From Recognition to Reasoning: Benchmarking and Enhancing MLLMs on Real-World Receipt Document Understanding

Yandi Wang*, Libin Zhan*, Ziwei Huang*, Tiancheng Luo,
Yuxuan Jiang, Wang Dong, Leilei Gan†, Jun Chen†
Zhejiang University, China
{yandiwang, zhanlibin, leileigan, chenjun332}@zju.edu.cn

Abstract

Extracting structured information from visual documents (Visual Information Extraction, VIE) is a cornerstone of business automation. While recent Multimodal Large Language Models (MLLMs) have shown promising capabilities, existing benchmarks suffer from critical limitations in scale and realism, lack semantic granularity, and fail to cover diverse document types. To bridge this gap, we introduce **ReceiptBench**, a large-scale, human-annotated benchmark consisting of 10k diverse receipts, organizing information extraction into four hierarchical sub-tasks: (1) *Basic Perception* for raw text spotting, (2) *Format Normalization* for strictly following standardization instructions, (3) *Semantic Reasoning* for inferring implicit attributes from context, and (4) *Structure Parsing* for handling nested line items. Furthermore, we propose a two-stage training framework incorporating *Metric-Aware Group Relative Policy Optimization (GRPO)*, which translates rigorous evaluation constraints into reinforcement learning signals to enhance structural consistency. Extensive experiments demonstrate that our method yields state-of-the-art performance, surpassing leading proprietary models on complex reasoning tasks. We release our datasets and code at https://github.com/wwwT0ri/ReceiptBench.

1 Introduction

Visual Information Extraction (VIE) serves as a cornerstone of enterprise automation, enabling the digitization of workflows in finance, logistics, and legal domains. The recent emergence of Multimodal Large Language Models (MLLMs) (Bai et al., 2023; Hurst et al., 2024; Chen et al., 2024) has shifted the paradigm from pipeline-based Optical Character Recognition (OCR) to end-to-end visual reasoning.

The advancement of VIE, particularly for Key Information Extraction (KIE), relies heavily on high-quality benchmarks to ensure the extraction reliability and logical consistency required for financial evidence. Receipts and invoices are critical in this domain due to their global ubiquity and layout diversity. However, as highlighted in Table 1, existing benchmarks struggle to simultaneously satisfy the demands of scale, diversity, and granularity. Early real-world datasets (Huang et al., 2019; Park et al., 2019; Sun et al., 2021; Wang et al., 2021; Xu et al., 2022) established essential baselines but are severely limited in scale (<2k images) and confined to narrow domains (e.g., retail and dining receipts), failing to represent the heterogeneous layouts found in broader real-world scenarios. While synthetic datasets like FATURA (Limam et al., 2025) address the data volume issue, they often rely on finite templates and lack the authentic semantic logic inherent in real transactions. Furthermore, current efforts (Abdallah et al., 2024; Huang et al., 2019; Park et al., 2019; Mathew et al., 2021; Jaume et al., 2019; Xu et al., 2022) predominantly focus on explicit text extraction; this shallow perception fails to capture the complexity of real-world financial processing, which demands format normalization, implicit reasoning, and structural parsing.

To bridge this gap, we introduce **ReceiptBench**, a large-scale benchmark designed to evaluate reasoning-aware information extraction from complex financial documents. ReceiptBench comprises **10,656** high-quality images collected from diverse real-world sources, covering multi-lingual regions and heterogeneous document types (e.g., taxi invoices, ferry tickets, hotel statements). Unlike previous works that treat extraction as a flat slot-filling task, we propose a hierarchical taxonomy of four capabilities: *Perception*, *Normalization*, *Reasoning*, and *Structure*. This taxonomy requires models to not only "read" the pixels but also "understand"

* Equal contribution.
† Corresponding authors.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 46007–46024

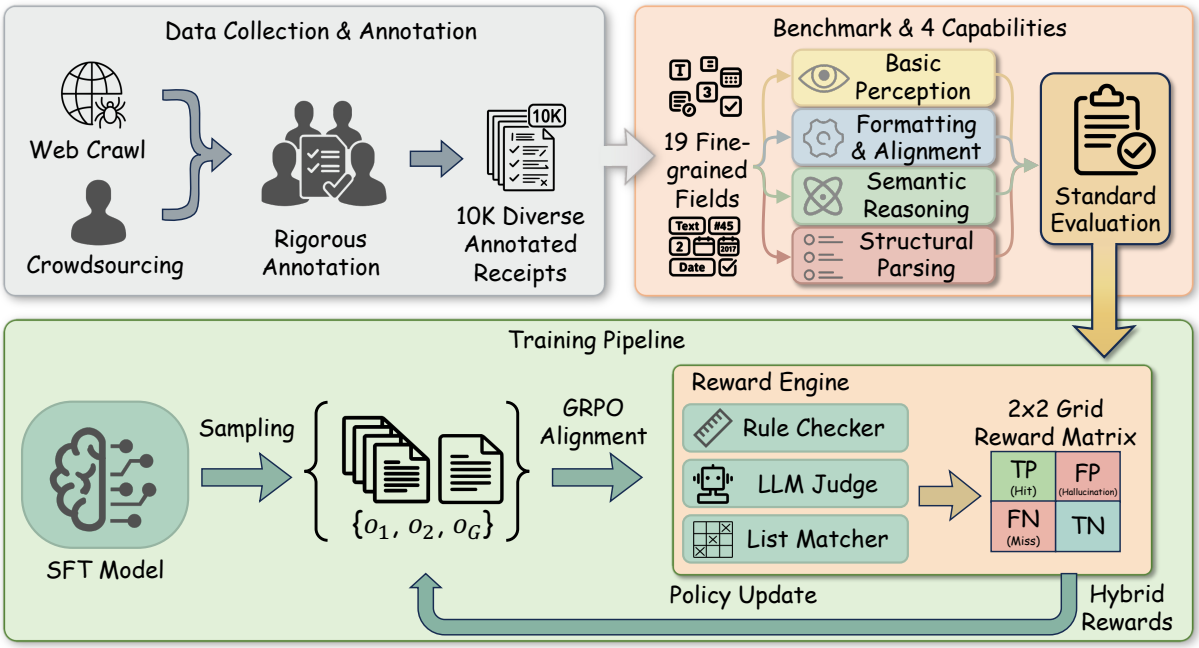


Figure 1: **Overview of the ReceiptBench Framework. (Top) Benchmark Construction:** We curate 10k diverse invoices via web crawling and crowdsourcing. The benchmark defines a hierarchical taxonomy covering four capabilities: *Basic Perception*, *Formatting*, *Semantic Reasoning*, and *Structural Parsing*. **(Bottom) Training Pipeline:** To master these capabilities, we propose a Metric-Aware GRPO framework. The SFT model acts as the policy, generating outputs that are evaluated by a hybrid Reward Engine (comprising Rule Checkers, LLM Judges, and List Matchers). Crucially, the evaluation results are mapped into a 2x2 Reward Matrix—rewarding hits (TP) while explicitly penalizing hallucinations (FP)—to align the model with rigorous auditing standards.

the business logic and "structure" the output rigorously.

However, our evaluation on ReceiptBench highlights significant deficiencies in current methodologies: general-purpose models (Bai et al., 2023; Hurst et al., 2024; Chen et al., 2024) often overlook fine-grained financial constraints, while specialized models (Chen et al., 2025; Cui et al., 2025; Huang et al., 2022) lack the generative reasoning required for complex extraction. Even establishing a competitive baseline via standard Supervised Fine-Tuning (SFT) proves non-trivial. SFT optimizes for local token probabilities rather than global logical consistency, frequently resulting in structural hallucinations (e.g., invalid JSON syntax) and arithmetic inconsistencies (e.g., line items not summing to the total). To address this, we propose a two-stage training framework. After initial instruction tuning, we introduce an alignment stage using Group Relative Policy Optimization (GRPO) (Shao et al., 2024). Specifically, we design a **Metric-Aware Reward Engine** that directly translates our rigorous evaluation protocols—arithmetic checks and schema adherence—into reinforcement learning signals. This approach explicitly penalizes hallucinations and rewards logical coherence, enabling the model to internalize the complex reasoning rules of the benchmark.

In summary, our contributions are as follows:

- We present **ReceiptBench**, a challenging benchmark with 10k real-world samples and 19 fine-grained fields, shifting the focus of VIE from literal extraction to cognitive reasoning and structural parsing.
- We design a robust **Hybrid Evaluation Protocol** that moves beyond simple string matching, incorporating LLM-based semantic judges and Hungarian matching algorithms (Kuhn, 1955) for nested lists to ensure fair and accurate assessment.
- We propose a **Metric-Aware GRPO** training framework. Extensive experiments demonstrate that this method significantly improves the reasoning and structural capabilities of open-source MLLMs (e.g., Qwen3-VL (Bai et al., 2025)), narrowing the gap with proprietary SOTA models like GPT-5.

2 Related Work

2.1 Benchmarks for VIE

Existing benchmarks face critical limitations in **scale and realism**. Early datasets like

SROIE (Huang et al., 2019), CORD (Park et al., 2019), and WildReceipt (Sun et al., 2021) are too small ($<$ 2k images) for data-hungry MLLMs. Comprehensive benchmarks like CC-OCR (Yang et al., 2025) offer limited fresh KIE challenges by partially aggregating existing datasets ($\sim$2k samples). While FATURA (Limam et al., 2025) addresses scalability via synthesis, it suffers from template bias and lacks authentic semantic logic.

Regarding **granularity and task alignment**, existing efforts often diverge from the needs of enterprise automation. ReceiptSense (Abdallah et al., 2024) provides sparse annotations that hinder complex reasoning. Meanwhile, benchmarks like DocVQA (Mathew et al., 2021), MP-DocVQA (Tito et al., 2023), and DUDE (Van Landeghem et al., 2023) frame document understanding primarily as open-ended Visual Question Answering (VQA) or generic layout analysis (e.g., FUNSD (Jaume et al., 2019), XFUND (Xu et al., 2022)) rather than structured schema-constrained extraction. Furthermore, while OCR-Reasoning (Huang et al., 2025) extensively evaluates visual reasoning, its taxonomy is heavily skewed toward academic problem-solving rather than the strict financial business logic required in real-world scenarios.

Finally, **document diversity** remains narrow; datasets are often skewed towards simple retail slips (SROIE) or exam papers (EPHOIE (Wang et al., 2021)), failing to represent heterogeneous financial documents like multi-page hotel statements. Consequently, the field lacks a unified benchmark balancing these critical dimensions, a gap **ReceiptBench** aims to fill.

2.2 Specialized Document Understanding Models

Early pipeline systems combined OCR with NLP models. While the LayoutLM series (Xu et al., 2020; Huang et al., 2022) embedded spatial semantics, they still **relied on external OCR**. Subsequently, Donut (Kim et al., 2022) and Nougat (Blecher et al., 2023) introduced OCR-free, end-to-end paradigms mapping pixels to text. Recent models prioritize efficiency: GOT-OCR (Wei et al., 2024) unifies OCR tasks under a general theory, while PaddleOCR-VL (Cui et al., 2025) utilizes a NaViT-style encoder to achieve SOTA performance with minimal consumption.

To address high-resolution token costs, DeepSeek-OCR (Wei et al., 2025) introduces optical context compression to minimize vision tokens. Similarly, UReader (Ye et al., 2023) and mPLUG-DocOwl 1.5 (Hu et al., 2024) employ shape-adaptive cropping. Moving towards agentic reasoning, DianJin-OCR (Chen et al., 2025) leverages Chain-of-Thought (CoT) (Wei et al., 2022b) for interleaved planning and tool use. However, current benchmarks lack evaluation for such **complex reasoning and structural parsing**, motivating **ReceiptBench**.

3 The ReceiptBench Benchmark

To address the limitations of existing benchmarks and support the training of end-to-end MLLMs for complex document understanding, we introduce **ReceiptBench**, a large-scale, real-world dataset designed with financial accounting standards and multi-dimensional capability evaluation in mind.

3.1 Data Collection and Annotation

Data Sources. The dataset comprises **10,656** images sourced from real-world scenarios to ensure diversity in layout, visual conditions, and content. Our collection followed a hybrid strategy: (1) *Public Web Crawling:* We gathered receipt images from publicly available repositories, prioritizing those with varied layouts and quality. (2) *Crowdsourced Solicitation:* To capture long-tail document types (e.g., specific regional taxi invoices or flight tickets) often absent in public collections, we conducted a paid, questionnaire-driven campaign, ensuring broad geographical and domain coverage. Each collected document was manually reviewed to verify its authenticity and legibility.

Annotation Process. We engaged a professional data annotation service to ensure high-quality labeling. The entire dataset was divided into 10 batches, each of which underwent the vendor's internal annotation and multi-stage review cycle. Upon delivery, we applied a stringent **acceptance protocol** comprising three validation stages:

1. Random Sampling Inspection: We randomly sampled a validation subset, in which domain experts manually verified the correctness of all annotated fields to ensure accuracy.

2. Automated Logic and Format Validation: We employed custom validation scripts to check compliance with the predefined schema. This included crucial **cross-field consistency** checks (e.g., verifying that the sum of line items in `detail` matches `std_total`) and **standard formatting**

Dataset	Size	Fields	Document Types Coverage	Key Limitations
SROIE (Huang et al., 2019)	1,000	4	Retail Receipts	Small Scale; Low Granularity
FUNSD (Jaume et al., 2019)	199	-	General Forms	Small Scale; Generic Entity Labels
CORD (Park et al., 2019)	1,000	8	Retail & Dining Receipts	Small Scale; Narrow Domain
WildReceipt (Sun et al., 2021)	1,765	25	Retail Receipts	Small Scale; Narrow Domain
EPHOIE (Wang et al., 2021)	1,494	10	Examination Papers	Small Scale; Education Domain
XFUND (Xu et al., 2022)	1,393	-	General Forms	Small Scale; Generic Entity Labels
DocVQA (Mathew et al., 2021)	12,767	-	Various Documents	Paradigm Divergence (QA vs. IE)
FATURA (Limam et al., 2025)	10,000	24	Invoices	Synthetic Logic; Finite Templates (50)
ReceiptSense (Abdallah et al., 2024)	20,000	5	Retail Receipts	Low Granularity; Narrow Domain
ReceiptBench (Ours)	**10,656**	**19**	**Purchasing, Hotel, Travel, etc.**	**-**

Table 1: **Comparison of our dataset with existing VIE benchmarks.** Existing datasets exhibit critical limitations in three key aspects: (1) **Scale and Realism**: they are either limited in size or rely on synthetic generation; (2) **Granularity and Task Alignment**: they suffer from low granularity or target divergent paradigms such as QA and generic layout analysis; and (3) **Document Diversity**: they are restricted to specific narrow domains like retail or non-financial domains. In contrast, our dataset balances scale, realism, granularity, and document diversity.

rules (e.g., ensuring `std_invoice_time` conforms to the standard date pattern).

3. Error Analysis and Iterative Refinement: Validation results were aggregated to compute field-specific accuracy rates and to summarize common error patterns. These findings, encompassing all detected errors, were documented in an audit report provided to the vendor. If the accuracy for any field within the validation subset fell below the **97%** threshold, the *entire* batch was rejected and required to undergo revision and re-annotation.

Through this multi-stage, iterative pipeline, the final dataset achieved an overall average annotation accuracy of **98.7%**, confirming its high quality for subsequent tasks.

3.2 Dataset Statistics

Document Type Diversity. ReceiptBench distinguishes itself by covering a wide spectrum of service-oriented financial documents, moving beyond the retail-centric distribution of prior works (Table 2). While *General Purchase & Dining* receipts account for the plurality (43.5%), the dataset features a substantial proportion of *Transportation* documents (Plane, Taxi, Train, Bus, etc.), totaling over 35%, as well as complex *Hotel Bills* (11.2%). This distribution introduces multi-page layouts and tabular structures significantly more challenging than standard supermarket receipts.

Language Distribution. As shown in Table 6, the dataset is predominantly English (98.0%) to align with the primary pre-training data of most MLLMs. However, it includes a "long-tail" of 213 samples covering 8 other languages. This inclusion allows for evaluating the model's robustness against linguistic noise and character variations in low-resource scenarios.

Category	Count	Pct.
Purchase & Dining	4,636	43.50%
Plane Ticket	2,175	20.42%
Hotel Bill	1,198	11.24%
Taxi Receipt	1,132	10.62%
Train Ticket	416	3.90%
Bus Ticket	249	2.34%
Ship/Ferry Ticket	150	1.41%
Fuel Receipt	132	1.24%
Metro Ticket	130	1.22%
Others	438	4.11%
Total	**10,656**	**100.0%**

Table 2: Distribution of document types in our dataset. "Others" includes Car Rental, Postage, Toll, Parking, Internet, Phone, Baggage, Water, Electricity, Medical, Education and Handling receipts.

3.3 Task Taxonomy and Schema

We define the information extraction problem as a set of four progressive sub-tasks targeting 19 distinct fields (e.g., `std_invoice_time`, `tax_number`). The selection of these fields is grounded in standard accounting principles, ensuring the benchmark's utility for real-world financial auditing. See Appendix A for detailed definitions.

Based on the cognitive capabilities required, we partition the fields into four sub-tasks:

Task 1: Basic Perception (8 fields). Evaluates Optical Character Recognition (OCR) and grounding. It targets explicit text such as `invoice_number` and raw timestamps. Success here indicates the model can accurately "read" visual tokens (Biten et al., 2019).

Task 2: Formatting & Normalization (4 fields). Tests instruction-following abilities. Models must convert raw text into standardized formats (e.g., converting "20 Oct, 23" to "2023-10-20" for `std_start_time`). This aligns with the instruction tuning paradigm critical for LLM usability (Wei et al., 2022a).

Task 3: Semantic Reasoning (6 fields). Requires extracting implicit information. For instance,

deducing `type="Hotel"` from room charges, or inferring `std_curr="USD"` from a "New York" address. This evaluates multi-modal reasoning beyond simple extraction (Xu et al., 2020).

Task 4: Structural Parsing (1 field). The `detail` field requires parsing complex, often nested tables into a list of dictionaries (content, amount, tax status). This represents the most challenging task, demanding an understanding of spatial structures similar to table extraction benchmarks (Zhong et al., 2019).

3.4 Evaluation Protocol

Evaluating information extraction from complex invoices presents unique challenges, such as valid OCR variations and permutation-invariant lists. To ensure a robust and fair comparison for ReceiptBench, we define a standardized hybrid evaluation protocol combining rule-based matching and semantic judgment.

Hierarchical Evaluation Logic. We categorize the 19 target fields into four types, applying specific metrics for each:

Type A: Exact Match Fields. For fields requiring strict adherence to visual evidence (e.g., `type`, `tax_number`, `std_invoice_time`), we use **Exact Match (EM)**. Both ground truth and predictions are normalized (lowercased, whitespace trimmed) before comparison to handle minor spacing differences.

Type B: Numeric Fields. For monetary values (e.g., `std_total`), we allow a floating-point tolerance of $\epsilon < 1e^{-6}$. Zero values (0) and empty strings are treated equivalently to handle format inconsistencies.

Type C: Semantic Fields. For fields where minor textual variations preserve meaning (e.g., `place`, `seller_name`), we employ a **Cascading Judge**: (1) *Exact Filter:* First, we check for normalized string equality. If they match, it is counted as a True Positive (TP). (2) *LLM Judge:* If the exact match fails, we employ a lightweight LLM (Qwen3-4B) as a semantic judge. The model is prompted with specific criteria **(see Table 8)** to determine if the predicted entity is semantically equivalent to the ground truth, explicitly allowing for abbreviations (e.g., "Co." vs. "Company") and synonyms while penalizing factual errors.

Type D: Structured List Fields. Evaluating the lists (e.g., `detail`, `orig_curr`) is the most challenging aspect due to order invariance and nested attributes. We formulate this as a **Maximum Bipartite Matching** problem. For a predicted list P and ground truth list G, we construct a cost matrix C where C_{ij} represents the dissimilarity between item P_i and G_j. The dissimilarity is derived from a composite similarity score S_{ij}, calculated as a weighted sum of four metrics to capture both lexical and semantic correspondence:

$$S_{ij} = \alpha \cdot S_{\text{lev}} + \beta \cdot S_{\text{sort}} + \gamma \cdot S_{\text{lcs}} + \delta \cdot S_{\text{sem}} \quad (1)$$

where S_{lev} denotes the Levenshtein ratio, S_{sort} is the Token Sort similarity (robust to word reordering), S_{lcs} is the Longest Common Subsequence ratio, and S_{sem} represents the cosine similarity of embeddings from a Sentence Transformer. The coefficients $\alpha, \beta, \gamma, \delta$ are hyperparameters empirically optimized via grid search on a human-annotated validation set to maximize alignment with human judgment. A comprehensive hyperparameter sensitivity analysis (detailed in Appendix D.3) confirms that our evaluation metric and the resulting model rankings are highly robust to these weight variations.

The cost is defined as $C_{ij} = 1 - S_{ij}$. For the `detail` field, we impose additional hard constraints: items with mismatched numerical amounts ($|\Delta| > 0.05$) are assigned an infinite cost. We then apply the **Hungarian Algorithm** to find the optimal assignment, accepting matches only if the cost $C_{ij} \leq 0.25$ and attributes align.

Primary Metrics. We mainly report **F1-score**. Given that many fields in ReceiptBench can be legitimately empty (e.g., `departure` for a restaurant receipt), correct identification of absent information is crucial. Therefore, our evaluation script explicitly accounts for True Negatives (TN) to avoid penalizing models that correctly predict "empty" for missing fields.

4 Methodology

While one-shot evaluation on proprietary models (e.g., GPT-5) provides a reference for upper-bound performance, it is crucial to establish strong open-source baselines to validate the learnability of ReceiptBench. In this section, we describe our two-stage training pipeline: Supervised Fine-Tuning (SFT) for instruction adherence and Group Relative Policy Optimization (GRPO) (Shao et al., 2024) for reasoning alignment.

4.1 Stage 1: Supervised Fine-Tuning (SFT)

To equip the model with the capability to handle the complex extraction rules of ReceiptBench, we construct a rigorous instruction-following dataset.

Instruction Schema Design. Unlike general captioning tasks, our task requires strict adherence to a predefined JSON schema. We designed a comprehensive system prompt (see Appendix C.1 for full text) that includes: (1) **Role Definition:** An AI assistant specialized in invoice processing. (2) **Field Constraints:** Explicit rules for 19 fields (e.g., `type` must be chosen from a fixed list of 8 categories; `std_total` must be rounded to 2 decimal places). (3) **Negative Constraints:** Instructions on how to handle missing fields (return empty string '""' rather than 'null').

Formally, for each image I, we construct the instruction prompt P and the ground truth JSON Y. The SFT objective is to minimize the negative log-likelihood of the output tokens given the image and instruction.

4.2 Stage 2: Alignment via GRPO

While SFT instills the basic instruction-following capabilities, models often struggle with the precise trade-off between extraction recall and hallucination suppression. To align the model's behavior with the strict standards of ReceiptBench, we employ Group Relative Policy Optimization (GRPO) (Shao et al., 2024).

Metric-Aware Reward Shaping. Unlike generic RLHF which relies on a separate reward model, we construct a rule-based reward function directly derived from our evaluation protocol (Section 3.4). For each field f in the invoice, let P_f be the predicted value and G_f be the ground truth. We define the similarity score $S(P_f, G_f) \in [0, 1]$ based on the field type (e.g., Exact Match score, or the LLM-Judge score for semantic fields, or the Hungarian matching score for lists).

To handle the sparsity of invoice fields, we introduce a **Reward Shaping** mechanism based on the confusion matrix states (TP, TN, FP, FN). The reward R_f for field f is defined as follows:

$$R_f(P_f, G_f) = \begin{cases} S(P_f, G_f) & \text{if } G_f \neq \emptyset \wedge P_f \neq \emptyset \\ \lambda_{\text{TN}} & \text{if } G_f = \emptyset \wedge P_f = \emptyset \\ \lambda_{\text{FP}} & \text{if } G_f = \emptyset \wedge P_f \neq \emptyset \\ \lambda_{\text{FN}} & \text{if } G_f \neq \emptyset \wedge P_f = \emptyset \end{cases} \quad (2)$$

where the hyperparameters are set as follows:

True Positive (TP): The reward is the alignment score $S \in [0, 1]$. For semantic fields, this S is provided by the LLM Judge (Section 3.4), encouraging semantically correct answers even if they aren't exact string matches.

True Negative (TN, $\lambda_{\text{TN}} = 0.3$): We assign a modest positive reward. This encourages the model to correctly identify missing fields but prevents "mode collapse" where the model learns to maximize rewards by simply outputting empty strings for everything (which would happen if λ_{TN} were too high).

False Positive (FP, $\lambda_{\text{FP}} = -0.5$): We impose a negative penalty to explicitly suppress hallucinations, which is critical for financial document processing.

False Negative (FN, $\lambda_{\text{FN}} = 0$): No reward is given when the model fails to extract existing information.

The final reward for an invoice is the average of rewards across all 19 fields. By optimizing this shaped reward, GRPO effectively fine-tunes the model's decision boundary between "answering" and "abstaining."

5 Experiments

5.1 Experimental Setup

Data Splitting. To ensure a rigorous evaluation that reflects the diversity of real-world scenarios, we partition the ReceiptBench dataset using stratified sampling based on receipt types. We reserve 2,000 images as the held-out test set to strictly maintain the distributional consistency with the full dataset. The remaining images are utilized for training.

Baselines. We compare our fine-tuned models against a comprehensive set of state-of-the-art models, categorized into three groups: (1) **General Proprietary MLLMs**, represented by GPT-5 and Gemini-3-Pro; (2) **Open-source General MLLMs**, including the Qwen3-VL(**?**) and InternVL3(Zhu et al., 2025) series; and (3) **Specialized Document Models**, such as DianJin-OCR-R1, DeepSeek-OCR, PaddleOCR-VL, and olmOCR-7B (Chen et al., 2025; Wei et al., 2025; Cui et al., 2025; Poznanski et al., 2025).

Implementation Details. For our fine-tuned baselines, we utilize **Qwen3-VL-4/8B** and

Category	Model	Size	Overall	Perception	Normalization	Reasoning	Structure
Proprietary & API-based	GPT-5	-	0.7076	0.7304	0.8743	0.8706	0.4893
	Gemini-3-Pro	-	0.7373	0.7360	0.9086	0.8714	0.5781
	InternVL3.5-241B	241B	0.6742	0.6853	0.8791	0.8024	0.5112
	Qwen3-VL-Plus	-	0.7210	0.7306	0.9000	**0.8787**	0.5484
General Open	InternVL3-2B	2B	0.3807	0.3851	0.5821	0.4637	0.3077
	InternVL3-8B	8B	0.5772	0.5696	0.7661	0.7371	0.4609
	InternVL3-78B	78B	0.6443	0.6486	0.8444	0.7874	0.4630
	Qwen3-VL-4B	4B	0.6261	0.6407	0.7503	0.7588	0.4909
	Qwen3-VL-8B	8B	0.6545	0.6664	0.8577	0.7936	0.4792
	Qwen3-VL-32B	32B	0.6751	0.6645	0.8310	0.8690	0.4852
Specialized	DianJin-OCR-R1	7B	0.4979	0.5038	0.7126	0.5992	0.2982
	DeepSeek-OCR-small	3B	0.3959	0.3688	0.5081	0.5517	0.3637
	olmOCR-7B	7B	0.6228	0.5736	0.8030	0.7854	0.4849
	PaddleOCR-VL	0.9B	0.6344	0.6133	0.8125	0.8067	0.4746
Ours (Fine-tuned)	InternVL3-2B (SFT)	2B	0.6496	0.7077	0.8514	0.7314	0.4509
	InternVL3-2B (SFT+GRPO)	2B	0.4466	0.6279	0.4696	0.3954	0.2995
	Qwen3-VL-4B (SFT)	4B	0.7723	0.8003	0.8964	0.8184	0.6478
	Qwen3-VL-4B (SFT+GRPO)	4B	0.7788	0.8226	0.9298	0.8417	0.6215
	Qwen3-VL-8B (SFT)	8B	0.7736	0.8155	0.9180	0.8273	**0.6462**
	Qwen3-VL-8B (SFT+GRPO)	8B	**0.7950**	**0.8488**	**0.9416**	0.8547	0.6373

Table 3: **Main evaluation results (F1-score) on ReceiptBench.** Our fine-tuned models significantly outperform general baselines. While GRPO enhances overall performance for Qwen3-VL models (4B/8B) by boosting perception and reasoning, it poses stability challenges for the smaller InternVL3-2B.

InternVL3-2B as backbones due to their efficiency. For detailed training configurations, hyperparameter settings and infrastructure specifications in Appendix B.

5.2 Main Results

As shown in Table 3, the results demonstrate that domain-specific alignment is a more decisive factor than raw parameter scale. Our fine-tuned **Qwen3-VL-8B** achieves an overall F1-score of **0.7950**, significantly outperforming proprietary state-of-the-art models like Gemini-3-Pro (0.7373) and GPT-5 (0.7076). This trend extends to data efficiency, where the compact **InternVL3-2B (SFT)** (0.6496) rivals the one-shot performance of the massive InternVL3.5-241B (0.6742). While Metric-Aware GRPO successfully boosts the overall performance of Qwen3-VL models (e.g., 8B improves from 0.7736 to 0.7950), it poses stability challenges for the smaller InternVL3-2B. In the training logs, we observe *reward collapse* and *policy drift* evidenced by an initial reward increase that sharply declines around steps 220–230, accompanied by a significant spike in KL divergence. This suggests a capacity threshold for effective RL alignment, as 2B-scale models struggle to balance complex structural constraints against fundamental linguistic coherence. Conversely, specialized document models like DeepSeek-OCR and olmOCR exhibit significant performance drops in reasoning and structure generation, revealing that strong perceptual capabilities alone are insufficient for complex logic extraction. Furthermore, parsing nested structures remains the bottleneck across all baselines.

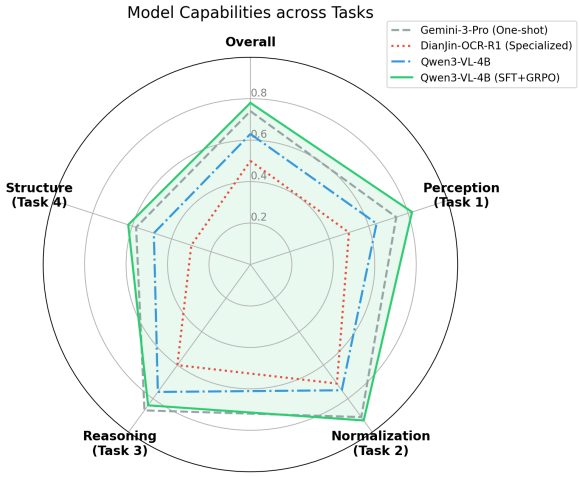


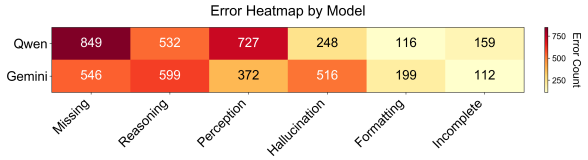
Figure 2: **Holistic Evaluation.** The chart compares model capabilities across the four sub-tasks. While proprietary models (gray) are balanced, our fine-tuned baseline (green) excels in domain-specific structure parsing.

While GPT-5 scores only 0.4893 on this metric, our SFT approach substantially improves this to **0.6478** (Qwen3-VL-4B), validating the effectiveness of our pipeline in handling heterogeneous layouts.

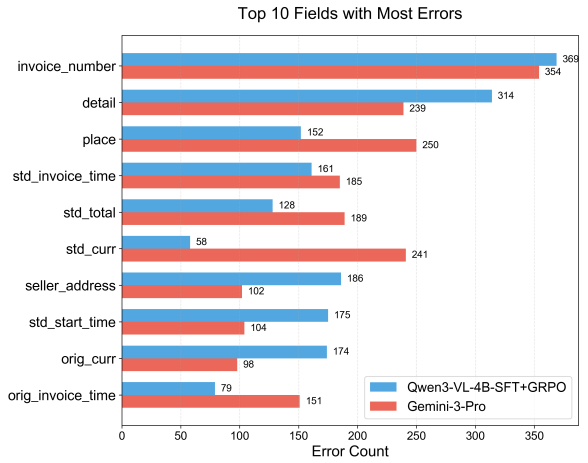
To ensure these findings are not skewed by the dominance of English or specific receipt types, we further evaluated our models on a curated category-balanced test set and a non-English subset. As detailed in Appendix D.1, the relative performance rankings remain strictly consistent, demonstrating the cross-lingual and cross-category robustness of our framework.

5.3 Analysis

To understand the capability boundaries of current MLLMs, we conducted a fine-grained error analysis comparing our fine-tuned **Qwen3-VL-4B**



(a) **Error Type Distribution.** Qwen3-VL (Ours) shows a "conservative" pattern with high *Missing* rates (849), whereas Gemini-3-Pro is "aggressive" with high *Hallucination* (516) and *Reasoning* errors.



(b) **Top 10 Error-Prone Fields.** `invoice_number` and `detail` are the hardest fields. Note the significant gap in `place`, indicating Gemini's tendency to over-interpret location context.

Figure 3: **Fine-grained Error Analysis on Receipt-Bench.** We compare the error patterns of our fine-tuned Qwen3-VL-4B against Gemini-3-Pro. (a) illustrates the divergent behavioral profiles, while (b) highlights the specific fields that pose the greatest challenges.

against the proprietary **Gemini-3-Pro**. As illustrated in Figure 3, we categorize the failure modes into three distinct patterns.

Perception Bottlenecks: Visual Ambiguity vs. Hallucination. As illustrated in Figure 3, the perception task reveals a fundamental behavioral divergence between the two models. Qwen3-VL suffers predominantly from *Missing* errors (849 cases) and *Perception* errors (727 cases). It struggles with fine-grained OCR in dense layouts, often failing to detect fields like `orig_curr` or misidentifying confusing digits (e.g.,"1"/"7") in `invoice_number`—which ranks as the most error-prone field for both models (Figure 3b). Conversely, Gemini exhibits a high tendency for *Hallucination* (Gemini: 516 cases vs. Qwen: 248). When a unique ID is visually absent, Gemini often fabricates a plausible string for `invoice_number` to satisfy the schema, rather than outputting an empty string. This highlights the challenge of grounding generation strictly in visual evidence.

Reasoning Gaps: Contextual Inference. Beyond perception, **Reasoning and Normalization** tasks account for the majority of Gemini's failures (599 Reasoning errors) exposing critical deficiencies in utilizing global context. The Field Error Ranking (Figure 3b) highlights a massive performance gap in the `place` field (Gemini: 250 vs. Qwen: 152). Gemini frequently hallucinates specific cities based on currency cues (e.g., inferring "London" from "£"), whereas Qwen tends to remain conservative when the address is ambiguous. Similarly, in `std_curr`, Gemini produces significantly more errors (241 vs. 58). Models often default to USD when the symbol "$" is ambiguous, failing to cross-reference the `seller_address` to correctly infer CAD or AUD. Ambiguous date formats (e.g., "02/03/24") lead to swapping Day/Month. Gemini's higher error count in *Formatting* (199 cases) suggests it often ignores specific normalization instructions compared to the fine-tuned Qwen.

Consistency Trap in Structural Parsing. The `detail` field ranks as the second most difficult field in Figure 3b. A critical finding in the **Structure** task is the phenomenon of *"Hallucination for Arithmetic Consistency."* Complex invoices imply constraints (e.g., $\sum$ items $=$ Total). Models, particularly Gemini, often tamper with visual data to satisfy these priors: **Value Tampering:** To force the sum of `detail` items to match the `std_total`, models occasionally alter the price of a line item or hallucinate a non-existent "Tax" item. **Unwanted Calculation:** When the total amount is visually missing, models attempt to manually sum up line items to generate a `std_total`, leading to calculation errors.

This underscores the value of our Metric-Aware GRPO arithmetic reward, which enforces logical consistency without compromising visual faithfulness.

5.4 Ablation Studies

Effect of Training Stages. We analyze the contribution of each stage using Qwen3-VL-4B. As shown in Table 4, **SFT** establishes a critical foundation, yielding massive gains in *Perception* (+16.0%) and *Normalization* (+14.6%) over the one-shot baseline by teaching schema adherence. The introduction of **Metric-Aware GRPO** further refines these capabilities. Interestingly, "GRPO Only" achieves the highest *Reasoning* score (0.8560), in-

dicating RL's potency in optimizing logic, yet it lags in visual grounding. Consequently, the combined **SFT + GRPO** strategy achieves the optimal balance, delivering state-of-the-art results in *Perception* (**0.8226**) and *Normalization* (**0.9298**) while maintaining strong reasoning gains. Crucially, this RL alignment explicitly suppresses hallucinations. Quantitative analysis confirms that Metric-Aware GRPO reduces False Positives (FPs) by up to 68.9% in complex fields while significantly boosting overall precision (see Appendix D.2).

Stage	Perception	Normalization	Reasoning
One-shot	0.6407	0.7503	0.7588
SFT Only	0.8003	0.8964	0.8184
GRPO Only	0.7758	0.8782	**0.8560**
SFT + GRPO	**0.8226**	**0.9298**	0.8417

Table 4: Ablation of training stages. SFT enables robust visual grounding and formatting, while GRPO is essential for maximizing reasoning capabilities.

6 Conclusion

We introduce **ReceiptBench**, a benchmark designed to propel Visual Information Extraction (VIE) from literal extraction toward cognitive reasoning. Moving beyond retail-centric datasets, ReceiptBench comprises 10k diverse overseas financial documents with a hierarchical taxonomy covering Perception, Normalization, Reasoning, and Structure. To tackle these challenges, we propose a two-stage training framework combining SFT with **Metric-Aware GRPO**. Experiments demonstrate that while SFT establishes a solid foundation, our RL alignment significantly mitigates hallucinations and improves arithmetic consistency. However, the persistent performance gap in structural parsing highlights that handling nested layouts remains an open research problem. We hope ReceiptBench serves as a rigorous testbed for next-generation multimodal agents, fostering advancements in autonomous financial auditing.

Limitations

While ReceiptBench represents a significant step forward, it has certain limitations.

First, although the dataset covers multiple languages, it is predominantly English-centric (97.9%), reflecting the data availability in open web sources. Future work should focus on scaling low-resource languages to improve multilingual robustness.

Second, to strictly protect privacy, all PII (Personally Identifiable Information) was masked. While necessary, this may slightly alter the visual distribution compared to raw private financial data found in internal corporate streams.

Third, we did not perform systematic visual data augmentation (e.g., rotation, gaussian blur, or noise injection) during the evaluation. While our dataset contains natural visual variations from real-world collection, we have not explicitly stress-tested the models' robustness against severe visual degradations or adversarial attacks.

Finally, our proposed GRPO training method, while effective, incurs a higher computational cost compared to standard SFT. Developing more data-efficient alignment strategies for MLLMs remains a valuable direction for future exploration.

Ethics Statement

Given that our data originates from real-world transactions, we enforced a strict de-identification policy where sensitive PII (e.g., personal names) was detected and masked with **irreversible black boxes** during annotation, ensuring effective anonymity by rendering private information visually and digitally inaccessible.

Acknowledgments

This work was supported in part by the Ningbo Youth Science and Technology Innovation Leading Talent Program (No. 2025QL059), CCF-1688 Yuanbao Collaborative Fund (No. CCF-Alibaba 2025004), the "Pioneer and Leading Goose" R&D Program of Zhejiang (No. 2025C02037), the Zhejiang Provincial Philosophy and Social Sciences Planning Project (No. 22QNYC04ZD), the National Social Science Fund of China (No. 24BGL071), and the Fundamental Research Funds for the Central Universities.

We gratefully acknowledge Ziman Li for her assistance in designing the annotation schema and guidelines; Xiaoqing Liu, Yongbo Wang, and Lufei Xu for their help with receipt image collection; Enci Zhang, Xiang Li, Wuyou Mao, Yingtian Hu, and Shujian Zhu for their contributions to annotated data validation; and Qi Yang and Yuan Liu, together with the above validation team, for error analysis during model iterations.

References

Abdelrahman Abdallah, Mohamed Mounis, Mahmoud Abdalla, Mahmoud SalahEldin Kasem, and 1 others. 2024. Receiptsense: Beyond traditional ocr - a dataset for receipt understanding. *arXiv preprint arXiv:2406.04493*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, and 1 others. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. 2019. Scene text visual question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4291–4301.

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

Qian Chen, Xianyin Zhang, Lifan Guo, Feng Chen, and Chi Zhang. 2025. Dianjin-ocr-r1: Enhancing ocr capabilities via a reasoning-and-tool interleaved vision-language model. *arXiv preprint arXiv:2508.13238*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198.

Council of the European Union. 2006. Council directive 2006/112/ec of 28 november 2006 on the common system of value added tax.

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*.

Financial Accounting Standards Board (FASB). 2010. Statement of financial accounting concepts no. 8: Conceptual framework for financial reporting. *Financial Accounting Series*. Chapter 1: The Objective of General Purpose Financial Reporting.

Anwen Hu, Haiyang Xu, Jiabo Ye, Ming Yan, Liang Zhang, Bo Zhang, Ji Zhang, Qin Jin, Fei Huang, and Jingren Zhou. 2024. mplug-docowl 1.5: Unified structure learning for ocr-free document understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3096–3120.

Mingxin Huang, Yongxin Shi, Dezhi Peng, Songxuan Lai, Zecheng Xie, and Lianwen Jin. 2025. Ocr-reasoning benchmark: Unveiling the true capabilities of mllms in complex text-rich image reasoning. *arXiv preprint arXiv:2505.17163*.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM international conference on multimedia*, pages 4083–4091.

Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and CV Jawahar. 2019. Icdar2019 competition on scanned receipt ocr and information extraction. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, pages 1516–1520. IEEE.

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

Guillaume Jaume, Hazım Kemal Ekenel, and Jean-Philippe Thiran. 2019. Funsd: A dataset for form understanding in noisy scanned documents. In *Proceedings of the International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, pages 1–6. IEEE.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. Ocr-free document understanding transformer. In *European Conference on Computer Vision*, pages 498–517. Springer.

Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

Mahmoud Limam, Marwa Dhiaf, and Yousri Kessentini. 2025. Information extraction from multi-layout invoice images using fatura dataset. *Engineering Applications of Artificial Intelligence*, 149:110478.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2200–2209.

Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. 2019. Cord: A consolidated receipt dataset for post-ocr parsing. In *Workshop on Document Intelligence at NeurIPS*.

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*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Hongbin Sun, Zhanghui Kuang, Xiaoyu Yue, Chenggang Lin, and Wayne Zhang. 2021. Spatial dual-modality graph reasoning for key information extraction. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5229–5237.

Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. 2023. Hierarchical multimodal transformers for multipage docvqa. *Pattern Recognition*, 144:109834.

Jordy Van Landeghem, Rubèn Tito, Łukasz Borchmann, Michał Pietruszka, Pawel Joziak, Rafal Powalski, Dawid Jurkiewicz, Mickaël Coustaty, Bertrand Anckaert, Ernest Valveny, and 1 others. 2023. Document understanding dataset and evaluation (dude). In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19528–19540.

Jiapeng Wang, Chongyu Lian, Wenjing Wang, Xudong Ying, and Baoyuan Wang. 2021. Towards robust visual information extraction in real world: New dataset and novel solution. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

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*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200.

Yiheng Xu, Tengchao Lv, Lei Cui, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, and Furu Wei. 2022. Xfund: A benchmark dataset for multilingual visually rich form understanding. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3214–3224.

Zhibo Yang, Jun Tang, Zhaohai Li, Pengfei Wang, Jianqiang Wan, Humen Zhong, Xuejing Liu, Mingkun Yang, Peng Wang, Shuai Bai, and 1 others. 2025. Cc-ocr: A comprehensive and challenging ocr benchmark for evaluating large multimodal models in literacy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21744–21754.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang, and 1 others. 2023. Ureader: Universal ocr-free visually-situated language understanding with multimodal large language model. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2841–2858.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 3: system demonstrations)*, pages 400–410.

Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. 2019. Publaynet: largest dataset ever for document layout analysis. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1015–1022. IEEE.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, and 1 others. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

A Dataset Details & Field Specifications

Our dataset defines 19 fields designed to provide a comprehensive evidence chain for financial auditing. The selection of these fields is grounded in the *Generally Accepted Accounting Principles (GAAP)* (Financial Accounting Standards Board (FASB), 2010) and international tax regulations (e.g., EU VAT Directive (Council of the European Union, 2006)), ensuring the benchmark's utility for real-world financial auditing. The annotation rules for each field are detailed below.

1. Entity Verification This dimension focuses on identifying the stakeholders involved in the transaction to establish legitimacy.

- **seller_name**: The name of the merchant or service provider. As these refer to public business entities, they are not considered PII. The annotation must be faithful to the visual information on the receipt (e.g., logos, headers).
- **seller_address**: The city that the merchant located, formatted as "Country-City" (e.g., *UK-London*). Inferring addresses via external search engines is strictly prohibited to ensure the dataset reflects only the information contained in the image.
- **invoice_number**: The unique identifier of the receipt or invoice. Common labels include "Invoice No.", "Receipt No.", "Confirmation No." or "Ticket No.". If multiple numbers exist (e.g., Order No. and Invoice No.), the Invoice Number takes precedence as the primary financial identifier.
- **tax_number**: The tax identification number of the merchant (e.g., VAT ID, GST No., TIN).

2. Financial Integrity This dimension captures critical financial data to verify calculations and amounts.

- **orig_total**: The total amount of the transaction as it appears visually in the raw text. This field captures the exact string from the document, including original separators (e.g., `1.000,00`), without any normalization.
- **std_total**: The normalized total amount for computational verification. The value is standardized to a decimal format with two decimal places (e.g., `1,000.00`). Thousands separators are unified to commas. Logic rules dictate that this should be the final amount payable, inclusive of taxes and tips.
- **orig_curr**: Visual evidence of the currency. This includes symbols (e.g., $, €), text abbreviations (e.g., USD, RMB), or geographic clues (e.g., "Toronto" implying Canadian Dollars) explicitly found on the image.
- **std_curr**: The standardized 3-letter ISO currency code (e.g., USD, EUR, GBP, CNY). This is inferred from the `orig_curr` evidence.
- **detail**: A structured list containing line items to verify the breakdown of the total amount. This is a complex field where each item is a JSON object containing three subcomponents:
  - `content`: The description of the product or service.
  - `amount`: The numerical value of the specific item.
  - `ifTax`: A boolean flag (`True`/`False`) indicating whether the item represents a tax charge (e.g., VAT, GST).

  Annotators ensure that the summation of these line items logically aligns with the `std_total`.

3. Spatio-Temporal Validation This dimension validates when and where the expense occurred to ensure the context matches the business trip or transaction claim.

- **place**: The location where the expense occurred, formatted as "Country-City" (e.g., *UK-London*). If the document only specifies a city, the country is added; if only the country is visible, the city is left blank.
- **departure**: The origin city for transportation tickets. This applies to cross-city travel (plane, train, bus). If a trip involves multiple segments (e.g., A-B-A), only the initial departure point is recorded.
- **arrival**: The destination city for transportation tickets. Similar to `departure`, this captures the endpoint of the travel service.
- **orig_start_time**: The raw text indicating the start of the service or event. It preserves the original date format found on the image (e.g., "15-July-24").

Field Name	Definition	Sub-task Category	Metric
orig_start_time	Start time of service as it appears visually (raw text)	Basic Perception	Semantic
orig_end_time	End time of service as it appears visually (raw text)	Basic Perception	Semantic
orig_invoice_time	Issuance time as it appears visually (raw text)	Basic Perception	Semantic
orig_total	Total amount as it appears visually (raw text)	Basic Perception	Numeric
orig_curr	Currency clues like symbol or city & country as it appears visually (e.g., $)	Basic Perception	Structured List
invoice_number	Unique identifier of the receipt/invoice	Basic Perception	Exact Match
tax_number	Tax identification number of the merchant	Basic Perception	Exact Match
seller_name	Name of the merchant or service provider	Basic Perception	Semantic
std_start_time	Start time normalized to YYYY-MM-DD format	Formatting & Normalization	Exact Match
std_end_time	End time normalized to YYYY-MM-DD format	Formatting & Normalization	Exact Match
std_invoice_time	Issuance time normalized to YYYY-MM-DD format	Formatting & Normalization	Exact Match
std_total	Total amount normalized to decimal format (e.g., 1,000.00)	Formatting & Normalization	Numeric
type	Classification of expense (e.g., Hotel, Train, Taxi)	Semantic Reasoning	Exact Match
place	Location where the expense occurred	Semantic Reasoning	Semantic
departure	Origin city (for transport tickets)	Semantic Reasoning	Semantic
arrival	Destination city (for transport tickets)	Semantic Reasoning	Semantic
std_curr	Standardized ISO currency code inferred from context (e.g., USD)	Semantic Reasoning	Exact Match
seller_address	City that the merchant locates	Semantic Reasoning	Semantic
detail	Structured list of line items (content, amount, tax status)	Structural Parsing	Structured List

Table 5: Basic definitions, categories and metrics of the 19 annotation fields in our dataset. These fields are categorized into four sub-tasks based on the required cognitive capability (see Section 3.3). The evaluation metrics are defined in Section 3.4.

Language	Count	Pct.
English	10,443	98.00%
French	60	0.56%
Spanish	51	0.49%
German	31	0.29%
Indonesian	31	0.29%
Portuguese	18	0.18%
Romanian	10	0.09%
Others	11	0.10%
Non-English Total	**213**	**2.00%**
Total	**10,656**	**100.0%**

Table 6: Language distribution of the dataset. "Others" includes Italian, Korean, and Japanese.

- **std_start_time**: The normalized start date converted to the ISO `YYYY-MM-DD` format (e.g., `2024-07-15`). This facilitates temporal reasoning. Logic rules handle ambiguous formats (e.g., 07/06/24) by cross-referencing the country's date convention.

- **orig_end_time**: The raw text indicating the end of the service (e.g., hotel check-out, flight arrival). If the transaction occurs on a single day, this field should be left empty.

- **std_end_time**: The normalized end date converted to `YYYY-MM-DD`.

- **orig_invoice_time**: The raw text indicating when the invoice/receipt was issued. For on-the-spot receipts (e.g., retail receipts), this is identical to the transaction time; for post-paid invoices, it may differ from the service period.

- **std_invoice_time**: The normalized issuance date converted to `YYYY-MM-DD`.

4. Expense Classification This dimension categorizes the nature of the transaction for accounting and reimbursement purposes.

- **type**: A classification label selected from a standardized list: *plane, train, ship, bus, taxi, metro, hotel*, or *other*. Annotators determine this based on explicit keywords (e.g., "Flight" $\rightarrow$ *plane*) or implicit logic (e.g., "Double Room" $\rightarrow$ *hotel*).

Table 5 shows the basic definitions, sub-task categories, and metrics of these 19 fields. Table 6 shows the language distribution of the dataset.

B Implementation Details

B.1 Implementation Details and Hyperparameters

We utilized the LLaMA-Factory framework (Zheng et al., 2024) to fine-tune the **Qwen3-VL** series and **InternVL-3** models. The training configurations were set as follows: In the **SFT stage**, models are trained for 2 epochs with a global batch size of 16 (achieved via gradient accumulation steps of 8 on single-device batches), a learning rate of $1e-5$ with a cosine decay scheduler, and BF16 mixed-precision. Notably, we set the maximum context length to **5,120 tokens** to accommodate receipts with long lists of items (the `detail` field), ensuring no information truncation during training. In the **GRPO stage**, we employ the reward function defined in Section 4, setting the KL coefficient to 0.01 and collecting 16 samples per prompt for

policy updates. All experiments are conducted on 4×NVIDIA A800 GPUs.

C Evaluation Details

C.1 Prompts for Instruction Tuning and Inference

To ensure the model adheres to the strict output schema required by ReceiptBench, we designed a comprehensive system prompt. Table 7 illustrates the exact prompt used during both the Supervised Fine-Tuning (SFT) and inference stages. The prompt consists of three components: (1) a role definition and format constraint, (2) detailed extraction rules for each field, and (3) a one-shot demonstration to guide the JSON structure.

C.2 Prompt for LLM Semantic Judge

For fields requiring semantic reasoning (e.g., `place`, `seller_name`), we employ a lightweight LLM as a judge when exact matching fails. Table 8 details the instruction provided to the judge model to determine semantic equivalence.

D Additional Results

D.1 Robustness across Languages and Categories

To address potential evaluation biases arising from data distribution, we conducted robustness checks on two specific subsets: a curated category-balanced test set and a non-English subset.

Category-Balanced Evaluation. We constructed a balanced test set comprising 1,387 samples by down-sampling dominant categories (e.g., Purchase, Dining) to match the frequency of minority classes. As shown in Table 9, while the absolute F1 scores shifted slightly due to the altered distribution, the relative ranking of the models remained highly consistent, with our SFT+GRPO framework maintaining its superior performance.

Cross-Lingual Robustness. We also evaluated the models on the 2% non-English subset. As detailed in Table 10, despite the data scarcity for these low-resource languages, our fine-tuned models exhibit significant improvements. Notably, the Qwen3-VL-8B (SFT+GRPO) achieves a leading F1 score of 0.7190, outperforming both GPT-5 (0.6441) and Gemini-3-Pro (0.6827). This demonstrates that our Metric-Aware GRPO method successfully enables the model to capture universal layout and structural patterns, effectively mitigating the impact of language barriers.

D.2 Quantitative Proof of Hallucination Suppression

Our Metric-Aware GRPO explicitly penalizes hallucinations through negative rewards for False Positives (FP). To quantitatively demonstrate this, we compared the Precision scores and the absolute FP counts before and after RL alignment.

As shown in Table 11, both Qwen3-VL-4B and 8B models exhibit significant improvements in Precision after GRPO training (e.g., the 8B model's overall Precision increased from 0.8319 to 0.8794). Furthermore, Table 12 illustrates a consistent reduction in the absolute number of False Positives across various fields. Notably, hallucinated predictions for the complex `detail` field dropped by 20.0%, and errors in `std_invoice_time` decreased sharply by 68.8%. These quantitative results confirm that the performance gains of our framework are heavily driven by substantial hallucination suppression.

D.3 Hyperparameter Sensitivity Analysis for Structural Similarity

To ensure the structural parsing similarity score in Equation (1) robustly reflects true semantic understanding, the weights $(\alpha, \beta, \gamma, \delta)$ were determined through a rigorous empirical validation process. Furthermore, as requested during the review phase, we conducted a sensitivity analysis to confirm that minor variations in these hyperparameters do not alter the relative rankings of the evaluated models.

Weight Optimization. We collected a validation set of 400 complex structural prediction samples. Three human annotators labeled whether the model predictions were semantically equivalent to the ground truth (accounting for acceptable variations where strict string matching fails). Through a grid search, we evaluated different weight configurations based on their alignment accuracy with human annotations.

Sensitivity and Ranking Stability. We selected four representative weight configurations to test the stability of our benchmark:

- **Config A (Optimal):** $\alpha = 0.3, \beta = 0.2, \gamma = 0.1, \delta = 0.4$. Achieves the highest human alignment accuracy (**92%**).
- **Config B (Equal Weights):** $\alpha = 0.25, \beta =$

System Instruction

You are an AI assistant specialized in extracting structured information from images of overseas travel and expense invoices. Your task is to analyze each invoice image and convert the relevant information into a structured JSON object. Follow the field definitions and annotation rules below precisely. The output must be a valid JSON object containing all required key-value pairs, using exact formats as described.

OUTPUT FORMAT

Your output must be a single **valid JSON object** for each invoice. Do not include any explanatory text or formatting like markdown. Below are the output rules for the key-value pairs in the json objects. Specifically, if a field is missing or cannot be identified, return it as the empty string `""`.

EXTRACTION RULES

- `type`: output a string, infer from keywords/context and choose within ["plane", "train", "ship", "bus", "taxi", "metro", "hotel", "other"].
- `orig_start_time`: output a string, start time of the event. Keep original format.
- `orig_end_time`: output a string, end time of the event. Keep original format. Return empty if within 1 day.
- `orig_invoice_time`: output a string, date of issuance. Evidence needed. Keep original format.
- `std_start_time`: output a string, "YYYY-MM-DD" format of `orig_start_time`.
- `std_end_time`: output a string, "YYYY-MM-DD" format of `orig_end_time`.
- `std_invoice_time`: output a string, "YYYY-MM-DD" format of `orig_invoice_time`.
- `place`: output a string, "Country-City" format. Only valid if clearly stated.
- `departure`: output a string, "Country-City" format. Only for intercity travel.
- `arrival`: output a string, "Country-City" format. Only for intercity travel.
- `orig_curr`: output a list of evidence strings (e.g., "$", "Toronto"). Keep original content.
- `std_curr`: output a string, 3-letter standard ISO code (e.g. USD, EUR, CAD).
- `orig_total`: output a string, total amount. Keep original content. No symbol needed.
- `std_total`: output a string, standard format with comma separator, 2 decimal places (e.g. "1,200.00").
- `detail`: output a list of dicts: `[ {"content": "...", "amount": "...", "ifTax": True/False} ]`. Exclude subtotals.
- `seller_name`: output a list of seller names. Valid only when explicitly present.
- `seller_address`: output a list of seller cities in ["Country-City"] format.
- `invoice_number`: output a string. Strip prefixes like "No.". Prefer official IDs.
- `tax_number`: output a string. Strip prefixes like "Tax ID".

FEW-SHOT EXAMPLE

```
{
  "type": "train",
  "std_start_time": "2024-07-06", "orig_start_time": "06 Jul 2024",
  "std_end_time": "", "orig_end_time": "",
  "std_invoice_time": "2024-06-03", "orig_invoice_time": "03 Jun 2024",
  "place": "Australia-Sydney",
  "departure": "Australia-Sydney", "arrival": "Australia-Canberra",
  "std_curr": "AUD", "orig_curr": ["$", "Sydney"],
  "std_total": "50.58", "orig_total": "50.58",
  "detail": [
    { "content": "Trip Fare", "amount": "45.00", "ifTax": False },
    { "content": "Tax fee", "amount": "5.58", "ifTax": True }
  ],
  "seller_name": ["NSW TrainLink"],
  "seller_address": ["Australia-Sydney"],
  "invoice_number": "0306202450122",
  "tax_number": "50 325 560 455"
}
```

USER INPUT

Extract now (JSON only, no explanation):

Table 7: The full system prompt used for SFT and inference on ReceiptBench. The prompt enforces schema constraints, defines normalization rules, and provides a one-shot demonstration to guide the model's output format.

System Instruction

You are an expert data quality analyst. Your task is to determine if the 'Predicted Value' is semantically equivalent to the 'Ground Truth Value' for a specific field extracted from a document.

Context

- Field Name: `<field_name>`

Equivalence Criteria

Consider the values **equivalent** if they represent the same real-world entity or meaning, even with minor differences like:

- Abbreviations (e.g., "Co." vs. "Company").
- Common synonyms or alternative names.
- Minor typos or spelling errors that do not change the meaning.
- Formatting differences (e.g., "1,234.50" vs. "1234.50").
- Presence or absence of trivial words (e.g., "The Grand Hotel" vs. "Grand Hotel").

Consider the values **NOT equivalent** if:

- They refer to different entities (e.g., "Pepsi" vs. "Coca-Cola").
- The core information is different (e.g., a different address or name).
- The prediction contains significant missing or extra information that changes the meaning.

Task

Based on the criteria above, evaluate the following pair:

- Ground Truth Value: "`<ground_truth>`"
- Predicted Value: "`<prediction>`"

Output

Respond ONLY with a valid JSON object containing two keys:

1. `"is_equivalent"`: A boolean value (true or false).
2. `"reasoning"`: A brief explanation for your decision.

Table 8: The full prompt used for the LLM-based Semantic Judge. This prompt is triggered only when the Levenshtein similarity between the prediction and ground truth falls below the exact match threshold.

Model	Overall	Perc.	Norm.	Reason.	Struct.
Gemini-3-Pro	0.6863	0.6817	0.9022	0.8019	0.5481
Qwen3-VL-8B	0.6544	0.6690	0.8554	0.7975	0.4770
Qwen3-VL-8B (SFT)	0.7639	0.8130	0.9138	0.8285	**0.6366**
Qwen3-VL-8B (SFT+GRPO)	**0.7861**	**0.8478**	**0.9377**	**0.8548**	0.6112

Table 9: Evaluation results on the category-balanced test set.

Model	Overall	Perc.	Norm.	Reason.	Struct.
GPT-5	0.6441	0.6968	0.8299	0.8112	0.3058
Gemini-3-Pro	0.6827	0.7123	0.8396	0.8092	**0.4305**
Qwen3-VL-4B	0.5906	0.6172	0.7593	0.7579	0.3143
Qwen3-VL-4B (SFT)	0.6574	0.7468	0.7212	0.7781	0.4081
Qwen3-VL-4B (+GRPO)	0.6832	0.7532	0.7681	0.8432	0.3429
Qwen3-VL-8B	0.6193	0.6504	0.7733	0.8011	0.2954
Qwen3-VL-8B (SFT)	0.6745	**0.7920**	0.7491	0.8011	0.3531
Qwen3-VL-8B (+GRPO)	**0.7190**	0.7912	**0.8620**	**0.8441**	0.3810

Table 10: Evaluation results on the non-English subset.

$0.25, \gamma = 0.25, \delta = 0.25$. Achieves **91%** human alignment.

- **Config C (Lexical-Heavy):** $\alpha = 0.4, \beta = 0.3, \gamma = 0.3, \delta = 0.0$. Drops semantic embeddings entirely. Achieves **88%** human alignment.
- **Config D (Semantic-Heavy):** $\alpha = 0.0, \beta = 0.3, \gamma = 0.3, \delta = 0.4$. Heavily penalizes Levenshtein distance, focusing on semantics and token matching. Achieves **90%** human alignment.

As shown in Table 13, we re-evaluated five leading models across these distinct configurations. While the absolute Overall F1 scores exhibit minor fluctuations depending on the strictness of the weights, the **relative ranking of the models remains strictly consistent** (Qwen3-VL-8B (+GRPO) > Qwen3-VL-8B (SFT) > Gemini-3-Pro

Model (Precision)	Overall	Perc.	Norm.	Reason.	Struct.
Qwen3-VL-4B (SFT)	0.8134	0.8527	0.9329	0.9271	0.6756
Qwen3-VL-4B (+GRPO)	0.8693	0.9022	0.9662	0.9664	0.7479
Qwen3-VL-8B (SFT)	0.8319	0.8590	0.9349	0.9502	0.7351
Qwen3-VL-8B (+GRPO)	**0.8794**	**0.9099**	**0.9690**	**0.9703**	**0.7664**

Table 11: Precision improvement after GRPO alignment.

Field Name	SFT (FP)	+GRPO (FP)	Change	Change (%)
`std_invoice_time`	410	128	-282	-68.8%
`invoice_number`	398	142	-256	-64.3%
`arrival`	53	27	-26	-49.1%
`std_curr`	68	38	-30	-44.1%
`seller_address`	147	116	-31	-21.1%
`detail`	2079	1664	-415	-20.0%

Table 12: Reduction of False Positives (FP) for the Qwen3-VL-8B model across representative fields after Metric-Aware GRPO training.

> Qwen3-VL-Plus > GPT-5) across all scenarios. This empirical proof firmly validates that our evaluation metric is robust, and the superior reasoning and structural capabilities of our Metric-Aware GRPO framework are not artifacts of hyperparameter selection.

Model	Overall F1 Score under Different Configs			
	Config A (Opt.)	Config B	Config C	Config D
GPT-5	0.7076	0.7015	0.6945	0.7070
Gemini-3-Pro	0.7373	0.7305	0.7306	0.7312
Qwen3-VL-Plus	0.7210	0.7176	0.7170	0.7183
Qwen3-VL-8B (SFT)	0.7736	0.7685	0.7672	0.7693
Qwen3-VL-8B (+GRPO)	**0.7950**	**0.7938**	**0.7937**	**0.7945**

Table 13: Hyperparameter sensitivity analysis on the Overall F1 score. The evaluation metric remains highly stable, with the relative performance rankings strictly preserved regardless of the weight distribution.

D.4 Qualitative Case Study

To visually demonstrate the challenges of Receipt-Bench and the effectiveness of our training pipeline, we present a detailed comparison between the **One-shot Base Model (Qwen3-VL-4B)** and our final **Fine-tuned Model (Ours, SFT+GRPO)** on a complex hotel receipt.

As shown in Figure 4, the input image is a hotel folio from *EconoLodge*. This sample features a scattered layout with multiple address blocks and a "Balance Due" table, posing significant cognitive hurdles. The comparison highlights four key improvements:

1. Spatial Reasoning and Disambiguation (Task 3). The document contains two distinct addresses: the hotel's physical address (top, "Ridgecrest") and the customer's billing address (bottom-left, "Carlsbad"). The Base Model creates a hallucination by concatenating "United States" with the distractor address "Carlsbad" for the `place` field. This is a typical spatial reasoning failure. Our model, aligned via SFT+GRPO, correctly identifies the semantic role of the top address block, accurately extracting "USA-Ridgecrest".

2. The "Balance Due" Trap (Task 2 & 3). For the `std_total` field, the Base Model extracts "0.00" because the receipt explicitly states "Total Balance Due: $0.00" (indicating the bill has been paid). This reveals a lack of financial logic in general-purpose models. Our model correctly reasons that the effective transaction amount is the sum of charges (or the payment amount), correctly extracting "79.33".

3. Semantic Mapping of Identifiers and Dates (Task 1). The receipt does not explicitly label an "Invoice Number" or "Invoice Date" using standard terminology. Instead, it uses the term "Account: 744376528" for the invoice identifier and presents the issuance date under the heading "Date". The Base Model fails to recognize these semantic synonyms, returning *Missing* for both fields. In contrast, our model successfully maps the semantically equivalent "Account" to the target `invoice_number` field and "date" to `orig_invoice_date` field, demonstrating robust domain adaptation and semantic understanding.

4. Structural Completeness (Task 4). In the `detail` list extraction, the Base Model misses the last line item ("Tourism Levy"), likely due to its visual separation from the main table body or its small font size. Our model achieves full recall, capturing all line items including the tax details. This structural completeness is crucial for the arithmetic consistency reward used during GRPO training.

In summary, this case illustrates that **One-shot General MLLMs often fail to distinguish semantic roles (e.g., Service vs. Billing address) and lack domain-specific financial logic (e.g., Total vs. Balance).** Our dataset and training pipeline effectively bridge these gaps.

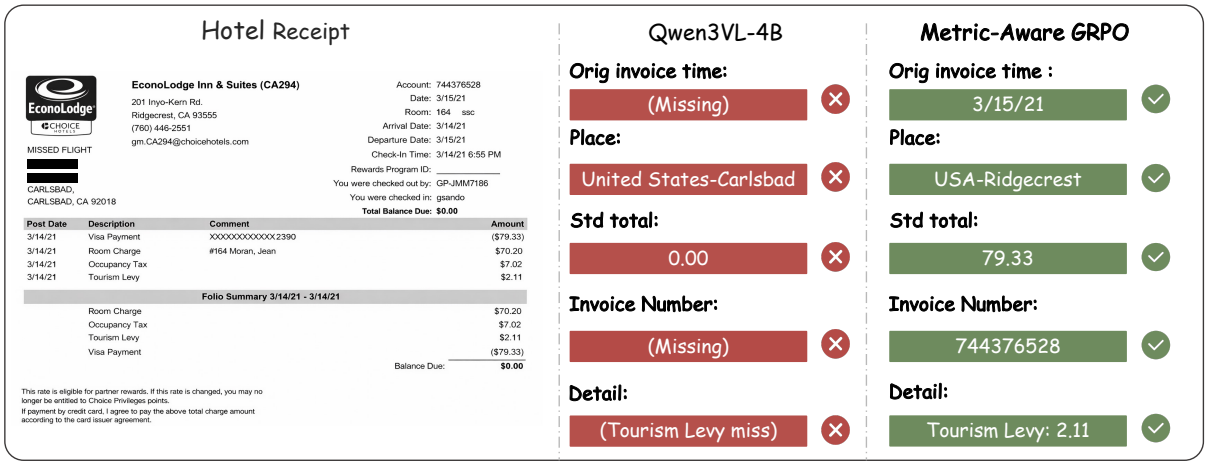


Figure 4: **Qualitative comparison on a complex hotel folio.** The **One-shot Base Model** (middle) falls into common visual and logical traps: extracting the billing address instead of the hotel location, and mistaking the "Balance Due" (0.00) for the total amount. In contrast, our **Fine-tuned Model** (right) correctly infers the semantic roles of fields and adheres to financial logic.