LeakageBench: Document-Level Leakage Risk for Redacting Personally Identifiable Information in Document Images

Vishnu Prasad Vijaya Kumar[1,2] **Santhosh Venkatesh**[2] **Ivan P. Yamshchikov**[1]
[1]Center for Artificial Intelligence and Robotics (CAIRO),
Technical University of Applied Sciences Würzburg-Schweinfurt (THWS),
Würzburg, Germany
[2]DataX, Frankfurt am Main, Germany
vishnuprasad.vijayakumar@thws.de santhosh.venkatesh@datax.me
ivan.yamshchikov@thws.de

Abstract

Real-world personally identifiable information (PII) redaction often operates on document images—scans, screenshots, and PDF renderings—where OCR errors, layout structure, and visual noise determine whether sensitive information is actually removed. Existing PII benchmarks are mostly text-centric and do not measure document-level redaction risk: a page remains unsafe if even one identifier is missed. We introduce LeakageBench, a challenge set of 500 document images with 11,954 GDPR-aligned PII annotations spanning direct identifiers, linkage keys, and contextual re-identification surfaces. We evaluate generic OCR pipelines, commercial and task-adapted OCR-dependent detectors, and OCR-free vision-language models using entity-level F1, group-wise leakage, and document-level leakage metrics. Code Interpreter raises GPT-5.5 localization F1 from 0.090 to 0.249, but critical page-level leakage remains 0.968. These results show that stronger detection and tool assistance improve localization without making most pages safe for release. LeakageBench provides a diagnostic benchmark for high-recall, spatially grounded PII redaction in document images.

1 Introduction

Organizations routinely redact personally identifiable information (PII) from scanned and rendered business documents. OCR errors, complex layouts, repeated identifiers, and visual noise make this more than a clean-text tagging problem. Most PII benchmarks measure span-level precision, recall, and F1 on text, while document-understanding benchmarks typically target selected key fields. Neither directly measures release safety: a page remains unsafe when any in-scope identifier remains visible. Evaluation must therefore capture both entity-level localization quality and residual page-level leakage.

Redaction presents an asymmetric operational trade-off. False positives unnecessarily obscure useful document content, whereas a single false negative can make a page unsafe to release. We therefore treat document-level leakage as a recall-oriented complement to, rather than a replacement for, localization and typing F1, which also penalize over-redaction through unmatched predictions.

We introduce **LeakageBench**, a challenge set of **500** document-page images containing **11,954** localized PII annotations. Its GDPR-aligned schema distinguishes *direct identifiers*, *linkage keys*, and *contextual re-identification surfaces*. Using a shared image-space interface, LeakageBench exposes a stark deployment gap: even the strongest evaluated configuration leaks critical Direct+Linkage PII on 96.8% of applicable pages. Our diagnostics attribute these failures to schema coverage, OCR recoverability, detection, interface validity, and spatial grounding.

Our contributions are: (1) a controlled, densely annotated document-image redaction challenge set; (2) a privacy-oriented schema covering direct, linkage, and contextual identifiers; (3) release-oriented document-level leakage metrics reported alongside entity-level F1; and (4) a unified diagnostic evaluation identifying why OCR-dependent and OCR-free systems continue to leak PII.

2 Related Work

Text-based PII detection and anonymization. Text-based benchmarks cover legal documents, regulatory categories, student writing, and clinical narratives (Pilán et al., 2022; Gambarelli et al., 2023; Mo et al., 2025; Holmes et al., 2024; Stubbs et al., 2015). They establish methods for span-level PII detection and anonymization, but exclude document-image artifacts, spatial localization, and page-level release safety.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 50519–50537
October 24-29, 2026 ©2026 Association for Computational Linguistics

Document-image understanding. FUNSD, SROIE, CORD, and DocVQA evaluate form understanding, receipt extraction, OCR, and visually grounded question answering (Jaume et al., 2019; Huang et al., 2021; Park et al., 2019; Mathew et al., 2020). BuDDIE broadens coverage to multiple business-document tasks and includes person- and address-related fields (Wang et al., 2025). However, these benchmarks target task-specific extraction rather than exhaustive PII localization and residual redaction leakage.

OCR-based and OCR-free document models. OCR-based models such as LayoutLM and LayoutLMv3 combine recognized text with two-dimensional layout (Xu et al., 2019; Huang et al., 2022), whereas OCR-free models such as Donut generate structured outputs directly from document images (Kim et al., 2022). For redaction, the former can propagate OCR and text-to-box alignment errors, while the latter must jointly learn visual reading and spatial localization. LeakageBench evaluates both through a common image-space prediction interface.

Leakage-oriented privacy evaluation. PRvL evaluates LLM-based PII redaction and residual exposure (Garza et al., 2025); ProPILE studies privacy leakage through targeted probing (Kim et al., 2023); and RedactBuster shows that redacted documents can remain vulnerable to downstream inference (Beltrame et al., 2024). These works motivate exposure-oriented evaluation, but do not provide a localized document-image benchmark with a page-level release criterion.

3 Task and Evaluation Protocol

3.1 Task definition

We evaluate *PII redaction on document images*. Given a page image—a scan, screenshot, or PDF rendering—a system must identify every visible region containing personally identifiable information, assign each region a schema type, and optionally extract its text value. Each gold instance is represented by an image-space bounding box, a fine-grained PII type, a risk group, and an optional verbatim value string.

The task is redaction-oriented rather than text-only: predictions must be spatially grounded so that the sensitive region can be masked on the image. This makes recall the primary safety concern. A page remains unsafe if even one in-scope identifier is missed, regardless of average entity-level performance. Inference is page-local: each rendered page is processed independently, and prediction–gold matching is performed only within that page. No cross-page context is provided. Accordingly, DocLeak is computed over document pages; evaluating multi-page contextual inference remains future work.

3.2 Prediction format

A system outputs a set of predicted instances

$$\hat{y} = \{(\hat{b}_j, \hat{t}_j, \hat{v}_j, \hat{s}_j)\}_{j=1}^{M},$$

where $\hat{b}_j = (x_{\min}, y_{\min}, x_{\max}, y_{\max})$ is an axis-aligned bounding box in pixels, $\hat{t}_j$ is a PII type, $\hat{v}_j$ is an optional extracted value string, and $\hat{s}_j \in [0, 1]$ is an optional confidence score. This unified interface supports both OCR-assisted pipelines, where boxes and values are derived from OCR spans, and OCR-free vision-language models.

3.3 Instance matching

We score predictions using one-to-one matching within each page. A predicted box $\hat{b}$ is eligible to match a ground-truth box b when $\mathrm{IoU}(\hat{b}, b) \geq \tau$. We enumerate all eligible pairs and greedily select them in descending IoU order, ensuring that each prediction and ground-truth instance is used at most once. A selected pair counts as a localization match regardless of type and as a typed match only when the predicted and ground-truth PII types agree. When values are available, value correctness is evaluated only for matched pairs.

Headline DocLeak uses type-aware matching: a gold instance is counted as protected only when an IoU-eligible prediction has the same PII type. This conservative criterion is used in Tables 3–4. Diagnostic attribution additionally uses type-agnostic matching because a wrongly typed but correctly localized box would still mask the identifier.

3.4 Metrics

We report entity-level F1 and document-level leakage. The former measures average detection quality; the latter measures whether any PII remains unmatched on a page.

Entity-level performance. $\mathrm{F1}_{\mathrm{loc}}$ is page-macro localization F1 using type-agnostic matching. $\mathrm{F1}_{\mathrm{type}}$ requires both $\mathrm{IoU} \geq \tau$ and the correct PII type, and is macro-averaged over ground-truth types. Unless stated otherwise, $\tau = 0.75$.

Document-level leakage. A page leaks if at least one in-scope ground-truth instance is unmatched:

$$\mathrm{DocLeak} = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \mathbb{1}[d \text{ leaks}], \qquad (1)$$

where $\mathcal{D}$ contains pages with at least one in-scope identifier. We report $\mathrm{DocLeak}_{\mathrm{all}}$ over all PII and $\mathrm{DocLeak}_{\mathrm{crit}}$ over Direct and Linkage identifiers.

Headline DocLeak uses type-aware matching. The error analyses additionally use type-agnostic matching to measure whether an identifier would be spatially masked despite an incorrect type. DocLeak penalizes unmatched gold instances but does not penalize extra predictions. We therefore report it jointly with $\mathrm{F1}_{\mathrm{loc}}$ and $\mathrm{F1}_{\mathrm{type}}$, whose precision terms count unmatched predictions as false positives. One-to-one IoU matching prevents a single prediction from receiving credit for multiple gold instances.

4 Dataset: Collection and Composition

4.1 Sources and generation process

LeakageBench contains **500** document-page images sampled from public operational-style business documents. We position the dataset as a challenge set for stress-testing image-based PII redaction under heterogeneous layouts, dense tables, scanned correspondence, mixed acquisition channels, and visual degradation. Rather than claiming to represent all deployment settings, LeakageBench covers a focused slice of business-document redaction workflows where identifiers appear across forms, invoices, emails, letters, and free-form correspondence.

Source A: OCR-IDL (public archive documents). We sample **103** documents (113) pages from OCR-IDL (Biten et al., 2022), which provides standardized OCR outputs for documents from the UCSF Industry Documents Library (IDL) [1], a large digital archive of industry documents hosted by UCSF. OCR-IDL is valuable for our setting because it includes heterogeneous, historically scanned/archived pages with substantial layout and rendering variation.

Source B: VRDU Ad-Buy Forms (structured, table-heavy invoices/receipts). We sample **107** documents (222) pages from the VRDU Ad-buy Forms corpus released by Google Research (Wang et al., 2023). These documents are public Federal Communications Commission (FCC) [2] filings and are dominated by structured invoices/receipts with multi-column layouts and repeated/nested table structures, providing a concentrated stress-test for localization within dense tabular regions.

Source C: Unstructured operational-style documents (free-form correspondence). We sample the remaining **81** documents (165) pages from FCC public filings, selecting free-form pages that resemble internal operational documents: email threads and attachments, letters and memos, scanned correspondence, and mixed-format "file" pages. Compared to invoice- or form-heavy sources, these pages exhibit less templated structure and place identifiers in narrative body text, signatures, headers/footers, and embedded attachments. This source therefore stresses redaction systems under realistic cross-field co-reference (the same individual referenced across multiple regions) and non-field PII placement.

4.2 Identifiability-first scope

Privacy regulation is triggered by *identifiability*, not by whether a string resembles a name. Under the GDPR, personal data includes identifiers such as names, identification numbers, and location data (Art. 4(1)) (European Union, 2016). Recital 26 emphasizes that identifiability must consider *all means reasonably likely to be used*, including *singling out* and linkage by other parties (European Union, 2016). Similarly, the CPRA/CCPA defines personal information as data that could reasonably be linked (directly or indirectly) to a consumer or household (California Legislature, 2025), and explicitly recognizes that personal information can exist in physical formats, including paper documents and printed images (California Legislature, 2025). In U.S. standards, PII is likewise defined to include information that is *linked or linkable* to an individual (National Institute of Standards and Technology, 2010).

Consequently, a redaction benchmark must treat as in-scope not only direct identifiers, such as names and contacts, but also *linkage keys* (policy numbers, case IDs, customer IDs) that resolve to a person through common workflows.

[1] https://industrydocuments.ucsf.edu/

[2] https://publicfiles.fcc.gov/

4.3 PII schema: re-identification surfaces

We annotate fine-grained labels for auditing, but report results using three privacy-aligned groups that reflect how identifiability arises in operational document workflows: (i) direct identification, (ii) lookup/linkage, and (iii) contextual inference. Table 1 summarizes the grouping used for reporting and analysis and lists the fine-grained labels covered under each group.

Rationale. **Direct identifiers** explicitly reveal identity or contact information (European Union, 2016; California Legislature, 2025). **Linkage keys** identify via lookup: they may appear innocuous on the page but act as stable keys into organizational systems or registries, consistent with standards that treat unique identifying numbers and codes as identifying (National Institute of Standards and Technology, 2010; U.S. Department of Health and Human Services, 2025) and with CPRA/CCPA "unique identifiers" (California Legislature, 2025). Finally, we annotate **contextual identifiers** (e.g., dates and entity fields) because they increase identifiability through co-occurrence and linkage in real documents (European Union, 2016; National Institute of Standards and Technology, 2010). We do not assume each contextual field is personal data in isolation; rather, it captures *re-identification surfaces* that amplify risk when combined with direct identifiers or linkage keys.

4.4 Dataset statistics and diagnostic views

Table 2 reports composition and label volume for **LeakageBench** (released) and for each source, including document and page counts and PII group totals. Redaction safety is determined by document-level worst cases. We therefore provide diagnostic views that expose the benchmark's risk profile: (i) the identifiers-per-page distribution (Fig. 1) and (ii) the spatial concentration of PII on the page (Fig. 2).

Identifiers-per-page distribution. Figure 1 shows the distribution of PII instances per page. This distribution is critical for safety: a page leaks if any identifier is missed, so pages with many identifiers place disproportionate pressure on redaction systems. Across the collected pool, we annotate **11,954** PII instances over **500** pages (**23.90** per page on average), including **4,315** critical identifiers from the Direct and Linkage categories (**8.63** per page on average). We therefore report document-level leakage alongside entity-level scores (Sec. 3.4).

Spatial PII heatmap. Figure 2 visualizes where identifiers appear on the page. We partition each image into a regular grid of non-overlapping 14×14-pixel patches (stride = 14), yielding 3,600 patches per page (60 × 60). For each patch, we record which PII categories overlap its area and aggregate the results across all pages. The resulting heatmap distinguishes benchmarks where PII concentrates in predictable regions-such as headers or signatures-from those where it is distributed throughout tables and body text. We report both an "all PII" heatmap and a "critical PII" heatmap restricted to Direct+Linkage identifiers.

Document types. LeakageBench is dominated by operational business documents: invoices, emails, forms, letters/correspondence, and file-like pages. Full page-type composition by source is reported in Appendix D.

4.5 Annotation procedure

All annotations were produced by the first two authors. The 500 page images were randomly split equally and annotated independently following the guidelines in Appendix A. Annotators labeled entity instances using tight bounding boxes around visible PII spans and transcribed `value` strings verbatim (no normalization); multi-line items use the delimiter "\n". We annotate fine-grained entity types and additionally report privacy-aligned group aggregates (Direct/Linkage/Contextual) for analysis.

4.6 Annotation quality control

To assess annotation consistency, each annotator contributed a 10% stratified sample of their assigned pages for cross-annotation by the other annotator (**50 images / pages total**). Disagreements were resolved by joint adjudication to form a single gold set.

Inter-annotator agreement is measured with Cohen's κ over entity instances after alignment. We align annotations via one-to-one bipartite matching using IoU $\geq \tau$ with τ=0.75. A matched pair is counted as agreement only if both boxes share the same fine-grained type; spatial overlaps with a type mismatch are counted as disagreements. Any annotation with no IoU≥ 0.75 counterpart from the other annotator is treated as a miss (disagreement). Under this protocol, we obtain $\kappa = 0.979$ on the cross-annotated subset.

Group	Fine-grained labels (annotated)	Basis (examples)
Direct identifiers	PERSON NAME, ADDRESS, EMAIL ADDRESS, PHONE NUMBER	GDPR; CPRA/CCPA (European Union, 2016; California Legislature, 2025)
Linkage keys	CUSTOMER ID, EMPLOYEE ID, POLICY NO., ID CARD NO., CASE/FILE/REFERENCE NO., CONTRACT/INVOICE/CHEQUE NO., REGISTRATION NO., CALL SIGN, DA CASE NUMBER	NIST; HIPAA; CPRA/CCPA (National Institute of Standards and Technology, 2010; U.S. Department of Health and Human Services, 2025; California Legislature, 2025)
Contextual identifiers	DATE, LOCATION, ORGANIZATION NAME, BUSINESS ID	GDPR Recital 26; NIST (European Union, 2016; National Institute of Standards and Technology, 2010)

Table 1: Privacy-aligned schema groups and supporting regulatory anchors. We annotate fine-grained labels and additionally report aggregated results by group.

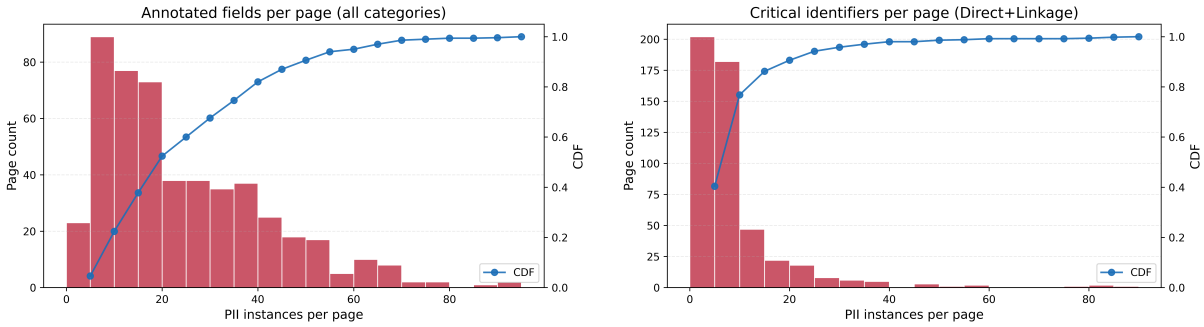

Figure 1: Per-page PII density in LeakageBench (histogram with CDF overlay). Left: all annotated PII. Right: safety-critical identifiers (Direct+Linkage).

5 Experimental Setup

We evaluate off-the-shelf systems under one unified image-level redaction protocol. The baselines cover two architectures. First, **OCR-dependent systems** read the page with OCR, detect PII in the recognized text, and project detections back to image coordinates using OCR word boxes. Within this family, we compare a generic redaction pipeline against stronger commercial and task-adapted privacy detectors. Second, **OCR-free vision-language models** receive the page image directly and predict PII regions without an explicit OCR stage. All systems are converted to the same prediction format (Sec. 3.2) and evaluated with entity-level localization, localization+typing, document-level leakage, schema coverage, and supported-type leakage (Sec. 3.4). No system is trained or tuned on LeakageBench.

5.1 OCR-dependent baselines

All OCR-dependent baselines use the same image-to-box pipeline: OCR extracts text and word-level boxes, a privacy detector predicts typed PII spans over the OCR text, and each span is mapped back to the page by merging the corresponding word boxes.

Generic OCR-dependent redaction. We use Microsoft Presidio Image Redactor (Microsoft, 2026) as the generic OCR-dependent baseline. Presidio applies OCR, detects PII spans with Presidio Analyzer, and returns image-space redaction boxes through OCR word alignment. We use Tesseract (Smith, 2007) for this baseline and map Presidio entity types to the LeakageBench schema with a fixed mapping.

Commercial and task-adapted OCR-dependent detectors. We evaluate stronger privacy-oriented detectors on top of Amazon Textract OCR (Amazon Web Services, 2026b): GLiNER-base (Zaratiana et al., 2024), Presidio-best (GLiNER restricted), GLiNER multi-PII (Zaratiana, 2024), NVIDIA GLiNER PII (NVIDIA, 2025), GLiNER2 (Zaratiana et al., 2025), Google DLP (Google Cloud, 2026), Amazon Comprehend PII (Amazon Web Services, 2026a) and an OpenAI Privacy Filter (OpenAI, 2026) pipeline. These systems test

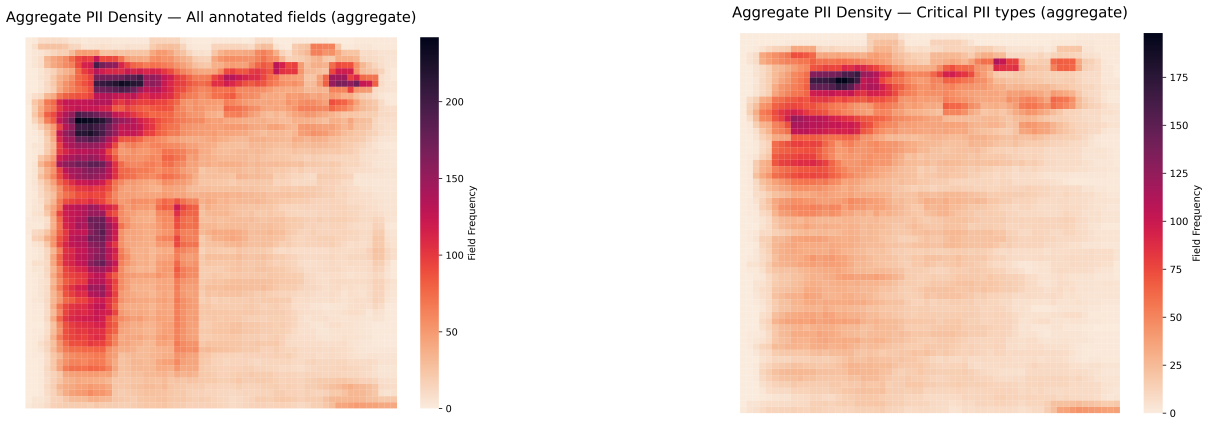

Figure 2: Spatial PII density aggregated over pages. Left: all annotated fields. Right: critical identifiers (Direct+Linkage).

Source	#Docs	#Pages	Direct	Linkage	Contextual	Other	Critical/Page
Source A (OCR-IDL)	103	113	1,143	109	445	17	11.08
Source B (VRDU Ad-Buy)	107	222	673	221	6,245	192	4.03
Source C (FCC free-form)	81	165	1,963	206	710	30	13.15
Total	291	500	3,779	536	7,400	239	8.63

Table 2: Source composition and label volume. **Other** aggregates non-group labels (e.g., Signature, Agency Code/Ref, Advertiser Ref). **Critical/Page** is (Direct+Linkage)/Pages.

whether stronger text-side PII detectors are sufficient once document-image constraints—OCR errors, dense layouts, repeated identifiers, span fragmentation, and box reconstruction—are introduced.

Schema alignment. All OCR-dependent systems are scored against the LeakageBench schema. Unsupported entity types count as missed detections under full-schema evaluation. We therefore report schema coverage and supported-type leakage to separate taxonomy limitations from empirical detection and localization failures.

Appendix B summarizes the OCR engines, detector settings, schema mappings, and conversion rules for each OCR-dependent baseline.

5.2 OCR-free vision-language baselines

OCR-free vision-language localization. We evaluate **Qwen3-VL-32B** (Bai et al., 2025) and **InternVL3-38B** (Zhu et al., 2025) as OCR-free vision-language baselines because they support explicit 2D grounding from image input. Unlike OCR-dependent systems, these models receive the page image directly and are prompted to return PII types, values, confidence scores, and bounding boxes. Predicted boxes are converted to image coordinates and evaluated under the same localization-and-typing protocol as all other systems. Coordinate conventions and parsing rules are summarized in Appendix B; full prompt templates are provided in Appendix C.

Closed OCR-free models. We additionally evaluate GPT-5.4 using single-pass vision and GPT-5.5 in two configurations: single-pass vision and Code Interpreter-assisted inspection. All configurations use the same PII taxonomy and semantic instructions and return structured predictions with integer bounding boxes in a normalized 0–999 coordinate system. For GPT-5.5, we use high image detail and medium reasoning effort; in the tool-assisted configuration, Code Interpreter is required to inspect the page and may crop or zoom regions before returning the final prediction. All outputs are processed using the same parsing, coordinate conversion, and evaluation pipeline as the other OCR-free models.

Prompt-driven schema coverage. For OCR-free systems, the full LeakageBench schema is supplied in the prompt. Their schema coverage is therefore 1.0 by configuration: the model is allowed to emit every benchmark type, but this does not imply reliable detection. Empirical failures are captured by entity-level F1, document-level leakage, group leakage, and output-validity rates.

Why not LayoutLM-style trained baselines. LeakageBench is evaluated as an off-the-shelf challenge set: systems must run without training on the benchmark and must return PII types with image-space boxes. LayoutLM-style document NER models typically require supervised fine-tuning, task-specific heads, OCR-token alignment, and a train/dev/test split. We therefore leave layout-aware trained models to a future trained-baseline track.

6 Results

Table 3 reports full-schema performance at IoU $\tau = 0.75$ under the unified prediction format. To assess sensitivity to box tightness, Appendix G reports results at $\tau \in \{0.50, 0.75, 0.95\}$. We report localization F1, localization+typing F1, document-level leakage over all identifiers, and leakage over safety-critical Direct+Linkage identifiers. Unless stated otherwise, unsupported schema types, invalid outputs, and out-of-bounds boxes are counted as missed detections.

For systems with restricted taxonomies, we additionally report supported-type leakage in Table 4, where gold labels are filtered to the entity types each system can emit. Schema coverage and full-schema group-wise leakage are reported in Appendix E and Appendix F. Valid denotes the fraction of pages producing parseable, schema-conformant outputs. Malformed or unusable outputs are scored as empty prediction sets; individual invalid boxes are rejected and receive no credit.

6.1 Analysis

Overall safety picture. Table 3 shows that stronger systems improve entity-level localization without achieving document-level safety. Amazon Comprehend PII obtains the highest localization F1 ($F1_{loc} = 0.304$), while GPT-5.5 with Code Interpreter obtains the highest typed F1 ($F1_{type} = 0.119$). Nevertheless, both have $DocLeak_{crit} = 0.968$, showing that better average detection does not guarantee that a page is safe to release.

This conclusion is robust to the matching scope. Crediting any IoU-eligible box regardless of its predicted type reduces the $DocLeak_{crit}$ range from 0.968–1.000 under headline type-aware matching to 0.945–1.000 under type-agnostic matching. Thus, the high leakage rates are not primarily caused by type-label errors.

Leakage is severe, not marginal. To complement binary DocLeak, we count unmatched critical identifiers per page. Across systems, leaking pages retain 5.3–8.7 critical identifiers on average. Even the lowest-severity system misses 5.3 per leaking page (median 4; P90 11), corresponding to approximately 66% of critical identifiers across applicable pages. Thus, leakage typically reflects multiple exposed identifiers rather than an isolated miss. Complete per-system severity statistics are reported in Table 13.

Schema coverage is not enough. Schema coverage diagnostics are reported in Appendix E. Restricted systems such as Presidio, Presidio-best (GLiNER restricted), and Amazon Comprehend PII cover only 20.8% of benchmark types, while Google DLP and the OpenAI Privacy Filter cover 25.0%. These restricted systems cover Direct identifiers but miss most Linkage keys under our schema mapping, making lookup-style identifiers a persistent coverage gap. However, full label coverage does not solve the task either: GLiNER-base, GLiNER multi-PII, and NVIDIA GLiNER PII are configured with full schema coverage, yet their Linkage leakage remains near-total in Appendix F. This suggests that sparse lookup keys, dense tables, OCR fragmentation, and layout variation remain difficult even when the label set is available.

Supported-schema leakage isolates detection failures. Table 4 filters gold labels to the types each system can emit and reports leakage by privacy group. Even under this easier setting, leakage remains high: full-schema OCR-dependent systems still leak on more than 94% of Direct pages and more than 96% of Linkage pages. This shows that failures are not only taxonomy mismatches; OCR errors, repeated identifiers, span fragmentation, and box reconstruction also contribute to residual exposure.

OCR-dependent failure attribution. Full results appear in Appendix Tables 14–15. Using type-agnostic matching, we attribute each leaked critical identifier to an OCR miss, a detector miss with zero box overlap, or a box/alignment failure with positive overlap below $\tau = 0.75$; one-to-one assignment conflicts account for at most 0.2%. Box/alignment failure is the largest category for seven of nine OCR-dependent systems, accounting for 45.9–84.5% of their attributable leaks, while detector misses vary from 15.5 to 48.8%. Independently

System	OCR	$F1_{loc}$ ↑	$F1_{type}$ ↑	$DocLeak_{all}$ ↓	$DocLeak_{crit}$ ↓	Valid↑
Generic baseline *(Tesseract OCR)*						
Presidio	Tesseract	0.137	0.048	0.994	0.990	0.996
Shared AWS Textract OCR *(detector varies; OCR is fixed)*						
Presidio-best (GLiNER restr.)	Textract	0.248	0.064	0.996	0.986	**1.000**
GLiNER-base	Textract	0.216	0.063	**0.988**	0.978	**1.000**
GLiNER-multi-PII	Textract	0.245	0.091	0.990	0.972	**1.000**
NVIDIA GLiNER PII	Textract	0.288	0.109	0.992	0.986	**1.000**
GLiNER2	Textract	0.156	0.065	1.000	0.994	0.998
Amazon Comprehend PII	Textract	**0.304**	0.085	0.992	**0.968**	**1.000**
OpenAI Privacy Filter	Textract	0.164	0.055	0.996	0.990	0.806
Vendor-managed OCR						
Google DLP	Vendor	0.076	0.013	1.000	1.000	**1.000**
OCR-free vision-language models *(no separate OCR stage)*						
Qwen3-VL-32B	—	0.073	0.046	1.000	0.998	0.666
InternVL3-38B	—	0.012	0.019	1.000	1.000	0.540
GPT-5.4	—	0.044	0.031	1.000	1.000	**1.000**
GPT-5.5	—	0.090	0.050	1.000	0.990	**1.000**
GPT-5.5 + Code Interpreter	—	0.249	**0.119**	0.990	**0.968**	**1.000**

Table 3: Full-schema results on **LeakageBench** at IoU $\tau = 0.75$. Textract systems share identical OCR, isolating detector differences. $F1_{loc}$ is page-macro type-agnostic localization F1; $F1_{type}$ also requires the correct type. $DocLeak_{all}$ and $DocLeak_{crit}$ are the fractions of pages leaking any identifier and any Direct+Linkage identifier, respectively. Unsupported types and invalid outputs count as misses; Table 4 controls for schema coverage. Valid denotes parseable, schema-conformant page outputs. Bold marks the best result, including ties.

of any detector, Textract cannot recover 623/4,280 value-bearing critical annotations (14.6%), compared with 765/4,280 (17.9%) for Tesseract. Performance differences among the Textract-based pipelines therefore arise primarily downstream of their shared OCR stage.

Formatting versus localization in OCR-free VLMs. Table 5 separates interface failures from failures on valid outputs using a common denominator of leaked critical identifiers. Interface failures account for 27.8% and 36.9% of leakage for Qwen3-VL-32B and InternVL3-38B, respectively, but none for GPT-5.4 or either GPT-5.5 configuration. Among residual leaked critical instances, sub-threshold boxes account for 51.7% for Qwen3-VL-32B, 83.5% for GPT-5.4, 96.1% for GPT-5.5, and 98.9% for GPT-5.5 with Code Interpreter. These percentages describe the composition of the errors that remain, not the overall localization-failure rate. A larger box-failure share can therefore coexist with improved $F1_{loc}$ when total leakage decreases and zero-overlap misses decline more sharply. InternVL3-38B instead exhibits a larger share of zero-overlap misses (35.1%). Thus, interface reliability is necessary but insufficient; spatial localization remains the dominant limitation for the GPT configurations and Qwen3-VL-32B.

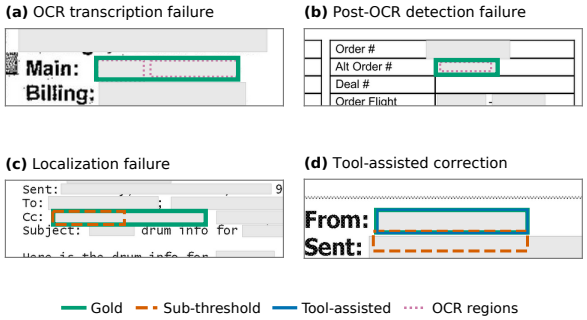

Figure 3: Representative failure modes. Panels (a)–(c) use GLiNER-multi-PII with Textract; panel (d) compares GPT-5.5 single-pass and tool-assisted localization. Green denotes gold boxes, orange dashed boxes sub-threshold predictions, blue tool-assisted predictions, and magenta dotted boxes OCR regions.

Tool assistance improves OCR-free localization, but leakage remains high. Figure 3 illustrates these failures and a tool-assisted localization correction. Within GPT-5.5, Code Interpreter raises $F1_{loc}$ from 0.090 to 0.249 and $F1_{type}$ from 0.050 to 0.119. It reduces $DocLeak_{crit}$ from 0.990 to 0.968 and reduces Direct and Contextual leakage from 0.956 to 0.887 and from 0.986 to 0.908, respectively. Linkage leakage nevertheless remains 0.964. Tool-assisted inspection therefore improves spatial grounding substantially, but does not yet provide release-level redaction safety.

System	OCR	Coverage↑	Unsup. pages↓	Direct^{sup} ↓	Linkage^{sup} ↓	Contextual^{sup} ↓	All^{sup} ↓
Restricted schema coverage *(unsupported gold types are excluded)*							
Presidio	Tesseract	0.208	0.900	0.973	–	0.923	0.990
Presidio-best	Textract	0.208	0.900	0.967	–	0.942	0.992
GLiNER2	Textract	**0.958**	**0.094**	0.971	0.985	0.963	1.000
Google DLP	Vendor	0.250	0.762	1.000	–	0.990	0.998
Amazon Comprehend PII	Textract	0.208	0.900	0.908	–	0.873	0.968
OpenAI Privacy Filter	Textract	0.250	0.850	0.971	0.994	0.991	0.990
Full schema coverage with shared AWS Textract OCR *(detector varies; OCR and schema are fixed)*							
GLiNER-base	Textract	1.000	0.000	0.954	**0.967**	0.924	**0.988**
GLiNER-multi-PII	Textract	1.000	0.000	0.950	0.973	**0.908**	0.990
NVIDIA GLiNER PII	Textract	1.000	0.000	**0.942**	**0.967**	0.916	0.992
Full schema coverage without a separate OCR stage *(OCR-free VLMs)*							
Qwen3-VL-32B	–	1.000	0.000	0.990	0.982	0.975	1.000
InternVL3-38B	–	1.000	0.000	1.000	1.000	0.998	1.000
GPT-5.4	–	1.000	0.000	0.998	0.988	0.990	1.000
GPT-5.5	–	1.000	0.000	0.956	0.994	0.986	1.000
GPT-5.5 + Code Interpreter	–	1.000	0.000	**0.887**	**0.964**	**0.908**	**0.990**

Table 4: Supported-schema leakage at IoU $\tau = 0.75$. Coverage is the fraction of benchmark types supported; Unsup. pages contain at least one unsupported gold type. Group scores retain only supported gold types. The full-schema Textract block fixes OCR and schema, isolating the detector. Bold marks the best coverage within the restricted-schema block and the lowest leakage within each full-schema block. Restricted-schema leakage is not ranked because supported type sets differ. Dashes denote no supported group type or no separate OCR stage.

System	Valid ↑	Leaked crit.	Interface ↓	Zero-IoU ↓	Box fail ↓	Assignment ↓
Qwen3-VL-32B	.666	4,013	27.8	20.5	51.7	0.0
InternVL3-38B	.540	4,274	36.9	35.1	28.0	0.0
GPT-5.4	1.000	4,060	0.0	16.5	83.5	0.0
GPT-5.5	1.000	3,787	0.0	3.9	96.1	0.0
GPT-5.5 + Code Interpreter	1.000	2,549	0.0	0.9	98.9	0.2

Table 5: Error attribution for leaked Direct and Linkage identifiers from OCR-free VLMs at $\text{IoU} = 0.75$, using type-agnostic matching. Attribution columns are percentages of the leaked critical instances shown in the second numeric column and sum to 100%. Interface denotes identifiers on pages without a fully valid output; Zero-IoU denotes no overlapping prediction; Box fail denotes positive overlap below the IoU threshold; and Assignment denotes one-to-one matching conflicts.

Takeaway. LeakageBench is best viewed as a redaction-safety challenge set rather than a leaderboard where systems differ only by small margins. The benchmark exposes complementary failure modes: restricted taxonomies omit identifier groups, OCR-dependent pipelines propagate recognition and alignment errors, and OCR-free VLMs remain limited by spatial grounding. Code Interpreter substantially improves GPT-5.5 localization, but the resulting system still leaks critical PII on 96.8% of applicable pages. These findings motivate systems that combine broad schema coverage, robust visual reading, and tight, high-recall localization.

Ethical, Legal, and Release Protocol

LeakageBench is intended for defensive research on document-redaction safety. Although the source documents originate from public collections, combining real document images with fine-grained PII annotations introduces privacy and dual-use risks. The document images and annotations will therefore be made available for research use under a privacy- and license-aware Data Use Agreement (DUA).

The benchmark is intended for evaluating privacy-preserving redaction, not for re-identification or PII mining. Its GDPR-aligned schema supports comparative evaluation but does not define legal compliance or prescribe which fields must be redacted in every jurisdiction or deployment setting.

Limitations

LeakageBench is a 500-page challenge set drawn from public business documents, not a representative sample of all redaction settings. It does not cover heavily handwritten, medical, legal-

discovery, multilingual, or non-U.S. documents. Evaluation is page-local and does not measure multi-page contextual inference.

The schema follows an identifiability-first view. Direct identifiers and Linkage keys are primary release-safety targets, whereas masking Contextual identifiers depends on the applicable policy.

Headline DocLeak uses conservative type-aware matching at IoU $\tau = 0.75$, so a wrongly typed but correctly localized box counts as a leak. However, type-agnostic matching yields the same substantive conclusion, with $\text{DocLeak}_{\text{crit}}$ remaining between 0.945 and 1.000. Axis-aligned boxes and IoU do not directly measure pixel coverage of skewed, handwritten, or irregular regions. Although F1 penalizes extra predictions, we do not define a document-level over-redaction or utility metric.

Error attribution is limited by the stored artifacts: 35 critical annotations lack usable verbatim values, span-level alignment logs are unavailable, and Google DLP's vendor OCR is not stored. Consequently, box and alignment errors cannot be separated fully. The OCR-free results also represent specific model and inference configurations whose API behavior may change.

References

Amazon Web Services. 2026a. Amazon comprehend: Detecting personally identifiable information. https://docs.aws.amazon.com/comprehend/latest/dg/pii.html. Accessed: 2026-05-17.

Amazon Web Services. 2026b. Amazon textract documentation. https://docs.aws.amazon.com/textract/latest/dg/what-is.html. Accessed: 2026-05-17.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025. Qwen3-vl technical report. *Preprint*, arXiv:2511.21631.

Mirco Beltrame, Mauro Conti, Pierpaolo Guglielmin, Francesco Marchiori, and Gabriele Orazi. 2024. Redactbuster: Entity type recognition from redacted documents. *Preprint*, arXiv:2404.12991.

Ali Furkan Biten, Rubèn Tito, Lluis Gomez, Ernest Valveny, and Dimosthenis Karatzas. 2022. Ocr-idl: Ocr annotations for industry document library dataset. *Preprint*, arXiv:2202.12985.

California Legislature. 2025. California civil code 1798.140 (definitions) (cpra/ccpa). California Legislative Information (Official). See 1798.140(v)(1); 1798.140(v)(4) (formats); 1798.140(aj) (unique identifier).

European Union. 2016. Regulation (eu) 2016/679 (general data protection regulation). EUR-Lex (Official Journal of the European Union). See Art. 4(1), Recital 14, Recital 26.

Gaia Gambarelli, Aldo Gangemi, and Rocco Tripodi. 2023. Is your model sensitive? spedac: A new resource for the automatic classification of sensitive personal data. *IEEE Access*, 11:10864–10880.

Leon Garza, Anantaa Kotal, Aritran Piplai, Lavanya Elluri, Prajit Das, and Aman Chadha. 2025. Prvl: Quantifying the capabilities and risks of large language models for pii redaction. *Preprint*, arXiv:2508.05545.

Google Cloud. 2026. Google cloud sensitive data protection / cloud data loss prevention. https://cloud.google.com/security/products/dlp. Accessed: 2026-05-17.

Langdon Holmes, Scott Crossley, Perpetual Baffour, Jules King, Lauryn Burleigh, Maggie Demkin, Ryan Holbrook, Walter Reade, and Addison Howard. 2024. The learning agency lab - pii data detection. https://kaggle.com/competitions/pii-detection-removal-from-educational-data. Kaggle.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. *Preprint*, arXiv:2204.08387.

Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and C. V. Jawahar. 2021. ICDAR2019 competition on scanned receipt OCR and information extraction. *CoRR*, abs/2103.10213.

Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 2019. Funsd: A dataset for form understanding in noisy scanned documents. In *2019 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 (ECCV)*.

Siwon Kim, Sangdoo Yun, Hwaran Lee, Martin Gubri, Sungroh Yoon, and Seong Joon Oh. 2023. Propile: Probing privacy leakage in large language models. *Preprint*, arXiv:2307.01881.

Minesh Mathew, Dimosthenis Karatzas, R. Manmatha, and C. V. Jawahar. 2020. Docvqa: A dataset for VQA on document images. *CoRR*, abs/2007.00398.

Microsoft. 2026. Microsoft presidio: Data protection and de-identification sdk. https://microsoft.github.io/presidio/. Accessed: 2026-05-17.

Fan Mo, Bo Liu, Yuan Fan, Kun Qin, Yizhou Zhao, Jinhe Zhou, Jia Sun, Jinfei Liu, and Kui Ren. 2025. DataSIR: A benchmark dataset for sensitive information recognition. In *NeurIPS 2025 Datasets and Benchmarks Track*. Poster.

National Institute of Standards and Technology. 2010. Guide to protecting the confidentiality of personally identifiable information (pii). Technical Report Special Publication 800-122, NIST.

NVIDIA. 2025. NVIDIA GLiNER PII. https://huggingface.co/nvidia/gliner-PII. Accessed: 2026-05-17.

OpenAI. 2026. Openai privacy filter. https://github.com/openai/privacy-filter. Accessed: 2026-05-17.

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*.

Ildikó Pilán, Pierre Lison, Lilja Øvrelid, Anthi Papadopoulou, David Sánchez, and Montserrat Batet. 2022. The text anonymization benchmark (TAB): A dedicated corpus and evaluation framework for text anonymization. *Computational Linguistics*, 48(4):1053–1101.

Ray Smith. 2007. An overview of the tesseract ocr engine. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR)*, volume 2, pages 629–633. IEEE Computer Society.

Amber Stubbs, Christopher Kotfila, and Özlem Uzuner. 2015. Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/uthealth shared task track 1. *Journal of Biomedical Informatics*, 58:S11–S19. Supplement: Proceedings of the 2014 i2b2/UTHealth Shared-Tasks and Workshop on Challenges in Natural Language Processing for Clinical Data.

U.S. Department of Health and Human Services. 2025. 45 cfr 164.514 — other requirements relating to uses and disclosures of protected health information. Electronic Code of Federal Regulations (eCFR). See §164.514(b)(2) de-identification safe harbor identifiers.

Dongsheng Wang, Ran Zmigrod, Mathieu J. Sibue, Yulong Pei, Petr Babkin, Ivan Brugere, Xiaomo Liu, Nacho Navarro, Antony Papadimitriou, William Watson, Zhiqiang Ma, Armineh Nourbakhsh, and Sameena Shah. 2025. BuDDIE: A business document dataset for multi-task information extraction. In *Proceedings of the Joint Workshop of the 9th Financial Technology and Natural Language Processing (FinNLP), the 6th Financial Narrative Processing (FNP), and the 1st Workshop on Large Language Models for Finance and Legal (LLMFinLegal)*, pages 35–47, Abu Dhabi, UAE. Association for Computational Linguistics.

Zilong Wang, Yichao Zhou, Wei Wei, Chen-Yu Lee, and Sandeep Tata. 2023. Vrdu: A benchmark for visually-rich document understanding. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 5184–5193. ACM.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2019. Layoutlm: Pre-training of text and layout for document image understanding. *CoRR*, abs/1912.13318.

Urchade Zaratiana. 2024. GLiNER multi pii model. https://huggingface.co/urchade/gliner_multi_pii-v1. Accessed: 2026-05-17.

Urchade Zaratiana, Gil Pasternak, Oliver Boyd, George Hurn-Maloney, and Ash Lewis. 2025. GLiNER2: Schema-driven multi-task learning for structured information extraction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 130–140, Suzhou, China. Association for Computational Linguistics.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2024. GLiNER: Generalist model for named entity recognition using bidirectional transformer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5364–5376, Mexico City, Mexico. Association for Computational Linguistics.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, and 32 others. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *Preprint*, arXiv:2504.10479.

A Annotation Instructions

These instructions describe how to annotate **LeakageBench** for privacy redaction. The goal is to mark *every visible region* on a page that contains an in-scope identifier, using tight bounding boxes and an exact transcription of the text.

A.1 What to label (schema)

Annotate **all** instances of the following fine-grained PII types (must match exactly):

- **Direct identifiers:** PERSON NAME, ADDRESS, EMAIL ADDRESS, PHONE NUMBER
- **Linkage keys:** CUSTOMER ID, EMPLOYEE ID, POLICY NUMBER, ID CARD NUMBER, FILE NUMBER, REFERENCE NUMBER,

CONTRACT NUMBER, INVOICE NUMBER, CHEQUE NUMBER, REGISTRATION NUMBER, CALL SIGN, DA CASE NUMBER

- **Operational references:** AGENCY CODE, AGENCY REF, ADVERTISER REF

- **Contextual identifiers:** DATE, ORGANIZATION NAME, BUSINESS ID, LOCATION

- **Other (visual PII):** SIGNATURE (handwritten signatures or signature marks)

Important: If a page contains multiple identifiers, annotate *all of them*. This benchmark is safety-driven: a page is considered leaked if *any* in-scope identifier is missed.

A.2 How to draw boxes (localization)

- Draw a **tight bounding box** around the **visible identifier text** (or signature mark). Boxes should cover the characters with minimal background.

- Do **not** box entire rows, table blocks, or paragraphs unless *the entire region is PII.*

- If an identifier spans multiple lines (e.g., an address block), draw **one box covering the full multi-line span** if the tool supports it. Otherwise, use multiple boxes and keep the transcription consistent across lines.

- If the identifier is repeated on the page (e.g., header and footer), annotate **each occurrence** separately.

A.3 How to enter the text value (transcription)

- Transcribe the **value exactly as it appears** in the image.

- Do **not** correct spelling, punctuation, casing, or spacing.

- Do **not** normalize formats (dates, phone numbers, IDs, etc.). Keep the original formatting.

- For multi-line values, use the literal token `\n` to indicate line breaks (e.g., `Line1\nLine2\nLine3`).

A.4 Type selection rules (common edge cases)

Use the most specific schema type that matches the *role* of the text on the page.

- **Person Name:** Names of individuals (signers, recipients, customers, employees). If the page contains initials, titles, or suffixes as part of the displayed name, include them as shown.

- **Organization Name:** Company, agency, or institution names. Do not label generic department words unless they clearly function as an entity name in context.

- **Address:** Postal addresses (street + city/state/zip) or address blocks. Include apartment/suite/unit numbers if present.

- **Email Address:** Email addresses exactly as written.

- **Phone Number:** Telephone numbers (including extensions if shown).

- **Date:** Dates and date ranges as written (including billing periods and coverage periods).

- **Invoice Number / Contract Number / Policy Number / Reference Number / File Number:** Choose the type based on the field label and context. If the page label says "Invoice #", annotate as INVOICE NUMBER. If it says "Policy No.", annotate as POLICY NUMBER, etc.

- **Customer ID / Employee ID / Business ID:** Use when the page explicitly indicates the identifier category (e.g., "Customer ID", "Employee ID", "UEN/ABN/VAT", "Business ID").

- **Call Sign:** Broadcast or station identifiers (e.g., `KXAS`, `WLAX/WEUX`) when used as a station/call sign.

- **Agency Code / Agency Ref / Advertiser Ref / DA Case Number:** Use when the page contains explicit labeled codes corresponding to these identifiers.

- **Signature:** Handwritten signature marks or signature scribbles. Do not label printed names near the signature line as SIGNATURE;

When unsure. If you cannot confidently map a span to one of the schema types, **do not guess**. Prefer leaving it unannotated over inventing a type.

A.5 Pre-marked / highlighted regions

Some pages may contain regions already highlighted or redacted (e.g., dark blocks, overlays, pre-annotations). These are **not reliable ground-truth evidence of PII**.

- Do **not** annotate pre-redacted regions where the text is no longer readable.
- If the text remains visible despite highlighting, annotate it normally based on what is visible.

A.6 Quality checklist before submitting a page

Before finalizing a page, quickly verify:

- All visible names, emails, phone numbers, and addresses are boxed.
- All visible linkage keys (invoice/contract/policy/reference/file/customer IDs, etc.) are boxed.
- Boxes are tight and not overly large.
- Transcriptions are verbatim and include \n for multi-line items.

B Baseline Reproducibility Details

This appendix records the implementation choices needed to reproduce the baseline evaluations. No baseline is trained or tuned on LeakageBench. All outputs are converted to the same format: PII type, optional value, confidence score, and image-space bounding box.

Type scope controls which LeakageBench labels a system can emit before scoring; it is used only for schema coverage and supported-type leakage, not for full-schema evaluation.

OCR-dependent conversion. OCR-dependent systems detect PII over recognized text and recover page coordinates from OCR word boxes. Presidio uses Tesseract OCR. The stronger OCR-dependent baselines use Amazon Textract word boxes, except for services that provide their own text recognition or span interface. For each detected text span, we align the span to OCR words and use the minimal enclosing rectangle over the matched word boxes. Unaligned detections are discarded.

Schema mapping and unsupported types. All predictions are mapped to the LeakageBench schema before scoring. For systems with built-in or restricted taxonomies, unmapped prediction labels are discarded. Under full-schema evaluation, unsupported gold types remain in scope and count as missed detections. Supported-type leakage filters gold labels to the entity types each system can emit.

OCR-free conversion and validation. OCR-free VLMs receive the page image directly and output normalized boxes, which we convert to pixel coordinates before scoring. The full LeakageBench schema is supplied in the prompt, so OCR-free schema coverage is 1.0 by configuration. Malformed outputs, labels outside the schema, invalid boxes, and out-of-bounds boxes are scored as empty predictions; we do not apply model-specific repair or manual correction.

C Prompt Templates

C.1 Shared prompt header

Listing 1: Shared prompt header used for all models.

```
PROMPT_HEADER = r"""
You are a privacy redaction detector for BUSINESS DOCUMENT IMAGES (emails, invoices, forms, letters).

TASK
Detect and localize ALL regions that contain PII (Personally Identifiable Information) on this single page.

DRAWING BOXES
Draw boxes tightly around the visible PII text (do not box entire rows/tables unless the whole region is PII).

SCORE
Set score to your confidence [0,1] that (a) the box tightly encloses the PII text and (b) the type is correct; use lower scores for uncertain cases, and if you include a prediction you must still provide a score

ALLOWED PII TYPES (MUST MATCH EXACTLY)
{schema_types}

TYPE CONSTRAINT (IMPORTANT)
- You MUST output ONLY the allowed PII types listed above.
- Do NOT output any other labels (no synonyms, no extra categories).
- If you are unsure of the correct allowed type, OMIT the prediction.

NO-PII CASE
```

System	OCR / boxes	Detector	Type scope
Presidio	Tesseract	spaCy en_core_web_lg	mapped built-in types
Presidio-best	Textract	GLiNER restricted	restricted PII types
GLiNER-base	Textract	GLiNER-base	full LeakageBench schema
GLiNER-multi-PII	Textract	GLiNER multi-PII	PII-specialized types
NVIDIA GLiNER PII	Textract	NVIDIA GLiNER PII	PII-specialized types
GLiNER2	Textract	GLiNER2	mapped schema (23/24 types)
Google DLP	service/OCR text	Google DLP	mapped built-in types
Amazon Comprehend PII	Textract	Amazon Comprehend PII	mapped built-in types
OpenAI Privacy Filter	Textract	OpenAI Privacy Filter	mapped built-in types
Qwen3-VL-32B	image	Qwen3-VL-32B	prompted full schema
InternVL3-38B	image	InternVL3-38B	prompted full schema
GPT-5.4	image	GPT-5.4	prompted full schema
GPT-5.5	image	GPT-5.5	prompted full schema
GPT-5.5 + Code Interpreter	image + tool inspection	GPT-5.5	prompted full schema

Table 6: Baseline inventory. "OCR / boxes" indicates the source of text and localization boxes. "Type scope" indicates whether the system uses built-in labels mapped to LeakageBench, a restricted PII label set, or the full benchmark schema.

```
If the page contains no PII of the allowed types, follow the NO-PII output rule in OUTPUT FORMAT.

RULES
- Output ONLY what is required by OUTPUT FORMAT (no prose, no Markdown).
- Do NOT invent text for "value". If unsure, use "".
"""
```

C.2 Qwen prompt template (JSON output)

Listing 2: Prompt template for Qwen-style models with strict JSON output.

```
PROMPT_TEMPLATE_QWEN = PROMPT_HEADER + r"""
OUTPUT FORMAT (STRICT)
Return exactly one JSON object:
{
  "predictions": [
    {"bbox_2d":[x1,y1,x2,y2], "type":"<one allowed type>", "score":0.0, "value":""}
  ]
}

COORDINATES
- bbox_2d MUST be normalized to [0,1000] with origin at top-left.
- Each box is [x1,y1,x2,y2] with x1<x2 and y1<y2.

NO-PII OUTPUT
{"predictions": []}
"""
```

C.3 InternVL prompt template (tagged blocks)

Listing 3: Prompt template for InternVL-style models with tagged block outputs.

```
PROMPT_TEMPLATE_INTERNVL = PROMPT_HEADER + r"""
EXAMPLE (one block):
<ref>Date</ref><box>[[120,340,210,360]]</box><value>01/28/20</value><score>0.95</score>

OUTPUT FORMAT (STRICT)
Return ONLY plain text consisting of zero or more repeated blocks:
<ref>TYPE</ref><box>[[x1,y1,x2,y2], ...]</box><value>...</value><score>...</score>


COORDINATES
- Boxes MUST be normalized to [0,1000] with origin at top-left.
- Each box is [x1,y1,x2,y2] with x1<x2 and y1<y2.

NO-PII OUTPUT
Output an EMPTY string (no characters).
"""
```

D Document-Type Composition

Table 7 reports the page-type distribution for each source. The benchmark is dominated by invoices, emails, forms, correspondence, and file-like pages, reflecting the operational-document focus of LeakageBench.

E Schema Coverage

Table 8 reports entity-type coverage under each baseline's schema mapping or prompting interface. Coverage indicates which LeakageBench types the evaluated configuration was allowed to emit; it does not measure intrinsic model capability or detection accuracy.

Source	Invoice	Forms	Email(s)	Letter/Corr.	File	Other
Source A (OCR-IDL)	3	2	22	46	30	10
Source B (VRDU Ad-Buy)	219	2	0	0	1	0
Source C (FCC free-form)	7	35	68	16	16	23
Total	229	39	90	62	47	33

Table 7: Page-type composition by source. "Email(s)" aggregates Email and Emails; "Letter/Corr." aggregates Letter and Correspondence.

System	All Types↑	Direct↑	Linkage↑	Contextual↑
OCR-dependent pipelines				
Presidio	0.208	1.000	0.000	0.333
Presidio-best	0.208	1.000	0.000	0.333
GLiNER-base	1.000	1.000	1.000	1.000
GLiNER-multi-PII	1.000	1.000	1.000	1.000
NVIDIA GLiNER PII	1.000	1.000	1.000	1.000
GLiNER2	0.958	1.000	1.000	1.000
Google DLP	0.250	1.000	0.000	0.667
Amazon Comprehend PII	0.208	1.000	0.000	0.333
OpenAI Privacy Filter	0.250	1.000	0.077	0.333
OCR-free VLMs				
Qwen3-VL-32B	1.000	1.000	1.000	1.000
InternVL3-38B	1.000	1.000	1.000	1.000
GPT-5.4	1.000	1.000	1.000	1.000
GPT-5.5	1.000	1.000	1.000	1.000
GPT-5.5 + Code Interpreter	1.000	1.000	1.000	1.000

Table 8: Schema coverage of each evaluated configuration after fixed schema mapping or prompting. Coverage reflects the prediction interface, not intrinsic model capability or detection accuracy.

F Group-Level Leakage

Table 9 reports conditional DocLeak for each privacy-aligned group, computed over pages containing at least one ground-truth identifier from that group.

G IoU Threshold Sweep

Table 10 reports a threshold sweep over the IoU matching criterion $\tau \in \{0.50, 0.75, 0.95\}$. Lowering τ relaxes box tightness requirements, while higher τ requires tighter spatial alignment. Although entity-level F1 changes substantially with τ, document-level leakage remains high across thresholds.

H Corpus and Entity Statistics

Table 11 reports corpus-level entity-category statistics by source, and Table 12 reports fine-grained entity-type counts.

I Additional Error Diagnostics

Table 13 reports per-system leakage severity using the headline type-aware matcher. Table 14 attributes OCR-dependent leakage using type-agnostic matching, while Table 15 measures detector-independent OCR recoverability using only gold values and stored OCR text.

OCR engine	Unrecoverable		Partial n
	n	% ↓	
AWS Textract	623	**14.6**	262
Tesseract	765	17.9	350

Table 15: OCR recoverability over 4,280 value-bearing critical annotations. Of 4,315 critical annotations, 35 lack a verbatim gold value and are excluded. Partial denotes unrecoverable values with OCR-window similarity ≥ 0.8.

System	Direct↓	Linkage↓	Contextual↓
OCR-dependent pipelines			
Presidio	0.973	1.000	0.982
Presidio-best	0.967	1.000	0.977
GLiNER-base	0.954	0.967	0.924
GLiNER-multi-PII	0.950	0.973	0.908
NVIDIA GLiNER PII	0.942	0.967	0.916
GLiNER2	0.971	0.985	0.963
Google DLP	1.000	1.000	0.990
Amazon Comprehend PII	0.908	1.000	0.971
OpenAI Privacy Filter	0.971	1.000	0.994
OCR-free vision-language models			
Qwen3-VL-32B	0.990	0.982	0.975
InternVL3-38B	1.000	1.000	0.998
GPT-5.4	0.998	0.988	0.990
GPT-5.5	0.956	0.994	0.986
GPT-5.5 + Code Interpreter	0.887	0.964	0.908

Table 9: Conditional document leakage by privacy group at IoU $\tau = 0.75$. Each column reports DocLeak over pages containing at least one ground-truth identifier from that group. All systems use full-schema scoring; unsupported types and invalid outputs count as misses.

System	IoU	$F1_{loc}$	$F1_{type}$	$DocLeak_{all}$ ↓	$DocLeak_{crit}$ ↓
Presidio	0.50	0.448	0.109	0.992	0.964
	0.75	0.137	0.048	0.994	0.990
	0.95	0.001	0.001	1.000	1.000
Presidio-best	0.50	0.529	0.110	0.992	0.968
	0.75	0.248	0.064	0.996	0.986
	0.95	0.020	0.004	1.000	1.000
GLiNER-base	0.50	0.403	0.136	0.988	0.949
	0.75	0.216	0.063	0.988	0.978
	0.95	0.031	0.004	1.000	1.000
GLiNER-multi-PII	0.50	0.432	0.184	0.974	0.939
	0.75	0.245	0.091	0.990	0.972
	0.95	0.030	0.005	1.000	1.000
NVIDIA GLiNER PII	0.50	0.567	0.251	0.992	0.972
	0.75	0.288	0.109	0.992	0.986
	0.95	0.026	0.004	1.000	1.000
GLiNER2	0.50	0.311	0.139	0.994	0.976
	0.75	0.156	0.065	1.000	0.994
	0.95	0.018	0.004	1.000	1.000
Google DLP	0.50	0.286	0.077	0.994	0.976
	0.75	0.076	0.013	1.000	1.000
	0.95	0.000	0.000	1.000	1.000
Amazon Comprehend PII	0.50	0.589	0.136	0.976	0.901
	0.75	0.304	0.085	0.992	0.968
	0.95	0.033	0.006	1.000	0.998
OpenAI Privacy Filter	0.50	0.272	0.086	0.990	0.964
	0.75	0.164	0.055	0.996	0.990
	0.95	0.024	0.005	1.000	1.000
Qwen3-VL-32B	0.50	0.224	0.134	0.992	0.978
	0.75	0.073	0.046	1.000	0.998
	0.95	0.001	0.000	1.000	1.000
InternVL3-38B	0.50	0.090	0.048	1.000	0.996
	0.75	0.012	0.019	1.000	1.000
	0.95	0.000	0.000	1.000	1.000
GPT-5.4	0.50	0.242	0.183	0.988	0.953
	0.75	0.044	0.031	1.000	1.000
	0.95	0.000	0.000	1.000	1.000
GPT-5.5	0.50	0.369	0.248	0.958	0.907
	0.75	0.090	0.050	1.000	0.990
	0.95	0.001	0.000	1.000	1.000
GPT-5.5 + Code Interpreter	0.50	0.614	0.380	0.904	0.775
	0.75	0.249	0.119	0.990	0.968
	0.95	0.009	0.003	1.000	1.000

Table 10: IoU threshold sweep under full-schema scoring. Lower IoU thresholds relax box alignment, while higher thresholds require tighter localization. Shaded metric cells correspond to the main-paper threshold $\tau = 0.75$.

	Corpus		Mentions		By Category				
Src	**Docs**	**Pgs**	**All**	**/Pg**	**DI**	**CI**	**LK**	**Sig**	**OR**
A	103	113	1714	15.2	1143	445	109	17	0
B	107	222	7331	33.0	673	6245	221	5	187
C	81	165	2909	17.6	1963	710	206	30	0
All	291	500	11954	23.9	3779	7400	536	52	187

Table 11: Corpus and entity-category statistics by source. DI = Direct Identifiers, CI = Contextual Identifiers, LK = Linkage Keys, Sig = Signature, OR = Operational References (Agency Code + Agency Reference + Advertiser Reference).

Group	**Entity Type**	**Src A**	**Src B**	**Src C**	**All**
Direct Identifiers (Subtotal)		1,143	673	1,963	3,779
	Person Name	804	238	895	1,937
	Address	156	226	318	700
	Phone Number	117	205	165	487
	Email Address	66	4	585	655
Contextual Identifiers (Subtotal)		445	6,245	710	7,400
	Date	207	5,784	499	6,490
	Organization Name	238	461	207	906
	Business ID	0	0	4	4
Linkage Keys (Subtotal)		109	221	206	536
	Linkage ID	20	110	109	239
	Contract Number	68	23	7	98
	Invoice Number	13	88	7	108
	ID Card Number	8	0	0	8
	Registration Number	0	0	32	32
	Reference Number	0	0	19	19
	File Number	0	0	9	9
	Cheque Number	0	0	8	8
	DA Case Number	0	0	5	5
	Call Sign	0	0	4	4
	Customer ID	0	0	4	4
	Employee ID	0	0	1	1
	Policy Number	0	0	1	1
Operational References (Subtotal)		0	187	0	187
	Agency Code	0	45	0	45
	Agency Reference	0	75	0	75
	Advertiser Reference	0	67	0	67
Signature (Subtotal)		17	5	30	52
	Signature	17	5	30	52
Total	**All Entities**	1,714	7,331	2,909	11,954

Table 12: Entity-type statistics by source. Casing is standardized to Title Case; acronyms (e.g., ID, DA) remain uppercase.

System	$\text{DocLeak}_{\text{crit}}$ ↓	Mean missed↓	Median↓	P90↓	Mean proportion missed↓
OCR-dependent pipelines					
Presidio	0.990	7.002	5	15	0.839
Presidio-best	0.986	6.639	5	14	0.805
GLiNER-base	0.978	6.768	4	14	0.785
GLiNER-multi-PII	0.972	7.013	4	14	0.772
NVIDIA GLiNER PII	0.986	6.680	4	14	0.767
GLiNER2	0.994	6.835	5	15	0.827
Google DLP	1.000	8.425	6	17.7	0.978
Amazon Comprehend PII	**0.968**	**5.308**	**4**	**11**	0.661
OpenAI Privacy Filter	0.990	6.196	5	12	0.813
OCR-free vision-language models					
Qwen3-VL-32B	0.998	8.183	5	18	0.935
InternVL3-38B	1.000	8.652	6	18	0.990
GPT-5.4	1.000	8.245	5	18	0.939
GPT-5.5	0.990	7.810	5	16	0.860
GPT-5.5 + Code Interpreter	**0.968**	**5.538**	**4**	**12**	**0.656**

Table 13: Severity of missed Direct and Linkage identifiers at IoU $\tau = 0.75$. DocLeak and missed counts use the headline type-aware matcher. Mean, median, and P90 are computed over leaking pages; mean proportion is computed over all pages containing critical identifiers. Bold marks the best value within each system family.

System	Attrib. n	Share of attributable critical leaks (%)			
		OCR miss	Detector miss	Box/alignment	Assignment
Tesseract OCR					
Presidio	3,371	21.4	21.8	56.8	0.0
Shared AWS Textract OCR *(OCR fixed; detector varies)*					
Presidio-best	3,182	16.9	37.2	45.9	0.1
GLiNER-base	3,052	18.9	29.0	52.1	0.0
GLiNER-multi-PII	3,204	16.3	48.8	34.9	0.1
NVIDIA GLiNER PII	3,127	18.2	22.7	59.0	0.0
GLiNER2	3,244	17.3	34.8	47.9	0.0
Amazon Comprehend PII	2,499	20.3	23.0	56.5	0.2
OpenAI Privacy Filter	2,984	18.9	44.4	36.6	0.1
Vendor-managed OCR *(OCR text unavailable)*					
Google DLP	4,127	–	15.5	84.5	0.0

System	Leaked critical	Value unavailable	Attrib. n	OCR partial
Presidio	3,397	26	3,371	312
Presidio-best	3,208	26	3,182	192
GLiNER-base	3,077	25	3,052	219
GLiNER-multi-PII	3,231	27	3,204	176
NVIDIA GLiNER PII	3,152	25	3,127	215
GLiNER2	3,271	27	3,244	205
Google DLP	4,160	33	4,127	–
Amazon Comprehend PII	2,518	19	2,499	162
OpenAI Privacy Filter	3,008	24	2,984	209

Table 14: Attribution of leaked Direct and Linkage identifiers for OCR-dependent systems using type-agnostic matching at IoU $\tau = 0.75$. The upper panel reports exclusive failure shares; the lower panel gives their denominators and secondary diagnostic counts. Value-unavailable instances remain in the leakage total but are excluded from attribution percentages. OCR partial is a secondary flag on OCR-miss instances. Dashes indicate unavailable vendor OCR.