

ESG-CID-Plus: A Stratified Benchmark for Disclosure Content Index Table Extraction from Corporate Sustainability Reports

Shafiuddin Rehan Ahmed, Reza Abdi, Tianyu Luan
Teja Kanchinadam, Christopher Clarke, Eugene Siow
Agentic AI Center of Excellence, Accenture, Mountain View, CA, USA
{shafiuddin.r.ahmed,c.a.clarke,eugene.siow}@accenture.com

Abstract

Corporate sustainability reports include a *content index* mapping each disclosure required by a reporting framework (e.g. GRI) to its supporting pages — a resource that potentially benefits nearly every downstream sustainability-NLP task, yet no existing benchmark scores its recovery as a framework-grounded, page-resolved output. We formalize Content Index Detection (CID) as three evaluable stages — index page retrieval (S1), table extraction and disclosure linking (S2), and page-label-to-index resolution (S3) — and release `esg_cid_plus`: 37 GRI reports and 7,294 rows stratified across three difficulty tiers, from unchanged real reports to an adversarial tier of naturally hard documents and synthetic perturbations. Our main benchmark sweeps open-weight vision-language models from sub-1B to 35B parameters (plus frontier-API references) as S2 extractors under shared, deterministic S1 and S3 stages: the best backends reach S2-F1 of 0.89–0.92 on the easier tiers, while standard OCR-benchmark scores explain only a third of the variance in extraction quality. Composing extraction with the deterministic resolver exposes S3 as the open frontier: under label-scheme resolution alone, where even best extraction caps end-to-end F1 at 0.11 on the hardest tier. `esg_cid_plus` will be released at https://huggingface.co/datasets/ahmeshaf/esg_cid_plus.

1 Introduction

Corporate sustainability reports (CSRs) are the primary public record of how organisations disclose their Environmental, Social, and Governance (ESG) practices. They are long, narrative-heavy PDFs whose structure is dictated by reporting frameworks such as the Global Reporting Initiative (GRI), the European Sustainability Reporting Standards (ESRS), the Sustainability Accounting Standards Board (SASB), or the recommendations of the Task Force on Climate-Related Financial Dis-

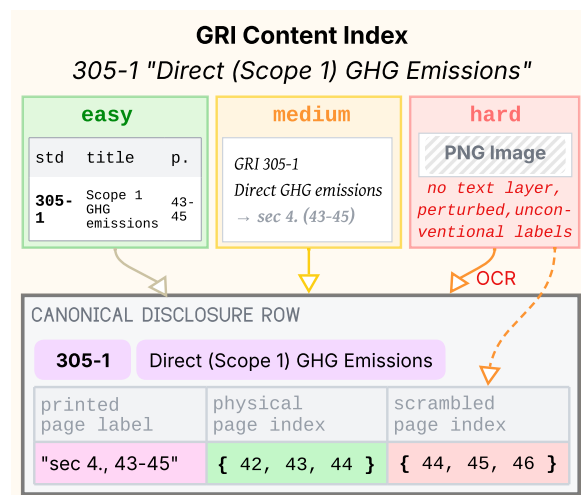


Figure 1: One canonical disclosure (GRI 305-1, “Direct (Scope 1) GHG Emissions”) rendered three different ways across the three difficulty tiers (`gri_easy`, `gri_medium`, `gri_hard`) must collapse to the same canonical row $r_i = (s_i, t_i, \tilde{t}_i, \ell_i, P_i)$. The footer separates the printed page label ℓ_i from the resolved 0-based PDF indices P_i ; the dashed arrow on `gri_hard` marks where text-only extraction fails and OCR is required.

closures (TCFD).¹ Most CSRs also include, often as an appendix, a *content index*: a table that maps each required disclosure (e.g. GRI 302-3, “Energy intensity”) to its corresponding pages.

This index, the report’s own disclosure-to-evidence map, is a resource that potentially benefits nearly every downstream sustainability-NLP application: retrieval-augmented generation (Ahmed et al., 2025; Ni et al., 2023), coverage assessment (Usmanova et al., 2025; Polignano et al., 2022), claim verification and grounded QA over CSRs (Vaghefi et al., 2023; Diggelmann et al., 2020), and longitudinal comparison of disclosure quality (Schimanski et al., 2024b; Ong et al., 2025). Yet the literature invests heavily in downstream disclo-

¹See <https://www.globalreporting.org/> (GRI); <https://www.efrag.org/> (ESRS); <https://sasb.org/> (SASB); and <https://www.fsb-tcfd.org/> (TCFD).

sure analysis while assuming the content index can already be recovered reliably.

We study *Content Index Detection (CID)*: given a CSR PDF $\mathcal{D}$ and a target framework $\mathcal{F}$, identify the pages $P_{\mathcal{F}}$ containing $\mathcal{F}$ ’s content index and produce a schema-aligned table $\text{CI}_{\mathcal{F}}(\mathcal{D})$ with one row per framework disclosure $(s_i, t_i, \tilde{t}_i, \ell_i, P_i)$ — the framework id and title, the title as it appears in the report, the verbatim page-reference cell text, and the resolved 0-based PDF page indices (Figure 1). We decompose CID into three stages: CI page retrieval (S1), CI table extraction (S2, comprising raw row extraction S2.1 and disclosure linking S2.2), and page reference resolution (S3) (Section 3), enabling systems to be evaluated by failure stage rather than only end-to-end.

Framed this way, CID is a *composition* of five fundamental document extraction tasks, each with its own literature and failure modes: (i) **Search** — retrieving the few index pages within a long report (S1); (ii) **Parse** — extracting table rows, including OCR and multi-page stitching (S2.1); (iii) **Bind** — disambiguating repeated *management-of-material-topics* entries (e.g., “3–3”) by binding each to its spanning section header; (iv) **Fill** — mapping the parsed, disambiguated rows into the framework template, one canonical slot per disclosure (S2.2); and (v) **Resolve** — converting human-facing page references to physical 0-based PDF indices (S3). Prior work studies these tasks in isolation or, in ESG-NLP, uses CID only to bootstrap retrieval supervision (Ahmed et al., 2025); no benchmark evaluates them jointly on a page-resolved, framework-grounded output, and *Bind* and *Resolve* remain essentially unaddressed.

Two of these tasks make CID markedly harder than generic table extraction (Mathew et al., 2021; Saad-Falcon et al., 2023; Schimanski et al., 2026). The index table is idiosyncratic across CSRs — variable column layouts, multi-page two-column spreads, and disclosure ids in many typographic variants (“GRI 302-3”, “302–3”, “GRI 302:3”) — and its page references are written for human readers as *printed page labels* (“iv”, “A-12”, “2.7”) that need not match 0-based PDF indices: numbering schemes restart or switch styles, the same label can recur across physical pages, and a reference may even be a section name or an external citation (“Annual Report 2023, p. 12”), so resolution must model the report’s own label scheme — or correctly mark a reference unresolvable. These failure modes are also not uniform: heuristics that work on

one report family break on another, so a benchmark for CID must stratify reports by difficulty — from unchanged real reports, through naturally adversarial ones, to synthetically perturbed documents — to expose where a single recipe fails and why.

Concretely, we make four contributions.

Task formalisation. We formalise CID as a framework-grounded, page-resolved problem over evaluable stages S1–S3 (Section 3).

Stratified benchmark. We release the dataset `esg_cid_plus`: 37 GRI reports and 7,294 CID rows across three tiers — `gri_easy`, `gri_medium`, and an adversarial `gri_hard` (§4).

Baselines. We implement a deterministic baseline for every stage — lexical S1 retrieval with OCR fallback, `Img2Table+RapidOCR` S2 extraction and linking, and a label-scheme S3 resolver — as fixed, swappable components (§5).

An S2 model sweep. We sweep open-weight vision-language models from sub-1B to 35B parameters (plus frontier-API references) as S2.1 extractors under oracle S1, so that extraction quality is isolated (§6): the best backends reach S2-F1 of 0.89–0.92 on the easier tiers but at most 0.79 on `gri_hard`, and standard OCR-benchmark scores explain only a third of the variance in extraction quality (§7). An ablation separating standard from topic-qualified identifiers isolates the cost of *Bind* as standing headroom.

Our results indicate that page resolution (S3) is the open frontier: under deterministic resolution, even gold-quality extraction caps end-to-end F1 at 0.11 on `gri_hard` (§6) — the remaining headroom sits in S3. `esg_cid_plus` thus lays the foundation for future work that models *Resolve* directly rather than treating page resolution as a deterministic post-process.

2 Related Work

We position `esg_cid_plus` against three bodies of work — table structure recognition, long-document VQA and key-information extraction, and ESG-NLP — and map each onto the five tasks CID composes (§1), as summarised in Table 1.

Table structure recognition. `FinTabNet` (Zheng et al., 2021) and `PubTables-1M` (Smock et al., 2022b) provide large-scale training signal for *Parse*, the latter with explicit spanning-cell classes,

Dataset / Benchmark	Document Extraction Tasks (§1)				
	Pg. Retr.	Table Understanding			Pg. X-Ref.
	Search	Parse	Bind	Fill	Resolve
<i>Table structure recognition</i>					
FinTabNet	✗	✓	▲	✗	✗
PubTables-1M	✗	✓	▲	✗	✗
HiTab	✗	▲	✓	▲	✗
TUTA	✗	▲	✓	▲	✗
<i>Long-document VQA and KIE</i>					
DocVQA	▲	▲	✗	✗	✗
MP-DocVQA	✓	▲	✗	✗	✗
Kleister	▲	✗	✗	✓	✗
DocILE	✗	✓	✗	✓	✗
<i>ESG / sustainability benchmarks</i>					
ESGDoc	✓	▲	✗	✗	▲
GRI Pipeline	▲	▲	✗	▲	✗
ReportGRI	▲	✗	✗	✓	✗
GRI-QA	▲	✓	✗	✓	✗
esg_cid	▲	✓	✗	▲	✗
ours	✓	✓	✓	✓	✓

Table 1: Coverage of the five document extraction tasks CID composes (§1) across the three bodies of prior work surveyed below — table structure recognition, long-document VQA and key-information extraction, and ESG / sustainability benchmarks. ✓ explicitly evaluated with released gold; ▲ present in the data or method but not scored; ✗ absent. Citations are given in the text.

and GriTS (Smock et al., 2022a) scores recovered topology, content, and location; Nunes et al. (2025), Soric et al. (2025), and Horn and Keuper (2026) benchmark multimodal LLMs against OCR pipelines on PDF tables, and OCRBench (Liu et al., 2024) and OCRBench v2 (Fu et al., 2025) rank VLMs on OCR-centric recognition, localisation, and parsing — they guided our open-weight roster selection but score isolated capabilities. Closest to *Bind*, HiTab (Cheng et al., 2022) and TUTA (Wang et al., 2021) formalise linking a data cell to its governing multi-level headers, but both assume the table is already parsed, bind cells to *in-table* headers rather than an external framework schema, and never recover spans from a rendered PDF. GRI content indices additionally span multiple physical pages as one logical table, so per-fragment structural metrics are uninterpretable; we capture location-text quality within S2.2 via fuzzy matching instead.

Long-document VQA and key-information extraction. DocVQA (Mathew et al., 2021) and PDFTriage (Saad-Falcon et al., 2023) show that modelling document structure beats flat-text re-

trieval, motivating our staged decomposition, and MP-DocVQA (Tito et al., 2023) makes evidence-page identification over multi-page documents explicit (*Search*) — but the located page is evidence for an answer, not a resolved output key. On the extraction side, Kleister (Stanisławek et al., 2021) extracts values for a fixed key schema from long PDFs and DocILE (Šimsa et al., 2023) localises line items on invoices (*Fill*); both fill flat key schemas, without hierarchical-header binding or per-row page resolution.

ESG and sustainability NLP. The field has grown quickly — domain-adapted language models (Webersinke et al., 2022; Schimanski et al., 2024b), retrieval-augmented report QA (Ni et al., 2023; Vaghefi et al., 2023), and climate-claim verification (Diggelmann et al., 2020; Schimanski et al., 2024a; Mankour et al., 2025) — but none targets framework alignment or is benchmarked page-resolved, though CID’s schema would readily sharpen their claims. On framework alignment specifically: Usmanova et al. (2025) automate GRI alignment via zero-shot prompting (five reports, no tables, no page grounding); Polignano et al. (2022) apply an NLP pipeline to GRI indexes across 134 reports but release no gold; GRI-QA (Contalbo et al., 2025) benchmarks table QA over disclosed values once located — the natural downstream complement to CID. ESGDoc (Wang et al., 2023) is the closest structural predecessor, extracting a report’s own chapter hierarchy rather than the compliance-framework table, with no label-to-index pagination — leaving *Resolve* without a dedicated benchmark anywhere.

Positioning. *esg_cid_plus* is the only benchmark in Table 1 scoring all five tasks jointly, so an end-to-end failure can be attributed to a stage: pages missed (*Search*), rows lost (*Parse*), topics unbound (*Bind*), slots mis-filled (*Fill*), or references unresolved (*Resolve*). The closest prior work, *esg_cid* (Ahmed et al., 2025), mines content indices to bootstrap retrieval supervision and supplies the PDFs for the bulk of our adapted subset, but collapses duplicate topic-qualified rows into a flat namespace and releases no page-resolved gold, so neither *Bind* nor *Resolve* can be measured on it.

3 Content Index Extraction Task (CID)

Before turning to dataset construction (§4), we formalize CID.

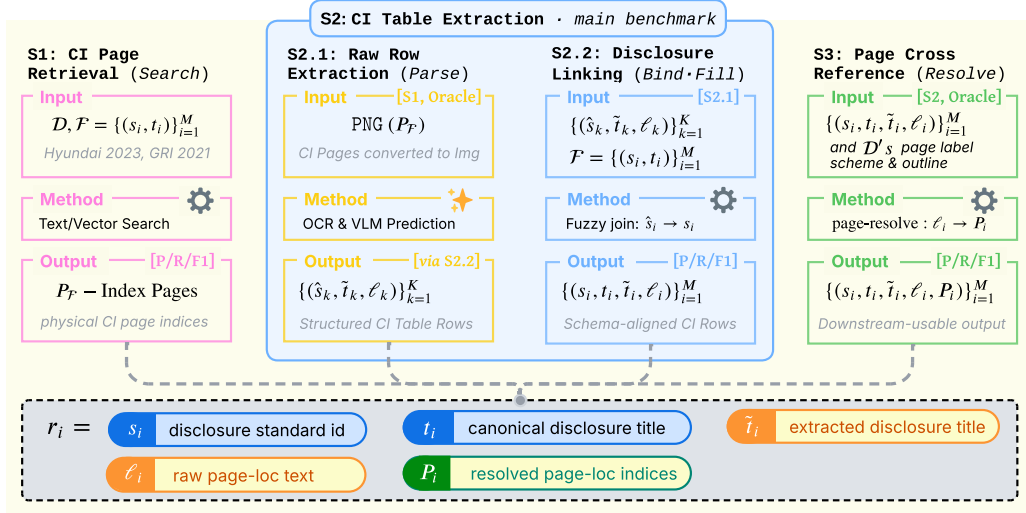


Figure 2: The three stages of CID extraction. Given a report $\mathcal{D}$ and a framework $\mathcal{F}$: (S1) retrieves index pages $P_{\mathcal{F}}$ (Search); (S2) extracts raw candidate rows (S2.1, *Parse*) and fuzzy-joins them to the canonical schema (S2.2, *Bind · Fill*); (S3) resolves verbatim page-location text ℓ_i to physical page indices P_i via $\mathcal{D}$'s printed-label scheme (*Resolve*). Icons mark each stage as model-inferred (✨) or deterministic (⚙️). S2.1 is the only inferred stage and the target of our main benchmark.

3.1 Task Formalization

Given a PDF report $\mathcal{D}$ (N pages) and a disclosure framework $\mathcal{F} = \{(s_i, t_i)\}_{i=1}^M$ — a canonical list of standard ids (e.g. "2-1") paired with titles (e.g. "Organizational details"), such as GRI 2021 ($M=147$) — CID requires producing two outputs. $P_{\mathcal{F}} \subseteq \{0, \dots, N-1\}$ is the set of PDF pages on which $\mathcal{F}$'s content index appears. $\text{CI}_{\mathcal{F}}(\mathcal{D})$ is the extracted, schema-aligned content index:

$$\text{CI}_{\mathcal{F}}(\mathcal{D}) = \{r_i : i = 1, \dots, M\},$$

with exactly one row per disclosure:

$$r_i = (s_i, t_i, \tilde{t}_i, \ell_i, P_i).$$

$\tilde{t}_i$ is the title as printed in the report (may differ from canonical t_i due to abbreviation, translation, or OCR noise); ℓ_i is the verbatim location-reference cell (numeric labels, e.g. "62, 78"; section names, e.g. "Governance chapter"; external citations, e.g. "Annual Report 2023, p. 12"; or inline notes, e.g. "Not applicable"); $P_i \subseteq \{0, \dots, N-1\}$ is the resolved set of physical page indices. When a report provides no evidence for s_i within $\mathcal{D}$, $P_i = \emptyset$. $\text{CI}_{\mathcal{F}}(\mathcal{D})$ is *M-complete*: every $s_i \in \mathcal{F}$ has a row in $\text{CI}_{\mathcal{F}}(\mathcal{D})$.

3.2 Subtasks

The task decomposes into three stages illustrated in Figure 2, with stage S2 further divided into two sub-stages.

S1: CI Page Retrieval (Search). Given $(\mathcal{D}, \mathcal{F})$, produce $P_{\mathcal{F}}$ — the set of pages on which the content index appears. This is a focused retrieval problem: the index shares surface vocabulary with tables of contents, assurance statements, and appendix tables.

S2: CI Table Extraction. Given $(\mathcal{D}, \mathcal{F}, P_{\mathcal{F}})$, produce the M -complete linked table in two sub-stages:

S2.1: Raw Row Extraction (Parse). Extract raw candidate rows $\{(\hat{s}_k, \hat{t}_k, \ell_k)\}_{k=1}^K$ from the index region, handling OCR, multi-page stitching, and multi-column layouts.

S2.2: Disclosure Linking (Bind · Fill). Fuzzy-join candidates against $\mathcal{F}$ on $\hat{s}_k \rightarrow s_i$: candidates with no canonical match are dropped, duplicates are merged, and schema entries with no candidate become ε rows, yielding the M -complete table $\{r'_i\} = \{(s_i, t_i, \tilde{t}_i, \ell_i)\}$.

S3: Page Cross Reference (Resolve). Given each r'_i and $\mathcal{D}$, resolve $\ell_i \rightarrow P_i$ by recovering the report's label-to-index function λ and resolving each token by type: numeric labels via λ^{-1} , section references via GoTo links or heading lookup, external citations to $P_i = \emptyset$.

We now turn to how this formalism is annotated, stratified, and evaluated in practice.

4 Dataset Construction

`esg_cid_plus` is built from the 73 GRI PDFs in `airefinery/esg_cid` (57 train, 6 dev, 10 test_gri), covering GRI Standards 2016 ($M=238$) and GRI Standards 2021 ($M=147$); the 11 ESRs documents are set aside as a separate future task. We select 26 of the 73 reports for three real-world splits, stratified by page-label scheme complexity and index layout, and annotate ten additional reports for an adversarial hard tier. The full per-report listing is in Appendix A.

Annotation protocol. The annotator starts from the framework template $\mathcal{F}$ and, per PDF: (i) locates the content index pages, recorded as the S1 ground truth; (ii) for each $s_i \in \mathcal{F}$, copies the location cell text ℓ_i verbatim from the matching content-index row (left empty if none exists); (iii) resolves each populated ℓ_i to physical page indices P_i , following section-name pointers where needed.

Row taxonomy. Every row of $CI_{\mathcal{F}}(\mathcal{D})$ is first *absent* ($\ell_i=\varepsilon$, count M^{abs} , inserted by step (ii) to enforce M -completeness) or *mentioned* ($\ell_i \neq \varepsilon$, count M^{men} ; gold target for S2), so $M = M^{\text{abs}} + M^{\text{men}}$. Mentioned rows further split into *page-indexed* ($P_i \neq \emptyset$, count M^{pg} ; gold target for S3 — numeric spans and resolvable section pointers) and *non-page-indexed* ($P_i = \emptyset$, count M^{pg} ; omission markers, external references, unresolvable section pointers), so $M^{\text{men}} = M^{\text{pg}} + M^{\text{pg}}$. Per-split totals are unpacked in Section A.3.

Standard vs. topic-qualified disclosures. Most disclosures are *standard* (e.g. 2-1): one row, one identity. Material-topic disclosures (103-1/2/3 under GRI 2016, 3-3 under GRI 2021) instead recur once per material topic under an identical ID and title — the same binding problem as a data cell under a multi-level header (§2), except the governing header sits in the surrounding section, not the table. GRI 2016 crosses 96 such disclosures against up to 32 topics; GRI 2021 has 32. We evaluate baselines on the standard disclosures and analyze topic-qualified rows separately.

Difficulty tiers. Tiers are defined by construction-side signals observable before any model is applied. We classify each PDF’s label-to-index function λ as *trivial* (declared and machine-readable), *nontrivial* (must be inferred empirically, e.g. from sampled headers), or *complex* (requires parsing document structure).

gri_easy (10 PDFs): trivial scheme, contiguous well-labelled index. **gri_medium** (10 PDFs): a nontrivial scheme (6/10), sidebar/multi-column layout, or non-standard headers. **gri_hard** (10 PDFs): six real-world adversarial reports (section-based location text, double-page-spread labeling, or a complex hierarchical scheme) plus four synthetic $_x$ perturbations (image-only rendering or scrambled /PageLabels) applied to easy/medium source reports; Table 4 (Appendix A) lists all ten.

5 Experimental Setup

Of the four stages formalized in §3.2 (Figure 2), only S2.1 is model-inferred (✱); S1, S2.2, and S3 are deterministic (⚙) and shared by all backends, so end-to-end differences reflect the S2.1 backend alone.

5.1 Baselines

S1: CI Page Retrieval (Search). We index each page with `lunr` (Diaz, 2018), a lightweight TF-IDF lexical search engine, and query it using batched disclosure ids and titles from $\mathcal{F}$. We take the top-10 retrieved pages as seeds and expand each by a ± 5 -page neighborhood. We then confirm candidate pages using *disclosure-id density*: we normalize framework-id regex matches by token count and accept a page if its density exceeds 0.03 or, for sparse pages, if it contains at least seven matches regardless of density. We tuned the five retrieval hyperparameters (seed count, neighborhood radius, density threshold, match-count floor, and the OCR fallback below) once on `train_small` (S1 P/R/F1 = .95/.98/.96) and kept them fixed across all evaluation splits. Finally, we close confirmed pages under contiguity by filling gaps between adjacent detections. When a page has an empty or sparse text layer, we transparently substitute **RapidOCR** (RapidAI, 2021) output before indexing and density scoring, allowing the same retrieval pipeline to handle rasterized reports without a separate code path.

S2: CI Table Extraction (Parse · Bind · Fill). We render confirmed CI pages as images and parse them with **Img2Table** (Canton, 2021)+**RapidOCR** (S2.1). We scan each extracted row for a disclosure id: an id match opens a new candidate row, while subsequent rows without an id extend the current candidate by appending their title and location-text cells until the next id match. This recovers disclosures whose title or location text spans multiple

Model	S2-F1	Model	S2-F1	Model	S2-F1
GPT-5.5	0.85	Qwen3.5-35B-A3B	0.65	Ministral-3-3B	0.48
GPT-5.4	0.80	Nemotron-Nano-v2-VL	0.65	Gemma-4-12B-it	0.45
GPT-5-mini	0.80	Ministral-3-8B	0.64	Nemotron-3-Nano-Omni	0.45
Qwen3.5-27B	0.79	Ovis2.5-9B	0.63	LLaVA-OneVision-2	0.42
Gemini-3.5-Flash	0.79	Claude Haiku 4.5	0.61	UI-TARS-1.5-7B	0.34
Gemini-3.1-Flash-Lite	0.78	GPT-5-nano	0.60	InternVL3.5-2B	0.31
Gemini-3-Flash	0.77	Qwen3.5-9B	0.59	Gemma-4-E4B-it	0.31
Qwen3.6-27B	0.77	MiniCPM-V-4.5	0.58	Qwen3.5-2B	0.28
Ovis2.6-30B-A3B	0.74	InternVL3.5-4B	0.56	Granite-Vision-4.1-4B	0.28
Qwen3.6-35B-A3B	0.74	InternVL3.5-8B	0.56	InternVL3.5-1B	0.27
Gemma-4-31B-it	0.71	Qwen3.5-4B	0.56	MiniCPM-V-4.6	0.23
Claude Sonnet 4.5	0.71	Baseline (Img2Table)	0.52	Ovis2.5-2B	0.21
GLM-4-6v-flash	0.69	Qianfan-OCR	0.52	Gemma-4-E2B-it	0.19
Gemma-4-26B-A4B-it	0.67	Step3-VL-10B	0.51	Qwen3.5-0-8B	0.16
olmOCR-2-7B-1025	0.66	InternVL3.5-14B	0.50	Ovis2-1B	0.13
Ministral-3-14B	0.66	InternVL3.5-30B-A3B	0.50	Granite-Vision-3.2-2B	0.06

Table 2: Mean S2-F1 across `gri_easy`, `gri_med`, and `gri_hard` (oracle S1). Closed-source models are indicated in blue; the Baseline (§5.1) is indicated in red.

table rows. We then normalize candidate ids (removing whitespace, dash variants, and framework prefixes) and join them to $\mathcal{F}$ by exact id match (S2.2). We discard candidates with no schema match and emit ε rows for schema disclosures with no matching candidate. Every S2.1 backend shares this normalize-then-join stage, so it absorbs backend-specific formatting noise identically before scoring.

S3: Page Cross Reference (*Resolve*). We read the PDF’s /PageLabels catalog (numbering style, prefix, and start page) and build a label→physical-index map, defaulting to implicit arabic-from-zero numbering when the catalog is absent. We then mine each row’s location text for numeric ranges, bare numbers, and roman or alphabetic labels after scrubbing disclosure ids (so, e.g., “2-1” is not misread as a page range). Finally, we resolve each mined label through the map to produce P_i , dropping labels that do not appear in the map. This explains why non-page-indexed mentioned rows (§4) exist even when ℓ_i is populated.

5.2 Metrics

We report three canonical metrics, one per stage boundary. **S1-P/R/F1** is macro P/R/F1 over per-report page tuples $(\mathcal{D}, p)$, $p \in P_{\mathcal{F}}$. **S2-P/R/F1** is scored at the end of S2.2: micro P/R/F1 over *mentioned* tuples $(\mathcal{D}, s_i, \ell_i)$ (§4), where a true positive requires both the correct disclosure identifier s_i and the extracted verbatim location text $\hat{\ell}_i$ to fuzzy-match the gold ℓ_i at $\theta \geq 85$ (rapidfuzz token-sort ratio). **S3-P/R/F1** is the end-to-end score after S3: micro P/R/F1 over $(\mathcal{D}, s_i, p)$ tuples, one tuple per

resolved page index $p \in P_i$; a predicted tuple is a true positive only if both s_i and that specific physical page index p exactly match a gold tuple.

Topic-qualified identifiers (*Bind*). Gold topic-qualified rows (e.g. 3-3:Materials, §4) are *included* in all reported scores. No evaluated system attempts topic binding — the deterministic extractor merges recurring 3-3 rows into one candidate, and the VLM prompt requests unqualified ids — so these gold rows are unreachable for every backend and uniformly bound recall. Scoring each configuration with and without topic-qualified identifiers therefore isolates the cost of *Bind* as a controlled ablation.

5.3 S2 Model Sweep Setup

For the sweep, the same rendered CI page images are given directly to a vision-language model prompted for the same (id, title, location-text) triples — no table-parsing step. S2.1 is the only stage swept across backends; every backend is composed with oracle S1 and the shared S2.2/S3 stages above (Table 2). The sweep evaluates 48 backends under one shared harness: open-weight VLMs are self-hosted on Kubernetes (AKS) with H100 GPU nodes, served via vLLM (Kwon et al., 2023); closed models are queried through their provider APIs behind the same gateway. Appendix B gives further detail.

Roster curation. We used OCRBench (Liu et al., 2024) — the OCR-centric benchmark most consistently reported across candidate models’ technical reports and model cards — as a guide: back-

Split	Gold	S1			S2			S3		
		P	R	F1	P	R	F1	P	R	F1
gri_easy	all	.93	.97	.93	.64	.51	.56	.73	.56	.64
	w/o topic	(same)			.64	.53	.58	.73	.67	.70
gri_med	all	.93	.96	.92	.73	.57	.64	.69	.46	.55
	w/o topic	(same)			.73	.59	.65	.69	.58	.63
gri_hard	all	.92	.78	.79	.36	.29	.32	.26	.06	.10
	w/o topic	(same)			.36	.30	.33	.26	.08	.12

Table 3: Deterministic baseline scored at each stage boundary, with S2/S3 composed on the baseline’s own detected S1. **all** scores against the full gold; **w/o topic** excludes topic-qualified identifiers — the *Bind* ablation (S1 is unaffected).

ends were selected to span its full published score range, not just current leaderboard leaders, so the sweep covers varying document-reading competence rather than only strong OCR readers. Twenty-five of the evaluated open-weight models report a comparable OCRBench number; the frontier APIs and several families (e.g. the ministral-3 models and most gemma-4 variants) do not publish one, as do four models whose only available figure comes from a third-party competitor table — all excluded from the OCRBench-conditioned analysis of §7, which checks whether that guide actually predicted extraction quality. Per-backend references are listed in Appendix B.3.

6 Results

6.1 Deterministic Baseline

Table 3 reports the fully deterministic pipeline (§5) at each stage boundary on the three evaluation splits. S1 detection is near solved on gri_easy; the OCR fallback is what keeps gri_hard viable (without it, hard-split S1-F1 drops from .79 to .60). The S2 and S3 scores compose the baseline’s own detected S1, so they inherit these S1 errors; gri_hard collapses at S3 (.10) — the page-resolution pathologies analysed below. The *Bind* ablation (§5.2) excludes topic-qualified identifiers from gold: S2-F1 moves by at most .02, but S3-F1 rises by .06/.08/.02 on the three splits — recall the baseline forfeits by never attempting topic binding.

6.2 Selecting the S2.1 Backend: A VLM Sweep

We sweep open-weight VLMs spanning sub-1B to 35B parameters (plus frontier API references) as S2.1 backends, all evaluated under oracle S1 with shared S2.2/S3 code. Unlike prior VLM-vs-

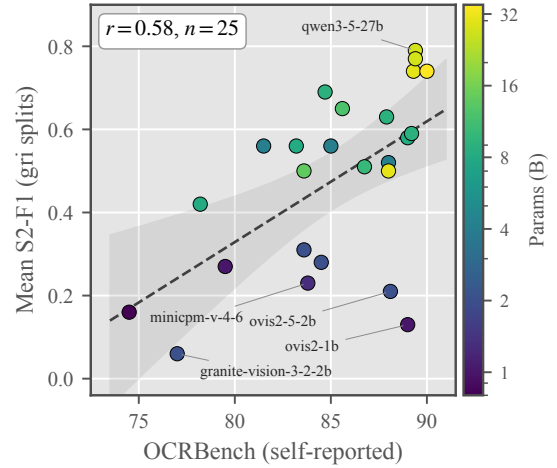


Figure 3: Mean S2-F1 vs. self-reported OCRBench (Liu et al., 2024), for the 25 open-weight backends with a comparable score. Points are colored by parameter count (total for mixture-of-experts); the dashed line is the OLS fit with an approximate 95% band.

OCR table-extraction benchmarks (Nunes et al., 2025; Soric et al., 2025; Horn and Keuper, 2026) (§2), which compare models on isolated single-page tables, GRI content indices routinely span multiple physical pages as a single logical table, so extraction quality here is measured by its effect on a composed, framework-grounded output.

Table 2 ranks all evaluated models by mean S2-F1 across the three splits, grouped by model family and ordered by parameter count (mixture-of-experts sizes shown as total/active).

7 Analysis

7.1 Relationship to Standard OCR Benchmarks

Figure 3 tests whether OCRBench, used to guide roster curation (§5.3), actually predicts S2 extraction quality, plotting mean S2-F1 against it for the 25 open-weight backends with a comparable score. The association is positive but loose: Pearson $r = 0.58$ ($p = 0.002$; Spearman $\rho = 0.61$), so OCRBench accounts for only about a third of the variance in S2-F1 ($r^2 = 0.34$). Reading competence is thus necessary but far from sufficient for content-index extraction.

The residuals are structured by model capacity: several sub-2B models (e.g. ovis2-1b, qwen3-5-2b) post a competitive OCRBench (77–89) yet collapse to S2-F1 below .25, and the signed residual increases with parameter count (Spearman $\rho = 0.60$). We read this as a task mis-

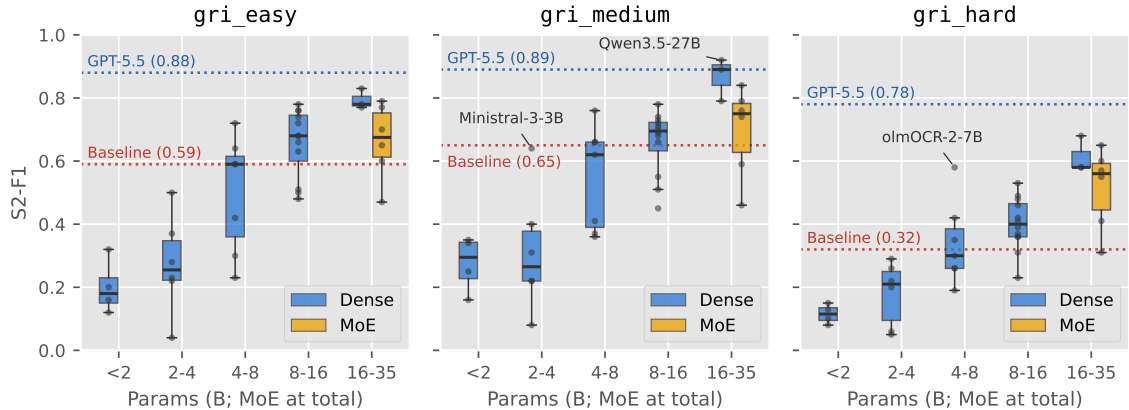


Figure 4: S2.1 location-extraction quality (S2-F1) by parameter bin, per split, for dense vs. mixture-of-experts open-weight VLMs under oracle S1 (closed models excluded, no published size); dotted lines mark GPT-5.5 and the deterministic Baseline. Boxes show median and IQR; dots are individual models. Annotated outliers: Ministral-3-3B, olmOCR-2-7B, and Qwen3.5-27B, the only open model to surpass GPT-5.5.

match, not noise: OCRBench rewards recognizing characters in single-region crops, already saturated at this size, while S2 demands multi-page, schema-faithful structured prediction and instruction following—these models can compose a table (many emit HTML natively), but our task’s specific structuring, a multi-page (id, title, location-text) triple matched against $\mathcal{F}$ ’s fixed schema, breaks that flow. Other OCR-centric benchmarks we collected show the same trend (DocVQA (Mathew et al., 2021), $r = 0.68$; CharXiv (Wang et al., 2024), $r = 0.70$), so the pattern is not specific to OCRBench. Since these small models’ character-recognition competence is already saturated, a natural line of future work is to repurpose them as pure OCR engines feeding the deterministic S2.1 pipeline (§5.1) — in place of RapidOCR — rather than asking them to also structure the output, and compare against the Img2Table+RapidOCR baseline directly.

7.2 S2 Extraction Quality

Figure 4 summarizes the S2.1 sweep (§6.2) as S2-F1 distributions over parameter bins, separating dense from mixture-of-experts models. Three patterns are consistent across splits. First, closed frontier models lead on the hard split (GPT-5.5 at .78 S2-F1 vs. .68 for the best open model), but the best open models are competitive or better on gri_med (Qwen3.5-27B at .92). Second, location-text fidelity scales with model capacity far more steeply than disclosure identification does: sub-4B models frequently extract the right rows but paraphrase or truncate location text, collapsing S2-F1 below .40.

Third, split difficulty dominates model choice on gri_hard: every backend degrades substantially, consistent with the pagination and rasterization pathologies described above.

8 Conclusion

We formalized Content Index Detection (CID) as a composition of five document-extraction tasks — Search, Parse, Bind, Fill, and Resolve — spanning three evaluable stages, and released esg_cid_plus: 37 GRI reports and 7,294 M -complete rows stratified into gri_easy, gri_med, and an adversarial gri_hard tier, each row carrying both verbatim page-location text and resolved physical page indices. Our deterministic baseline and 35B-parameter VLM sweep converge on the same finding from opposite directions: page-location resolution (S3) is the binding bottleneck, not table extraction (S2.1). Even oracle-quality S2.1 rows only lift gri_hard S3-F1 to .11, and improvements in S2-F1 among strong backends stop translating into S3-F1 gains once extraction quality clears a moderate threshold — what predicts end-to-end success past that point is whether a backend’s page references resolve cleanly, not how closely its location text matches the annotator’s phrasing. This decomposition lets future work target *Resolve* directly — e.g. modelling a report’s label scheme jointly with extraction, rather than treating page resolution as a deterministic post-process — and the stage-scored, page-resolved gold in esg_cid_plus makes that progress measurable rather than inferred from end-to-end scores alone.

Limitations

esg_cid_plus covers 37 GRI reports (10/10/10 across gri_easy/gri_med/gri_hard, plus 7 dev), substantially smaller than table-structure-recognition corpora such as PubTables-1M. Annotating page-resolved content indices requires a human to locate the index pages, transcribe location text verbatim, and resolve it against each report’s own page-label scheme — a slow, per-report process that does not scale the way automatically mined tables do. This is most acute for gri_hard: ten reports is enough to surface the label-scheme and rasterization failure modes we report, but too few to estimate how S3 performance varies across the long tail of adversarial page-labeling conventions in the wild.

Ethics Statement

The corporate sustainability reports underlying esg_cid_plus are publicly disclosed by the companies themselves, contain no personal or sensitive information, and are used here strictly for their stated purpose: making ESG disclosures auditable. Annotation was carried out by the authors; no external crowd labor was used. We see this work as supporting, not substituting for, sustainability accountability: automating content-index recovery lowers the cost of checking whether a report actually discloses what a framework requires, which benefits auditors, investors, NGOs, and regulators assessing compliance at scale, and the stage-level errors we surface (particularly the S3 page-resolution bottleneck) are meant to caution against deploying automated disclosure-checking tools without accounting for where they still fail. We do not foresee a dual-use risk specific to this task: the same extraction pipeline that helps a third party verify a report’s disclosures could in principle help a reporting company game a content index, but the dataset and models here only recover disclosure structure that a report already states, and do not evaluate the truthfulness of underlying ESG claims. esg_cid_plus will be released under the same license as its source PDFs’ host repository, airefinery/esg_cid, at https://huggingface.co/datasets/ahmeshaf/esg_cid_plus.

Disclaimer

This content is provided for general information purposes and is not intended to be used in place of consultation with our professional advisors. This

document may refer to marks owned by third parties. All such third-party marks are the property of their respective owners. No sponsorship, endorsement or approval of this content by the owners of such marks is intended, expressed or implied.

Copyright © 2026 Accenture. CC BY-NC-ND. All rights reserved. Accenture and its logo are registered trademarks of Accenture.

References

- Shafiuddin Rehan Ahmed, Ankit Shah, Quan Hung Tran, Vivek Khetan, Sukryool Kang, Ankit Mehta, Yujia Bao, and Wei Wei. 2025. [Enhancing retrieval for ESGLLM via ESG-CID: A disclosure content index finetuning dataset for mapping GRI and ESRS](#). In *Proceedings of the 2nd Workshop on Natural Language Processing Meets Climate Change (ClimateNLP 2025)*, pages 1–16, Vienna, Austria. Association for Computational Linguistics.
- AIDC-AI Ovis Team. 2026. Ovis2.6-30B-A3B: Mixture-of-experts multimodal large language model. Hugging Face model card, <https://huggingface.co/AIDC-AI/Ovis2.6-30B-A3B>. Official release artifact for the Ovis2.6 MoE VLM (30B total / 3B active parameters); no dedicated Ovis2.6 technical report exists.
- Xiang An, Yin Xie, Feilong Tang, Yunyao Yan, Huajie Tan, Didi Zhu, Changrui Chen, Xiuwei Zhao, Bin Qin, Kaicheng Yang, and 1 others. 2026. LLaVA-OneVision-2: Towards next-generation perceptual intelligence. *arXiv preprint arXiv:2605.25979*.
- Anthropic. 2025a. [Claude Haiku 4.5 system card](#). Accessed: July 2026.
- Anthropic. 2025b. System card: Claude Sonnet 4.5. <https://assets.anthropic.com/m/12f214efcc2f457a/original/Claude-Sonnet-4-5-System-Card.pdf>. Accessed: July 2026.
- Xavier Canton. 2021. img2table: Table identification and extraction from PDFs and images. <https://github.com/xavctn/img2table>.
- Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. 2022. HiTab: A hierarchical table dataset for question answering and natural language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1094–1110. Association for Computational Linguistics.
- Michele Luca Contalbo, Sara Pederzoli, Francesco Del Buono, Venturelli Valeria, Francesco Guerra, and Matteo Paganelli. 2025. [GRI-QA: a comprehensive benchmark for table question answering over environmental data](#). In *Findings of the Association for*

- Computational Linguistics: ACL 2025*, pages 15764–15779, Vienna, Austria.
- Amala Sanjay Deshmukh, Kateryna Chumachenko, Tuomas Rintamaki, Matthieu Le, Tyler Poon, Danial Mohseni Taheri, Ilia Karmanov, Guilin Liu, Jarno Seppanen, Guo Chen, and 1 others. 2025. NVIDIA Nemotron Nano V2 VL. *arXiv preprint arXiv:2511.03929*.
- Amala Sanjay Deshmukh, Kateryna Chumachenko, Tuomas Rintamaki, Matthieu Le, Tyler Poon, Danial Mohseni Taheri, Ilia Karmanov, Guilin Liu, Jarno Seppanen, Arushi Goel, and 1 others. 2026. Nemotron 3 Nano Omni: Efficient and open multimodal intelligence. *arXiv preprint arXiv:2604.24954*.
- Yeray Diaz Diaz. 2018. Lunr.py. <https://github.com/yeraydiazdiaz/lunr.py>. Python port of Lunr.js.
- Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leipold. 2020. CLIMATE-FEVER: A dataset for verification of real-world climate claims. *Preprint*, arXiv:2012.00614.
- Daxiang Dong, Mingming Zheng, Dong Xu, Chunhua Luo, Bairong Zhuang, Yuxuan Li, Ruoyun He, Haoran Wang, Wenyu Zhang, Wenbo Wang, and 1 others. 2026. Qianfan-OCR: A unified end-to-end model for document intelligence. *arXiv preprint arXiv:2603.13398*.
- Ling Fu, Zhebin Kuang, Jiajun Song, Mingxin Huang, Biao Yang, Yuzhe Li, Linghao Zhu, Qidi Luo, Xinyu Wang, Hao Lu, 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. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*.
- Gemma Team. 2026. Gemma 4 technical report. *Preprint*, arXiv:2607.02770.
- Google. 2025. Gemini 3 Flash: Frontier intelligence built for speed. <https://blog.google/products/gemini/gemini-3-flash/>. Model card: <https://deepmind.google/models/model-cards/gemini-3-flash/>. Accessed July 2026.
- Google DeepMind. 2026a. Gemini 3.1 Flash-Lite model card. <https://deepmind.google/models/model-cards/gemini-3-1-flash-lite/>. Published March 3, 2026. Accessed July 2026.
- Google DeepMind. 2026b. Gemini 3.5 Flash model card. <https://deepmind.google/models/model-cards/gemini-3-5-flash/>. Published May 19, 2026. Accessed July 2026.
- Granite Vision Team, Leonid Karlinsky, Assaf Arbelle, Abraham Daniels, Ahmed Nassar, Amit Alfassi, Bo Wu, Eli Schwartz, Dhiraj Joshi, Jovana Kondic, and 1 others. 2025. Granite Vision: a lightweight, open-source multimodal model for enterprise Intelligence. *arXiv preprint arXiv:2502.09927*.
- Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, and 1 others. 2025. GLM-4.5V and GLM-4.1V-Thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*.
- Pius Horn and Janis Keuper. 2026. Beyond string matching: Semantic evaluation of PDF table extraction. *arXiv preprint arXiv:2603.18652*.
- Ailin Huang, Chengyuan Yao, Chunrui Han, Fanqi Wan, Hangyu Guo, Haoran Lv, Hongyu Zhou, Jia Wang, Jian Zhou, Jianjian Sun, and 1 others. 2026. Step3-VL-10B technical report. *arXiv preprint arXiv:2601.09668*.
- IBM Granite Vision Team. 2026. Granite 4.1 Vision. Hugging Face model card, released April 29, 2026; accessed July 2026.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th ACM SIGOPS Symposium on Operating Systems Principles (SOSP)*.
- Alexander H Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, and 1 others. 2026. Ministral 3. *arXiv preprint arXiv:2601.08584*.
- Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. 2024. OCR-Bench: On the hidden mystery of OCR in large multimodal models. *Science China Information Sciences*, 67(12):220102.
- Shiyin Lu, Yang Li, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Han-Jia Ye. 2024. Ovis: Structural embedding alignment for multimodal large language model. *arXiv preprint arXiv:2405.20797*.
- Shiyin Lu, Yang Li, Yu Xia, Yuwei Hu, Shanshan Zhao, Yanqing Ma, Zhichao Wei, Yinglun Li, Lunhao Duan, Jianshan Zhao, and 1 others. 2025. Ovis2.5 technical report. *arXiv preprint arXiv:2508.11737*.
- R. Mankour, Y. Chafai, H. Saleh, G. Ben Hassine, T. Barreau, and P. Tankov. 2025. Climate finance bench. https://github.com/Pladifes/climate_finance_bench. Working paper, PLADIFES.

- Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. 2021. DocVQA: A dataset for VQA on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.
- Jingwei Ni, Julia Bingler, Chiara Colesanti-Senni, Mathias Kraus, Glen Gostlow, Tobias Schimanski, Dominik Stammach, Saeid Ashraf Vaghefi, Qian Wang, Nicolas Webersinke, Tobias Wekhof, Tingyu Yu, and Markus Leippold. 2023. *CHATREPORT: Democratizing sustainability disclosure analysis through LLM-based tools*. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 21–51, Singapore. Association for Computational Linguistics.
- Guilherme Nunes, Vitor Rolla, Duarte Pereira, Vasco Alves, Andre Carreiro, and Márcia Baptista. 2025. *Benchmarking table extraction: Multimodal LLMs vs traditional OCR*. In *Proceedings of the 1st Joint Workshop on Large Language Models and Structure Modeling (XLLM 2025)*, pages 8–15, Vienna, Austria. Association for Computational Linguistics.
- Keane Ong, Rui Mao, Ranjan Satapathy, Ricardo Shirota Filho, Erik Cambria, Johan Sulaeman, and Gianmarco Mengaldo. 2025. *Explainable natural language processing for corporate sustainability analysis*. *Information Fusion*, 115:102726.
- OpenAI. 2026a. *GPT-5.4 thinking system card*. Published March 5, 2026; accessed July 2026.
- OpenAI. 2026b. *GPT-5.5 system card*. Published April 23, 2026; accessed July 2026.
- OpenBMB. 2026. MiniCPM-V 4.6. <https://huggingface.co/openbmb/MiniCPM-V-4.6>. Hugging Face model card. Accessed 2026-07-10.
- Marco Polignano, Nicola Bellantuono, Francesco Paolo Lagrasta, Sergio Caputo, Pierpaolo Pontrandolfo, and Giovanni Semeraro. 2022. *An NLP approach for the analysis of global reporting initiative indexes from corporate sustainability reports*. In *Proceedings of the First Computing Social Responsibility Workshop within the 13th Language Resources and Evaluation Conference*, pages 1–8, Marseille, France. European Language Resources Association.
- Jake Poznanski, Luca Soldaini, and Kyle Lo. 2025. olmOCR 2: Unit test rewards for document OCR. *arXiv preprint arXiv:2510.19817*.
- Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, and 1 others. 2025. UI-TARS: Pioneering automated GUI interaction with native agents. *arXiv preprint arXiv:2501.12326*.
- Qwen Team. 2026a. *Qwen3.5: Towards native multimodal agents*. Official release blog post; accessed July 2026.
- Qwen Team. 2026b. *Qwen3.6-27B: Flagship-level coding in a 27B dense model*. Official release blog post. No Qwen3.6 technical report exists as of July 2026 (searched arXiv and Google Scholar; the QwenLM/Qwen3.6 GitHub README lists “Paper” as unlinked placeholder text and documentation as “coming soon”). Accessed July 2026.
- Qwen Team. 2026c. *Qwen3.6-35B-A3B: Agentic coding power, now open to all*. <https://qwen.ai/blog?id=qwen3.6-35b-a3b>. Official release blog post. No Qwen3.6 technical report exists as of July 2026 (searched arXiv and Google Scholar). Cited as the closest official artifact for the evaluated checkpoint listed as “Qwen3.6-35B-A3B v2”: no official v2/-2 variant of Qwen3.6-35B-A3B is documented by Qwen anywhere (Hugging Face, ModelScope, GitHub, or blog); only the single April 2026 release exists. Accessed July 2026.
- RapidAI. 2021. RapidOCR: A lightweight, cross-platform OCR toolkit based on ONNX runtime. <https://github.com/RapidAI/RapidOCR>.
- Jon Saad-Falcon, Joe Barrow, Alexa Siu, Ani Nenkova, David Seunghyun Yoon, Ryan A. Rossi, and Franck Dernoncourt. 2023. *PDFTriage: Question answering over long, structured documents*. *Preprint*, arXiv:2309.08872.
- Tobias Schimanski, Imene Kolli, Yu Fan, Ario Saeid Vaghefi, Jingwei Ni, Elliott Ash, and Markus Leippold. 2026. *pdfQA: Diverse, challenging, and realistic question answering over PDFs*. *Preprint*, arXiv:2601.02285.
- Tobias Schimanski, Jingwei Ni, Roberto Spacey, Nicola Ranger, and Markus Leippold. 2024a. *ClimRetrieve: A benchmarking dataset for information retrieval from corporate climate disclosures*. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17492–17516.
- Tobias Schimanski, Andrin Reding, Nico Reding, Julia Bingler, Mathias Kraus, and Markus Leippold. 2024b. *Bridging the gap in ESG measurement: Using NLP to quantify environmental, social, and governance communication*. *Finance Research Letters*, 61:104979.
- Štěpán Šimsa, Milan Šulc, Michal Uříčář, Yash Patel, Ahmed Hamdi, Matěj Kocián, Matyáš Skalický, Jiří Matas, Antoine Doucet, Mickaël Coustaty, and Dimosthenis Karatzas. 2023. *DocILE benchmark for document information localization and extraction*. In *Document Analysis and Recognition – ICDAR 2023*, volume 14188 of *Lecture Notes in Computer Science*, pages 147–166. Springer.
- Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, and 1 others. 2025. OpenAI GPT-5 system card. *arXiv preprint arXiv:2601.03267*.

- Brandon Smock, Rohith Pesala, and Robin Abraham. 2022a. GriTS: Grid table similarity metric for table structure recognition. *arXiv preprint arXiv:2203.12555*.
- Brandon Smock, Rohith Pesala, and Robin Abraham. 2022b. PubTables-1M: Towards comprehensive table extraction from unstructured documents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4634–4642.
- Marijan Soric, Cécile Gracianne, Ioana Manolescu, and Pierre Senellart. 2025. [Benchmarking table extraction from heterogeneous scientific documents](#). *arXiv preprint arXiv:2511.16134*.
- Tomasz Stanisławek, Filip Graliński, Anna Wróblewska, Dawid Lipiński, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemysław Biecek. 2021. [Kleister: Key information extraction datasets involving long documents with complex layouts](#). In *Document Analysis and Recognition – ICDAR 2021*, volume 12821 of *Lecture Notes in Computer Science*, pages 564–579. Springer.
- Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. 2023. [Hierarchical multimodal transformers for multi-page DocVQA](#). *Pattern Recognition*, 144:109834.
- Aida Usmanova, Rana Abdullah, Debayan Banerjee, Markus Leippold, and Ricardo Usbeck. 2025. [ReportGRI: Automating GRI alignment and report assessment](#). In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25)*.
- Saeid Ashraf Vaghefi, Dominik Stammach, Veruska Muccione, Julia Binger, Jingwei Ni, Mathias Kraus, Simon Allen, Chiara Colesanti-Senni, Tobias Wekhof, Tobias Schimanski, Glen Gostlow, Tingyu Yu, Qian Wang, Nicolas Webersinke, Christian Huggel, and Markus Leippold. 2023. [ChatClimate: Grounding conversational AI in climate science](#). *Communications Earth & Environment*, 4(1):480.
- Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, and 1 others. 2025. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.
- Xinyu Wang, Lin Gui, and Yulan He. 2023. [A scalable framework for table of contents extraction from complex ESG annual reports](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13215–13229, Singapore. Association for Computational Linguistics.
- Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. 2021. TUTA: Tree-based transformers for generally structured table pre-training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD)*, pages 1780–1790.
- Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Hao-tian Liu, Sadhika Malladi, Alexis Chevalier, Sanjeev Arora, and Danqi Chen. 2024. CharXiv: Charting gaps in realistic chart understanding in multimodal LLMs. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2024 Datasets and Benchmarks Track)*.
- Nicolas Webersinke, Mathias Kraus, Julia Anna Binger, and Markus Leippold. 2022. ClimateBERT: A pretrained language model for climate-related text. In *Tackling Climate Change with Machine Learning Workshop, NeurIPS*.
- Tianyu Yu, Zefan Wang, Chongyi Wang, Fuwei Huang, Wenshuo Ma, Zhihui He, Tianchi Cai, Weize Chen, Yuxiang Huang, Ranchi Zhao, and 1 others. 2026. MiniCPM-V 4.5: Cooking efficient MLLMs via architecture, data, and training recipe. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11704–11715.
- Xinyi Zheng, Doug Burdick, Lucian Popa, Xu Zhong, and Nancy Xin Ru Wang. 2021. Global table extractor (GTE): A framework for joint table identification and cell structure recognition using visual context. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Introduces FinTabNet.

A Dataset Details

A.1 gri_hard Composition

Table 4 lists the ten gri_hard reports underlying the difficulty tiers of §4.

A.2 Page-Label Schemes

Table 7 reports, for each of the 37 PDFs, the declared /PageLabels scheme (trivial: no entry,

Report	Category	Mod.
HELLENIC_TELECOM_2021	Section-based location	text
OLYMPUS_2021	Section-based location	text
PANASONIC_2022	Section-based location	text
FUJITSU_2023	Complex label scheme	text
ADANI_2022	Double-paged	text
ADAPTEO_2022	Double-paged	text
FUJITSU_2023_x	Scrambled labels	text
HYUNDAI_2020_x	Image-only CI pages	mixed
TOYOTA_2023_x	Image-only	image
TRAIN_187_x	Scrambled labels	text

Table 4: The ten gri_hard reports. _x suffix = synthetic perturbation of a source report; the rest are real-world adversarial documents. Mod.: *mixed* = only GRI index pages rasterized; *image* = full document rasterized.

Split	PDFs	Pages/PDF median (Q1, Q3)	M	M^{abs}	M^{men}	M^{pg}	$M^{\overline{\text{pg}}}$	$ P_{\mathcal{F}} /\mathcal{D}$ mean
gri_easy	10	96 (74, 166)	1,936	1,105	831	790	41	5.8
gri_medium	10	128 (92, 148)	1,937	1,007	930	917	13	4.1
gri_hard	10	131 (62, 290)	2,028	1,169	859	823	36	8.2
train_small	7	125 (62, 166)	1,393	895	498	497	1	4.0

Table 5: Per-split statistics. M = total schema rows; $M^{\text{abs}} / M^{\text{men}}$ = absent / mentioned rows; $M^{\text{pg}} / M^{\overline{\text{pg}}} =$ page-indexed / non-page-indexed mentioned rows; $|P_{\mathcal{F}}|/\mathcal{D}$ = mean index pages per report (S1 scale).

implicit $i = l - 1$; nontrivial: missing or structurally irregular; complex: non-numeric labels requiring custom parsing — see §4 for the tier definitions these feed) and the resulting label-to-index function $i = f(l)$.

A.3 Descriptive Statistics

Table 5 summarises each split along the dimensions that drive the four evaluation stages.

Three patterns stand out. First, absent rows dominate: roughly 55–57% of schema slots are unpopulated across all splits, reflecting real variation in what companies choose to disclose. Second, gri_hard has the highest mean index-page count (8.2 vs 4–6 elsewhere), a direct consequence of selecting reports with complex or multi-part content indices. Third, non-page-indexed rows are rare (91 of 3,118 mentioned rows, 2.9%) but must be retained for S2 eval: they carry valid location or status information even when no physical page index can be produced.

Table 6 breaks down rows by disclosure kind. Standard disclosures dominate in both count and mention rate; the topic-qualified family is smaller but carries a distinct difficulty — every mentioned topic-qualified row is page-indexed ($M_{\text{top}}^{\text{men}} = M_{\text{top}}^{\text{pg}}$), so all non-page-indexed rows in the dataset come from standard disclosures. Baselines are evaluated over standard disclosures only.

Split	Standard			Topic-qualified		
	M_{std}	$M_{\text{std}}^{\text{men}}$	$M_{\text{std}}^{\text{pg}}$	M_{top}	$M_{\text{top}}^{\text{men}}$	$M_{\text{top}}^{\text{pg}}$
gri_easy	1,305	774	733	631	57	57
gri_medium	1,305	850	837	632	80	80
gri_hard	1,334	798	762	694	61	61
train_small	928	401	400	465	97	97

Table 6: Rows by disclosure kind. $M_{\text{std}}/M_{\text{top}}$ = standard / topic-qualified total rows; superscripts men / pg = mentioned / page-indexed subsets. $M_{\text{top}}^{\text{men}} = M_{\text{top}}^{\text{pg}}$ in all splits: all topic-qualified rows are page-indexed.

A.4 Qualitative Analysis

The common case — a numeric label resolving cleanly to a physical page — is well represented across all splits (e.g. $\ell_i = "62" \rightarrow P_i = \{61\}$, or a range "12–15" $\rightarrow \{11, 12, 13, 14\}$). Three harder cases illustrate why the benchmark is difficult.

Section-based location. In OLYMPUS_2021 (GRI 102-1), the content index gives $\ell_i = "Consolidated Company Outline"$ with no page number. The verbatim cell is retained as ℓ_i for auditability, but $P_i = \emptyset$ — the row is mentioned but non-page-indexed. This is the defining challenge for the three section-based reports in gri_hard: the location text is present and correct, but no numeric index can be produced without resolving the section pointer.

Topic-qualified disambiguation. In KERRY GROUP_2023, the GRI 3-3 row recurs once per material topic in the content index with no topic label — the topic is given only by the surrounding section header. The gold resolves each occurrence to its qualified identifier (e.g. $s_i = 3-3: \text{Biodiversity}$), yielding thirteen distinct 3-3 disclosures that the flat namespace in airefinery/esg_cid collapses to a single row.

Image-only perturbation. In TOYOTA_2023_x, all pages are rasterised and carry no text layer. The gold annotation is intact ($s_i = 2-1$, $\ell_i = "3"$, $P_i = \{2\}$), but the disclosure is inaccessible to any text-only system. The same row is trivially extractable from the unperturbed TOYOTA_2023 in gri_medium, making the pair a controlled ablation of what image-only rendering costs.

B Implementation Details

B.1 Evaluation Harness (model-*iq*)

We run the S2.1 backend sweep (§6.2, §5.3) on **model-*iq***, an internal Kubernetes-native control plane for serving and governing both self-hosted and API-based large language models as one fleet.

Report (Company)	Ctry	SASB Industry	Year	Layout	$ P_F $	G	label scheme	$i = f(l)$
gri_easy								
<i>GRI 2016 ($M = 238$)</i>								
TRAIN_170 (Coupa)	US	Software and IT Services	2021	single	6	76	trivial	$i = l - 1$
TRAIN_20 (Pfizer)	US	Biotechnology and Pharmaceuticals	2021	single	10	75	trivial	$i = l - 1$
TRAIN_212 (Molson Coors Beverage)	US	Alcoholic Beverages	2022	single	3	61	trivial	$i = l - 1$
TRAIN_225 (Green Plains)	US	Biofuels	2021	single	10	115	trivial	$i = l - 1$
TRAIN_70 (TJX Companies)	US	Apparel Accessories and Footwear	2022	single	5	76	trivial	$i = l - 1$
<i>GRI 2021 ($M = 147$)</i>								
AB SKF_2023	SE	Industrial Machinery and Goods	2023	single	5	84	trivial	$i = l - 1$
KERRY GROUP_2023	IE	Processed Foods	2023	single	5	115	trivial	$i = l - 1$
KIA_2024	KR	Automobiles	2024	single	3	102	trivial	$i = l - 1$
KPN_2023	NL	Telecommunications	2023	single	9	63	trivial	$i = l - 1$
TRAIN_90 (SK Inc.)	KR	Asset Management and Custody Activities	2023	single	2	86	trivial	$i = l - 1$
gri_medium								
<i>GRI 2016 ($M = 238$)</i>								
TRAIN_143 (Kumbra Iron Ore)	ZA	Metals and Mining	2020	single	5	132	trivial	$i = l + 1$
TRAIN_153 (NTT DOCOMO)	JP	Telecommunications	2020	single	4	148	trivial	$i = l$
TRAIN_187 (Ferrexpo)	CH	Metals and Mining	2020	single	7	84	trivial	$i = l$
TRAIN_194 (Banca Transilvania)	RO	Commercial Banks	2020	single	6	150	nontrivial	$i = l - 1$
TRAIN_307 (Taishin Financial)	TW	Commercial Banks	2021	single	5	114	trivial	$i = l + 1$
<i>GRI 2021 ($M = 147$)</i>								
HYUNDAI_2023	KR	Automobiles	2023	single	3	108	trivial	$i = l - 1$
HYUNDAI_2024	KR	Automobiles	2024	single	3	108	nontrivial	$i = l - 1$
LACTALIS_2023	FR	Meat, Poultry and Dairy	2023	single	3	57	trivial	$i = l + 1$
TOYOTA_2023	JP	Automobiles	2023	single	3	46	nontrivial	$i = l - 1$
TRELLEBORG_2023	SE	Industrial Machinery and Goods	2023	single	2	83	nontrivial	$i = l + 1$
gri_hard								
<i>GRI 2016 ($M = 238/241$)</i>								
ADANI_2022	IN	Engineering and Construction Services	2022	double	8	129	nontrivial	$i = \lfloor l/2 \rfloor$
HELLENIC_2021	GR	Telecommunications	2021	single	14	122	trivial	$i = l - 1$
HYUNDAI_2020_x [†]	KR	Automobiles	2020	single	5	93	trivial	$i = l - 1$
OLYMPUS_2021	JP	Medical Equipment and Supplies	2021	single	9	148	nontrivial	$i = l$
PANASONIC_2022	JP	Electrical and Electronic Equipment	2022	single	5	137	trivial	$i = l$
TRAIN_187_x [†]	CH	Metals and Mining	2020	single	7	84	complex	—
<i>GRI 2021 ($M = 147/148$)</i>								
ADAPTEO_2022	SE	Real Estate Owners, Developers and Investment Trusts	2022	double	3	74	nontrivial	$i = \lfloor l/2 \rfloor$
FUJITSU_2023	JP	Software and IT Services	2023	single	14	117	complex	—
FUJITSU_2023_x [†] (syn)	JP	Software and IT Services	2023	single	14	117	complex	—
TOYOTA_2023_x [†] (syn)	JP	Automobiles	2023	single	3	46	trivial	$i = l - 1$

Table 7: Per-report characteristics for all 37 PDFs. $|P_F|$ = number of content-index pages; G = true rows ($\ell_i \neq \varepsilon$); /PageLabels scheme: trivial = no /PageLabels entry; nontrivial = missing or structurally irregular; complex = non-numeric labels requiring custom parsing. $i = f(l)$ is the 0-based page index as a function of the printed label l . [†] = synthetic perturbation of the preceding real-world report.

This appendix details the parts of its architecture relevant to reproducing the sweep; model-iq’s broader scope (cost analytics, RBAC, budget-scoped API keys) is out of scope here.

Unified serving. model-iq deploys open-weight VLMs on GPU-backed infrastructure via **KServe**, with the serving runtime pluggable per model (we use **vLLM** (Kwon et al., 2023) and **SGLang** de-

pending on backend support); GPU scheduling and autoscaling are handled by the GPU Operator and KEDA. API-based closed models (used for the frontier-API references in Table 2) bypass the serving layer entirely and are proxied through the same gateway with provider credential injection. This means every backend — self-hosted or API-based — is reachable through one OpenAI-

compatible request path, so the extraction prompt, image payload, and response parsing code in our S2.1 harness is identical for all 48 backends; only the routed endpoint differs.

Model cards. Each evaluated backend is registered as a *model card*: a versioned record of its identity, weight source (Hugging Face repo id, or provider/model id for closed models), and deployment configuration (serving runtime, GPU/replica count, context-window and generation parameters). Every reported score is tagged with the model-card id and version that produced it, so a score can always be traced back to the exact serving configuration used — important given that quantization, context length, and runtime choice can all shift VLM output quality independently of the underlying checkpoint.

Request path and failure handling. The harness issues one request per confirmed CI page image (§5.3) through the model-iq gateway, which handles auth, rate limiting, and per-backend routing; multi-page CI regions are stitched at the application layer after all page-level responses return. Backends that time out, return malformed output, or exceed context are logged and excluded from that report’s aggregate rather than imputed, so per-model N can vary slightly across splits (Appendix A reports per-report coverage).

Why a unified fleet, not per-provider scripts. Prior VLM-vs-OCR benchmarks (§2) typically compare a handful of models, often via each provider’s own SDK or a single self-hosted runtime. At 48 backends spanning sub-1B self-hosted VLMs to frontier closed APIs, per-provider integration code would itself become a confound — subtle differences in retry logic, image encoding, or prompt formatting across scripts could shift scores independently of model quality. Routing every backend through one gateway and one model-card schema removes that confound and is what makes the sweep’s cross-backend comparisons attributable to the model alone.

Decoding parameters. We do not pin sampling parameters across the sweep: each self-hosted backend reuses its own model family’s recommended hyperparameters (temperature, top- p , etc., as specified in the checkpoint’s `generation_config.json`), falling back to vLLM/SGLang’s defaults (temperature = top- p

= 1.0) when a checkpoint specifies none. Closed-model backends use their provider’s own default.

B.2 Extraction Prompt and Structured Output

Every S2.1 backend is given the same system prompt and constrained to the same structured-output schema (a Pydantic `LLMExtractedTable` of `ExtractedCIRow` objects), enforced via `response_format` on every request regardless of backend.

System Prompt

You are extracting a GRI Content Index table from a sustainability report page.

For each row extract:

- `standard_id`: the GRI standard number, e.g. "GRI 302-1", "2-1", "102-1"
- `disclosure_text`: the disclosure title or description
- `page_location_text`: the page reference verbatim, e.g. "42", "p.42-43", "N/A"
- `note`: any extra remark shown in the row, e.g. "partially reported", "omitted"

Rules:

- Include all rows, even those with no page reference.
- Do not invent standard IDs – only extract what is explicitly present.
- Normalise `standard_id` to the form "NNN-N" or "GRI NNN-N" as printed.
- If the page has multiple index tables (e.g. side-by-side columns or separate topic sections), collapse them into one flat list of rows.

Expected Output

```
{
  "rows": [
    {
      "standard_id": "305-1",
      "disclosure_text": "Direct (Scope 1) GHG Emissions",
      "page_location_text": "44-45",
      "note": ""
    },
    {
      "standard_id": "2-1",
      "disclosure_text": "Organizational details",
      "page_location_text": "12",
      "note": "partially reported"
    }
  ]
}
```

B.3 Roster Curation

Table 8 reports the same mean S2-F1 ranking as Table 2 in the main text, with each backend’s provider affiliation added as a field; those affiliations line up with the family groupings below. The sweep’s

Model	Affiliation	loc-F1	Model	Affiliation	loc-F1
GPT-5.5	OpenAI	0.85	InternVL3.5-4B	Shanghai AI	0.56
GPT-5.4	OpenAI	0.80	InternVL3.5-8B	Shanghai AI	0.56
GPT-5-mini	OpenAI	0.80	Qwen3.5-4B	Alibaba	0.56
Qwen3.5-27B	Alibaba	0.79	Baseline	Ours	0.52
Gemini-3.5-Flash	Google	0.79	Qianfan-OCR	Baidu	0.52
Gemini-3.1-Flash-Lite	Google	0.78	Step3-VL-10B	StepFun	0.51
Gemini-3-Flash	Google	0.77	InternVL3.5-14B	Shanghai AI	0.50
Qwen3.6-27B	Alibaba	0.77	InternVL3.5-30B-A3B	Shanghai AI	0.50
Ovis2.6-30B-A3B	Alibaba	0.74	Ministral-3-3B	Mistral AI	0.48
Qwen3.6-35B-A3B	Alibaba	0.74	Gemma-4-12B-it	Google	0.45
Gemma-4-31B-it	Google	0.71	Nemotron-3-Nano-Omni	NVIDIA	0.45
Claude Sonnet 4.5	Anthropic	0.71	LLaVA-OneVision-2	LLaVA Team	0.42
GLM-4-6v-flash	Zhipu AI	0.69	UI-TARS-1.5-7B	ByteDance	0.34
Gemma-4-26B-A4B-it	Google	0.67	InternVL3.5-2B	Shanghai AI	0.31
olmOCR-2-7B-1025	AI2	0.66	Gemma-4-E4B-it	Google	0.31
Ministral-3-14B	Mistral AI	0.66	Qwen3.5-2B	Alibaba	0.28
Qwen3.5-35B-A3B	Alibaba	0.65	Granite-Vision-4.1-4B	IBM	0.28
Nemotron-Nano-v2-VL	NVIDIA	0.65	InternVL3.5-1B	Shanghai AI	0.27
Ministral-3-8B	Mistral AI	0.64	MiniCPM-V-4.6	OpenBMB	0.23
Ovis2.5-9B	Alibaba	0.63	Ovis2.5-2B	Alibaba	0.21
Claude Haiku 4.5	Anthropic	0.61	Gemma-4-E2B-it	Google	0.19
GPT-5-nano	OpenAI	0.60	Qwen3.5-0-8B	Alibaba	0.16
Qwen3.5-9B	Alibaba	0.59	Ovis2-1B	Alibaba	0.13
MiniCPM-V-4.5	OpenBMB	0.58	Granite-Vision-3.2-2B	IBM	0.06

Table 8: Mean loc-F1 across gri_easy, gri_med, and gri_hard (oracle S1). Closed-source models are indicated in blue; the Baseline (§5.1) is indicated in red.

48 backends are drawn from the following model families:

- **Claude:** claude-haiku-4-5 and claude-sonnet-4-5 (Anthropic, 2025a,b)
- **Gemini:** gemini-3-flash-preview (Google, 2025), gemini-3.1-flash-lite (Google DeepMind, 2026a), and gemini-3.5-flash (Google DeepMind, 2026b)
- **Gemma:** the gemma-4 family (gemma-4-e2b-it, gemma-4-e4b-it, gemma-4-12b-it, gemma-4-26b-a4b-it, and gemma-4-31b-it) (Gemma Team, 2026)
- **GLM:** glm-4-6v-flash (Hong et al., 2025)
- **GPT:** gpt-5-mini and gpt-5-nano (Singh et al., 2025), gpt-5-4-1 (OpenAI, 2026a), and gpt-5-5 (OpenAI, 2026b)
- **Granite Vision:** granite-vision-3-2-2b (Granite Vision Team et al., 2025) and granite-vision-4-1-4b (IBM Granite Vision Team, 2026)
- **InternVL:** the internvl3-5 family (internvl3-5-1b, internvl3-5-2b, internvl3-5-4b, internvl3-5-8b, internvl3-5-14b, and internvl3-5-30b-a3b) (Wang et al., 2025)
- **LLaVA:** llava-onevision-2-8b-instruct (An et al., 2026)
- **MiniCPM-V:** minicpm-v-4-5 (Yu et al., 2026) and minicpm-v-4-6 (Yu et al., 2026; OpenBMB, 2026)

- **Ministral:** the ministral-3 family (ministral-3-3b, ministral-3-8b, and ministral-3-14b) (Liu et al., 2026)
- **Nemotron:** nemotron-nano-v2-vl (Deshmukh et al., 2025) and nemotron-3-nano-omni-reasoning (Deshmukh et al., 2026)
- **olmOCR:** olmocr-2-7b-1025 (Poznanski et al., 2025)
- **Ovis:** ovis2-1b (Lu et al., 2024), the ovis2-5 models (Lu et al., 2025), and ovis2-6-30b-a3b (AIDC-AI Ovis Team, 2026; Lu et al., 2025)
- **Qianfan:** qianfan-ocr (Dong et al., 2026)
- **Qwen:** the qwen3-5 family (qwen3-5-0-8b, qwen3-5-2b, qwen3-5-4b, qwen3-5-9b, qwen3-5-27b, and qwen3-5-35b-a3b) (Qwen Team, 2026a) and the qwen3-6 models (qwen3-6-27b and qwen3-6-35b-a3b-2) (Qwen Team, 2026b,c)
- **Step3:** step3-vl-10b (Huang et al., 2026)
- **UI-TARS:** ui-tars-1-5-7b (Qin et al., 2025)

Table 8 reports the same mean S2-F1 ranking as Table 2 in a two-column layout with affiliation as an added field. We print it in §B.3 rather than repeating it here, since the affiliation column pairs directly with the family-by-family roster listed there.