can reproduce the same degraded input to compare strategies and tool effects. It also perturbs severity and parameters within the same or similar degradation compositions, generates instances from different source documents, and increases their sampling probability. As the policy improves, the environment reassesses performance and updates its sampling focus while maintaining coverage of

other degradation patterns. The training distribution therefore follows current policy weaknesses rather than a predetermined increase in degradation severity.

A.3 Construction of the Action-Level Teacher Distribution

Budget-constrained return estimation. Let h_t denote the remaining enhancement budget and $\mathcal{S}(s)$ the parsing score obtained from the frozen downstream parsing models. For an enhancement action a, let $\widehat{\Xi}_t(a; h_t)$ contain the evaluated search paths starting from s_t with action a and comprising at most h_t enhancement operations. Search permits early termination, so any evaluated intermediate state can serve as a path endpoint. We estimate the return of a by

$$\widehat{U}_t(a) = \max_{\xi \in \widehat{\Xi}_t(a;h_t)} \left[\mathcal{S}(s_\xi^{\mathrm{end}}) - \mathcal{S}(s_t) - 0.05\,|\xi| \right], \tag{9}$$

where s_ξ^{end} is the terminal state of path ξ, and $|\xi|$ counts its enhancement operations. The cost is set to $c_T = 0.05$ per complete enhancement call. The stopping action, denoted by a_{parse}, corresponds to an empty enhancement path and therefore has return $\widehat{U}_t(a_{\mathrm{parse}}) = 0$.

An evaluated action remains eligible even if its successors have not been expanded, since stopping immediately after its execution provides a valid path. Equation equation 9 thus accounts for both immediate and subsequent improvements within the remaining budget. It estimates the best return among explored paths without assuming exhaustive search or global optimality.

Teacher probability assignment. Let $\mathcal{A}_t^{\mathrm{eval}} \subseteq \mathcal{A}_t$ contain the actions with valid return estimates, including a_{stop}. We define the best estimated return and the near-optimal action set as

$$\widehat{U}_t^\star = \max_{a \in \mathcal{A}_t^{\mathrm{eval}}} \widehat{U}_t(a), \quad \mathcal{N}_t = \left\{ a \in \mathcal{A}_t^{\mathrm{eval}} : \widehat{U}_t(a) \geq \widehat{U}_t^\star - \delta \right\}, \tag{10}$$

where $\delta \geq 0$ is the return tolerance. The availability of the stopping action ensures that $\mathcal{N}_t$ is nonempty.

We reserve a small positive probability $\varepsilon_t(a)$ for each candidate, with $\varepsilon_t^\Sigma = \sum_{a \in \mathcal{A}_t} \varepsilon_t(a) < 1$, and allocate the remaining mass over $\mathcal{N}_t$:

$$Q_t(a) = \varepsilon_t(a) + (1 - \varepsilon_t^\Sigma) \begin{cases} \dfrac{\exp\left((\widehat{U}_t(a) - \widehat{U}_t^\star)/\tau \right)}{\sum_{a' \in \mathcal{N}_t} \exp\left((\widehat{U}_t(a') - \widehat{U}_t^\star)/\tau \right)}, & a \in \mathcal{N}_t, \\ 0, & a \notin \mathcal{N}_t, \end{cases} \tag{11}$$

where $\tau > 0$ controls the concentration of the return-based allocation. This construction gives a normalized distribution with positive support over $\mathcal{A}_t$. Actions outside $\mathcal{N}_t$, including those without valid return estimates, receive only their reserved probabilities. Missing evaluations are therefore not treated as zero returns. All probability allocations operate on complete tool–parameter configurations.

A.4 Semantic Action Trie: Construction Details and Probability Preservation

The construction uses a fixed parameter ordering in $\mathrm{canon}(a)$. We check that complete candidate encodings are pairwise distinct and that none is a prefix of another, ensuring that each candidate can be uniquely identified by its token path. Candidates sharing a tool name remain distinct until their parameter configurations are resolved. Accordingly, trie construction stops along a path only when the complete tool–parameter configuration is uniquely identified within $\mathcal{A}_t$.

We establish the probability-preservation property of the conditional targets in Equation equation 6. The definitions of b, $\mathcal{T}_b$, $\mathcal{A}_b$, and $\mathcal{A}_{b,v}$ are inherited directly from the main text.

Proposition: Teacher probability preservation. For any $a \in \mathcal{A}_t$ with $Q_t(a) > 0$, let $\mathcal{B}(a)$ denote the decision nodes traversed until a is uniquely identified, and let $v_{a,b}$ be its corresponding token at node b. Under the distinguishability conditions above,

$$Q_t(a) = \prod_{b \in \mathcal{B}(a)} Q_b(v_{a,b}). \tag{12}$$

Proof. Order the nodes in $\mathcal{B}(a)$ as $(b_1, \ldots, b_K)$. Let $\mathcal{C}_0 = \mathcal{A}_t$ and $\mathcal{C}_i = \mathcal{A}_{b_i, v_{a,b_i}}$ be the compatible candidate set after the i-th decision. For any candidate subset $\mathcal{C}$, write its teacher probability mass as $M_t(\mathcal{C}) = \sum_{a' \in \mathcal{C}} Q_t(a')$.

Between consecutive decision nodes, all compatible candidates share the same tokens, so their set remains unchanged. Hence, $\mathcal{A}_{b_i} = \mathcal{C}_{i-1}$. At the point of unique identification, $\mathcal{C}_K = \{a\}$. Since a belongs to every set along its path and $Q_t(a) > 0$, all relevant probability masses are positive. Substituting Equation equation 6 gives

$$\prod_{i=1}^{K} Q_{b_i}(v_{a,b_i}) = \prod_{i=1}^{K} \frac{M_t(\mathcal{C}_i)}{M_t(\mathcal{C}_{i-1})} = \frac{M_t(\{a\})}{M_t(\mathcal{A}_t)} = Q_t(a). \tag{13}$$

For a singleton candidate set, the empty product equals 1, matching its sole action's teacher probability. This completes the proof.

At a non-decision position, the compatible set is unchanged and the corresponding conditional probability is 1. The same holds for the canonical continuation after a complete action has been uniquely identified. Omitting these positions therefore leaves the factorization unchanged. Such omission concerns only distillation supervision: generation must still complete the call, and preceding tokens remain part of the prediction context.

A.5 Training and Implementation Details

Critic-free PPO. We adopt a critic-free variant of PPO that retains the clipped probability-ratio objective and uses returns computed from sampled trajectories as policy-update weights, without learning a value function. Process rewards combine changes in parsing scores, enhancement costs, and penalties for invalid calls. Since the agent can revisit historical images, returns are accumulated along the image-dependency path leading to the final output, preventing gains from unrelated enhancement branches from being attributed to the current action. Actions on discarded branches retain their local negative feedback. We normalize returns by their running root mean square without mean subtraction, preserving the signs of positive and negative returns. Generated tokens within the same assistant response share the corresponding return weight, while environment observations are excluded from the policy loss.

This design is suited to the short interaction horizon of our task: each trajectory contains at most three enhancement operations, and the environment can evaluate parsing performance before and after each operation. Returns can therefore be computed directly from observed execution outcomes, without bootstrapping unobserved future returns from a learned value function. This avoids the training overhead and approximation errors associated with an additional value model while retaining PPO's clipping mechanism for moderating policy updates.

Optimization objective. A2T-OPD uses an equally weighted mixture of forward and reverse KL divergences. Following the notation in the main text, the local distillation loss at each valid decision position b matched along a student-generated trajectory is

$$\ell_b = \frac{1}{2} D_{\mathrm{KL}}(Q_b \,\|\, P_{\theta,b}) + \frac{1}{2} D_{\mathrm{KL}}(P_{\theta,b} \,\|\, Q_b)\,. \tag{14}$$

Both terms are computed over the candidate token set $\mathcal{T}_b$. The forward term penalizes insufficient student probability on candidates favored by the teacher, while the reverse term penalizes excessive student probability on candidates assigned low teacher probability.

Averaging the local losses ℓ_b over all valid decision positions yields

$$\mathcal{L}_{\text{A2T-OPD}} = \frac{1}{N_{\text{valid}}} \sum_{i=1}^{N} \sum_{t=1}^{T_i} \sum_{b \in \mathcal{B}_{i,t}^{\text{vis}}} \ell_b. \tag{15}$$

Here, N is the number of sampled trajectories, T_i is the number of states retained for distillation in trajectory i, and $\mathcal{B}_{i,t}^{\text{vis}}$ contains the valid visited decision positions at its t-th retained state. N_{valid} denotes the total number of these positions across the batch.

Table 5: Descriptive results for counterfactual enhancement-action order.

Actions	Counterfactual path pairs	Observation	Summary
2	94 pairs; mean $\Delta = 0.028$, maximum $\Delta = 0.312$	AB outperforms BA in 47.2% of comparisons, so the two directions have similar win rates	No stable global directional preference across images
3	150 pairs; mean $\Delta = 0.027$, maximum $\Delta = 0.501$	Some adjacent action pairs show clear local order preferences, but the preferred direction varies with the action pair and image	Pair-specific local preferences exist, but there is no universal global ordering

The resulting $\mathcal{L}_{\text{A2T-OPD}}$ is jointly optimized with $\mathcal{L}_{\text{PPO}}$ as specified in Equation equation 8. Teacher targets remain fixed during optimization, and gradients propagate only through the student distributions. The KL regularizer relative to the frozen reference policy is included in the PPO objective and is computed separately from this teacher–student distribution alignment.

Training configuration. Starting from the SFT initialization, we perform full-parameter training using AdamW with a learning rate of 10^{-5}. Each batch of newly sampled trajectories is reused for two PPO update epochs. In the joint objective, we set $\lambda_{\text{PPO}} = 1$ and $\lambda_{\text{A2T-OPD}} = 0.01$.

B Effect of Enhancement Action Order

To support the statement in the main text that the composition and execution order of enhancement operations can affect the final result, we conduct a counterfactual analysis using 488 stored recovery search trees. Search uses a beam width of 3 and a maximum depth of 3. Among these trees, 449 have a best trajectory containing at least two enhancement actions. In 208 of them (46%), search also explores trajectories with the same action multiset as the best trajectory but a different execution order, yielding 244 comparable counterfactual path pairs.

Let $\Delta = Q(\tau_{\text{best}}) - Q(\tau_{\text{perm}})$ denote the final-score difference between the best trajectory and a trajectory that permutes the same action set. For two-action trajectories, we compare AB with BA. For three-action trajectories, we also compare adjacent swaps while holding the other action and positions fixed. Table 5 reports the corresponding descriptive statistics.

Table 5 shows that a fixed direction does not provide a stable benefit for two-action trajectories. With three actions, some adjacent action pairs exhibit clearer local order preferences. Enhancement actions are therefore generally non-commutative, but the data do not support a fixed global order that applies to every image. The appropriate execution order depends on the current image state and the preceding actions. Because permutations pruned by beam search are not included, these observations apply only to the counterfactual trajectories actually explored by search.

C Implementation Details of the Online Degradation–Interaction Environment

C.1 Online Degradation Generation

At the beginning of each episode, $\mathcal{G}$ samples a clean document $(I^\star, y) \sim \mathcal{D}_{\text{clean}}$ and a degradation configuration $\eta \sim P_{\text{deg}}$. It produces the initial image $I_0 = D_\eta(I^\star)$ using the composite degradation operator D_η. For the current execution, η specifies the degradation types, levels, and concrete parameters required to generate the image. Persistent records used for subsequent adaptive sampling retain only the degradation types and level ranges; they do not store exact hyperparameters for replaying the same degradation instance. The sampling distribution P_{deg} can be updated using training feedback to generate additional examples around the model's current weaknesses. All degradation operators share image input/output and parameter configuration interfaces and can be

Table 6: Taxonomy of the eight degradation categories and 20 subtypes, together with the real-world conditions they simulate.

Category	Degradation subtype	Real-world condition
Blur	`gaussian_blur`	Uniform softening introduced by low-quality optics, scanner smoothing, or post-processing.
	`motion_blur`	Directional streaking caused by camera shake or relative motion between the camera and document during exposure.
	`camera_psf_blur`	Optical blur caused by lens aberrations and a non-ideal camera point-spread function.
	`defocus_disk`	Out-of-focus blur when the document lies outside the camera's focal plane.
Noise	`poisson_gaussian_noise`	Combined photon shot noise and sensor read noise in low-light or high-ISO capture.
Compression	`jpeg_compress`	Blocking and ringing introduced by messaging platforms, cloud services, or repeated JPEG saving.
Rotation	`right_angle_rotation`	A page stored at 90°, 180°, or 270° after orientation metadata is lost during scanning or capture.
Geometry	`light_rotation`	Small page skew caused by misaligned scanner feeding or slight camera roll.
	`light_perspective`	Trapezoidal perspective distortion from oblique smartphone capture.
	`uv_flow_warp`	Non-rigid deformation caused by curled or folded paper, or by curvature near a book binding.
Illumination	`brightness_contrast_gamma`	Global brightness and contrast changes caused by underexposure, overexposure, or camera tone mapping.
	`soft_shadow`	A soft local shadow cast by a hand, the capture device, or another object blocking the light source.
	`lighting_map_overlay`	Spatially non-uniform illumination, highlights, or intensity gradients caused by lamps, window light, or reflections.
	`vignetting`	Attenuation near page borders and corners caused by lens shading or partial occlusion.
Sampling/aliasing	`downsample`	Resolution loss from low-resolution capture, thumbnail generation, or bandwidth-saving rescaling.
	`aliasing`	Jagged character strokes and stair-step edges caused by insufficient sampling or repeated resizing.
	`moire`	Interference patterns produced when photographing a display or printed halftone.
Document medium	`paper_texture`	Fibers and background patterns from aged, recycled, or rough paper.
	`ink_bleed`	Show-through from the reverse side of thin paper, or stroke spreading caused by ink diffusion and penetration.
	`dirty_scanner`	Dust, fingerprints, stains, or scratches on the scanner platen or document surface.

uniformly composed into single- or multi-degradation generation pipelines. Degradation is applied only during initialization; subsequent image changes result from the agent's tool actions.

C.2 Degradation Taxonomy

We organize the degradations in the online environment into eight categories and 20 subtypes. Each subtype models a plausible failure mode arising during document capture, transmission, resampling, or physical preservation, rather than serving only as an abstract image perturbation. Table 6 summarizes the taxonomy and its real-world correspondence.

C.3 Automatic Degradation-Level Calibration

Calibration set. We select 200 clean document images from data disjoint from the test set and require both PaddleOCR and OvisOCR2 to parse every image perfectly. This criterion prevents pre-existing parsing errors from confounding the estimated effect of a degradation.

Parameter-based operators. For each degradation operator controlled by explicit hyperparameters, we first partition its parameter range into fine-grained buckets. We then apply only one degradation with one parameter bucket to the clean images and parse the resulting images with the two fixed models. Based on the parsing performance drop relative to the clean inputs, parameter buckets that are adjacent or have similar effects are automatically merged into three levels: L1 denotes no noticeable impact, L2 a mild impact, and L3 a marked impact.

Asset-based operators. For degradations driven by external assets, such as warping and illumination conditions, we apply each asset independently to 10 clean calibration images. The asset is

Table 7: Preliminary degradation-level settings.

Operator	Category	L1 (no noticeable impact)	L2 (mild impact)	L3 (marked impact)
gaussian_blur	Blur	$\sigma = 0.15$–0.80 kernel_size= 3–31 (odd)	$\sigma = 1.10$–1.50 kernel_size= 33–47 (odd)	$\sigma = 2.00$–3.00 kernel_size= 49–71 (odd)
motion_blur	Blur	$\sigma = 2.0$–30.0 strength= 6.0	$\sigma = 30.1$–45.0 strength= 6.0	$\sigma = 45.1$–70.0 strength= 6.0
camera_psf_blur	Blur	–	Asset-based grading	
defocus_disk	Blur	(radius to be measured)	–	–
poisson_gaussian_noise	Noise	–	–	–
jpeg_compress	Compression	quality= 15–94	–	–
right_angle_rotation	Rotation	–	–	–
light_rotation	Geometry	$\lvert$angle$\rvert = 0.2$–4.0°	$\lvert$angle$\rvert = 4.01$–11.0°	$\lvert$angle$\rvert = 11.01$–30.0°
light_perspective	Geometry	–	–	–
uv_flow_warp	Geometry		Asset-based grading	
soft_shadow	Illumination	–	–	–
lighting_map_overlay	Illumination	–	–	–
vignetting	Illumination	–	–	–
downsample (restore_size=true)	Sampling/ aliasing	scale= 0.45–0.90 restore_size=true min_side=32	scale= 0.35–0.44 restore_size=true min_side=32	scale= 0.25–0.35 restore_size=true min_side=32
downsample (restore_size=false)	Sampling/ aliasing	scale= 0.68–0.90 restore_size=false min_side=32	scale= 0.45–0.67 restore_size=false min_side=32	scale= 0.35–0.44 restore_size=false min_side=32
aliasing	Sampling/ aliasing	–	–	–
moire	Sampling/ aliasing	–	–	–
paper_texture	Document medium		Used for realism; not graded by parsing impact	
ink_bleed	Document medium	–	–	–
dirty_scanner	Document medium	–	–	–

assigned to L1–L3 according to the parsing-performance drop of the two models across these 10 images. The full calibration pipeline—from degradation application and model inference to score aggregation and level assignment—is automated and involves no manual visual inspection.

Table 7 summarizes the ranges obtained from the preliminary experiments. A dash indicates that no usable fixed parameter interval has yet been obtained; it does not imply that the operator has no effect on parsing performance.

C.4 Enhancement Operators

We use an *enhancement operator* to denote a family of image transformations with the same functional objective, while a concrete algorithm or model is treated as an implementation of that operator. The current environment contains six operator categories: Sharpen, Tone, Dewarp, Rotate, Denoise, and Resize. Sharpen has two implementations—Laplacian sharpening and unsharp masking (USM)—while every other operator uses one fixed implementation. Table 8 lists all implementations and settings.

D Data Construction Details

D.1 Data Sources and Filtering

We assemble document images from publicly available open-source datasets covering newspapers, books, and other sources with diverse page layouts. To efficiently retain images that are recognizable before degradation, we parse every image with MinerU2.5-Pro and PaddleOCR-VL-1.5 and compute the normalized edit distance (NED) between their outputs. Let the two parsing results be $\hat{y}_{\mathrm{MinerU}}$ and $\hat{y}_{\mathrm{Paddle}}$. We retain only images satisfying

$$\mathrm{NED}(\hat{y}_{\mathrm{MinerU}}, \hat{y}_{\mathrm{Paddle}}) < 0.02.$$

Table 8: Implementations and hyperparameter settings of the enhancement operators.

Operator	Implementation	Hyperparameters or fixed settings
Sharpen	Laplacian sharpening	8-neighborhood Laplacian operator
	Unsharp masking (USM)	Radius fixed to 5
Tone	Adaptive brightness and contrast enhancement	Brightness and contrast parameters determined adaptively from the input image
Dewarp	Trained lightweight dewarping network	Fixed trained model weights at inference time
Rotate	In-plane image rotation	Rotation direction fixed to clockwise
Denoise	Bilateral filtering	Filter kernel fixed to 3×3
Resize	Bilinear interpolation	Interpolation method fixed to bilinear

This high agreement serves as a fast proxy for the recognizability of the original image. The filter reduces the chance that an image is already hard to parse before degradation, so that subsequent decreases in parsing performance can be attributed more reliably to the applied visual degradation.

D.2 SFT Data Construction

We use 1,000 clean document images as SFT base images. Each base image produces one or two degraded images, and one trajectory is constructed for every degraded image, yielding approximately 1k SFT trajectories. The data are constructed using the pre-calibrated degradation levels and simple prior knowledge about the enhancement tools. Based on the number of degradation operators applied to an image, we divide the data into single-degradation and compound-degradation examples.

For a single-degradation example, we apply one L2 or L3 degradation to the clean image and design an enhancement action using a prior judgment about that degradation. For a compound-degradation example, we apply one strong L2 or L3 degradation together with several randomly sampled mild L1 degradations. Its target trajectory contains one enhancement action designed for the strong degradation and one action designed for a randomly selected mild degradation.

Here, "designed for" denotes only a coarse construction prior rather than a strict one-to-one mapping between degradation and enhancement. For example, blur is usually paired with `sharpen`, while downsampling is usually paired with `resize`. However, document-medium degradations and some illumination degradations do not have directly corresponding actions in the current tool set. These SFT trajectories therefore provide basic behavioral priors rather than optimal enhancement sequences or fixed degradation–action mappings. We additionally include images with only mild degradation or no degradation and use a direct `parse` call as the target decision, teaching the model to stop when enhancement is unnecessary.

D.3 RL Data Construction

During RL, several degradation operators are randomly selected and composed for each clean image to produce the initial image for online interaction. The environment records each operator's type and severity level, where a level corresponds to an automatically calibrated hyperparameter range. These records support subsequent targeted sampling of the policy's weak degradation modes. The environment retains the degradation levels and their parameter ranges, not the exact hyperparameters used for a particular instance. Later samples are therefore drawn again from the same type and level ranges rather than strictly replaying an earlier degradation configuration. RL data contain no pre-annotated optimal enhancement trajectory: tool selection, execution order, and stopping are explored online by the current policy.

D.4 Data Format and GPT-Assisted Annotation

The SFT construction rules above produce one seed trajectory per degraded sample, with the enhancement actions or direct parsing decision determined by the stated rules and tool priors. We use GPT to annotate these trajectories, primarily by describing the visible image state and degradation phenomena, rather than by replacing the rules that determine the action trajectory. The resulting

single-trajectory examples serve as SFT seed data for learning to recognize visual degradation patterns and acquire basic knowledge of the available enhancement tools.

E QUALITATIVE ANALYSIS

This section presents three examples enhanced by AgenticDoc and reports the corresponding changes in image parsing scores. For each image, the parsing score is computed as the mean of the scores produced by PaddleOCR-VL-1.6, MinerU2.5-Pro, Unlimited-OCR, and OvisOCR2.

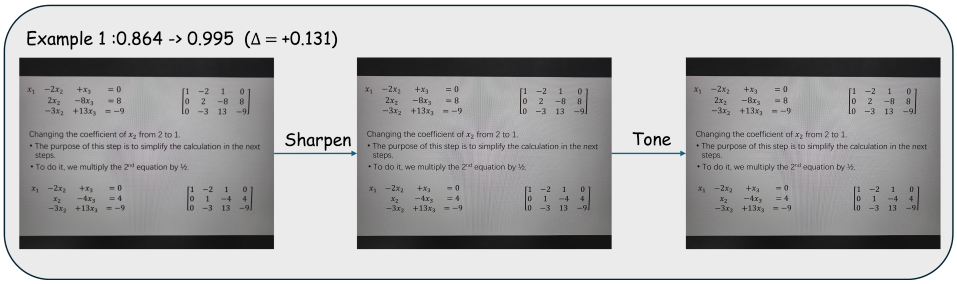

Figure 4: Qualitative case 1

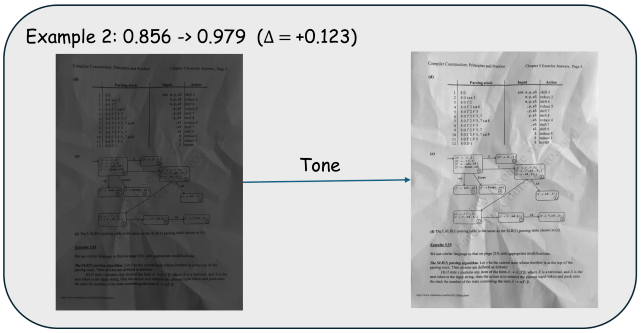

Figure 5: Qualitative case 2

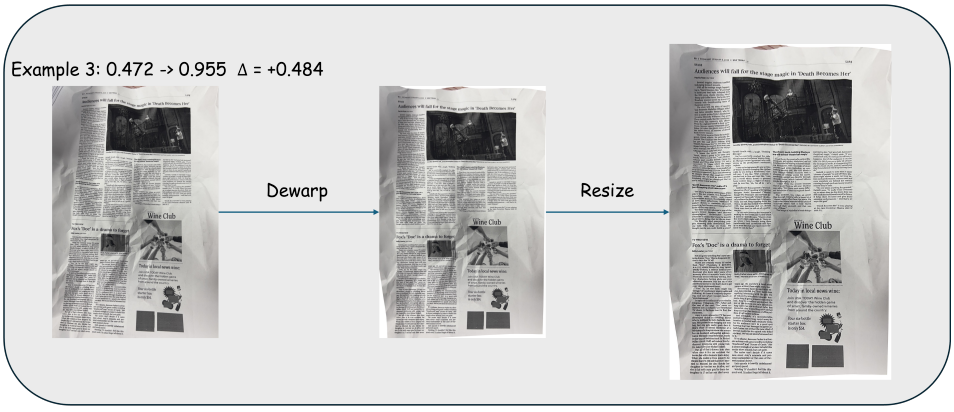

Figure 6: Qualitative case 3