

MeDocVL: Noise-Aware Post-Training of Visual Language Models for Medical Document Parsing

Wei Wu, Liang Diao*, Yuan Zhao*, Wenjie Wang, Ying Liu, Xiaole Lv

Ping An Property & Casualty Insurance Company of China, Ltd.

Ping An Finance Center, No. 5033 Yitian Road, Futian District,

Shenzhen, Guangdong, China

*Corresponding author: zhaoyuan940711@gmail.com, diaoliang91@gmail.com

Abstract

High-precision document parsing in domain-specific scenarios, such as medical documents, remains challenging due to complex layouts, specialized terminology, and strict field-level exact-match requirements under noisy supervision. As a result, existing OCR pipelines and general-purpose vision-language models (VLMs) often fail to deliver reliable key-value extraction in practice. We propose a noise-aware post-training framework that adapts general-purpose VLMs for high-precision document parsing without assuming clean annotations. The framework learns robust extraction behavior from noisy supervision through training-driven label refinement and a staged RL-to-SFT post-training strategy that balances robustness and precision. Experiments demonstrate that VLMs adapted with the proposed framework consistently outperform strong baselines under noisy supervision.

1 Introduction

Optical Character Recognition (OCR) and document parsing have long been fundamental to document digitization and information processing (Bautista and Atenza, 2022; Rang et al., 2023). Recent document-specific architectures and OCR-free VLMs have achieved strong results on broad document understanding benchmarks (Xu et al., 2021; Kim et al., 2022a; Mathew et al., 2021; Ouyang et al., 2025; Liu et al., 2026).

Nevertheless, achieving high-precision document parsing remains challenging in domain-specific scenarios, particularly for medical documents (Xu et al., 2021; Cui et al., 2021). In these settings, the objective is to extract structured key-value information from visually complex documents under strict exact-match requirements (Ouyang et al., 2022; Zhou et al., 2023). As illustrated in Figure 1, this capability does not directly translate to high-precision domain-specific

parsing, where strict exact-match requirements and noisy large-scale annotations introduce additional challenges (Li et al., 2025b; Nacson et al., 2025; Poznanski et al., 2025; Niu et al., 2026).

Although domain-specific post-training can reduce this gap, it typically relies on large-scale, clean annotations, which are costly and often impractical to obtain (Guo et al., 2025; Han et al., 2018; Li et al., 2020). In real-world scenarios, large volumes of domain data are available, but their annotations are often incomplete (Team et al., 2025; Wang et al., 2023), inconsistent, or noisy due to human errors and imperfections in automatic labeling pipelines. This leads to a fundamental tension where high-precision document parsing requires strict supervision, while real-world supervision is inherently imperfect.

In this work, we study high-precision, end-to-end document parsing under realistic noisy supervision, using medical invoices as a representative domain-specific setting. Our goal is not to model medical-specific knowledge or clinical reasoning, but to adapt general-purpose VLMs into reliable document parsers under imperfect annotations.

To this end, we propose a noise-aware post-training framework, summarized in Figure 2. The framework is built on three core ideas. First, we formulate annotation refinement as a learnable correction task, training a domain-adapted model to revise noisy labels and mitigate systematic errors introduced by automatic annotation pipelines. Second, we introduce a dynamic token-wise reinforcement learning strategy (DT-GRPO) (Schulman et al., 2017; Ouyang et al., 2022; Guo et al., 2025) that suppresses noisy and trivial optimization signals while progressively emphasizing informative hard samples, drawing inspiration from curriculum and self-paced learning paradigms (Bengio et al., 2009). Finally, we adopt an RL-to-SFT training strategy, using supervised fine-tuning (SFT) on limited expert annotations as a final precision

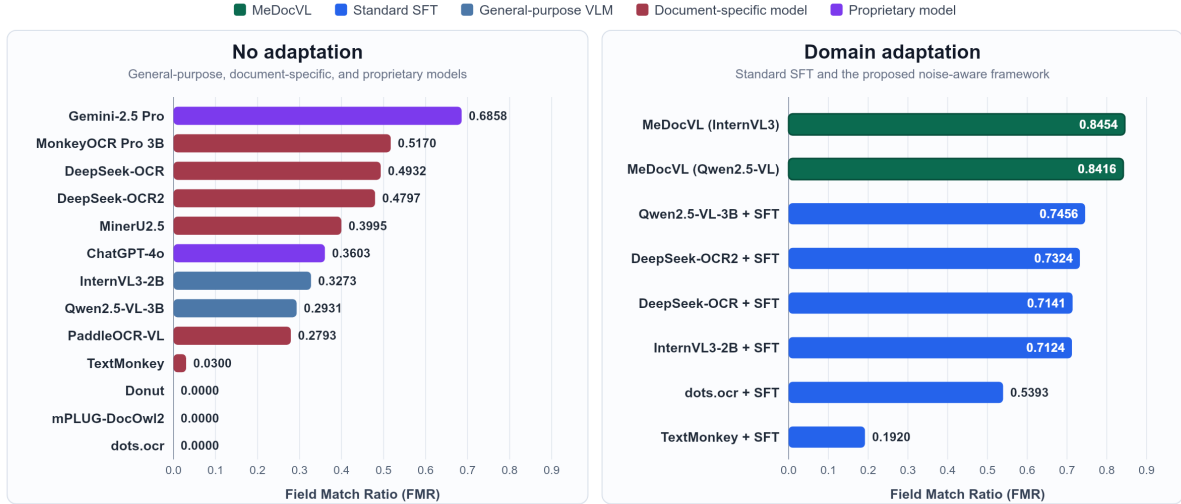


Figure 1: **Comparison of document parsing performance.** Despite strong general reasoning ability, both open-source and proprietary VLMs remain substantially behind domain-adapted models in high-precision document parsing.

consolidation stage, complemented by dynamic prompt augmentation to further improve robustness.

The proposed approach is a model-agnostic post-training framework applicable to a wide range of multimodal backbones. For convenience, we refer to VLMs adapted through this framework as **MeDocVL**. Unless otherwise specified, MeDocVL denotes an instantiation of the proposed training paradigm on a particular backbone, rather than a new standalone model. Extensive experiments demonstrate that our framework delivers large-margin improvements over the baseline, boosting FMR by over 50 percentage points.

Our main contributions are summarized as follows:

- We study high-precision, query-driven document parsing under noisy supervision (Mathew et al., 2021; Xu et al., 2021; Wan et al., 2024) as a practical setting that combines flexible field queries with strict field-level exact matching and imperfect large-scale annotations.
- We propose a noise-aware post-training framework integrating Training-driven Label Refinement, Dynamic Token-wise GRPO, and RL-to-SFT optimization for robust and precise VLM adaptation under noisy annotation.
- We expand and re-annotate a medical invoice dataset, constructing a challenging benchmark with over 40 key-value pairs on average.

- Extensive experiments on medical invoice parsing and a standard noisy-label classification benchmark validate the framework under noisy supervision.

2 Related Work

Early document parsing systems typically decompose document understanding into a sequence of specialized modules, such as layout analysis, reading order prediction, OCR, formula recognition, and table structure recognition (Kim et al., 2022a; Xu et al., 2021; Niu et al., 2026). This modular design enables fine-grained optimization and remains effective across diverse document formats. However, cascading errors introduced at early stages often propagate downstream, and the lack of holistic semantic modeling limits robustness in complex real-world scenarios.

Large language models (LLMs) (Hui et al., 2024; Liu et al., 2024; Touvron et al., 2023) partially alleviate these issues by providing strong text understanding and reasoning capabilities. Several works incorporate LLMs to aggregate OCR outputs or perform structured extraction (Kim et al., 2022b; Blecher et al., 2024; Wei et al., 2024; Peng et al., 2022; Liu et al., 2023; Wan et al., 2024). Despite improved semantic flexibility, such approaches operate primarily on textual representations and remain sensitive to OCR and layout errors, particularly for documents with dense structures.

Document understanding has advanced through structure-aware modeling, Document VQA, key in-

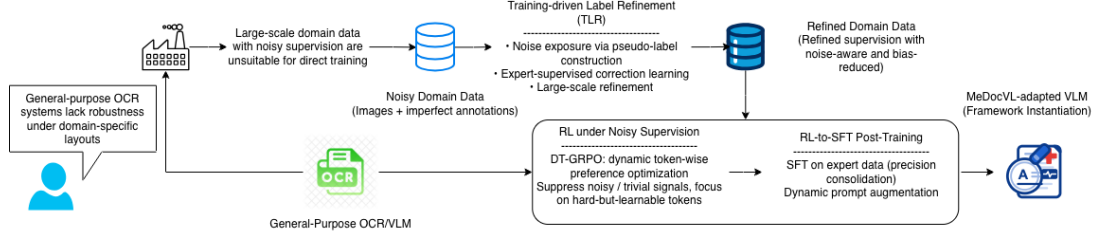


Figure 2: **Overall pipeline of the proposed framework.** Large-scale domain data are highly relevant but unsuitable for training due to imperfect annotations. MeDocVL converts raw domain data into refined supervision and adapts a general-purpose VLM into a high-precision document parsing model through noise-aware post-training.

formation extraction, OCR-free architectures, and document-specific VLMs, including LayoutLMv2, DocVQA, Donut, mPLUG-DocOwl2, TextMonkey, MonkeyOCR, dots.ocr, and olmOCR (Xu et al., 2021; Mathew et al., 2021; Kim et al., 2022a; Hu et al., 2025; Liu et al., 2026; Li et al., 2024, 2025a; Poznanski et al., 2025). These methods improve document representation, visual text recognition, or task-specific understanding. Our work addresses a complementary problem by adapting existing VLMs to high-precision domain-specific parsing when large-scale supervision is noisy.

Learning under noisy supervision has been widely studied through sample selection, co-training, label correction, semi-supervised learning, and robust loss design (Han et al., 2018; Yu et al., 2019; Patrini et al., 2017; Li et al., 2020; Liu et al., 2020; Karim et al., 2022). These studies primarily improve downstream optimization on noisy data, motivating our complementary focus on refining supervision before post-training (Section 4).

3 Task Formulation

We study high-precision key–value document parsing under noisy supervision. Our query-driven formulation is closely related to Document VQA and key information extraction. Conventional Document VQA typically answers an individual natural-language question, while our interface requests a set of fields and returns them in one structured key–value output, making it closer to key information extraction. Queried keys provide a flexible interface for instruction-following VLMs, and the complete requested structure is evaluated using field-level exact matching after normalization. Given a document image x and a queried key set $\mathcal{K} = \{k_1, \dots, k_K\}$, the goal is to generate structured key–value outputs with strict field-level exact-match accuracy.

We assume access to two datasets: a large

pseudo-labeled dataset $\mathcal{D}_u = \{(x_i, \tilde{\mathcal{Y}}_i)\}_{i=1}^N$, where annotations may contain token corruption, missing fields, or spurious entries; and a small expert-annotated dataset $\mathcal{D}_l = \{(x_j, \mathcal{Y}_j)\}_{j=1}^M$, where annotations are accurate and strictly validated, $M = |\mathcal{D}_l|$ is the number of expert-annotated samples, and $M \ll N$.

A vision–language model f_θ takes $(x, \mathcal{K})$ as input and predicts

$$\hat{\mathcal{Y}} = f_\theta(x, \mathcal{K}) = \{(\hat{k}_m, \hat{v}_m)\}_{m=1}^{M_x},$$

where each value $\hat{v}_m$ is generated autoregressively. Since generative document parsing may produce missing, spurious, or misaligned fields, the predicted key set is not guaranteed to match $\mathcal{K}$ exactly.

Our objective is to learn a robust mapping from $(x, \mathcal{K})$ to structured outputs that preserve correct key alignment and field-level exactness under imperfect supervision. Accordingly, the main contribution of this work is not a new document-understanding task, but a noise-aware post-training framework that adapts general-purpose VLMs to domain-specific query-driven parsing under imperfect supervision.

4 Training-driven Label Refinement

Large-scale document parsing datasets are commonly built from automated labeling pipelines based on OCR systems or MLLMs (Kim et al., 2022a; Xu et al., 2021; Niu et al., 2026). While such pseudo annotations provide broad coverage, they contain systematic errors, including token corruption, missing fields, key–value misalignment, and hallucinated entries (Zhou et al., 2023; Wang et al., 2023). Under strict field-level exact-match requirements, directly optimizing on these labels often yields unstable training and brittle extraction behavior (Zhu et al., 2022; Ouyang et al., 2022).

Training-driven Label Refinement (TLR) treats refinement as a learnable correction prob-

lem (Karim et al., 2022; Zhang et al., 2024). TLR follows three steps: (i) exposing systematic noise produced by the labeling pipeline, (ii) learning correction behavior from expert supervision, and (iii) applying the learned correction at scale to construct refined supervision. Additional visualization of the TLR framework is provided in Appendix A.

Unlike methods that directly make downstream optimization robust to noisy labels, TLR operates before downstream post-training and explicitly improves the supervision itself. The refinement model is trained independently to correct structured textual annotations conditioned on the document image and queried fields. The resulting refined annotations can subsequently supervise either SFT or noise-aware RL objectives.

4.1 Noise Exposure via Pseudo-label Construction

Rather than suppressing noise, TLR explicitly exposes systematic errors induced by the OCR/MLLM-based labeling pipeline (Northcutt et al., 2021; Song et al., 2022). Given an expert-annotated sample $(x_j, \mathcal{Y}_j) \in \mathcal{D}_l$, we regenerate a noisy structured pseudo annotation by applying the same OCR/VLM-based labeling pipeline used to construct $\mathcal{D}_u$: $\tilde{\mathcal{Y}}_j = \mathcal{T}(x_j)$, where $\mathcal{T}(\cdot)$ maps a document image to a noisy structured annotation. This step is not intended to approximate clean labels, but to preserve realistic error patterns observed in large-scale industrial annotation pipelines, which the correction model must learn to revise.

4.2 Correction Learning with Expert-supervised Behavior

We train a refinement model f_ϕ , initialized from a pre-trained VLM backbone and adapted via LoRA fine-tuning, to correct noisy annotations. Conditioned on the document image x_j , target key set $\mathcal{K}_j$, and pseudo annotation $\tilde{\mathcal{Y}}_j$, the model predicts a corrected structure: $\hat{\mathcal{Y}}_j = f_\phi(x_j, \mathcal{K}_j, \tilde{\mathcal{Y}}_j)$.

A key principle of TLR is that expert annotations supervise *how to correct* noisy outputs, rather than serving as large-scale supervision for downstream extraction (Pham et al., 2022; Gou et al., 2021). Accordingly, f_ϕ is optimized to map noisy but informative structures to expert-consistent outputs, instead of learning extraction from scratch.

We train f_ϕ with token-level supervision over

corrected value sequences:

$$\mathcal{L}_{\text{corr}} = \frac{1}{M} \sum_{j=1}^M \frac{1}{T_j} \sum_{t=1}^{T_j} \ell_{\text{tok}}(\hat{y}_{j,t}, y_{j,t}), \quad (1)$$

Here, $M = |\mathcal{D}_l|$ denotes the number of expert-annotated samples, T_j is the number of target tokens in sample j , and ℓ_{tok} is the token-level cross-entropy loss. This objective distills correction behavior that suppresses systematic bias while preserving valid information already present in $\tilde{\mathcal{Y}}_j$.

4.3 Large-scale Refinement for Supervision Construction

Once trained, f_ϕ defines a reusable correction function that is applied to the large-scale raw dataset. For each sample $(x_i, \tilde{\mathcal{Y}}_i) \in \mathcal{D}_u$, refined supervision is obtained as $\hat{\mathcal{Y}}_i = f_\phi(x_i, \mathcal{K}_i, \tilde{\mathcal{Y}}_i)$. We denote the refined dataset as $\hat{\mathcal{D}}_u = \{(x_i, \hat{\mathcal{Y}}_i)\}_{i=1}^N$, which serves as the supervision source for subsequent noise-aware RL.

This train-once, apply-at-scale paradigm reduces dominant systematic errors while retaining controlled residual noise that reflects realistic extraction uncertainty. By reshaping noisy annotations into a more stable supervision distribution, TLR establishes a scalable foundation for downstream noise-aware post-training.

5 Reinforcement Learning under Noisy Supervision

Although TLR substantially improves supervision quality, residual annotation noise still destabilizes strict token-level optimization in high-precision document parsing. Direct SFT assumes uniform label reliability and amplifies residual annotation errors through exact-match supervision (Zhu et al., 2022; Wang et al., 2023; Ouyang et al., 2022; Rafailov et al., 2023). This often causes overfitting to noisy supervision while under-optimizing genuinely difficult fields. We therefore introduce reinforcement learning (RL) as an intermediate optimization stage. Instead of enforcing exact correspondence to imperfect annotations, RL optimizes extraction behavior through reward-based supervision, improving robustness under residual noise (Team et al., 2025; Schulman et al., 2017; Ouyang et al., 2022).

5.1 Limitations of Sequence-level GRPO

We build upon Group Relative Policy Optimization (GRPO) (Guo et al., 2025), which stabilizes

RL training via group-relative advantage normalization. In standard GRPO, a single sequence-level advantage is uniformly applied to all tokens within a rollout. While effective for long-horizon reasoning, this formulation is ill-suited for structured document parsing, where correctness depends on localized token predictions (e.g., digits, dates, and codes).

Under residual noisy supervision, sequence-level optimization suffers from three limitations: (i) noisy and genuinely difficult fields receive indistinguishable optimization signals; (ii) trivially easy fields continue contributing gradients after convergence; and (iii) optimization capacity cannot adapt to heterogeneous token-level difficulty. These issues stem from coarse-grained credit assignment and cannot be resolved by simple reward scaling or heuristic masking.

5.2 Dynamic Token-wise GRPO (DT-GRPO)

To address these limitations, we introduce Dynamic Token-wise GRPO (DT-GRPO). DT-GRPO redistributes optimization emphasis based on historical agreement at the key-value level between rollout predictions and TLR-refined supervision, allowing the model to distinguish easy, hard, and noisy regions without explicit noise modeling (Bengio et al., 2009; Lin et al., 2017; Karim et al., 2022).

We perform RL on the TLR-refined dataset $\hat{\mathcal{D}}_u = \{(x_i, \hat{\mathcal{Y}}_i)\}_{i=1}^N$. At training epoch n , for each sample i , the current policy π_θ generates K rollout candidates

$$\{\bar{\mathcal{Y}}_i^{(n,1)}, \dots, \bar{\mathcal{Y}}_i^{(n,K)}\}.$$

5.2.1 Value-correction reward.

The primary reward evaluates field-level extraction correctness conditioned on the queried key set $\mathcal{K}_i$:

$$r_i^{(k)} = \frac{1}{|\mathcal{K}_i|} \sum_{k_m \in \mathcal{K}_i} \mathbf{1}[\bar{v}_i^{(n,k)}(k_m) = \hat{v}_i(k_m)]. \quad (2)$$

This reward directly aligns RL with field-level exact-match accuracy under refined supervision.

5.2.2 Auxiliary structural rewards.

To encourage structurally valid outputs, we incorporate auxiliary rewards, including (i) a format reward evaluating key-value layout correctness and (ii) a key-length reward measuring consistency between predicted and queried key sets. Detailed formulations are provided in the appendix. Their weighted combination, denoted by $r_{i,\text{total}}^{(k)}$, is used

for policy optimization, and the complete definition and coefficients are given in Appendix G. Besides, an ablation reported in Appendix G.7 shows that the value-correction reward alone achieves a mean FMR of 0.8173, while the complete six-component reward improves it to 0.8216, indicating that the value term provides the primary signal and the auxiliary terms offer complementary gains.

5.2.3 Dynamic token-wise weighting.

For each queried key $k_m \in \mathcal{K}_i$, we define a rollout correctness indicator:

$$I_{i,m}^{(e,k)} = \mathbf{1}[\bar{v}_i^{(e,k)}(k_m) = \hat{v}_i(k_m)]. \quad (3)$$

If a key-value pair is correctly predicted in all K rollouts at epoch n , it is treated as an easy case and excluded from further optimization. Otherwise, its importance is determined by historical agreement:

$$w_{i,m}^{(n)} = \begin{cases} 0, & \text{if } I_{i,m}^{(n,k)} = 1, \forall k, \\ \frac{\sum_{e=1}^{n-1} \sum_{k=1}^K I_{i,m}^{(e,k)}}{K(n-1)}, & \text{otherwise.} \end{cases} \quad (4)$$

Structured outputs $\bar{\mathcal{Y}}_i^{(n,k)}$ are serialized into token sequences $\bar{y}_i^{(n,k)}$. Let $\mathcal{S}_{i,m}$ denote token positions corresponding to the value span of key k_m . Token-level weights are assigned as

$$w_{i,t}^{(n)} = w_{i,m}^{(n)}, \quad \forall t \in \mathcal{S}_{i,m}. \quad (5)$$

5.2.4 Group-relative advantage.

Rollout advantages are computed via group-wise normalization:

$$A_i^{(k)} = \frac{r_{i,\text{total}}^{(k)} - \text{mean}_k(r_{i,\text{total}}^{(k)})}{\text{std}_k(r_{i,\text{total}}^{(k)}) + \epsilon}, \quad (6)$$

where ϵ ensures numerical stability.

5.2.5 DT-GRPO objective.

$$\mathcal{L}_{\text{DT-GRPO}} = -\mathbb{E}_{i,k} \left[A_i^{(k)} \sum_{t=1}^{T_i} w_{i,t}^{(n)} \frac{\pi_\theta(\bar{y}_{i,t}^{(n,k)} \mid x_i, \bar{y}_{i,<t}^{(n,k)})}{\pi_{\text{old}}(\bar{y}_{i,t}^{(n,k)} \mid x_i, \bar{y}_{i,<t}^{(n,k)})} \right]. \quad (7)$$

This expression highlights the proposed token-wise weighting. Standard GRPO ratio clipping and reference-policy KL regularization are retained unchanged in implementation and omitted here for clarity.

6 RL-to-SFT Post-training

After reinforcement learning under noisy supervision, the model reaches a behavior-aligned intermediate state. At this stage, the policy has learned robust extraction behavior from large-scale refined data and is tolerant to residual annotation noise, but token-level precision has not yet been fully consolidated. Additional visualization of the RL-to-SFT pipeline is provided in Appendix B.

6.1 Why RL before SFT

SFT imposes hard token-level constraints and can amplify residual annotation errors when applied before RL (Ouyang et al., 2022; Zhou et al., 2023; Guo et al., 2025; Rafailov et al., 2023). In contrast, RL first establishes noise-tolerant extraction behavior, after which SFT on expert data consolidates exact-match precision. The ablation in Table 5 verifies this ordering.

6.2 Precision consolidation via SFT

Following RL, we perform a final SFT stage using the expert-annotated dataset $\mathcal{D}_l$. This stage serves solely as precision consolidation, sharpening token-level accuracy, reducing variance, and calibrating outputs toward exact-match requirements (Zhou et al., 2023). To avoid reintroducing noise, SFT operates exclusively on curated expert data. All expert samples are jointly used in a single SFT stage to ensure balanced performance across heterogeneous document formats.

6.3 Dynamic prompt-based precision augmentation

Hard supervision during SFT may cause overfitting to prompt patterns (Zhou et al., 2023; Wei et al., 2022). To mitigate this risk, we introduce dynamic prompt-based precision augmentation (Dyn-Prompt) during SFT. Augmentation is performed at the instruction level, consistent with the key-conditioned extraction formulation. Specifically, queried keys are randomly removed or reordered within prompts, with targets adjusted accordingly. This regularizes hard supervision, encourages independent visual verification for each field, and further reduces hallucinated outputs when combined with RL.

7 Experiment

We conduct extensive experiments to evaluate the effectiveness of the proposed framework for high-

precision document parsing.

7.1 Experiment Settings

7.1.1 Dataset and Training Splits.

We conduct experiments on a public medical invoice dataset released by Alibaba Cloud (Alibaba Tianchi, 2023), which contains 950 real-world invoices (800 for training and 150 for testing) collected from diverse practical scenarios. Representative dataset samples are provided in Appendix C.

To better reflect real-world document parsing requirements, we substantially expand the structured annotations (Mathew et al., 2021; Ouyang et al., 2025). Redundant fields are consolidated, and additional fine-grained fields commonly required in medical invoice processing are introduced. As a result, the 800 training documents contain 36,800 annotated key-value pairs in total, corresponding to 46 pairs per document on average, compared with 16,342 pairs before expansion. Since the task is query-driven, each training or evaluation instance corresponds to an arbitrary subset of fields, yielding a combinatorially large effective query space even for a fixed document image (Mathew et al., 2021; Wan et al., 2024). All methods are evaluated using the same expanded annotations to ensure fair comparison.

Although all documents are manually annotated with high-quality ground truth, we construct a realistic noisy supervision setting for training. Specifically, the 800 training documents are split into 700 pseudo-labeled samples $\mathcal{D}_u$ and 100 expert-annotated samples $\mathcal{D}_l$. Applying Training-driven Label Refinement (TLR) to $\mathcal{D}_u$ produces the refined-label dataset $\hat{\mathcal{D}}_u$, which is used consistently in experiments involving noisy supervision. The held-out test set of 150 documents is never used during training or refinement.

7.1.2 Metrics:

All models are evaluated under a unified key-conditioned extraction setting. Given a document image and a queried key set, the model is required to generate structured key-value outputs conditioned on the prompt. We report a task-specific Field Match Ratio (FMR), defined in this work as the proportion of queried fields whose keys are correctly aligned and whose normalized values exactly match the references. Unlike field-level F1 or ANLS, FMR uses strict exact matching after normalization.

Table 1: Comparison under different adaptation settings on medical invoice parsing.

Method	FMR↑
<i>No adaptation: general-purpose VLMs</i>	
Qwen2.5-VL-3B (Bai et al., 2025)	0.2931
InternVL3-2B (Zhu et al., 2025)	0.3273
<i>No adaptation: document-specific models</i>	
DeepSeek-OCR (Wei et al., 2025)	0.4932
DeepSeek-OCR2 (Wei et al., 2026)	0.4797
PaddleOCR-VL (Cui et al., 2025)	0.2793
MonkeyOCR Pro 3B (Li et al., 2024)	0.5170
MinerU2.5 (Niu et al., 2026)	0.3995
Donut (Kim et al., 2022a)	0.0000
mPLUG-DocOwl2 (Hu et al., 2025)	0.0000
TextMonkey (Liu et al., 2026)	0.0300
dots.ocr (Li et al., 2025a)	0.0000
<i>Proprietary models</i>	
ChatGPT-4o (OpenAI, 2024)	0.3603
Gemini-2.5 Pro (Google, 2025)	0.6858
<i>Standard domain adaptation</i>	
Qwen2.5-VL-3B + SFT	0.7456
InternVL3-2B + SFT	0.7124
DeepSeek-OCR + SFT	0.7141
DeepSeek-OCR2 + SFT	0.7324
TextMonkey + SFT	0.1920
dots.ocr + SFT	0.5393
<i>Noise-aware domain adaptation</i>	
MeDocVL (Qwen2.5-VL)	0.8416
MeDocVL (InternVL3)	0.8454

Formally, FMR is defined as

$$\text{FMR} = \frac{\sum_{d \in D} \sum_{k \in \mathcal{K}_d} \mathbf{1} \left[\phi(\hat{v}_{d,k}) = \phi(v_{d,k}) \wedge \hat{k}_d = k \right]}{\sum_{d \in D} |\mathcal{K}_d|}. \quad (8)$$

Before evaluation, each generated JSON or structured response is parsed to extract the textual value associated with each queried key. The function $\phi(\cdot)$ applies the same field-specific normalization to predictions and references. Text values are lowercased and stripped of leading and trailing whitespace, dates are converted to a unified canonical format, and monetary amounts are normalized by removing currency symbols, thousand separators, and insignificant trailing zeros. These output conventions are also specified in the post-training prompts. Exact matching is performed on the normalized values, while missing keys and unparsable responses are counted as incorrect.

7.2 Comparison with the SOTA Models

Table 1 compares general-purpose and document-specific models under no adaptation, standard SFT, and noise-aware adaptation using the same query-driven protocol and expanded annotations. The near-zero zero-shot FMR of several task-specific models reflects limited compliance with the required complete and canonically formatted key-value output, rather than an inability to recognize document content.

Table 2: Comparison of RL objectives on document parsing using the refined-label dataset $\hat{\mathcal{D}}_u$. Results are reported as mean $\pm$ standard deviation over three independent runs.

RL Algorithm	FMR↑ (Mean $\pm$ Std)
Vanilla GRPO (Guo et al., 2025)	0.8019 $\pm$ 0.0006
DAPO (Yu et al., 2026)	0.8205 $\pm$ 0.0004
GSPO (Zheng et al., 2025)	0.8192 $\pm$ 0.0005
SAPO (Gao et al., 2025)	0.8198 $\pm$ 0.0005
DT-GRPO	0.8216 $\pm$ 0.0004

Table 3: Impact of annotation noise on post-training and the effectiveness of TLR.

Training Data	FMR↑
20% Noisy Data	0.7662
30% Noisy Data	0.7296
50% Noisy Data	0.5820
70% Noisy Data	0.2601
Refined Data (from 30% noise)	0.7833

Standard SFT on the same expert subset improves domain adaptation, but remains below the proposed framework. MeDocVL achieves the strongest results on both Qwen2.5-VL and InternVL3 backbones, supporting the role of noise-aware post-training rather than a particular document architecture. A controlled all-clean comparison in Table 5 further separates the effects of post-training strategy and supervision quality. Unless otherwise specified, Qwen2.5-VL is used as the default backbone in subsequent experiments.

7.3 Comparison with Different RL Algorithms

Table 2 compares DT-GRPO with representative RL objectives under the same training configuration on $\hat{\mathcal{D}}_u$. To evaluate training stability, we repeat every method using three independent random seeds. The complete per-seed results are provided in Appendix J. DT-GRPO achieves the best result in all three runs while maintaining low variance. Although the numerical differences among modern RL objectives are modest, the consistent ranking indicates that dynamic token-wise optimization provides a stable improvement under noisy supervision. More importantly, the subsequent RL-to-SFT strategy amplifies the gains of noise-aware optimization, indicating that the overall post-training paradigm is more consequential than the choice of RL objective alone.

7.4 Noisy Effectiveness Analysis

This experiment investigates the impact of annotation noise on SFT and evaluates the effective-

Table 4: Accuracy (%) on CIFAR-10 with 50% symmetric label noise.

Method	Acc.	Method	Acc.
CE	79.4	DivideMix	94.6
Co-teaching+	85.7	ELR+	94.8
F-correction	79.8	MOIT	90.0
PENCIL	89.1	SeICL+	93.9
LossModelling	92.0	TCL	93.9
CLIPCleaner	95.7	TLR+DT-GRPO	95.9

ness of Training-driven Label Refinement (TLR) in mitigating noise-induced performance degradation. Starting from the same foundation model, we inject controlled noise into the pseudo-labeled training set $\mathcal{D}_u$ by corrupting a fixed proportion of key-value labels, with noise ratios set to 20%, 30%, 50%, and 70%. All four settings quantify the sensitivity of direct SFT to supervision noise, and the 30% setting is additionally selected for the TLR refinement experiment as a representative moderate-noise regime. Construction and data-separation details are provided in Appendix E.

As shown in Table 3, model SFT on noisy data degrades rapidly as annotation noise increases. Even moderate noise (30%) leads to a clear drop in FMR, while higher noise levels cause severe performance collapse, highlighting the brittleness of SFT under strict exact-match supervision. Applying TLR to the 30% noisy dataset substantially recovers performance and achieves an FMR of 0.7833, outperforming all models trained directly on noisy annotations. This demonstrates that TLR effectively suppresses systematic annotation errors and provides more reliable supervision for downstream optimization.

7.5 Comparison on a Standard Noisy-label Benchmark

To test whether our noise-aware framework transfers beyond document parsing, we compare it with established noisy-label methods on CIFAR-10 with 50% symmetric label noise following CLIP-Cleaner (Feng et al., 2024). Implementation and result-source details are provided in Appendix E.

As shown in Table 4, TLR+DT-GRPO achieves 95.9% accuracy, compared with 95.7% for CLIP-Cleaner and 94.8% for ELR+. This provides framework-level cross-task evidence, rather than an isolated evaluation of TLR or a strictly supervision-matched comparison, because our complete framework uses limited clean correction supervision.

Table 5: Ablation of supervision quality and post-training strategy. The realistic setting combines limited expert annotations with refined pseudo labels, while all-clean supervision uses expert annotations for all samples. Results are single-run, with multi-seed RL statistics in Table 2.

Training Configuration	FMR $\uparrow$
<i>No domain adaptation</i>	
Baseline (no post-training)	0.2931
<i>Realistic supervision: single-stage ablations</i>	
SFT on $\mathcal{D}_u$ (pseudo labels)	0.7290
SFT on $\mathcal{D}_l$ (expert labels)	0.7530
SFT on $\mathcal{D}_u + \mathcal{D}_l$	0.7662
SFT on $\hat{\mathcal{D}}_u$ (refined labels)	0.7833
Vanilla GRPO	0.8017
DT-GRPO	0.8214
<i>Realistic supervision: training-order ablations</i>	
SFT $\rightarrow$ SFT	0.7952
SFT $\rightarrow$ Vanilla GRPO	0.8028
SFT $\rightarrow$ DT-GRPO	0.8124
Vanilla GRPO $\rightarrow$ SFT	0.8125
TLR $\rightarrow$ DT-GRPO $\rightarrow$ SFT	0.8384
TLR $\rightarrow$ DT-GRPO $\rightarrow$ SFT + DynPrompt	0.8416
<i>All-clean supervision: controlled comparison</i>	
SFT	0.8291
SFT $\rightarrow$ DT-GRPO	0.8403
DT-GRPO $\rightarrow$ SFT	0.8582
DT-GRPO $\rightarrow$ SFT + DynPrompt	0.8617

7.6 Ablation Study

Table 5 disentangles the effects of supervision quality, post-training objective, and training order. The realistic pipeline uses 700 TLR-refined samples for RL and 100 expert samples for SFT, while the all-clean setting uses expert annotations for all 800 samples. All comparisons use matched training and inference configurations, and DynPrompt is applied only where marked.

7.6.1 Baseline and Single-Stage Training.

The baseline model achieves only 0.2931 FMR, showing that foundation VLMs are insufficient for high-precision document parsing without domain adaptation. Single-stage SFT on pseudo-labeled data $\mathcal{D}_u$ improves performance but remains sensitive to annotation noise. Using expert-labeled data $\mathcal{D}_l$ yields better results, while jointly training on $\mathcal{D}_u + \mathcal{D}_l$ further improves performance by combining large-scale supervision with clean annotations.

Applying SFT on refined labels $\hat{\mathcal{D}}_u$ achieves additional gains, confirming that TLR effectively improves supervision quality. Under the same realistic supervision setting, RL on refined labels further improves over the corresponding SFT variants, and DT-GRPO surpasses vanilla GRPO by providing

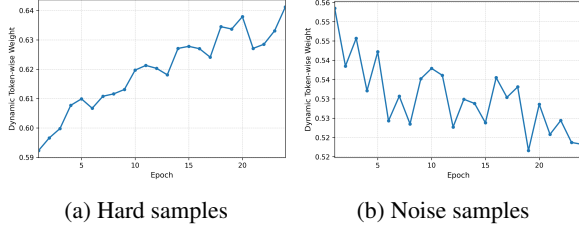


Figure 3: Trend of dynamic token-wise weights in DT-GRPO for hard and noisy samples.

more robust optimization under residual noise.

7.6.2 Training Order under Realistic Supervision.

The realistic multi-stage block compares different training orders using refined noisy supervision followed or preceded by limited expert supervision. Methods that apply RL before SFT consistently outperform those that apply SFT first, indicating that RL better tolerates residual annotation noise before strict token-level supervision is enforced.

Replacing vanilla GRPO with DT-GRPO further improves performance across different training orders, showing that token-wise optimization remains beneficial even after supervised initialization. These results support using RL as the primary behavior alignment stage, followed by SFT for precision consolidation.

7.6.3 Controlled Comparison with All-Clean Supervision.

The all-clean SFT result of 0.8291 is a single-stage baseline rather than an upper bound. With supervision fixed, DT-GRPO $\rightarrow$ SFT reaches 0.8582, outperforming SFT $\rightarrow$ DT-GRPO at 0.8403. Dyn-Prompt provides modest gains in both the realistic setting, from 0.8384 to 0.8416, and the all-clean setting, from 0.8582 to 0.8617. The realistic complete pipeline reaches 0.8416 using only 100 expert-annotated samples, leaving a 0.0201 gap to the all-clean complete pipeline. Its advantage over all-clean single-stage SFT therefore reflects the stronger post-training strategy rather than an advantage of noisy supervision.

7.7 Effect of DT-GRPO.

Using the controlled corruption from Section 7.4, we categorize supervision at the key-value level through comparison with ground-truth annotations. Injected corruptions are treated as noisy samples, while the remaining incorrect-but-consistent predictions are treated as hard samples.

Figure 3 visualizes the evolution of dynamic token-wise weights in Equation 4 during training. For hard samples in Figure 3a, token weights increase steadily over epochs, indicating that DT-GRPO progressively allocates more optimization capacity to difficult yet learnable predictions. In contrast, tokens associated with noisy supervision exhibit persistently low or oscillatory weights in Figure 3b, reflecting suppressed gradient contribution due to inconsistent historical agreement.

These dynamics demonstrate that DT-GRPO implicitly distinguishes between hard and noisy labels through historical rollout behavior, without explicit noise modeling or heuristic filtering. By decreasing gradients from noisy tokens while focusing optimization on hard cases, DT-GRPO stabilizes training and improves field-level exact-match performance under noisy supervision.

8 Conclusion

This work studies high-precision document parsing under realistic noisy supervision and presents a noise-aware post-training framework for adapting general-purpose VLMs without relying on large-scale clean annotations. By treating supervision noise as a learnable factor, MeDocVL refines imperfect labels and decouples robustness-oriented behavior alignment from precision consolidation, enabling stable training under strict field-level exactness constraints. Beyond medical-invoice parsing, results on noisy-label image classification provide complementary evidence that the framework’s core noise-aware training principle transfers across tasks.

Limitations

While the proposed framework demonstrates strong performance under noisy supervision, several limitations remain. First, our end-to-end document parsing experiments are limited to medical invoices and 2B–3B VLM backbones. Although the CIFAR-10 experiment provides cross-task evidence for the core noise-aware principle, it does not establish generalization to other document domains, including long-form reports, handwritten forms, multilingual documents, or multi-page documents. Future work should evaluate the complete framework on more diverse document parsing and DocVQA-style benchmarks. Second, the proposed Training-driven Label Refinement (TLR) relies on the availability of a small amount of high-quality expert annota-

tions to learn correction behavior. In scenarios where expert annotations are extremely limited or unavailable, the effectiveness of refinement may degrade. Third, DT-GRPO introduces additional reinforcement learning overhead compared with standard supervised fine-tuning. Although the method improves robustness under noisy supervision, RL-based optimization increases training complexity and computational cost, particularly for large-scale VLM backbones. Finally, our work focuses primarily on improving field-level exact-match accuracy under noisy supervision. Broader aspects of document understanding, such as reasoning across multiple pages, table structure reconstruction, or long-context semantic consistency, are not explicitly addressed in the current framework.

Acknowledgements

We would like to sincerely thank Wenfeng Xie, Ziling Lin, Zengjian Fan, De Shi, Bin Pei, Jinping Liu, and Lin Huang for their valuable guidance and support throughout this work, including their contributions to research direction and problem formulation, solution design, computational resource support, and valuable feedback on practical challenges encountered in real-world applications. We greatly appreciate their insights, encouragement, and continuous support throughout the research process.

References

- Alibaba Tianchi. 2023. Task for ocr key-element extraction of medical invoices and itemized bills. <https://tianchi.aliyun.com/dataset/131815>. Accessed: 2026-01-29.
- Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. 2019. Unsupervised label noise modeling and loss correction. In *International conference on machine learning*, pages 312–321. PMLR.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.
- Darwin Bautista and Rowel Atienza. 2022. Scene text recognition with permuted autoregressive sequence models. In *European Conference on Computer Vision (ECCV)*, pages 178–196. Springer.
- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.
- Lukas Blecher, Guillem Cucurull Preixens, Thomas Scialom, and Robert Stojnic. 2024. Nougat: Neural optical understanding for academic documents. In *International Conference on Learning Representations*, volume 2024, pages 37646–37663.
- Cheng Cui, Ting Sun, Suyin Liang, Tingquan Gao, Zelun Zhang, Jiaxuan Liu, Xueqing Wang, Changda Zhou, Hongen Liu, Manhui Lin, and 1 others. 2025. Paddleocr-vl: Boosting multilingual document parsing via a 0.9 b ultra-compact vision-language model. *arXiv preprint arXiv:2510.14528*.
- Lei Cui, Yiheng Xu, Tengchao Lv, and Furu Wei. 2021. Document ai: Benchmarks, models and applications. *arXiv preprint arXiv:2111.08609*.
- Chen Feng, Georgios Tzimiropoulos, and Ioannis Patras. 2024. Clipcleaner: Cleaning noisy labels with clip. In *Proceedings of the 32nd ACM international conference on multimedia*, pages 876–885.
- Chang Gao, Chujie Zheng, Xiong-Hui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. 2025. Soft adaptive policy optimization. *arXiv preprint arXiv:2511.20347*.
- Google. 2025. Gemini 2.5 pro. <https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-5-pro>. Accessed: 2026-03-03.
- Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International journal of computer vision*, 129(6):1789–1819.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.
- Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31.
- Anwen Hu, Haiyang Xu, Liang Zhang, Jiabo Ye, Ming Yan, Ji Zhang, Qin Jin, Fei Huang, and Jingren Zhou. 2025. mplug-docowl2: High-resolution compressing for ocr-free multi-page document understanding. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5817–5834.
- Zhizhong Huang, Junping Zhang, and Hongming Shan. 2023. Twin contrastive learning with noisy labels. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11661–11670. IEEE.

- Binyuan Hui, Jian Yang, Zeyu Cui, Jiayi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, and 1 others. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.
- Nazmul Karim, Mamshad Nayeem Rizve, Nazanin Rahnavard, Ajmal Mian, and Mubarak Shah. 2022. Unicon: Combating label noise through uniform selection and contrastive learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9666–9676. IEEE.
- Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022a. Ocr-free document understanding transformer. In *European Conference on Computer Vision*, pages 498–517. Springer.
- Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022b. Ocr-free document understanding transformer. In *European Conference on Computer Vision*, pages 498–517. Springer.
- Junnan Li, Richard Socher, and Steven CH Hoi. 2020. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*.
- Shikun Li, Xiaobo Xia, Shiming Ge, and Tongliang Liu. 2022. Selective-supervised contrastive learning with noisy labels. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 316–325. IEEE.
- Yumeng Li, Guang Yang, Hao Liu, Bowen Wang, and Colin Zhang. 2025a. dots. ocr: Multilingual document layout parsing in a single vision-language model. *arXiv preprint arXiv:2512.02498*.
- Zhang Li, Yuliang Liu, Qiang Liu, Zhiyin Ma, Ziyang Zhang, Shuo Zhang, Biao Yang, Zidun Guo, Jiarui Zhang, Xinyu Wang, and 1 others. 2025b. Monkeyocr: Document parsing with a structure-recognition-relation triplet paradigm. *arXiv preprint arXiv:2506.05218*.
- Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. 2024. Monkey: Image resolution and text label are important things for large multi-modal models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26753–26763. IEEE.
- Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.
- Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. 2020. Early-learning regularization prevents memorization of noisy labels. *Advances in neural information processing systems*, 33:20331–20342.
- Yuliang Liu, Biao Yang, Qiang Liu, Zhang Li, Zhiyin Ma, Shuo Zhang, and Xiang Bai. 2026. Textmonkey: An ocr-free large multimodal model for understanding document. *IEEE transactions on pattern analysis and machine intelligence*.
- Yuliang Liu, Jiaxin Zhang, Dezhi Peng, Mingxin Huang, Xinyu Wang, Jingqun Tang, Can Huang, Dahua Lin, Chunhua Shen, Xiang Bai, and 1 others. 2023. Spts v2: single-point scene text spotting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):15665–15679.
- Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2199–2208. IEEE.
- Mor Shpigel Nacson, Aviad Aberdam, Roy Ganz, Elad Ben Avraham, Alona Golts, Yair Kittenplon, Shai Mazor, and Ron Litman. 2025. Docvlm: Make your vlm an efficient reader. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 29005–29015.
- Junbo Niu, Zheng Liu, Zhuangcheng Gu, Bin Wang, Linke Ouyang, Zhiyuan Zhao, Tao Chu, Tianyao He, Fan Wu, Qintong Zhang, and 1 others. 2026. Mineru2. 5: A decoupled vision-language model for efficient high-resolution document parsing. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL 2026)*, pages 13–42.
- Curtis Northcutt, Lu Jiang, and Isaac Chuang. 2021. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411.
- OpenAI. 2024. Hello GPT-4o. <https://openai.com/index/hello-gpt-4o/>. Accessed: 2026-03-03.
- Diego Ortego, Eric Arazo, Paul Albert, Noel E. O’Connor, and Kevin McGuinness. 2021. Multi-objective interpolation training for robustness to label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6606–6615.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

- Long Ouyang and 1 others. 2025. Omnidocbench: Benchmarking diverse pdf document parsing with comprehensive annotations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1944–1952.
- Dezhi Peng, Xinyu Wang, Yuliang Liu, Jiaxin Zhang, Mingxin Huang, Songxuan Lai, Jing Li, Shenggao Zhu, Dahua Lin, Chunhua Shen, and 1 others. 2022. Spts: single-point text spotting. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4272–4281.
- Minh Pham, Minsu Cho, Ameya Joshi, and Chinmay Hegde. 2022. Revisiting self-distillation. *arXiv preprint arXiv:2206.08491*.
- 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*.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.
- Miao Rang, Zhenni Bi, Chuanjian Liu, Yunhe Wang, and Kai Han. 2023. An empirical study of scaling law for ocr. *arXiv preprint arXiv:2401.00028*.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2022. Learning from noisy labels with deep neural networks: A survey. *IEEE transactions on neural networks and learning systems*, 34(11):8135–8153.
- 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*.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Jianqiang Wan, Sibong Song, Wenwen Yu, Yuliang Liu, Wenqing Cheng, Fei Huang, Xiang Bai, Cong Yao, and Zhibo Yang. 2024. Omniparser: A unified framework for text spotting key information extraction and table recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15641–15653.
- Song Wang, Zhen Tan, Ruocheng Guo, and Jundong Li. 2023. Noise-robust fine-tuning of pretrained language models via external guidance. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12528–12540.
- Haoran Wei, Chenglong Liu, Jinyue Chen, Jia Wang, Lingyu Kong, Yanming Xu, Zheng Ge, Liang Zhao, Jianjian Sun, Yuang Peng, and 1 others. 2024. General ocr theory: Towards ocr-2.0 via a unified end-to-end model. *arXiv preprint arXiv:2409.01704*.
- Haoran Wei, Yaofeng Sun, and Yukun Li. 2025. Deepseek-ocr: Contexts optical compression. *arXiv preprint arXiv:2510.18234*.
- Haoran Wei, Yaofeng Sun, and Yukun Li. 2026. Deepseek-ocr 2: Visual causal flow. *arXiv preprint arXiv:2601.20552*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. 2021. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Kun Yi and Jianxin Wu. 2019. Probabilistic end-to-end noise correction for learning with noisy labels. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7010–7018. IEEE.
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2026. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244.
- Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. 2019. How does disagreement help generalization against label corruption? In *International Conference on Machine Learning*, pages 7164–7173.
- Jingfeng Zhang, Bo Song, Haohan Wang, Bo Han, Tongliang Liu, Lei Liu, and Masashi Sugiyama. 2024. Badlabel: A robust perspective on evaluating

and enhancing label-noise learning. *IEEE transactions on pattern analysis and machine intelligence*, 46(6):4398–4409.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. 2025. [Group sequence policy optimization](#). *Preprint*, arXiv:2507.18071.

Chunting Zhou and 1 others. 2023. Lima: Less is more for alignment. In *NeurIPS*.

Dawei Zhu, Michael A Hedderich, Fangzhou Zhai, David Ifeoluwa Adelani, and Dietrich Klakow. 2022. Is bert robust to label noise? a study on learning with noisy labels in text classification. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pages 62–67.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, and 1 others. 2025. Internv13: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

A Training-driven Label Refinement Visualization

Figure 4 illustrates the overall Training-driven Label Refinement (TLR) framework. TLR formulates noisy annotation refinement as a learnable correction problem. Expert annotations supervise correction behavior rather than direct large-scale extraction learning. The trained refinement model is subsequently applied to large-scale pseudo-labeled data to produce refined supervision with reduced systematic bias and controlled residual noise.

B RL-to-SFT Post-training Visualization

Figure 5 presents the RL-to-SFT post-training strategy used in MeDocVL. DT-GRPO first aligns extraction behavior under residual noisy supervision using refined labels, while the subsequent SFT stage consolidates exact-match precision using curated expert annotations. Dynamic prompt augmentation is further applied during SFT to improve robustness against prompt dependency and hallucinated outputs.

C Representative Dataset Samples

Figure 6 shows representative samples from the medical invoice dataset used in experiments. The dataset contains diverse layouts, dense structured fields, domain-specific terminology, handwritten elements, and visually complex formatting, making

strict field-level exact-match parsing particularly challenging.

D Notation Summary

We summarize the main symbols and notation used throughout the paper.

- $\mathcal{D}_u = \{(x_i, \tilde{\mathcal{Y}}_i)\}_{i=1}^N$: large pseudo-labeled dataset, where $\mathcal{Y}_i$ is produced by OCR or multimodal models and may contain noisy, missing, or spurious annotations.
- $\mathcal{D}_l = \{(x_j, \mathcal{Y}_j)\}_{j=1}^M$: small expert-annotated dataset comprising $M = |\mathcal{D}_l|$ accurately validated samples, where $M \ll N$.
- x : model input, consisting of a document image and a structured instruction prompt with queried keys.
- $\mathcal{K} = \{k_1, \dots, k_K\}$: queried key set for a given document.
- $\mathcal{Y} = \{(k_n, v_n)\}$: ground-truth structured annotation, where each k_n is a target field and v_n is its corresponding value sequence.
- $\tilde{\mathcal{Y}}$: noisy pseudo annotation produced by the automatic labeling pipeline.
- $\hat{\mathcal{Y}}$: refined structured annotation produced by Training-driven Label Refinement (TLR).
- $\mathcal{T}(\cdot)$: automatic OCR/VLM-based labeling function that maps a document image to a noisy structured annotation.
- f_ϕ : refinement model in TLR, which takes a document image, queried keys, and noisy annotation as input, and predicts a corrected structured annotation.
- $\hat{\mathcal{D}}_u = \{(x_i, \hat{\mathcal{Y}}_i)\}_{i=1}^N$: refined training dataset obtained by applying TLR to $\mathcal{D}_u$.
- f_θ : key-conditioned document parsing model, which maps $(x, \mathcal{K})$ to structured key-value predictions.
- π_θ : policy model used during reinforcement learning.
- π_{old} : behavior policy used to generate rollout samples in GRPO/DT-GRPO training.
- $\bar{\mathcal{Y}}_i^{(n,k)}$: the k -th sampled structured output for sample i at training epoch n .

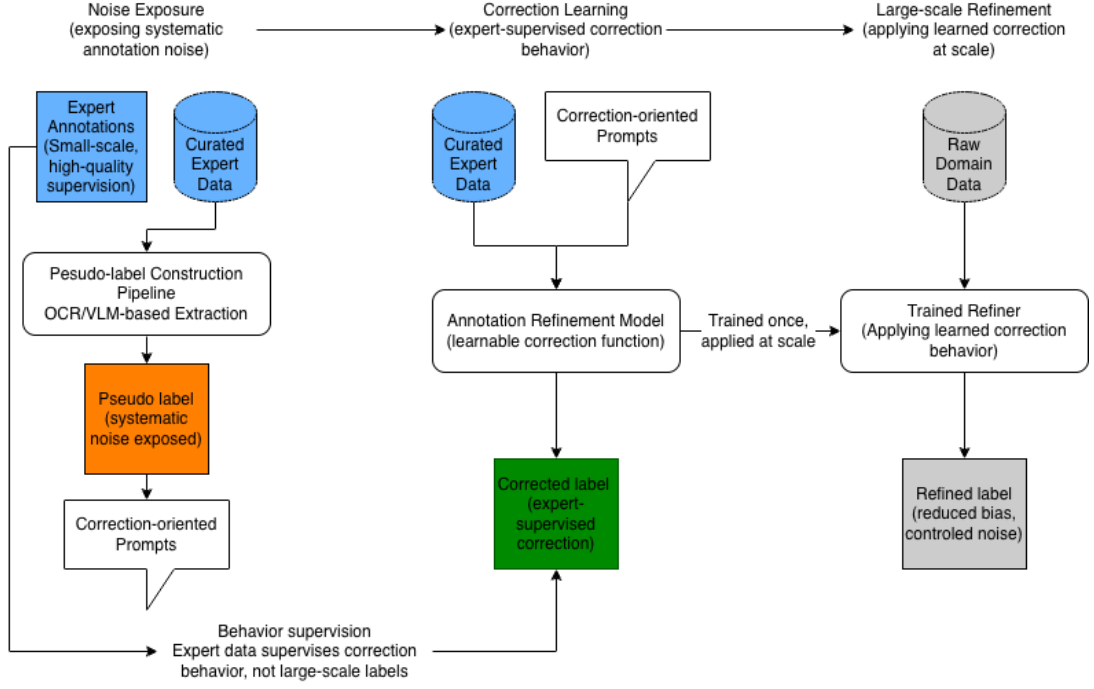


Figure 4: **Training-driven Label Refinement (TLR)**. Expert annotations supervise correction behavior, while the trained refiner is applied at scale to produce refined supervision with reduced bias and controlled residual noise.

- $\bar{v}_i^{(n,k)}(k_m)$: predicted value of queried key k_m in the k -th sampled output.
- $\hat{v}_i(k_m)$: refined target value of queried key k_m used for reward computation.
- $r_i^{(k)}$: sequence-level reward of the k -th sampled output for sample i .
- $A_i^{(k)}$: group-relative advantage of the k -th sampled output.
- $I_{i,m}^{(e,k)}$: correctness indicator of queried key k_m in the k -th rollout at epoch e .
- $w_{i,m}^{(n)}$: dynamic weight assigned to queried key k_m for sample i at epoch n .
- $\mathcal{S}_{i,m}$: token positions corresponding to the value span of queried key k_m in the serialized output.
- $w_{i,t}^{(n)}$: dynamic token-wise weight assigned to token position t at epoch n .
- $\rho_{i,t}^{(k)}(\theta)$: token-level probability ratio between the current policy and the old policy for the k -th sampled output.
- $\mathcal{C}_i$: set of clean token positions in the target sequence of sample i .
- $\mathcal{N}_i$: set of noisy token positions in the target sequence of sample i .
- $\mathcal{L}_{\text{corr}}$: correction learning objective used to train the TLR refinement model.
- $\mathcal{L}_{\text{SFT}}$: supervised fine-tuning objective.
- $\mathcal{L}_{\text{DT-GRPO}}$: dynamic token-wise GRPO objective used in the reinforcement learning stage.

E Implementation Details

We provide detailed implementation settings for reproducibility. All symbols follow the notation defined in Appendix D.

E.1 Backbone:

The key-conditioned document parsing model f_θ is initialized from general-purpose vision-language backbones. We use **Qwen2.5-VL-3B-Instruct** and **InternVL3-2B** as the primary backbones. Unless otherwise specified, both backbones share identical training protocols.

E.2 Document-specific baselines:

Donut, mPLUG-DocOwl2, TextMonkey, and dots.ocr are evaluated using their official checkpoints under the same query-driven key-value extraction protocol. TextMonkey and dots.ocr are

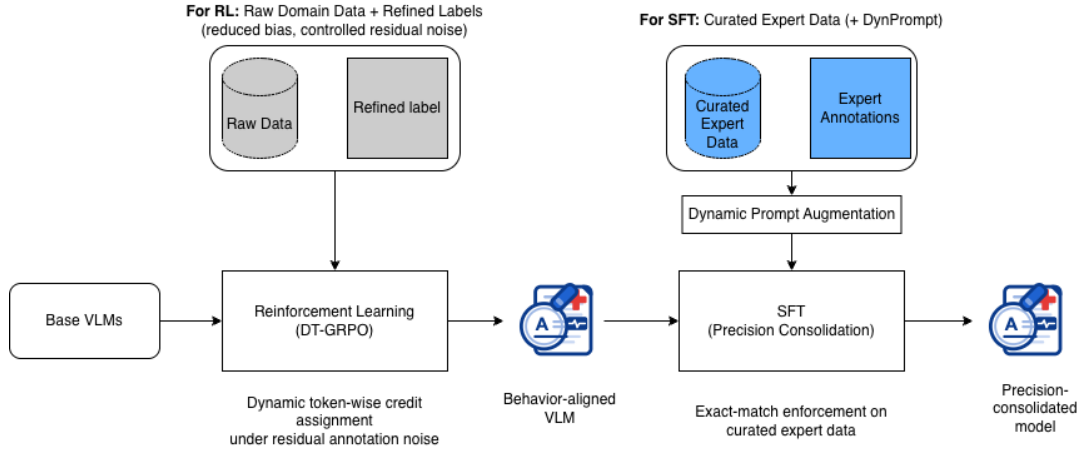


Figure 5: **RL-to-SFT post-training pipeline.** RL (DT-GRPO) aligns model behavior under residual annotation noise using raw domain data and refined labels. SFT consolidates exact-match precision using curated expert data with dynamic prompt augmentation.



Figure 6: **Representative samples from the medical invoice dataset.** The invoices exhibit diverse layouts, domain-specific terminology, mixed text styles, and complex visual elements, posing challenges for accurate document parsing.

additionally fine-tuned using their official training recipes on the same 100-sample expert subset $\mathcal{D}_l$ used for SFT in our framework. Outputs from OCR-based systems are converted to structured JSON and parsed by a fixed Qwen3-8B model using the same prompt configuration. All outputs are evaluated using the same response parsing, normalization, and FMR implementation.

E.3 Controlled-noise construction:

Noise is applied to annotations rather than images, and corrupted values are replaced with plausible but incorrect alternatives. Clean annotations corresponding to $\mathcal{D}_u$ are withheld from model training and used only to construct and analyze the controlled-noise experiment. TLR correction supervision is provided exclusively by the disjoint expert subset $\mathcal{D}_l$.

E.4 Input resolution and token budget:

Document images are resized with aspect ratio preserved. Visual inputs are controlled via pixel constraints, where the maximum number of pixels is 1,003,520 and the minimum is 627,200. The maxi-

mum text sequence length is 2,048 tokens and the maximum generation length is 1,024 tokens.

E.5 Parameter-efficient fine-tuning:

We adopt LoRA for all training stages. LoRA is applied to all linear layers with rank $r = 8$ and scaling factor $\alpha = 32$. The vision encoder and modality aligner are frozen during fine-tuning. Gradient checkpointing is enabled for memory efficiency.

E.6 Training settings:

We consider two training settings in our experiments:

- **Supervised fine-tuning (SFT).** The parsing model f_θ is optimized with $\mathcal{L}_{\text{SFT}}$ on structured annotations. SFT is trained for 70 epochs with a per-device batch size of 12 and gradient accumulation of 2 steps.
- **Reinforcement learning (RL).** The policy π_θ is optimized with $\mathcal{L}_{\text{DT-GRPO}}$. Each input samples $K = 8$ candidate outputs for advantage estimation. RL is trained for 15 epochs with

a per-device batch size of 8, and the behavior policy π_{old} is periodically updated.

E.7 Optimization:

All stages use the AdamW optimizer. Learning rates are set to 1×10^{-5} for SFT and 1×10^{-6} for RL. Cosine learning rate decay with a warmup ratio of 5% is applied. Gradient clipping with maximum norm 1.0 is used to stabilize RL training.

E.8 RL rollout configuration:

Rollout generation is accelerated using a vLLM server backend. Sampling is performed with temperature 1.0, top- $p = 0.85$, and top- $k = 50$. Each prompt generates 8 candidates.

E.9 Reward configuration:

We adopt the composite reward defined in Appendix G. The final reward is a weighted combination of value-correction, format consistency, key-structure alignment, balanced empty/non-empty accuracy, false-empty penalty, and length regularization terms. All reward components are computed at the structured key-value level and aggregated for sequence-level policy optimization.

E.10 Batching and hardware:

Training is conducted using distributed data parallelism on 8 NVIDIA A800 GPUs (80GB). Mixed-precision training with bfloat16 is adopted to improve memory efficiency.

E.11 Training cost:

Full training takes approximately 3 GPU-hours for SFT and 16 GPU-hours for RL on $8 \times \text{A800}$ GPUs.

E.12 Training stability:

Reinforcement learning without token-wise weighting is observed to cause unstable optimization under noisy supervision, including oscillating rewards and malformed structured outputs. DT-GRPO improves stability by suppressing unreliable token updates and enhancing reward consistency.

E.13 Noisy-label classification benchmark:

For the CIFAR-10 experiment in Section 7.5, we follow the standard 50% symmetric-noise construction used by CLIPCleaner (Feng et al., 2024), randomly replacing half of the training labels with uniformly sampled incorrect classes. We use the standard 50,000-image training set and 10,000-image clean test set. From the training set, 6,250 images

and their original labels form the limited clean subset used to construct paired noisy/clean correction supervision for TLR, and the remaining 43,750 images expose only the corrupted labels during training. Our method uses Qwen2.5-VL-3B-Instruct and generates one of the ten class names as a textual output. The classification reward is one when the generated class exactly matches the refined target and zero otherwise, while the dynamic weight follows historical class-prediction correctness. The reported 95.9% accuracy is obtained from a single run. Results for CE, Co-teaching+ (Yu et al., 2019), F-correction (Patrini et al., 2017), PENCIL (Yi and Wu, 2019), LossModelling (Arazo et al., 2019), DivideMix (Li et al., 2020), ELR+ (Liu et al., 2020), MOIT (Ortego et al., 2021), SelCL+ (Li et al., 2022), and TCL (Huang et al., 2023) are adopted from the CLIPCleaner benchmark under the same dataset and noise setting, while CLIPCleaner is rerun using the authors’ official implementation.

F Visualization of Document Parsing Samples

We provide qualitative examples to illustrate the input-output format of our document parsing task and to analyze how supervision noise affects structured learning in practice. Specifically, we present representative document images together with their instruction prompts, expert annotations, pseudo labels, and TLR-refined labels.

These examples clarify: (i) how structured key-value annotations are represented in realistic document parsing scenarios, (ii) what pseudo supervision looks like under automatic labeling pipelines, and (iii) how Training-driven Label Refinement (TLR) improves supervision quality.

We further include a hard example to distinguish supervision noise from inherent visual ambiguity. In such cases, the primary challenge arises from limited visual readability (e.g., blur and near-confusable characters) rather than annotation errors.

F.1 Noisy Sample Visualization and TLR Refinement

As our dataset is collected from Chinese medical invoice documents, we retain the original Chinese text to preserve real-world authenticity and provide English translations for accessibility.

Table 6 presents a complete example of the prompt, expert annotation, pseudo label, and TLR-

refined label for invoice field extraction. The table explicitly demonstrates how supervision noise manifests at the structured key–value level and how TLR refines unreliable annotations.

Several typical noise patterns can be observed in the pseudo labels. First, numeric fields may suffer from severe corruption due to OCR digit confusion, as shown by the incorrect total amount prediction. Second, semantic fields may exhibit key–value mismatches caused by structural parsing errors, such as the incorrect insurance type. These issues frequently arise in dense financial regions where complex layouts and small character spacing amplify recognition difficulty.

TLR mitigates these problems by explicitly modeling annotation inconsistency and correcting unreliable fields while preserving valid supervision. As shown in Table 6, the self-paid amount is successfully corrected to match the expert annotation exactly. The insurance type is refined toward the correct insurance category, although minor terminology differences remain. The total amount is substantially improved compared with the pseudo label but still contains residual numeric errors, indicating the difficulty of precise digit recovery under heavy noise.

Figure 7 further visualizes this example with spatial grounding between key fields and their supporting visual regions. Noisy fields in pseudo labels are highlighted in red, while fields corrected by TLR are highlighted in green. The visualization highlights how inaccurate supervision in critical regions can propagate errors through the structured prediction pipeline, and how refinement improves supervision reliability.

This example highlights that supervision noise is often heterogeneous, requiring refinement methods that selectively correct unreliable fields while preserving trustworthy supervision signals.

F.2 Hard Sample Visualization

Hard cases in medical document parsing often arise from intrinsic visual factors and complex textual content (e.g., professional medical terminology), as illustrated in Figure 8. Typical sources of difficulty include blurred or low-resolution text regions, compression artifacts, and near-confusable characters (e.g., visually similar digits or Chinese characters). These factors introduce localized recognition uncertainty, which can lead to incorrect field extraction even when supervision is accurate.

This example highlights a different source of

errors compared with the noisy sample shown in Figure 7. In the noisy case, errors originate from incorrect pseudo annotations, whereas in this example, errors are mainly caused by poor visual quality despite correct supervision. This contrast shows that supervision noise and visual difficulty are fundamentally different challenges. It motivates our noise-aware training design, which explicitly corrects unreliable annotations while maintaining robustness to unavoidable visual ambiguity.

G Reward Design for Dynamic Token-wise GRPO

We adopt the value-correction reward defined in the main paper as the primary optimization objective:

$$r_i^{(k)} = \frac{1}{|\mathcal{K}_i|} \sum_{k_m \in \mathcal{K}_i} \mathbf{1}[\bar{v}_i^{(n,k)}(k_m) = \hat{v}_i(k_m)]. \quad (9)$$

This reward directly measures key–value correctness at the field level.

To further improve structural validity and behavioral reliability during reinforcement learning (RL), we introduce several auxiliary reward components.

G.1 Format reward:

To encourage structurally valid outputs, we introduce a format reward

$$r_{i,\text{fmt}}^{(k)}, \quad (10)$$

which evaluates whether the rollout prediction follows the required structured key–value layout. This reward favors properly formatted outputs, including correct key–value organization and parsable structured responses, and helps reduce malformed predictions during RL.

G.2 Key-length reward:

We further introduce a key-length reward to encourage consistency between the predicted key structure and the queried key set. Let $\bar{\mathcal{K}}_i^{(n,k)}$ denote the set of keys appearing in the k -th rollout. We define

$$r_{i,\text{klen}}^{(k)} = \frac{|\bar{\mathcal{K}}_i^{(n,k)} \cap \mathcal{K}_i|}{|\bar{\mathcal{K}}_i^{(n,k)} \cup \mathcal{K}_i|}. \quad (11)$$

This term encourages predicted key sets that align with the queried fields while discouraging missing or spurious keys.

Table 6: Example of prompt and multi-stage annotations for invoice field extraction.

Item	Chinese	English
Prompt	请识别下图发票内容, 并返回字段: 票据号码、合计金额小写、个人自费、医保类型。请严格按照JSON 格式返回。 <image>	Please recognize the invoice content in the image and return the following fields: Invoice Number, Total Amount (in digits), Self-paid Amount, and Insurance Type. Please strictly return the results in JSON format.
Expert annotation (GT)	票据号码: 00024920; 合计金额小写: 670448.20; 个人自费: 204375.59; 医保类型: 新型农村合***。	Invoice Number: 00024920; Total Amount (digits): 670448.20; Self-paid Amount: 204375.59; Insurance Type: New Rural Cooperative Medical Scheme.
Pseudo label ($\tilde{Y}$)	票据号码: 00024920; 合计金额小写: 18.0 ; 个人自费: 204378.59 ; 医保类型: 自费 。	Invoice Number: 00024920; Total Amount (digits): 18.0 ; Self-paid Amount: 204378.59 ; Insurance Type: Self-funded .
TLR-refined label ($\hat{Y}$)	票据号码: 00024920; 合计金额小写: 870448.20 ; 个人自费: 204375.59 ; 医保类型: 新农合 。	Invoice Number: 00024920; Total Amount (digits): 870448.20 ; Self-paid Amount: 204375.59 ; Insurance Type: New Rural Cooperative Scheme .

Noisy fields in pseudo labels are highlighted in **red**, while fields corrected by TLR are highlighted in **green**.

G.3 Empty/non-empty balanced reward:

In practice, queried fields are often imbalanced where some fields are genuinely empty, while others contain valid values. To prevent optimization from being dominated by only one case, we separately evaluate non-empty and empty fields and combine them using a balancing coefficient.

We define

$$\text{acc}_{i,\text{non}}^{(k)} = \frac{\sum_{k_m \in \mathcal{K}_i} \mathbf{1}_{\left[\wedge \begin{smallmatrix} \hat{v}_i(k_m) \neq \emptyset \\ \bar{v}_i^{(n,k)}(k_m) = \hat{v}_i(k_m) \end{smallmatrix}\right]}}{\sum_{k_m \in \mathcal{K}_i} \mathbf{1}_{[\hat{v}_i(k_m) \neq \emptyset]}}. \quad (12)$$

and

$$\text{acc}_{i,\text{emp}}^{(k)} = \frac{\sum_{k_m \in \mathcal{K}_i} \mathbf{1}_{\left[\wedge \begin{smallmatrix} \hat{v}_i(k_m) = \emptyset \\ \bar{v}_i^{(n,k)}(k_m) = \hat{v}_i(k_m) \end{smallmatrix}\right]}}{\sum_{k_m \in \mathcal{K}_i} \mathbf{1}_{[\hat{v}_i(k_m) = \emptyset]}}. \quad (13)$$

The corresponding balanced reward is

$$r_{i,\text{bal}}^{(k)} = \alpha \text{acc}_{i,\text{non}}^{(k)} + (1 - \alpha) \text{acc}_{i,\text{emp}}^{(k)}, \quad (14)$$

where $\alpha \in [0, 1]$ controls the relative importance of non-empty fields.

G.4 False-empty penalty:

A particularly undesirable failure mode is predicting an empty value for a field whose target value is actually non-empty. We therefore introduce a false-empty penalty:

$$r_{i,\text{fe}}^{(k)} = - \frac{\sum_{k_m \in \mathcal{K}_i} \mathbf{1}_{\left[\wedge \begin{smallmatrix} \hat{v}_i(k_m) \neq \emptyset \wedge \\ \bar{v}_i^{(n,k)}(k_m) = \emptyset \end{smallmatrix}\right]}}{\sum_{k_m \in \mathcal{K}_i} \mathbf{1}_{[\hat{v}_i(k_m) \neq \emptyset]}}. \quad (15)$$

This term explicitly penalizes missed extraction of valid fields.

G.5 Length-based penalty:

To discourage pathological under-generation or over-generation of field values, we further introduce a length-based penalty:

$$r_{i,\text{len}}^{(k)} = - \frac{1}{|\mathcal{K}_i|} \sum_{k_m \in \mathcal{K}_i} \tanh \left(\frac{||\bar{v}_i^{(n,k)}(k_m)| - |\hat{v}_i(k_m)||}{\max(|\hat{v}_i(k_m)|, 1)} \right). \quad (16)$$

This term regularizes output behavior by discouraging implausible value lengths.

G.6 Overall reward:

Collectively, these auxiliary rewards regularize structural validity, semantic correctness, and behavioral plausibility, leading to more stable and reliable reinforcement learning for document parsing. The final reward is defined as a weighted combination of the value-correction reward and the auxiliary reward terms:

$$\begin{aligned} r_{i,\text{total}}^{(k)} = & \lambda_{\text{val}} r_i^{(k)} + \lambda_{\text{fmt}} r_{i,\text{fmt}}^{(k)} \\ & + \lambda_{\text{klen}} r_{i,\text{klen}}^{(k)} + \lambda_{\text{bal}} r_{i,\text{bal}}^{(k)} \\ & + \lambda_{\text{fe}} r_{i,\text{fe}}^{(k)} + \lambda_{\text{len}} r_{i,\text{len}}^{(k)}. \end{aligned} \quad (17)$$

G.7 Reward Ablation and Coefficient Selection

The reward coefficients were set heuristically according to task priorities before the reported evalua-

Table 7: Reward ablation of DT-GRPO. Results are mean FMR over three independent runs.

Reward Configuration	Mean FMR
Value-correction reward only	0.8173
Complete six-component reward	0.8216

tions and kept fixed across all experiments. The test set was not used for coefficient selection. We assign the dominant weight to value correction, with $\lambda_{\text{val}} = 1.0$. The auxiliary coefficients are $\lambda_{\text{fmt}} = 0.15$, $\lambda_{\text{fe}} = 0.12$, $\lambda_{\text{klen}} = 0.10$, $\lambda_{\text{bal}} = 0.05$, and $\lambda_{\text{len}} = 0.05$.

The value-correction reward alone achieves a mean FMR of 0.8173, while the complete six-component reward reaches 0.8216. This result identifies value correction as the primary optimization signal and shows that the auxiliary components provide a complementary gain.

H Gradient-Level Analysis under Noisy Supervision

In this section, we analyze how noisy supervision influences the optimization dynamics of SFT, GRPO, and DT-GRPO at the gradient level. The key distinction lies in how annotation noise enters the training signal and propagates through parameter updates.

SFT directly fits observed target tokens. GRPO updates the policy using sequence-level rewards. DT-GRPO further introduces token-level weighting to modulate the contribution of unreliable updates.

H.1 Preliminaries:

For each sample i , let x_i denote the input and

$$\hat{y}_i = (\hat{y}_{i,1}, \dots, \hat{y}_{i,T_i})$$

denote the target sequence derived from the refined annotation.

Although TLR improves annotation quality, residual noise may still remain. We therefore partition token positions into clean and noisy subsets:

$$\{1, \dots, T_i\} = \mathcal{C}_i \cup \mathcal{N}_i, \quad \mathcal{C}_i \cap \mathcal{N}_i = \emptyset. \quad (18)$$

H.2 SFT:

The SFT objective is

$$\mathcal{L}_{\text{SFT}} = -\mathbb{E}_i \left[\sum_{t=1}^{T_i} \log \pi_{\theta}(\hat{y}_{i,t} \mid x_i, \hat{y}_{i,<t}) \right]. \quad (19)$$

Its gradient is

$$\nabla_{\theta} \mathcal{L}_{\text{SFT}} = -\mathbb{E}_i \left[\sum_{t=1}^{T_i} \nabla_{\theta} \log \pi_{\theta}(\hat{y}_{i,t} \mid x_i, \hat{y}_{i,<t}) \right]. \quad (20)$$

Thus, every observed token contributes directly to optimization. For noisy positions $t \in \mathcal{N}_i$, the model is still pushed toward the incorrect token. Annotation noise, therefore, enters SFT gradients directly at the token level.

H.3 GRPO:

Let π_{old} denote the behavior policy. For each sample i , GRPO samples K candidate outputs

$$\bar{y}_i^{(k)} \sim \pi_{\text{old}}, \quad k = 1, \dots, K,$$

and computes a sequence-level reward $r_i^{(k)}$. The group-relative advantage is

$$A_i^{(k)} = \frac{r_i^{(k)} - \text{mean}_k(r_i^{(k)})}{\text{std}_k(r_i^{(k)}) + \epsilon}. \quad (21)$$

For clarity, the analysis below focuses on the policy-ratio term. Standard ratio clipping and reference-policy KL regularization are retained unchanged because DT-GRPO does not modify them, and are therefore omitted from the displayed objective. Using the ratio form, GRPO optimizes

$$\mathcal{L}_{\text{GRPO}} = -\mathbb{E}_{i,k} \left[A_i^{(k)} \sum_{t=1}^{T_i} \rho_{i,t}^{(k)}(\theta) \right], \quad (22)$$

where

$$\begin{aligned} \rho_{i,t}^{(k)}(\theta) &= \frac{\pi_{\theta}(\bar{y}_{i,t}^{(k)} \mid x_i, \bar{y}_{i,<t}^{(k)})}{\pi_{\text{old}}(\bar{y}_{i,t}^{(k)} \mid x_i, \bar{y}_{i,<t}^{(k)})} \\ &= \exp \left(\log \pi_{\theta}(\bar{y}_{i,t}^{(k)} \mid x_i, \bar{y}_{i,<t}^{(k)}) \right. \\ &\quad \left. - \log \pi_{\text{old}}(\bar{y}_{i,t}^{(k)} \mid x_i, \bar{y}_{i,<t}^{(k)}) \right). \end{aligned} \quad (23)$$

Since π_{old} does not depend on θ ,

$$\nabla_{\theta} \rho_{i,t}^{(k)}(\theta) = \rho_{i,t}^{(k)}(\theta) \nabla_{\theta} \log \pi_{\theta}(\bar{y}_{i,t}^{(k)} \mid x_i, \bar{y}_{i,<t}^{(k)}). \quad (24)$$

Therefore,

$$\begin{aligned} \nabla_{\theta} \mathcal{L}_{\text{GRPO}} &= -\mathbb{E}_{i,k} \left[A_i^{(k)} \sum_{t=1}^{T_i} \rho_{i,t}^{(k)}(\theta) \right. \\ &\quad \left. \cdot \nabla_{\theta} \log \pi_{\theta}(\bar{y}_{i,t}^{(k)} \mid x_i, \bar{y}_{i,<t}^{(k)}) \right]. \end{aligned} \quad (25)$$

This shows that GRPO, like SFT, is ultimately expressed in terms of token-level score functions $\nabla_{\theta} \log \pi_{\theta}(\cdot)$. However, SFT directly fits annotated target tokens, whereas GRPO scales token-level updates by the sequence-dependent factor $A_i^{(k)} \rho_{i,t}^{(k)}(\theta)$.

Consequently, GRPO does not fit the annotated sequence token by token. Instead, annotation noise influences optimization indirectly through the reward and advantage signals.

H.4 DT-GRPO:

DT-GRPO retains the sequence-level advantage of GRPO while further introducing token-level weights $w_{i,t}^{(n)}$:

$$\mathcal{L}_{\text{DT-GRPO}} = -\mathbb{E}_{i,k} \left[A_i^{(k)} \sum_{t=1}^{T_i} w_{i,t}^{(n)} \rho_{i,t}^{(k)}(\theta) \right]. \quad (26)$$

Its gradient is

$$\nabla_{\theta} \mathcal{L}_{\text{DT-GRPO}} = -\mathbb{E}_{i,k} \left[A_i^{(k)} \sum_{t=1}^{T_i} w_{i,t}^{(n)} \rho_{i,t}^{(k)}(\theta) \cdot \nabla_{\theta} \log \pi_{\theta} \left(\bar{y}_{i,t}^{(k)} \mid x_i, \bar{y}_{i,<t}^{(k)} \right) \right]. \quad (27)$$

Compared with GRPO, DT-GRPO introduces an explicit token-level modulation factor. As a result, unreliable tokens can contribute less even within the same rewarded sequence.

H.5 Unified view:

All three methods can be expressed in a unified gradient form:

$$\nabla_{\theta} \mathcal{L} = -\mathbb{E} \left[\sum_t c_{i,t} \nabla_{\theta} \log \pi_{\theta}(\cdot) \right], \quad (28)$$

with coefficients

$$c_{i,t} = \begin{cases} 1, & \text{for SFT,} \\ A_i^{(k)} \rho_{i,t}^{(k)}(\theta), & \text{for GRPO,} \\ A_i^{(k)} w_{i,t}^{(n)} \rho_{i,t}^{(k)}(\theta), & \text{for DT-GRPO.} \end{cases} \quad (29)$$

This unified view clarifies how noisy supervision influences optimization differently. In SFT, noise directly determines token-level gradients. In GRPO, noise affects training indirectly through sequence-level reward and advantage. In DT-GRPO, this influence is further modulated by token-level weighting.

Table 8: Noise taxonomy and distribution in pseudo annotations $\tilde{\mathcal{Y}}$.

Noise Type	Description	Ratio (%)
Value corruption	Incorrect numeric/textual content	31.8
Key-value misalignment	Correct value paired with wrong key	19.6
Missing field	Target key absent in output	17.4
Spurious field	Non-queried key generated	13.2
Format error	Non-parsable structured output	10.5
Other noise	Mixed or rare patterns	7.5
Total	–	100.0

Table 9: Refinement effectiveness of TLR on different noise types.

Noise Type	Corr. Rate (%)	Residual (%)
Value corruption	73.5	26.5
Key-value misalignment	76.1	23.9
Missing field	79.8	20.2
Spurious field	83.7	16.3
Format error	96.4	3.6

I Noise Statistics and Refinement Analysis

This section provides detailed statistics of the supervision noise used in our experiments and evaluates the effectiveness of Training-driven Label Refinement (TLR).

I.1 Noise Taxonomy and Distribution

Pseudo annotations $\tilde{\mathcal{Y}}$ contain heterogeneous noise patterns introduced by automatic labeling pipelines. Following the noise construction protocol described in the main paper, we categorize supervision noise at the key-value level, as in Table 8.

Value corruption and structural inconsistency account for the majority of supervision noise.

I.2 TLR Refinement Effectiveness

We evaluate how Training-driven Label Refinement improves pseudo supervision quality. For each noise category, we measure the proportion of instances successfully corrected by the refinement model f_{ϕ} .

TLR significantly reduces systematic supervision noise across all major categories, while residual errors remain, as in Table 9.

Table 10: Per-seed FMR results for different RL objectives.

Method	Seed 1	Seed 2	Seed 3	Mean $\pm$ Std
Vanilla GRPO	0.8017	0.8028	0.8013	0.8019 $\pm$ 0.0006
DAPO	0.8204	0.8211	0.8201	0.8205 $\pm$ 0.0004
GSPO	0.8191	0.8187	0.8198	0.8192 $\pm$ 0.0005
SAPO	0.8198	0.8204	0.8191	0.8198 $\pm$ 0.0005
DT-GRPO	0.8214	0.8221	0.8212	0.8216 $\pm$ 0.0004

J Per-Seed Results for RL Objective Comparison

Table 10 reports the complete results of the three independent runs summarized in Table 2. All methods use the same backbone, refined-label dataset, and training configuration. DT-GRPO ranks first in every run, showing that its improvement is consistent across random seeds rather than being driven by a single initialization.

