

Vision is Not Enough: Enhancing Bounding Box Grounding for Document Key Information Extraction with OCR Anchors

Zhaochen Li, Rathika Suresh Kumar, Tao Bai,
Zhijie Zhang, Hongxin Shao, Daniel Dahlmeier
SAP

{zhaochen.li, rathika.suresh.kumar, tao.bai, zhijie.zhang, hongxin.shao, d.dahlmeier}@sap.com

Abstract

Vision Language Model (VLM)-based OCR-free pipelines are increasingly adopted for industrial Key Information Extraction (KIE) systems due to their simpler end-to-end architecture and reduced dependence on potentially noisy OCR outputs. However, their reliability in grounding extracted values to precise bounding boxes remains unclear, raising questions about their suitability for grounding-critical use cases. In this paper, we present a systematic study of bounding box grounding capability in both general and document-specific VLMs. Our findings reveal a consistent extraction-grounding gap across model families. We revisit the OCR system as a grounding primitive and find that delegating the grounding step to Large Language Models (LLMs), rather than VLMs, in a zero-shot manner substantially improves region-level grounding accuracy. We introduce a fine-tuned 0.6B LLM that further improves the predicted box alignment, achieving high overlap with the target field for in-domain data while maintaining low serving latency. These results position OCR-anchored grounding as a practical deployment choice for industrial KIE systems that require both extraction and precise localization.

1 Introduction

Vision-Language Models (VLMs) are increasingly replacing traditional OCR-based pipelines for document Key Information Extraction (KIE) (Bai et al., 2025; Comanici et al., 2025), offering unified end-to-end architectures that avoid error propagation from standalone OCR modules (Yang et al., 2025b; Fu et al., 2025). Beyond extracting field values, certain industrial use cases such as redaction, audit tracing and automated form-filling require grounding extracted values to precise bounding boxes (grounded KIE), where misalignment may lead to compliance failures or costly manual re-verification. Unlike earlier encoder-based approaches (Xu et al., 2020; Huang et al., 2022) that

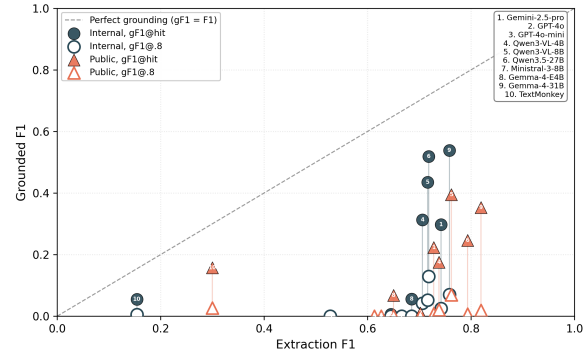


Figure 1: Extraction-grounding gap across 10 VLMs on internal (○, dark) and public (△, orange) benchmarks. The dashed diagonal is the upper bound where grounded F1 equals extraction F1 (i.e., perfect grounding). Detailed per-model results in Appendix A.4.2.

treat KIE as a sequence tagging task with inherent token-level positions as input, VLMs generate coordinates as output, introducing uncertainty absent in prior architectures.

In this paper, we systematically evaluate 10 general-purpose and document-specific VLMs on both internal and public benchmarks, adopting a two-regime evaluation protocol with extraction F1 as the natural upper bound. The strict regime (grounded F1@0.8) is designed for use cases requiring precise localization, while the lenient regime (grounded F1@hit) focuses on scenarios where identifying the correct region is sufficient. Our analysis reveals a consistent extraction-grounding gap across all model families (Figure 1): high-performing models show a degradation of $\sim 20\%$ under the lenient-regime, with the gap widening significantly to $\sim 50\text{--}70\%$ under the strict-regime. This substantial drop renders many value-correct extractions unsuitable for grounding-critical use cases, despite strong overall KIE extraction performance. Token-level entropy analysis further reveals a co-occurring pattern: coordinate digit tokens are generated with significantly higher

uncertainty than content tokens, particularly at fine-grained digit positions where precise grounding fails.

We address this gap with a decoupled grounding pipeline that anchors bounding box grounding to OCR tokens, rather than image features. We find that simply replacing images with OCR tokens in the grounding step brings substantial improvements, especially to lenient-regime grounding quality (Centroid Hit Accuracy increases from ~ 0.7 to ~ 0.9 on both benchmarks). We further design and delegate the grounding step to a fine-tuned 0.6B LLM, which improves strict-regime grounding quality on in-domain internal benchmarks (grounded F1@.8 increases from 0.084 to 0.417) and adds only 2.4s mean latency on top of extraction.

Our main contributions are as follows:

- We conduct a systematic evaluation across 10 VLMs and reveal a consistent extraction-grounding gap: even the strongest extractors suffer severe localization failures under strict IoU criteria.
- We investigate the gap by studying the decoding behavior and find that vision-conditioned coordinate generation introduces high uncertainty.
- We introduce an OCR-anchored grounding pipeline with zero-shot (training-free) and fine-tuned (0.6B LLM) variants that substantially improves both lenient and strict grounding.

2 Related Work

OCR-free Document KIE Methods. Visual document inputs naturally contain layout and reading-order cues that are crucial for document KIE. Recent work focuses on visual encoder designs to better capture dense text regions through feature aggregation (Park et al., 2024) and improve efficiency by pruning irrelevant visual tokens (Liu et al., 2024; Zhang et al., 2025; Liu et al., 2026c). A parallel line of research extends OCR-centric pipelines with broader downstream capabilities including KIE (Tang et al., 2023; Hu et al., 2025; Dong et al., 2026). Despite these advances, many studies formulate extraction as single-field question answering. This setup is less suitable for industrial use cases that require extracting many fields under an often complex schema, where per-field prompting and decoding can be costly and difficult to scale.

Grounding in Document KIE. Prior grounded KIE methods generally follow two paradigms: generation-based approaches that decode bounding box coordinates (Wang et al., 2023a; Feng et al., 2024; Perot et al., 2024; Liu et al., 2026c), and regression-based approaches that attach localization heads to predict bounding boxes (Okamoto et al., 2024; Zhu et al., 2025; Liu et al., 2026b). Existing regression-based designs are often tied to field-specific supervision (Okamoto et al., 2024) and single-field QA task formulation (Zhu et al., 2025), limiting their applicability to schema-based industrial workflows. In addition, benchmarking of grounding quality is still limited and inconsistent: prior work validates against line-level rather than field-level boxes (Perot et al., 2024), uses very lenient IoU thresholds (10^{-3} , 10^{-2} , 10^{-1}) that only reflect coarse region coverage (Liu et al., 2026b), or cross-checks faithfulness without requiring coordinate precision (Ji et al., 2026). As a result, the practical grounding quality of generation-based methods in production workflows remains unclear. Our pipeline is conceptually close to text-based generation methods such as LMDX (Perot et al., 2024), but we keep the upstream extraction module intact and delegate the grounding step to an LLM, which allows us to combine the extraction strength of VLMs with the spatial reliability of OCR tokens.

3 Methodology

3.1 Dataset and Metrics

Our internal industrial dataset consists of 1,490 single-page business documents sourced from procurement, supply chain, and financial transaction processes. The bounding box annotations were manually curated by drawing the minimum enclosing rectangle that tightly wraps around the ground-truth text. For the public dataset, we use VRDU Ad-Buy Form and Registration Form (Wang et al., 2023b). The dataset details can be found in Appendix A.1.

We evaluate grounded KIE under two industry-relevant regimes that reflect distinct downstream tolerances, with the ungrounded extraction F1 serving as the natural upper bound.

Strict Regime targets use cases such as redaction and automated form-filling, where bounding boxes must tightly overlap their target. We report **mIoU** (mean Intersection over Union between the predicted and the ground-truth boxes, sensitive to both position and size error) and **gF1@.8**

(end-to-end grounded F1 in which a prediction is considered correct only if its IoU with the ground-truth box ≥ 0.8).

Lenient Regime targets use cases such as audit highlighting and human-in-the-loop review, where bounding boxes can be more loosely defined as long as they point to the correct region. We report **CHA** (Centroid Hit Accuracy: the percentage of predicted boxes whose centroid falls within the ground-truth boxes, tolerating over-sized boxes as long as they are correctly positioned) and **gF1@hit** (end-to-end grounded F1 where a prediction is correct only when both the value matches and the centroid hits).

To isolate localization quality from errors propagated from incorrect value extractions, we evaluate mIoU and CHA only on value-correct predictions. The full matching procedure is detailed in Appendix A.2.

3.2 Preliminaries: Extraction-Grounding Gap

Following the practice in previous work (Bai et al., 2025; Liu et al., 2026c), grounded KIE was formulated as a schema-guided generation task, where the model is prompted to generate a schema-aligned prediction JSON that contains both the extracted value content and the corresponding bounding box coordinates. Coordinates are represented as a list of $[x_0, y_0, x_1, y_1]$, with (x_0, y_0) denoting the top-left and (x_1, y_1) denoting the bottom-right corners. The bounding box generation instruction was also standardized by adopting the normalized coordinate system scaled to the range $[0, 1000]$ (ablation on other coordinate systems in Appendix A.4.4).

Figure 1 shows results across 10 VLMs spanning proprietary (GPT-4o series (OpenAI et al., 2024), Gemini-2.5-pro (Comanici et al., 2025)), open-source (Qwen-VL series (Bai et al., 2025; Team, 2026), Gemma-4 series (Google DeepMind, 2026), Ministral 3 (Liu et al., 2026a)), and document-specific (TextMonkey (Liu et al., 2026c)) model families. We include TextMonkey as a representative OCR-free model in the single-field QA paradigm. The detailed results are documented in Appendix A.4.2. Despite substantial variation in extraction F1, all models exhibit a consistent extraction-grounding gap. The best-performing models exhibit significant gF1@.8 collapse (Gemma-4-31B drops from 0.819 to 0.019 on the public benchmark and from 0.758 to 0.070 on the internal benchmark).

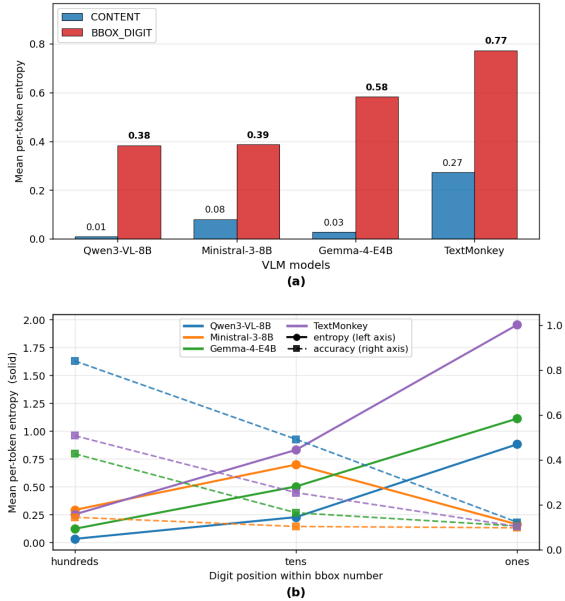


Figure 2: (a) Mean token entropy of BBOX_DIGIT and CONTENT across model families. (b) Digit-level entropy and accuracy for BBOX_DIGIT tokens, showing higher uncertainty and lower accuracy at finer digit positions.

To understand this gap, we look into decoding entropy in vision-conditioned grounded KIE as a diagnostic signal.

Coordinate Tokens Exhibit Higher Prediction Uncertainty.

We measure per-token entropy over the top-20 logits across four model families (Qwen3-VL-8B, Ministral-3-8B, Gemma-4-E4B, and TextMonkey) on the public dataset, categorizing decoded tokens as CONTENT (field values, e.g., “INV-NUM-012”) or BBOX_DIGIT (coordinate digits). Structural tokens (punctuation, field names) are excluded. For TextMonkey, which uses a single-field QA-style prompt interface, we reformulate our task into single-field queries to enable comparison. As shown in Figure 2a, BBOX_DIGIT tokens exhibit substantially higher entropy than CONTENT tokens across all families. Qwen3-VL-8B shows the largest gap, with a $38\times$ difference, suggesting that models are uncertain in coordinate generation. This pattern also holds for single-field QA-style TextMonkey, indicating that the entropy gap is not an artifact of the schema-guided prompt design.

Uncertainty is Concentrated at Fine Digit Positions.

Since content and coordinate tokens represent different prediction categories, we further decompose the entropy within coordinate tokens across individual digit positions. All four models

tokenize numbers digit-by-digit and coordinates are normalized to $[0, 1000]$: the hundreds digit represents coarse position (10% of image dimension), whereas the ones digit represents fine precision (0.1%). As shown in Figure 2b, entropy increases monotonically from hundreds to ones (solid lines), while accuracy follows the inverse pattern (dashed lines). Ministral-3-8B is an exception: we observe that its outputs are degenerate (the ones digit is nearly always “0”), so its entropy values are uninformative and are excluded from the trend characterization. This co-occurs with the precision failures: gF1@hit (loose alignment) shows moderate degradation while gF1@.8 (tight overlap) suffers significant collapse.

3.3 Problem Formulation

Motivated by the observed high uncertainty, we revisit the OCR system as a grounding primitive: OCR tokens provide a deterministic source of spatial information, enabling coordinates to be reliably looked up and composed. To realize this, we adopt a decoupled *two-step* design where a separate module handles grounding with OCR tokens as input, making it deployable as a drop-in on top of any existing extraction pipeline. The grounding task involves aligning predicted field values with their corresponding locations in a document image. Specifically, given (1) **OCR Output**: A set of word-level OCR results $\mathcal{O} = \{(w_i, b_i)\}_{i=1}^N$, where w_i is the text of the i -th OCR word and b_i its bounding box, and (2) **Predicted Field Values**: A set of predicted field values $\{v_j\}_{j=1}^M$ in schema format from an upstream extraction model, the goal is to assign each predicted field value v_j a bounding box $\hat{b}_j$. This bounding box $\hat{b}_j$ is computed as the minimum enclosing rectangle over a subset of OCR words $\mathcal{S}_j \subseteq \{w_i\}_{i=1}^N$, where the concatenated text of the words in $\mathcal{S}_j$ best matches the field value v_j .

3.4 LLM as Grounding Module

The OCR-anchored grounding step faces two practical challenges: (1) **Representation Ambiguity**: Variations may be present in how text is represented, due to differences in formatting (e.g., “1,000” vs. “1000”) or errors propagated from OCR module (e.g., parsing “o” as “0”). (2) **Semantic Ambiguity**: Multiple occurrences of the same value belonging to different fields may exist (e.g., “1,000” as both “Subtotal Amount” and “Total Amount”), requiring contextual understanding to disambiguate.

We delegate the grounding module to an LLM, which leverages its inherent language understanding ability to resolve the ambiguities and compose coordinates from input OCR spans. We demonstrate our approach in two ways: (1) **Zero-shot LLM**: We prompt the LLM decoder of the same VLM used for extraction to re-generate bounding boxes in a second pass, replacing the image input with OCR tokens and requiring no training data. (2) **Fine-tuned LLM**: We further fine-tune a lightweight LLM on our internal dataset for the grounding step with task-specific supervision.

For each training instance, the input x contains: (1) the schema populated with ground-truth extraction values, where each target bounding box is replaced by a placeholder string “[x0, y0, x1, y1]”, and (2) the OCR token sequence $\mathcal{O}$ as a list of (w_i, b_i) tuples, with b_i normalized to $[0, 1000]$. The target y is the same schema with placeholders replaced by normalized coordinates in the same coordinate system. An example input-output pair is shown in Figure 7 in Appendix A.5. This formulation encourages the model to resolve ambiguous matches using both lexical context and OCR geometry while keeping inference efficient.

We optimize standard autoregressive next-token cross-entropy on the serialized target sequence:

$$\mathcal{L} = - \sum_{t=1}^T \log p(y_t \mid y_{<t}, x) \quad (1)$$

4 Experiments and Results

4.1 Implementation Details

For the Fine-tuned LLM, we initialized its weight from Qwen-3-0.6B (Yang et al., 2025a) and utilized internal annotated data with 50,000 samples for training (ablation on data scaling is documented in Appendix A.4.3). The training samples share similar schemas (in-domain) with the 1,490 evaluation documents. Generalization to unseen schemas (cross-domain) was evaluated using public benchmarks. We utilized an internal OCR system (documented in Appendix A.3) to extract OCR tokens for both training and inference. Training was conducted on 8 NVIDIA A100-80GB GPUs. The full training recipe is documented in Appendix A.4.1.

4.2 Baselines

As shown in Table 2, we compare three baselines along three dimensions: grounding task formulation (joint vs. decoupled), grounding step input

Base Model	Method	Input	Internal					Public				
			mIoU	CHA	gF1@hit	gF1@.8	F1	mIoU	CHA	gF1@hit	gF1@.8	F1
Qwen3-VL-8B	One-step VLM	<i>I</i>	.431	.635	.435	.052		.438	.676	.395	.069	
	Two-step VLM	<i>I</i>	.429	.637	.457	.057		.439	.689	.408	.064	
	Two-step Heuristics	<i>O</i>	.568	.788	.563	.153	.716	.464	.736	.568	<u>.095</u>	.762
	Two-step Zero-shot LLM	<i>O</i>	<u>.650</u>	<u>.930</u>	<u>.635</u>	.148		<u>.530</u>	<u>.891</u>	<u>.512</u>	<u>.064</u>	
	Two-step Fine-tuned LLM	<i>O</i>	.788	.966	.676	.409		.602	.906	.512	.113	
Gemma-4-31B	One-step VLM	<i>I</i>	.472	.715	.538	.070		.433	.792	.353	.019	
	Two-step VLM	<i>I</i>	.522	.790	.592	.082		.443	.808	.363	.015	
	Two-step Heuristics	<i>O</i>	.569	.790	.586	.158	.758	.381	.706	.297	.024	.819
	Two-step Zero-shot LLM	<i>O</i>	<u>.677</u>	<u>.952</u>	<u>.650</u>	<u>.166</u>		<u>.544</u>	.958	.411	<u>.043</u>	
	Two-step Fine-tuned LLM	<i>O</i>	.790	.964	.707	.437		.560	<u>.912</u>	<u>.392</u>	.068	
Qwen3.5-27B	One-step VLM	<i>I</i>	.541	.748	.519	.129		.285	.601	.246	.007	
	Two-step VLM	<i>I</i>	.541	.743	.517	.129		.300	.585	.240	.010	
	Two-step Heuristics	<i>O</i>	.577	.799	.543	.142	.718	.379	.708	.281	.021	.793
	Two-step Zero-shot LLM	<i>O</i>	<u>.675</u>	<u>.950</u>	<u>.638</u>	<u>.156</u>		<u>.531</u>	.945	.402	<u>.039</u>	
	Two-step Fine-tuned LLM	<i>O</i>	.794	.965	.646	.407		.553	<u>.909</u>	<u>.371</u>	.064	
Average	One-step VLM	<i>I</i>	.481	.699	.497	.084		.385	.690	.331	.032	
	Two-step VLM	<i>I</i>	.498	.723	.522	.089		.394	.694	.337	.030	
	Two-step Heuristics	<i>O</i>	.571	.793	.564	.151	.731	.408	.717	.382	.047	.792
	Two-step Zero-shot LLM	<i>O</i>	<u>.668</u>	<u>.944</u>	<u>.641</u>	<u>.157</u>		<u>.535</u>	.931	.442	<u>.049</u>	
	Two-step Fine-tuned LLM	<i>O</i>	.791	.965	.676	.417		.572	<u>.909</u>	<u>.425</u>	.082	

Table 1: Comparison of grounding methods across three base models on the internal and public benchmarks. Input: *I* = Image (OCR-free), *O* = OCR-anchored. The rightmost **F1** column in each benchmark shows the ungrounded extraction F1, which serves as the natural upper bound for both gF1@hit and gF1@.8. **Bold** = best; underlined = second-best.

Baseline	Input	Grounding mechanism
One-step VLM	Image	Joint generation
Two-step VLM	Image	Second-pass generation with the same VLM
Two-step Heuristic	OCR	Fuzzy + beam search

Table 2: Baseline grounding methods. Baseline 1 and 2 test task decoupling within the VLM. Baseline 3 serves as a rule-based matching baseline to compare with model-parametric matching.

modality (image vs. OCR tokens) and grounding mechanism (model-parametric vs. heuristic). The extraction results from one-step VLM baseline are re-used for two-step baselines to ensure a controlled comparison. The heuristic baseline is an internal proprietary implementation (description is documented in Appendix A.3) that matches extracted values to OCR tokens using a beam search with fuzzy string similarity and span matching.

4.3 Main Results

Table 1 compares OCR-free and OCR-anchored approaches on three best-performing extractors, selected by average F1 across internal and public benchmarks (Qwen3-VL-8B, Gemma-4-31B, Qwen3.5-27B, full ranking in Appendix Table 11).

Joint-task Interference Has Minimal Impact on Grounding Quality. To isolate the effect of joint-task interference, we conduct a second-step

inference where the model is prompted to generate bounding box coordinates given the first step’s value extraction results and the image (*Two-step VLM* vs. *One-step VLM*). This decoupling yields only modest improvements on internal benchmarks (0.025 and 0.005 improvement in gF1@hit and gF1@.8 respectively) and has near-to-zero impact on public benchmarks (0.006 improvement on gF1@hit and 0.002 drop on gF1@.8). The results rule out joint-task interference as the dominant cause of grounding collapse in VLMs.

OCR Modality Narrows the Extraction-Grounding Gap. Switching the grounding signal from image to OCR tokens (*Two-step VLM* vs. *Two-step Zero-shot LLM*) without any task-specific training yields substantial improvements in both strict and lenient regimes on both internal and public benchmarks, showing improvements brought by the OCR tokens itself. Average gF1@hit improves from 0.522 to 0.641 and from 0.337 to 0.442 on the internal and public benchmarks respectively. Average CHA reaches above 0.930 on both benchmarks. Notably, *Two-step Zero-shot LLM* also outperforms the heuristic baseline, which often mismatches similar text (Figure 3b vs. c). Strict-regime improvements from modality alone are smaller (average gF1@.8 increases from 0.089 to 0.157 on internal and from 0.030 to 0.049 on public), indicating that zero-shot prompting might not be sufficient for tight box

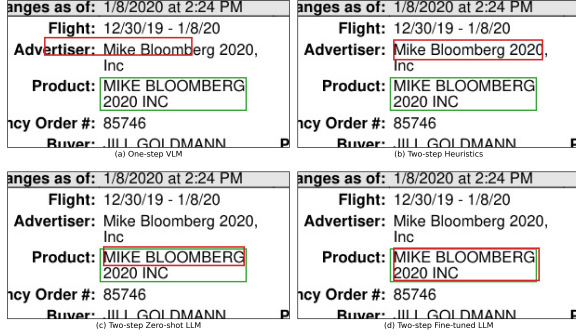


Figure 3: Qualitative comparison on VRDU Ad-Buy data. (a) *One-step VLM* locates the coarse region but drifts away from the target field. (b) *Two-step Heuristics* mismatches the bounding box to a similar text. (c) *Two-step Zero-shot LLM* correctly grounds to the right region, but the box doesn’t cover the full text. (d) *Two-step Fine-tuned LLM* generates a tight box.

overlap (Figure 3c).

Task-specific Fine-tuning Narrows the Strict-Regime Gap. Our *Two-step Fine-tuned LLM* consistently improves both lenient and strict-regime grounding on in-domain internal benchmarks across all three base extractors, with substantial gains on strict-regime ($\text{gF1@.8}=0.417$), producing tight field-level boxes (Figure 3d). On the cross-domain public benchmark, gains are smaller and concentrated in the strict regime (mIoU and gF1@.8), while the model occasionally trails *Two-step Zero-shot LLM* on the lenient metrics.

Our findings suggest a natural deployment flow for industrial KIE pipelines with the two approaches: *Zero-shot LLM* serves as an immediately deployable solution, requiring no labeled data, while *Fine-tuned LLM* becomes viable once sufficient in-domain annotations are available, further tightening strict-regime overlap.

Grounding Step	Δ Latency (s)	
	Mean	p95
Two-step VLM	+26.0	+48.8
Two-step Heuristic	<1	<1
Two-step Zero-shot LLM	+24.6	+50.7
Two-step Fine-tuned LLM	+2.4	+7.6

Table 3: Added latency of the grounding step.

4.4 Latency Analysis

We use Qwen3-VL-8B as the base extraction model and report mean and p95 added latency over 200 internal documents on a single NVIDIA L40S GPU

with batch size 1, served via vLLM (Table 3). The OCR system runs concurrently with the upstream extraction model and finishes well before extraction completes (OCR p95 0.67s vs. extraction p95 45.0s), thus adding negligible latency. Our fine-tuned 0.6B LLM adds a mean latency of 2.4s while substantially improving grounding quality.

Method	Params	gF1@.8	Latency
Two-step Fine-tuned VLM	0.8B	0.315	3.6 s
Two-step Fine-tuned LLM (ours)	0.6B	0.409	2.4 s

Table 4: Quality and latency comparison on task-specific fine-tuning of OCR-free VLM vs. OCR-anchored LLM on internal benchmark.

4.5 The Role of Task-specific Fine-tuning

An alternative to fine-tuning an LLM is to fine-tune a VLM that takes image input directly (two-step OCR-free grounding). We conducted an ablation study by fine-tuning Qwen3.5-0.8B (Team, 2026) (slightly more parameters compared to our proposed 0.6B LLM) for grounding with the document image. Table 4 reports the comparison. While fine-tuning substantially improves VLMs’ grounding performance compared to zero-shot LLM and heuristic baselines, confirming the importance of fine-tuning, it still trails behind the fine-tuned LLM by nearly 10 absolute points (0.315 vs. 0.409 on gF1@.8). The image-based inference incurs additional latency from the vision encoder and image-token prefill, resulting in $1.5\times$ latency compared to OCR-anchored LLM (3.6s vs. 2.4s). These findings suggest that, within the latency budget of industrial deployment, an OCR-anchored LLM design is more practical than the image-based VLM design on both grounding quality and serving cost.

4.6 Robustness to OCR Quality

Since our decoupled design uses OCR as the spatial anchor, we analyze its quality as a variable along two axes: engine choice, and noise level.

Engine axis We replace our internal OCR engine with Google Vision¹ and Tesseract² in the two-step methods. As shown in Table 5, the two-step OCR-anchored methods outperform the OCR-free *One-step VLM* baseline across all three engines. Notably, our *Two-step Fine-tuned LLM*, transfers to both external engines without retraining, despite being trained exclusively on the internal engine.

¹<https://cloud.google.com/vision>

²<https://github.com/tesseract-ocr/tesseract>

OCR engine	Method	mIoU	CHA	gF1@hit	gF1@.8
–	One-step VLM	.431	.635	.435	.052
Internal	Two-step Zero-shot LLM	.650	.930	.635	.148
	Two-step Fine-tuned LLM	.788	.966	.676	.409
Google Vision	Two-step Zero-shot LLM	.611	.904	.619	.115
	Two-step Fine-tuned LLM	.697	.906	.644	.262
Tesseract	Two-step Zero-shot LLM	.559	.776	.530	.154
	Two-step Fine-tuned LLM	.616	.798	.569	.234

Table 5: Grounding performance under different OCR engines on the internal dataset (Qwen3-VL-8B as base extractor).

Setting	mIoU	CHA	gF1@hit	gF1@.8
One-step VLM	.431	.635	.435	.052
Two-step Fine-tuned LLM (clean)	.788	.966	.676	.409
Two-step Fine-tuned LLM (5% noise)	.754	.924	.657	.396
Two-step Fine-tuned LLM (15% noise)	.732	.895	.636	.387
Two-step Fine-tuned LLM (30% noise)	.711	.873	.621	.369
Two-step Fine-tuned LLM (50% noise)	.675	.828	.588	.353

Table 6: Grounding performance of the Two-step Fine-tuned LLM under OCR noise injection on the internal dataset (Qwen3-VL-8B as base extractor).

Noise axis We inject noise into OCR tokens at different noise levels (the percentage of tokens subjected to noise injection). Each affected token undergoes a single-character edit such as substitution, deletion or insertion. While grounding scores decrease as noise level increase, *Two-step Fine-tuned LLM* stays well above the OCR-free baseline even at 50% noise (Table 6), and the LLM grounding modules continue to outperform the heuristic matcher under noise conditions (Table 7). Together, these results confirm the robustness of the decoupled design to OCR degradation.

4.7 Failure-Mode Decomposition

To characterize the types of grounding errors and how our method addresses them, we decompose the value-correct predictions of Qwen3-VL-8B on the internal dataset into *Strict Pass* (IoU ≥ 0.8), *Near Miss* (centroid inside GT, IoU < 0.8), and *Truly Incorrect* (centroid outside GT). As shown in Table 8, the *One-step VLM* baseline exhibits a substantial rate of *Truly Incorrect* failures (36.5%), with a further 55.2% being near-misses that locate the correct region but fail to tightly bound it. Our *Two-step Fine-tuned LLM* raises *Strict Pass* to 58.9% and reduces *Truly Incorrect* to 3.4%, demonstrating that the decoupled design largely eliminates gross localization failures and converts borderline near-misses into tight matches.

Method (5% noise)	mIoU	CHA	gF1@hit	gF1@.8
Two-step Heuristics	.518	.771	.551	.105
Two-step Zero-shot LLM	.636	.911	.624	.144
Two-step Fine-tuned LLM	.754	.924	.657	.396

Table 7: Two-step methods comparison under 5% OCR noise on the internal dataset (Qwen3-VL-8B as base extractor).

Failure mode	One-step VLM	Two-step Fine-tuned LLM
<i>Strict Pass</i>	8.3%	58.9%
<i>Near Miss</i>	55.2%	37.7%
<i>Truly Incorrect</i>	36.5%	3.4%

Table 8: Failure-mode decomposition of value-correct predictions on the internal dataset with Qwen3-VL-8B as the base extractor. *Strict Pass* and *Near Miss* together equal the Centroid Hit Accuracy.

5 Conclusion

VLMs and OCR systems are often framed as competing paradigms in document KIE, our study suggests that they are complementary. Across 10 VLMs, we identified a consistent extraction-grounding gap. We show that anchoring grounding to OCR tokens narrows this gap. A zero-shot LLM with OCR token input substantially improves grounding quality, while a fine-tuned 0.6B LLM further tightens strict-regime overlap on in-domain data at +2.4s mean added latency. Together, these results position OCR not as a legacy component but as a reliable spatial anchor that complements vision-based extraction in industrial KIE pipelines.

Limitations

Document scope. Our evaluation focuses on single-page and machine-printed documents. Multi-page, handwritten and chart-heavy documents are out of scope and likely require richer OCR-token representation to handle. **OCR dependence.** Our decoupled grounding step is dependent on the upstream OCR system. (1) It operates under the assumption that a reliable OCR system is available. (2) While it is able to handle OCR errors and noise, complete OCR failures on certain documents, caused by factors such as non-standard fonts, might negatively impact grounding quality. **LLM scaling.** Our fine-tuned LLM is initialized with 0.6B parameters, future studies on the performance-latency trade-off for LLMs of different sizes would bring valuable information to practitioners.

References

- Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhi-fang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025. [Qwen3-VL Technical Report](#). *CoRR*, abs/2511.21631.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Naveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. [Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities](#). *CoRR*, abs/2507.06261.
- Daxiang Dong, Mingming Zheng, Dong Xu, Chunhua Luo, Bairong Zhuang, Yuxuan Li, Ruoyun He, Hao-ran Wang, Wenyu Zhang, Wenbo Wang, Yicheng Wang, Xue Xiong, Ayong Zheng, Xiaoying Zuo, Ziwei Ou, Jingnan Gu, Quanhao Guo, Jianmin Wu, Dawei Yin, and Dou Shen. 2026. [Qianfan-OCR: A Unified End-to-End Model for Document Intelligence](#). *CoRR*, abs/2603.13398.
- Hao Feng, Qi Liu, Hao Liu, Jingqun Tang, Wengang Zhou, Houqiang Li, and Can Huang. 2024. [DocPe-dia: unleashing the power of large multimodal model in the frequency domain for versatile document understanding](#). *Sci. China Inf. Sci.*, 67(12).
- Ling Fu, Biao Yang, Zhebin Kuang, Jiajun Song, Yuzhe Li, Linghao Zhu, Qidi Luo, Xinyu Wang, Hao Lu, Mingxin Huang, Zhang Li, Guozhi Tang, Bin Shan, Chunhui Lin, Qi Liu, Binghong Wu, Hao Feng, Hao Liu, Can Huang, and 5 others. 2025. [OCRBench v2: An Improved Benchmark for Evaluating Large Multimodal Models on Visual Text Localization and Reasoning](#). *CoRR*, abs/2501.00321.
- Google DeepMind. 2026. [Gemma 4](#). Accessed: 2026-06-05.
- 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)*, ACL 2025, Vienna, Austria, July 27 - August 1, 2025, pages 5817–5834. Association for Computational Linguistics.
- 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 *MM '22: The 30th ACM International Conference on Multimedia, Lisboa, Portugal, October 10 - 14, 2022*, pages 4083–4091. ACM.
- Yifan Ji, Zhipeng Xu, Zhenghao Liu, Zulong Chen, Qian Zhang, Zhibo Yang, Junyang Lin, Yu Gu, Ge Yu, and Maosong Sun. 2026. [UNIKIE-BENCH: Benchmarking Large Multimodal Models for Key Information Extraction in Visual Documents](#). *CoRR*, abs/2602.07038.
- Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, and 101 others. 2026a. [Ministral 3](#). *CoRR*, abs/2601.08584.
- Chaohu Liu, Kun Yin, Haoyu Cao, Xinghua Jiang, Xin Li, Yinsong Liu, Deqiang Jiang, Xing Sun, and Linli Xu. 2024. [HRVDA: High-Resolution Visual Document Assistant](#). In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 15534–15545. IEEE.
- Shuhang Liu, Zhenrong Zhang, Pengfei Hu, Jiefeng Ma, Jun Du, Qing Wang, Jianshu Zhang, and Chenyu Liu. 2026b. [See then tell: Enhancing key information extraction with vision grounding](#). *Neurocomputing*, 673:132858.
- Yuliang Liu, Biao Yang, Qiang Liu, Zhang Li, Zhiyin Ma, Shuo Zhang, and Xiang Bai. 2026c. [TextMon-key: An OCR-Free Large Multimodal Model for Understanding Document](#). *IEEE Trans. Pattern Anal. Mach. Intell.*, 48(5):6008–6019.
- Yamato Okamoto, Youngmin Baek, Geewook Kim, Ryota Nakao, DongHyun Kim, Moonbin Yim, Seunghyun Park, and Bado Lee. 2024. [CREPE: Coordinate-Aware End-to-End Document Parser](#). In *Document Analysis and Recognition - ICDAR 2024 - 18th International Conference, Athens, Greece, August 30 - September 4, 2024, Proceedings, Part IV*, volume 14807 of *Lecture Notes in Computer Science*, pages 3–20. Springer.
- OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, and 400 others. 2024. [GPT-4o System Card](#). *CoRR*, abs/2410.21276.
- Jaeyoo Park, Jin Young Choi, Jeonghyung Park, and Bohyung Han. 2024. [Hierarchical Visual Feature Aggregation for OCR-Free Document Understanding](#).
- Vincent Perot, Kai Kang, Florian Luisier, Guolong Su, Xiaoyu Sun, Ramya Sree Boppana, Zilong Wang, Zifeng Wang, Jiaqi Mu, Hao Zhang, Chen-Yu Lee, and Nan Hua. 2024. [LMDX: Language Model-based Document Information Extraction and Localization](#). In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual*

meeting, August 11-16, 2024, volume ACL 2024 of *Findings of ACL*, pages 15140–15168. Association for Computational Linguistics.

Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. 2023. [Unifying Vision, Text, and Layout for Universal Document Processing](#). In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 19254–19264. IEEE.

Qwen Team. 2026. [Qwen3.5: Accelerating Productivity with Native Multimodal Agents](#).

Yonghui Wang, Wengang Zhou, Hao Feng, Keyi Zhou, and Houqiang Li. 2023a. [Towards Improving Document Understanding: An Exploration on Text-Grounding via MLLMs](#). *CoRR*, abs/2311.13194.

Zilong Wang, Yichao Zhou, Wei Wei, Chen-Yu Lee, and Sandeep Tata. 2023b. [VRDU: A Benchmark for Visually-rich Document Understanding](#). In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pages 5184–5193. ACM.

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 *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 1192–1200. ACM.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. [Qwen3 Technical Report](#). *CoRR*, abs/2505.09388.

Zhibo Yang, Jun Tang, Zhaohai Li, Pengfei Wang, Jianqiang Wan, Humen Zhong, Xuejing Liu, Mingkun Yang, Peng Wang, Shuai Bai, Lianwen Jin, and Junyang Lin. 2025b. [CC-OCR: A Comprehensive and Challenging OCR Benchmark for Evaluating Large Multimodal Models in Literacy](#). In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pages 21744–21754. IEEE.

Jiaxin Zhang, Wentao Yang, Songxuan Lai, Zecheng Xie, and Lianwen Jin. 2025. [DocKylin: A Large Multimodal Model for Visual Document Understanding with Efficient Visual Slimming](#). In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 9923–9932. AAAI Press.

Zhaoqing Zhu, Chuwei Luo, Zirui Shao, Feiyu Gao, Hangdi Xing, Qi Zheng, and Ji Zhang. 2025. [A Simple yet Effective Layout Token in Large Language Models for Document Understanding](#). In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 14472–14482. Computer Vision Foundation / IEEE.

A Appendix

A.1 Dataset Details

Image size (MP)	Internal	Public
0–3	1	285
3–5	527	0
5–10	637	0
10–20	24	33
20–30	262	0
30–50	2	0
≥50	37	0
Total	1,490	318

Table 9: Image-size (megapixels) distribution of internal and public evaluation data.

Table 9 shows the image resolution distribution of our internal and public evaluation datasets. We pass raw images directly to each VLM and rely on its native image preprocessor for resizing or downsampling. Table 10 shows the schema density statistics of our evaluation benchmarks. All schemas are formulated as header fields and line-item fields (a table structure with column headers as field name, each row is represented as a JSON dictionary in a line-item list). VRDU Reg. Form has no table structure.

A.2 Evaluation Details

Extraction is scored with field-level micro-averaged F1 over flattened (*key, value*) tuples, with line-item rows indexed (e.g., *lineitem.0.qty*) such that wrong-row extractions are penalized. Multi-occurrence ground-truth fields (e.g., a value printed twice in one document) are collapsed to a single slot whose ground-truth is a set of alternatives, and a prediction matching any of the alternatives is a true positive. Text values are normalized (whitespace cleanup and case normalization) before matching.

A.3 Internal System Implementation Details

A.3.1 OCR System

The OCR system consists of three main stages: preprocessing, text detection, and text recognition.

Dataset	Header	LineItem	Row
Internal	5.81	3.48	4.54
Public VRDU Ad-Buy	7.37	3.20	5.07
Public VRDU Reg. Form	5.11	0	0

Table 10: Per-document averages on schema density of evaluation benchmarks. **Header**: number of header fields; **LineItem**: number of line-item fields (table column headers); **Row**: number of table rows.

In the preprocessing stage, input documents are converted to grayscale images and deskewing is applied to correct orientation issues. For text detection, a pre-trained convolutional neural network (CNN) is used to identify text regions, producing bounding boxes around detected areas. These regions are then cropped and passed to a text recognition model to extract the textual content. The final output combines each recognized word text with its corresponding bounding box coordinates, structured as JSON.

A.3.2 Heuristic Grounding Method

The heuristic-based grounding method matches extracted values to OCR tokens using a best-first beam search with fuzzy string similarity for robustness to OCR errors with span matching for precise alignment. The method maintains a search state comprising the already-aligned (OCR word, annotation word) pairs, the next annotation word to be aligned, and the set of candidate OCR words. At each step, the current annotation word is aligned to a candidate OCR word using a cost function with three components: (1) Levenshtein distance between the annotation word and the OCR word, (2) minimum Euclidean distance between the candidate OCR box and the previously aligned OCR boxes, and (3) a weighted sum of these two normalized distances. The best-first beam search iteratively expands the lowest-cost partial alignments, and the bounding box for the extracted value is computed as the minimum enclosing rectangle of the aligned OCR tokens. The cost function weights and the beam width were tuned specifically for OCR-token grounding.

A.4 Experiment Settings and Results

A.4.1 Training Recipe

For Qwen-3-0.6B LLM and Qwen3.5-0.8B VLM, we utilized full-parameter tuning with learning rate of 1×10^{-5} and cosine decay scheduler. The maximum sequence length was set to 8192. The models

were trained for 3 epochs with BF16 precision. Per-device batch size and gradient accumulation step were set to 2 and 16 respectively.

A.4.2 Extraction-Grounding Gap Results

Table 11 provides the per-model breakdown of results referenced in Section 3.2. We report mIoU, CHA, gF1@hit, gF1@.8, and ungrounded extraction F1 separately for internal and public benchmarks, plus the average F1 across the two. The extraction-grounding gap is consistent across all 10 models: even the strongest extractor (Gemma-4-31B, average F1 0.788) achieves gF1@.8 of only 0.070 on internal and 0.019 on public benchmarks. We use the top-3 models by average F1 (Qwen3.5-27B, Gemma-4-31B, and Qwen3-VL-8B) as the base extraction models in our main results in Table 1.

A.4.3 Ablation on Training Data Size

We conducted a data scaling experiment to understand how the amount of fine-tuning data impacts the performance of the Two-step Fine-tuned LLM approach. Table 12 shows that grounding performance on internal benchmarks gradually improves with more training data (from 10,000 to 50,000). Even with 10,000 training data, our fine-tuned approach still outperforms the *Two-step Zero-shot LLM* by a large margin (0.387 vs. 0.157 in gF1@.8). On the cross-domain public benchmark, the performance is less sensitive to data scale and shows no clear trend.

A.4.4 Ablation on Coordinate System

The coordinate system of $[0, 1000]$ is documented in Qwen-VL series (Bai et al., 2025) and TextMonkey (Liu et al., 2026c). For other VLM families, the exact system used in pre-training is not explicitly disclosed. While we used $[0, 1000]$ system in VLM evaluation, we compared two other conventions (normalized float in $[0, 1]$ and raw pixel coordinate with width and height explicitly provided in prompt) on one representative model from each model family to rule out the implication on grounding quality. As shown in Table 13, Gemini-2.5-Pro and Gemma-4-E4B exhibit the best performance in $[0, 1000]$ system, GPT-4o and Ministral-3-8B achieve only marginally higher score in $[0, 1]$ system with clear overall grounding failure, suggesting that $[0, 1000]$ is a reasonable system for prompting VLMs in our evaluation setting.

Model	Internal					Public					Avg F1
	mIoU	CHA	gF1@hit	gF1@.8	F1	mIoU	CHA	gF1@hit	gF1@.8	F1	
Gemini-2.5-pro	.239	.395	.297	.025	.742	.259	.490	.223	.008	.728	.734
GPT-4o	.008	.011	.006	.000	.645	.029	.031	.010	.000	.702	.673
GPT-4o-mini	.001	.001	.000	.000	.528	.000	.000	.000	.000	.613	.571
Qwen3-VL-4B	.509	.734	.313	.042	.706	.405	.661	.175	.021	.738	.722
Qwen3-VL-8B	.431	.635	.435	.052	.716	.438	.676	.395	.069	.762	<u>.739</u>
Qwen3.5-27B	.541	.748	.519	.129	.718	.285	.601	.246	.007	.793	<u>.756</u>
Ministral-3-8B	.002	.002	.001	.000	.666	.001	.001	.000	.000	.626	.646
Gemma-4-E4B	.066	.107	.056	.000	.685	.094	.222	.068	.001	.650	.667
Gemma-4-31B	.472	.715	.538	.070	.758	.433	.792	.353	.019	.819	.788
TextMonkey	.234	.367	.055	.005	.154	.299	.484	.157	.027	.300	.227

Table 11: Model performance comparison across internal and public benchmarks. **Avg F1** is the mean of internal and public extraction F1; the **top-3** models by Avg F1 (**bold** / underlined) are used as base models in Table 1.

Train size	Internal				Public			
	mIoU	CHA	gF1@hit	gF1@.8	mIoU	CHA	gF1@hit	gF1@.8
10,000	.772	.954	.663	.387	.555	.901	.412	.071
20,000	.778	.958	.671	.397	.554	.892	.413	.074
50,000	.791	.965	.676	.417	.572	.909	.425	.082

Table 12: Effect of training data size on the 0.6B LLM’s grounding quality, averaged across three baseline models (Qwen3-VL-8B, Qwen3.5-27B, Gemma-4-31B). Internal metrics remain within 1-2% points, with no consistent trend.

A.5 Prompt and Training Templates

Following Yang et al. (2025b), we frame our prompts into schema-guided format. Figure 4, 5, and 6 show the prompt templates used in the *One-step VLM*, *Two-step VLM* and *Two-step Zero-shot LLM* experiments. Figure 7 shows the input-output pair for Supervised Fine-Tuning.

Model	[0,1] Format				[0,1000] Format				Pixel Format			
	mIoU	CHA	gF1@hit	gF1@.8	mIoU	CHA	gF1@hit	gF1@.8	mIoU	CHA	gF1@hit	gF1@.8
Gemini-2.5-Pro	.079	.156	.068	.000	.259	.490	.223	.008	.000	.001	.000	.000
GPT-4o	.029	.051	.016	.000	.029	.031	.010	.000	.015	.029	.009	.000
Gemma-4-E4B	.000	.000	.000	.000	.094	.222	.068	.001	.000	.001	.000	.000
Ministral-3-8B	.018	.027	.009	.000	.001	.001	.000	.000	.004	.011	.004	.000

Table 13: Ablation on coordinate normalization scale in instruction prompts across model families, evaluated on public VRDU Ad-Buy and Registration Form dataset.

<image>
 Suppose you are an information extraction expert. Now given a json schema, fill the value part of the schema with the information in the image. For each field, provide both the extracted text content and its bounding box location. The bounding box should be in [x0, y0, x1, y1] format, where (x0, y0) is the top-left corner and (x1, y1) is the bottom-right corner. Coordinates are normalized to a 0-1000 scale. Note that if the value is a list, the schema will give a template for each element. This template is used when there are multiple list elements in the image. Finally, only legal json is required as the output. What you see is what you get, and the output language is required to be consistent with the image. No explanation is required. Please output the results as required. The input json schema content is as follows:
 <schema>

Figure 4: Prompt template for One-step VLM extraction and grounding.

<image>
 Suppose you are an information extraction expert. The following JSON contains the values that have already been extracted from the document image. Each scalar leaf is shown in the form {"content": <extracted value>, "bbox": [x0, y0, x1, y1]}, where the bbox is currently a placeholder. Your task is to fill in each bbox with the actual location of that value in the image. Strict requirements:
 1. Preserve every 'content' value exactly as given. Do not change, normalize, translate, or truncate any value. Do not add or remove any field
 2. Replace each [x0, y0, x1, y1] placeholder with the bounding box of the corresponding value in the image. (x0, y0) is the top-left corner and (x1, y1) is the bottom-right corner. Coordinates are normalized to a 0-1000 scale.
 3. Preserve the exact structure of the input JSON, including nested objects and the order of 'lineItems' rows.
 4. Output only legal JSON. No explanation, no markdown fences, no extra fields.
 The input JSON to fill in is as follows:
 <schema with extracted values>

Figure 5: Prompt template for Two-step VLM grounding.

Suppose you are an information extraction expert. The following JSON contains the values that have already been extracted from a document. Each scalar leaf is shown as {"content": <extracted value>, "bbox": [x0, y0, x1, y1]} where the bbox is a placeholder. Your task is to fill in each bbox with the actual location of that value, using the OCR token list provided below. You will NOT see the image. You will see the representation of the OCR output:
 (a) a list of (bbox, text) tuples for every OCR-detected word
 Use these to locate each extracted value's bounding box. Coordinates are normalized to a 0-1000 scale.
 Strict requirements:
 1. Preserve every 'content' value exactly as given. Do not change, normalize, translate, or truncate. Do not add or remove any field.
 2. For each value, find the OCR tokens that span it (it may span multiple words on the same line) and emit a single bbox covering them. If a value is not present in the OCR, return [0, 0, 0, 0].
 3. Preserve the exact structure of the input JSON, including nested objects and the order of 'lineItems' rows.
 4. Output only legal JSON. No explanation, no markdown fences, no extra fields.
 === OCR token list ([x0, y0, x1, y1] text) ===
 <OCR token list>
 === JSON to fill in ===
 <schema with extracted values>

Figure 6: Prompt template for Two-step Zero-shot LLM grounding.

Input:

Suppose you are an information extraction expert. The following JSON contains the values that have already been extracted from a document. Each scalar leaf is shown as {"content": <extracted value>, "bbox": [x0, y0, x1, y1]} where the bbox is a placeholder. Your task is to fill in each bbox with the actual location of that value, using the OCR token list provided below.

You will NOT see the image. You will see the representation of the OCR output:

(a) a list of (bbox, text) tuples for every OCR-detected word

Use these to locate each extracted value's bounding box. Coordinates are normalized to a 0-1000 scale.

Strict requirements:

1. Preserve every 'content' value exactly as given. Do not change, normalize, translate, or truncate. Do not add or remove any field.
2. For each value, find the OCR tokens that span it (it may span multiple words on the same line) and emit a single bbox covering them. If a value is not present in the OCR, return [0, 0, 0, 0].
3. Preserve the exact structure of the input JSON, including nested objects and the order of 'lineItems' rows.
4. Output only legal JSON. No explanation, no markdown fences, no extra fields.

=== OCR token list ([x0, y0, x1, y1] text) ===

<OCR token list>

=== JSON to fill in ===

```
{
  "flight_to": {"content": "SUNDAY, SEPTEMBER 23, 2012", "bbox": [x0, y0, x1, y1]},
  "flight_from": {"content": "MONDAY, SEPTEMBER 17, 2012", "bbox": [x0, y0, x1, y1]},
  "property": {"content": "KUCW-TV", "bbox": [x0, y0, x1, y1]},
  "gross_amount": {"content": "$445.00", "bbox": [x0, y0, x1, y1]},
  "line_items": [
    {
      "program_desc": {"content": "TWO & HALF MEN", "bbox": [x0, y0, x1, y1]},
      "sub_amount": {"content": "$300.00", "bbox": [x0, y0, x1, y1]}
    },
    {
      "program_desc": {"content": "BIG BANG THEORY", "bbox": [x0, y0, x1, y1]},
      "sub_amount": {"content": "$145.00", "bbox": [x0, y0, x1, y1]}
    }
  ]
}
```

Output:

```
{
  "flight_to": {"content": "SUNDAY, SEPTEMBER 23, 2012", "bbox": [86, 916, 448, 928]},
  "flight_from": {"content": "MONDAY, SEPTEMBER 17, 2012", "bbox": [352, 143, 449, 158]},
  "property": {"content": "KUCW-TV", "bbox": [618, 260, 644, 271]},
  "gross_amount": {"content": "$445.00", "bbox": [289, 413, 390, 423]},
  "line_items": [
    {
      "program_desc": {"content": "TWO & HALF MEN", "bbox": [42, 322, 143, 334]},
      "sub_amount": {"content": "$300.00", "bbox": [293, 322, 336, 334]}
    },
    {
      "program_desc": {"content": "BIG BANG THEORY", "bbox": [47, 367, 268, 381]},
      "sub_amount": {"content": "$145.00", "bbox": [238, 367, 253, 379]}
    }
  ]
}
```

Figure 7: Example input-output pair used for Supervised Fine-Tuning. Schema and values are from the VRDU Ad-Buy dataset.