Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

VGA-KIE: Reliable Layout-Aware Key Information Extraction from Visually Rich Documents via Evidence-Grounded Verification and Repair

Yuepeng Shi

shiyuepeng@stu.shmtu.edu.cn

Henan University of Animal Husbandry and Economy

Ning Kong

Henan University of Animal Husbandry and Economy

Bo Peng

Henan University of Animal Husbandry and Economy

Meng Wang

Henan University of Animal Husbandry and Economy

Research Article

Keywords:

Posted Date: November 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10766973/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

VGA-KIE: Reliable Layout-Aware Key Information Extraction from Visually Rich Documents via Evidence-Grounded Verification and Repair

Yuepeng Shi[1*], Ning Kong[1†], Bo Peng[1†], Meng Wang[1†]

[1*]School of Energy and Intelligent Engineering, Henan University of Animal Husbandry and Economy, Street, Zhengzhou, 450046, Henan, China.

*Corresponding author(s). E-mail(s): shiyuepeng@stu.shmtu.edu.cn;
Contributing authors: dlutdanielkong@gmail.com; 195285840@qq.com;
sunstone333@hotmail.com;
†These authors contributed equally to this work.

Abstract

Key information extraction (KIE) from visually rich documents must identify field values, establish that each value comes from the correct document region, test schema and cross-field constraints, and support interpretable correction after verification failure. Existing document models improve candidate generation, but usually manage candidate values, locatable evidence, verification states, and repair actions separately. We introduce VGA-KIE, a modular post-hoc reliability method that couples three mechanisms as field-level state transitions. An evidence-grounded field state defines the object to be diagnosed. Typed failure diagnosis converts constraint violations into repair conditions. Failure-conditioned repair with re-verification permits a state transition only after renewed evidence linking and verification against the same constraints. Under a shared protocol, LayTokenLLM-8B with VGA-KIE achieved Field-F1 scores of 94.91, 99.17, and 98.86 on FUNSD, SROIE, and CORD. Against direct extraction from the same frozen candidate cache, mean Field-F1 increased from 93.49 to 97.65. VGA-KIE released 4,475 of 4,720 target fields, reached 92.8% Evidence Accuracy and 87.9% Verification Failure F1, and produced a 2.7% Unsupported Output Rate, 70.5% Repair Success, and 2.1% False Repair Rate at 1.32× post-hoc latency. The paired comparison and component ablations support the combined reliability layer under the evaluated protocol. They do not

establish per-class optimal diagnosis, candidate-generator-independent gains, or priority for any individual grounding, verification, or repair operation.

Keywords: key information extraction; visually rich documents; layout-aware document understanding; evidence grounding; verification; failure-conditioned repair

1 Introduction

Visually rich documents, including forms, receipts, invoices, purchase orders, contracts, and business reports, remain structurally difficult objects in document analysis and recognition. FUNSD formulates noisy scanned forms as a form-understanding task [1], whereas SROIE evaluates optical character recognition (OCR) and KIE from scanned receipts [2]. Graph-based extraction and LayoutLM show that relative position, visual layout, and two-dimensional coordinates affect field interpretation [3, 4]. FormNet further shows that serializing forms as plain text loses local structure and key-value relations [5]. CORD frames products, prices, and summary fields as a post-OCR parsing problem [6]. KIE from visually rich documents therefore depends on the interaction among physical structure, logical structure, and semantic interpretation.

Document understanding has progressed from sequence labeling to layout-aware pretraining, generative document language models, and OCR-free multimodal models. DocLLM combines text tokens with bounding boxes so that a language model can reason over spatial layout [7]. LayoutLLM brings visual, textual, and layout information into a large language model through document-understanding instruction tuning [8]. Donut removes the external OCR stage to reduce propagation of recognition errors into structured generation [9]. mPLUG-DocOwl 1.5 emphasizes unified structure learning for OCR-free document understanding [10]. LMDX and See then Tell connect extracted entities to document regions or image evidence [11, 12]. These advances establish modern KIE as a document-structure problem rather than a text-only extraction task.

Stronger candidate generation does not by itself provide auditable reliability. A generative model can return valid JSON while selecting the wrong region, confusing adjacent fields, violating table structure, or generating a value absent from the document. RealKIE, UNIKIE-BENCH, and re-evaluations of text-and-layout models show that schema variation, long-tail fields, complex layouts, and real semantic boundaries amplify these errors [13–15]. Candidate values, evidence, failure reasons, and repair decisions also tend to reside in separate components. This separation prevents them from forming a unified field state with auditable transition rules.

VGA-KIE (Verification-Guided Agent for Key Information Extraction) addresses this coupling gap. VGA-KIE is a modular post-hoc reliability layer, not another candidate encoder. Its method-level distinction is to use evidence support as a state condition, the dominant failure type as the condition for a local action, and re-verification after renewed grounding as the final acceptance gate. OCR-based extractors, layout-aware language models, and OCR-free multimodal large language models (MLLMs) can enter through a common candidate interface. This interface

provides architectural compatibility; the performance claims remain limited to the evaluated candidate condition.

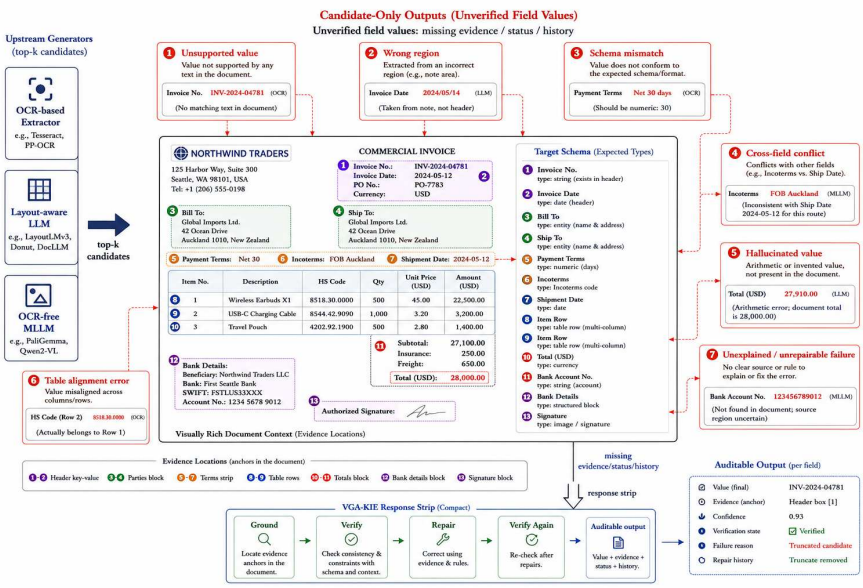


Fig. 1 Candidate-only KIE and the VGA-KIE reliability loop. Panel a shows OCR-based, layout-aware, and OCR-free extractors producing field candidates and generator scores without a complete field-level evidence, verification, or decision trajectory. Panel b groups the missing reliability information into evidence-grounding, schema or layout, and consistency or decision failures. Panel c shows VGA-KIE constructing an evidence-grounded field state, applying typed failure diagnosis, and updating the state through failure-conditioned repair with re-verification. Each final field state records a value, evidence, reliability score, verification state, failure reason, and repair history.

VGA-KIE follows an Extract, Ground, Verify, Repair, and Verify Again loop. The parser first converts a document into OCR tokens, text lines, text blocks, table cells, and layout regions, then organizes these observations as a layout-aware heterogeneous graph [3, 16]. A schema-guided extraction component supplies the top-k candidates for each target field.

The grounding module links every candidate to a token span, text block, table cell, or image region [11, 12]. The verifier tests format, spatial, semantic, evidence, cross-field, and table-structure constraints. When verification fails, the repair module applies a local action conditioned on the dominant failure reason, repeats evidence grounding, and verifies the updated candidate. The output records the value, evidence, reliability score, verification state, failure reason, and repair history.

The three contributions form one design chain that links evidence state, failure conditions, and re-verification gating:

1. **AEvidence-grounded field state.:**VGA-KIE stores the field value, locatable evidence, reliability score, verification state, failure reason, and repair history as

one object, making evidence a precondition for state transition. Evidence Accuracy, Unsupported Output Rate, and the evidence-linking ablation test this role.

2. **Typed failure diagnosis.** VGA-KIE converts format, spatial, evidence, semantic, cross-field, and table-structure constraints into a verification state and dominant failure condition. The evidence consists of binary Verification Failure F1 and the typed-verification ablation; it does not establish per-class F1 for all six failure types.
3. **Failure-conditioned repair with re-verification:**VGA-KIE maps the failure condition to a budgeted local action and uses renewed evidence linking and re-verification to limit harmful updates. Repair Success, False Repair Rate, and the repair and verify-again ablations test its benefit and risk.

2 Related Work

2.1 Layout-aware representation and structural modeling for KIE

Early work on KIE from visually rich documents addressed the loss caused by reducing a two-dimensional page to a one-dimensional text sequence. Graph convolution for multimodal information extraction showed that relative positions among text segments, table regions, and visual layout relations directly affect entity recognition [3]. This result motivates representing a document as a structured object with textual, visual, and layout relations.

FormNet uses Rich Attention and Super-Tokens to recover local structure around tokens, reducing the information loss caused by serializing complex forms [5]. BROS strengthens text-layout pretraining for KIE [17]. LayoutLMv2 advances multimodal pretraining across visual, textual, and layout signals, while LayoutLMv3 unifies text and image masking objectives [18, 19].

The same line of work extends to multilingual and unified document modeling. LayoutXLM supports multilingual visually rich document understanding [20]. TILT and DocFormer combine text, images, and layout [21, 22]. StrucTexT and UDOP target structured text understanding and general document processing [23, 24]. These models support VGA-KIE's upstream assumption that field extraction depends on layout relations, but they do not constitute a verification and repair loop.

Layout-aware models elevate layout from an external feature to an internal representation. Their main objective remains the accuracy of entity labels, field values, or key-value pairs. They do not necessarily determine whether a prediction has locatable evidence, whether that evidence satisfies schema and contextual constraints, or whether a structural error can be diagnosed and repaired. VGA-KIE uses the document graph for candidate search, evidence linking, constraint verification, and repair decisions. Layout relations therefore become inputs to reliability decisions as well as encoding features.

2.2 Document language models, prompting, and candidate generation

Large language models extend KIE from specialized taggers to generative document reasoning. DocLLM combines text tokens with bounding boxes and uses disentangled spatial attention and infilling pretraining to reason over complex layouts [7]. LayoutLLM connects visual, textual, and layout information to a language model through instruction tuning for classification, extraction, and question answering [8]. A Bounding Box is Worth One Token represents each bounding box as a language-model token [25]. LayTokenLLM further develops layout tokens as a spatial representation for document language models [26].

Other methods improve extraction without training a new foundation model. SAIL selects in-context examples using entity-level textual and layout similarity in a training-free setting [27]. Its results show that layout-similar examples can help a language model distinguish entities with similar geometry but different meanings. Language priors alone are therefore insufficient; document extraction also depends on structure-sensitive context.

These methods show that LLM and MLLM extraction benefits from explicit document structure. Their generated outputs nevertheless remain candidate hypotheses. Valid JSON does not ensure that a field comes from the correct region, satisfies the schema, or remains consistent with related fields. VGA-KIE places LLMs and MLLMs at the candidate-generation interface. Grounding, verification, and repair then determine which candidates enter the structured output.

2.3 OCR-free multimodal understanding and evidence localization

OCR-free document understanding generates structured answers directly from document images. Donut shows that end-to-end image-to-text modeling can reduce the propagation of external OCR errors [9]. DocParser applies this approach to information extraction from visually rich documents [28]. mPLUG-DocOwl 1.5 emphasizes unified structure learning [10], and mPLUG-DocOwl2 targets high-resolution compression and multi-page document understanding [29].

Evaluations of GPT-4V and OCRBench show that general multimodal models remain limited on complex OCR, table-structure recognition, and document information extraction [30, 31]. Nougat and Pix2Struct represent neural academic-document parsing and screenshot parsing, respectively [32, 33]. Image-to-structure generation is an important route, but it does not replace field-level evidence verification.

OCR-free generation also exposes a sharper reliability problem: a fluent field value can lack an auditable evidence chain. See then Tell uses visual grounding to localize the relevant image region before answering [11]. LMDX identifies missing layout encoding and grounding as central obstacles to document extraction with language models, then uses segment identifiers to support entity localization and hallucination filtering [12]. These methods make region-level localization a core requirement for document extraction.

VGA-KIE follows the evidence-localization principle and assigns grounding a broader role. Evidence is an input to verification rather than an attached visualization. A candidate must satisfy text matching, layout-path, schema-compatibility, cross-field, and table-structure constraints. Grounding failure, spatial mismatch, unsupported value, and table-alignment error can therefore condition different repair actions.

2.4 Real-world evaluation and field-level reliability

High scores on standard KIE benchmarks do not directly imply reliable extraction from real documents. EC-FUNSD shows that models can exploit spurious correlations when annotation granularity differs from real semantic entities [15]. DocVQA extends document-image understanding to visual question answering, where layout, text, and semantic reasoning interact [34]. Kleister targets long documents with complex layouts[35]. RealKIE adds enterprise documents, sparse fields, and realistic OCR noise [13]. UNIKIE-BENCH reports that schema variation, long-tail fields, and complex layouts reduce multimodal-model performance, while grounding accuracy and layout-aware reasoning remain open challenges [14].

This evidence shifts evaluation from asking which model has the highest F1 to asking why a model fails and whether the failure can be detected. Most benchmarks emphasize field exact match, entity F1, or document accuracy. They less often report evidence localization, verification-failure detection, repair success, and harmful repair. KIEval likewise argues that KIE quality requires more detailed evaluation than a single field-matching score [36]. In receipts, contracts, purchase orders, and financial documents, an erroneous field without evidence or a failure state can be riskier than an omitted field.

VGA-KIE complements existing KIE methods after candidate generation. Layout-aware models supply structured candidates [3]–[5], grounded extraction links values to document regions [11, 12], and real-world benchmarks reveal reliability failures under complex layouts and long-tail schemas [13]–[15]. VGA-KIE adds a unified transition rule across these functions: evidence determines whether a field has a diagnosable state, the failure type restricts the permitted repair actions, and re-verification determines whether the repaired state is released, rejected, or withheld as uncertain.

3 Task Definition

3.1 Atomic field instances and the initial candidate interface

VGA-KIE is a post-hoc reliability layer for KIE from visually rich documents. Let the input document be $D = (I_1, \ldots, I_P)$, where I_p is the image of page p, P is the page count, and a single-page document has $P = 1$. The jth atomic field request is $q_j = (D, s_j, \kappa_j)$. Here, κ_j is a reproducible instance key. A single-valued field uses $\kappa_j = 1$, while repeated entities, table rows, and hierarchical fields use an entity index, row index, or canonical structural path. The atomic request sequence is $Q_D = (q_1, \ldots, q_m)$. Multiple instances of one schema field use different κ_j values, allowing the same definition to represent fixed, repeated, and hierarchical fields.

The parser converts the document into the multigranular observations $X = \mathcal{P}(D) = (X_{\text{tok}}, X_{\text{line}}, X_{\text{block}}, X_{\text{cell}}, X_{\text{vis}})$. In an OCR-based setting, token i is $x_i^{\text{tok}} = (w_i, b_i, o_i, p_i)$. Its four components are the recognized text, bounding box $b_i = (x_i^1, y_i^1, x_i^2, y_i^2)$, OCR score, and page index, with $o_i \in [0, 1]$. Text lines, text blocks, table cells, and reading order form the structural observations. Visual regions or proposals from OCR-free models enter the visual-observation component. The interface can receive token-level layout analysis [37], table-structure parsing [38, 39], and pretrained OCR [40], but treats every parser output as a noisy observation rather than field truth.

The instantiated schema aligned with the atomic request sequence is $S = (s_1, \ldots, s_m)$. Request $q_j = (D, s_j, \kappa_j)$ uses the field descriptor s_j, defined as $s_j = (n_j, A_j, \tau_j, \Phi_j, \Psi_j, \Pi_j)$. Its six components are the canonical field name, alias set, value type, formatting and normalization constraints, cross-field or table-structure constraints, and spatial priors. Repeated instances can share one descriptor and use κ_j to identify each occurrence.

A replaceable upstream adapter $C_j^0 = \Gamma(D, X, q_j)$, $K_j \leq K_c$ supplies an ordered initial candidate pool for every atomic request. K_j is the actual number of initial candidates for request j, and K_c is the common maximum candidate budget. The minimum representation of candidate l is $c_{jl} = (v_{jl}, B_{jl}, \pi_{jl})$, which stores its value, initial region hint, and upstream ranking score π_{jl}. Section 2.1 adds schema-trigger evidence and local context. When an upstream system returns only one candidate, $K_c = 1$. The adapter normalizes upstream outputs and is not part of the post-hoc operator parameters. Its replaceability establishes interface compatibility, not verified gains across candidate generators.

3.2 Evidence-bearing outputs and release semantics

VGA-KIE builds a document graph G from X and returns $\boldsymbol{y}_j = (v_j, E_j, \rho_j, z_j, r_j, H_j)$ for each atomic request from the locatable evidence space. The tuple contains the normalized field value v_j, supporting evidence $E_j \subseteq E(G)$, final reliability score $\rho_j \in [0, 1]$, verification state $z_j \in \{\text{pass}, \text{fail}, \text{uncertain}\}$, dominant failure reason $r_j \in R_{\text{fail}} \cup \{\text{none}\}$, and repair history H_j. The upstream score π_{jl}, grounding support score g_{jl}, and final score ρ_j serve candidate ranking, evidence gating, and release decisions, respectively. None is interpreted as a probability of correctness unless calibrated separately.

Each item in E_j satisfies $e_{jk} \in E_j$, $1 \leq k \leq K_j^e$. The evidence item $e_{jk} = (p_{jk}, b_{jk}, x_{jk}, a_{jk})$ records the page index, evidence region, evidence text or visual-region description, and source type. Source types include token span, line, block, table cell, key-value path, and image region. Evidence must be relocatable through document coordinates or a structural path. A language-model explanation without a locatable region is not valid provenance.

For a candidate-evidence pair, the grounding gate produces the internal support label $u_{jl} \in \{\text{supported}, \text{unsupported}\}$. The unsupported label means that the current candidate cannot be linked to evidence that satisfies the localization requirements. It is an intermediate candidate label rather than a final verification state z_j. It can

trigger evidence expansion, candidate replacement, or local repair, but cannot directly trigger release.

Each final state has one action semantics. Pass means that the value, evidence, and constraints satisfy the release conditions. Fail denotes a definite violation of a critical constraint or the absence of an acceptable candidate after exhausting the repair budget. Uncertain denotes evidence insufficient for release but insufficient for a definite failure decision. The downstream action is $\text{Release}(\boldsymbol{y}_j) = (v_j, E_j, \rho_j)$ if $z_j = \text{pass} \wedge v_j \neq \emptyset \wedge E_j \neq \emptyset$ for pass and $\text{Release}(\boldsymbol{y}_j) = \emptyset$ if $z_j \in \{\text{fail}, \text{uncertain}\}$ for fail or uncertain. Fail rejects the field and uncertain abstains. Both retain internal candidates, failure reasons, and repair trajectories for auditing or human review.

3.3 Inference objective and evaluation boundary

Let the initial candidate pools for all atomic requests be $C^0 = (C_1^0, \ldots, C_m^0)$ and the verification-rule family be Ω. For request j, conditional repair forms the feasible candidate domain $\tilde{C}_j = C_j^0 \cup R_j(C_j^0, G, \Omega)$, $K_j^r \leq K_r$ under budget K_r, where K_j^r is the actual number of repair candidates. The core post-hoc reliability operator is

$$\boldsymbol{Y} = F(D, X, S, C^0, \Omega; \Theta) = (y_1, \ldots, y_m)$$

The parameter set Θ contains only the internal parameters for graph construction, evidence linking, verification gating, and the repair policy. It excludes parser $\mathcal{P}$ and initial candidate adapter Γ. The full framework first uses $\mathcal{P}$ and Γ to produce X and the initial candidates, then uses F to select or repair candidate-evidence pairs. A paired experiment with a shared frozen candidate cache estimates the post-hoc contribution of F without attributing candidate-generation quality to that contribution.

For evaluation, a frozen protocol maps annotations and predictions to the atomic key (D, s, κ) and constructs the test inventory Q. Single-valued fields use fixed instance keys, repeated entities use entity or row keys, and hierarchical fields use canonical structural paths. An unmatched gold instance is a missing prediction, and an unmatched predicted instance is a false positive. The release set is $A = \{q_j \in Q : z_j = \text{pass} \wedge v_j \neq \emptyset \wedge E_j \neq \emptyset\}$. Field-F1 treats $z_q \in \{\text{fail}, \text{uncertain}\}$, $v_q = \emptyset$, and empty evidence as missing predictions. Acceptance Coverage reports the released scope separately. Section 5.4 specifies the mapping rules and denominators.

The internal label u_{jl} controls grounding and repair only. Unsupported Output Rate is independently audited over released, non-empty outputs by a frozen external evidence matcher and does not reuse that label. An initially wrong field that becomes correct with external evidence is wrong→correct. An initially correct field that becomes wrong after post-hoc processing is correct→wrong. These transitions define Repair Success and False Repair Rate in Section 5.

4 VGA-KIE Method

4.1 From one-shot extraction to verification-constrained inference

Section 3 defines VGA-KIE as a post-hoc reliability layer over atomic field requests. Given document $D = (I_1, \ldots, I_P)$, multigranular observations X, target schema $S = (s_1, \ldots, s_m)$, initial candidate pools $C^0 = (C_1^0, \ldots, C_m^0)$, and verification rules Ω, the core operator returns $\boldsymbol{y}_j = (v_j, E_j, \rho_j, z_j, r_j, H_j)$ for each request. A field is released only when its evidence is non-empty and relocatable, its constraints are testable, and its state is pass. Fail causes definite rejection and uncertain causes abstention; both retain diagnostic information.

The post-hoc reliability operator is

$$\boldsymbol{Y} = F(D, X, S, C^0, \Omega; \Theta) = (y_1, \ldots, y_m).$$

Θ collects the internal parameters for graph construction, evidence linking, verification gating, and the repair policy. Parser $\mathcal{P}$ and initial candidate adapter Γ remain replaceable upstream components. The full system first produces X and the initial candidate pools, then executes Graph Build, Ground, Verify, Repair, and Verify Again. Candidate values come from the external interface; the document graph, evidence state, and repair transitions belong to F. The paired experiment with a shared frozen candidate cache estimates only the post-hoc increment of F.

$$\begin{aligned} Y^* = \underset{Y \in \mathcal{Y}(D,S)}{\arg\max} \sum_{j=1}^{m} \Big[& \lambda_c S_c\big(c_j \mid G, s_j\big) \\ & + \lambda_g S_g\big(c_j, E_j \mid G, s_j\big) + \lambda_v S_v\big(c_j, E_j, s_j; \Omega\big) - \lambda_h C_{\mathrm{rep}}(H_j) \Big] \end{aligned} \tag{1}$$

$$\text{s.t.} \quad c_j \in C_j, \;\; E_j \subseteq \varepsilon(G), \;\; z_j = V_z(c_j, E_j, s_j; \Omega), \;\; j = 1, \ldots, m. \tag{2}$$

In Equations (1) and (2), S_c measures candidate-generation compatibility, S_g measures grounding support from document evidence, S_v measures verification-rule satisfaction, and C_{rep} measures repair cost or trajectory complexity. The weights $\lambda_c, \lambda_g, \lambda_v, \lambda_h$ control their relative contributions. The objective combines candidate score, locatable evidence, constraint satisfaction, and interpretable repair. Candidate generation does not decide final acceptance by itself.

The three mechanisms form a functional dependency chain. The evidence-grounded field state defines a diagnosable object, typed failure diagnosis supplies the condition for action selection, and re-verification decides the post-repair transition. Equations (1)–(14) define the state, structure, and evidence interfaces; Equations (15)–(18) define verification gating; and Equations (19)–(29) define repair transitions and stopping. The common candidate interface provides architectural compatibility but does not establish stable gains across candidate generators.

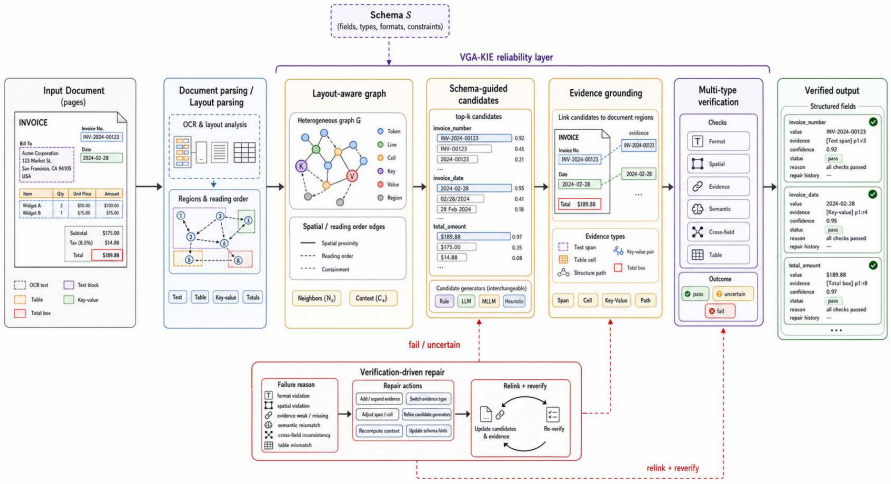


Fig. 2 VGA-KIE framework. Replaceable interfaces supply parsed observations and an initial candidate pool. The candidate module denotes interface adaptation and candidate caching, not a newly validated general candidate generator. The core reliability operator constructs a layout-aware graph, links candidates to textual, tabular, or visual evidence, and derives states and failure reasons from format, spatial, evidence, semantic, cross-field, and table-structure constraints. Conditional repair expands the local candidate domain only for fail or uncertain requests, and every repaired candidate undergoes renewed grounding and verification.

4.2 Multigranular OCR and layout observations

The parser converts a visually rich document into $X = \mathcal{P}(D) = (X_{\text{tok}}, X_{\text{line}}, X_{\text{block}}, X_{\text{cell}}, X_{\text{vis}})$. In an OCR-based setting, page I_p contains the token sequence $X_{\text{tok}} = (x_1^{\text{tok}}, \ldots, x_{N_p}^{\text{tok}})$, where $x_i^{\text{tok}} = (w_i, b_i, o_i, p_i)$. The normalized bounding box is $b_i = (x_i^1, y_i^1, x_i^2, y_i^2)$, the OCR score satisfies $o_i \in [0, 1]$, and p_i is the page index. OCR-free models do not need to fabricate tokens; their visual regions and proposals enter X_{vis}. Graph construction, grounding, and verification then share the same observation interface.

For a raw bounding box $b_i^{\text{raw}} = (x_i^1, y_i^1, x_i^2, y_i^2))$, the normalized geometric representation is

$$
\begin{aligned}
b_i &= \left(\frac{x_i^1}{W_p}, \frac{y_i^1}{H_p}, \frac{x_i^2}{W_p}, \frac{y_i^2}{H_p} \right), \\
a_i &= (x_i^c, y_i^c, \hat{w}_i, \hat{h}_i), \\
x_i^c &= \frac{x_i^1 + x_i^2}{2W_p}, \quad y_i^c = \frac{x_i^1 + y_i^2}{2H_p}, \\
\hat{w}_i &= \frac{x_i^2 - x_i^1}{W_p}, \quad \hat{h}_i = \frac{y_i^2 - y_i^1}{H_p}.
\end{aligned}
\tag{3}
$$

Text lines and blocks follow from geometry and reading order rather than string concatenation. Tokens i and k enter the same candidate line when they satisfy

vertical-overlap, baseline-distance, and horizontal-gap conditions. Indentation, vertical spacing, and column structure then group lines into text blocks.This process produces the partial reading-order relation $\prec_r$.A partial order represents tables, multi-column layouts, receipt regions, and footers more accurately than one global sequence.

$$i \sim_{\text{line}} k \iff \text{IoU}_y(b_i, b_k) \geq \delta_y \ \wedge \ |\text{base}(i) - \text{base}(k)| \leq \delta_b \ \wedge \ \text{gap}_x(i,k) \leq \delta_x. \quad (4)$$

The relation $l_a \prec_r l_b$) depends on page index, the two-dimensional ordering function order_{2D}, and column structure. It is therefore a partial order rather than a one-dimensional concatenation.

Table regions are represented by the cell set $X_{\text{cell}} = \{(u, \text{row}(u), \text{col}(u), b_u, T_u)\}$. When an external parser supplies table boundaries, row and column indices, or merged-cell relations, these structures enter the graph directly. When it supplies only text tokens, VGA-KIE estimates candidate cells from ruling lines, alignment patterns, and repeated spacing. Detected boxes, attention regions, or visual proposals from OCR-free and visual multimodal models can form X_{vis}, but each field candidate must still map to a traceable page region.

The parser outputs an uncertain observation layer rather than plain text. OCR errors, omissions, table-structure mismatches, and region offsets remain available to later verification and repair. This design reflects real document-analysis settings, where parsing, layout, and semantic errors interact.

4.3 A layout-aware heterogeneous document graph

The core operator builds the layout-aware heterogeneous document graph $G = (V, L, \phi, \psi, X_V, W)$. V is the node set, L is the link set, $\phi : V \to A_v$ and $\psi : L \to A_e$ assign node and link types, $X_V = (x_u : u \in V)$ is the node-attribute family, and W contains relation weights. Nodes represent OCR tokens, text lines, text blocks, table cells, candidate keys, candidate values, page regions, and visual proposals. Equation (5) uses o_u for OCR information. Token nodes retain their original OCR score, line, block, and cell nodes use an aggregate over their tokens, and visual or non-text nodes use an explicit missing marker. X_V denotes graph-node attributes and remains distinct from document-level observations X.

$$\begin{aligned} V &= V_{\text{tok}} \cup V_{\text{line}} \cup V_{\text{block}} \cup V_{\text{cell}} \cup V_{\text{key}} \cup V_{\text{val}} \cup V_{\text{reg}}, \\ x_u &= [\text{emb}_{\text{txt}}(u); \text{emb}_{\text{geo}}(b_u); \text{emb}_{\text{type}}(\phi(u)); \text{emb}_{\text{page}}(p_u); \text{emb}_{\text{ocr}}(o_u)]\,. \end{aligned} \quad (5)$$

To avoid collision with the field evidence set E_j, graph links are partitioned by relation type as $L = L_{\text{spa}} \cup L_{\text{read}} \cup L_{\text{hier}} \cup L_{\text{tab}} \cup L_{\text{sem}} \cup L_{\text{sch}}$. Spatial links encode left-of, right-of, above, below, near, and overlap. Hierarchical links connect token, line, block, cell, and page nodes. Table links represent same-row, same-column, same-cell, and header-to-value relations. Schema links connect field aliases to potential key and

value triggers.

$$
\begin{aligned}
w_{uv}^{(r)} &= \sigma\big(\gamma_r^\top f_r(u,v)\big), \\
f_r(u,v) &= \big[\mathrm{IoU}(b_u,b_v); \exp\big(-d_{\mathrm{geo}}(u,v)\big); \mathrm{sim}_{\mathrm{txt}}(u,v); \mathbb{1}\big[\psi(u,v)=r\big]\big].
\end{aligned}
\tag{6}
$$

For each field s_j, VGA-KIE constructs the schema-conditioned h-hop neighborhood $N_h(s_j)$ to obtain field-relevant local context. The key nodes that match field aliases are

$$
O_j = \big\{u \in V_{\mathrm{key}} : \mathrm{sim}\big(\mathrm{alias}(u), A_j\big) \geq \tau_{\mathrm{alias}}\big\}. \tag{7}
$$

$$
N_h(s_j) = \big\{u \in V : \min_{k \in O_j} \mathrm{dist}_G(u,k) \leq h\big\}. \tag{8}
$$

The symbol h denotes neighborhood radius and remains distinct from failure reason r_j. The neighborhood restricts candidate search to regions related to field semantics and expected layout while preserving relations across lines, cells, and blocks.

When the downstream model supports graph encoding, relation-aware message passing can update node representations. When the candidate generator is an LLM or MLLM, the graph remains a retrieval, constraint, and verification layer. The structural context at layer l is

$$
\begin{aligned}
h_u^{(l+1)} &= \sigma\left(W_0 h_u^{(l)} + \sum_{r \in A_e} \sum_{v \in N_r(u)} \alpha_{r,u,v} W_r h_v^{(l)}\right), \\
\alpha_{r,u,v} &= \underset{v \in N_r(u)}{\mathrm{softmax}}\left(a^\top \left[h_u^{(l)}; h_v^{(l)}; \mathrm{emb}(r); \Delta b_{u,v}\right]\right).
\end{aligned}
\tag{9}
$$

This representation makes layout a computable and query able structure. Candidate extraction, evidence linking, and verification operate over the same graph rather than maintaining separate document representations.

4.4 Schema-guided candidates preserve alternative hypotheses

For atomic request $q_j = (D, s_j, \kappa_j)$, adapter Γ supplies the traceable initial ranked candidate pool $C_j^0 = \Gamma(D, X, q_j)$, $K_j \leq K_c$. Candidate l is extended to $c_{jl} = (v_{jl}, B_{jl}, \pi_{jl}, a_{jl}, \xi_{jl})$, whose five components are the field value, candidate region or region set, generator score, schema or key trigger evidence, and a local context summary. K_j is the actual candidate count, K_c is the maximum candidate budget, and K_r is the repair budget. Γ retains ranked candidates within the initial budget, exposing upstream uncertainty. VGA-KIE uses the repair budget to expand the candidate domain only under a diagnosed failure condition. When the top candidate fails evidence or constraint tests, a lower-ranked or locally repaired candidate enters re-verification.

Schema $s_j = (n_j, A_j, \tau_j, \Phi_j, \Psi_j, \Pi_j)$ supplies semantic aliases, a field type, formatting rules, cross-field constraints, and spatial priors. A_j matches key phrases. τ_j defines the type domain, including dates, amounts, addresses, and identifiers. Φ_j specifies normalization and format constraints. Ψ_j declares cross-field dependencies such

as total $\approx$ subtotal + tax $-$ discount, and Π_j declares key-value relations such as right-of-key, below-key, and same-row-in-table.

$$\begin{aligned} C_j &= \mathrm{TopK}_{K_c}\big\{c \in C_j^0 \,\big|\, q_\theta(c \mid G, s_j; N_h(s_j))\big\}, \\ q_\theta(c \mid G, s_j; N_h(s_j)) &\propto \exp\Big\{ f_\theta(c, G, s_j; N_h(s_j)) + \mu_1 \kappa_{\mathrm{type}}(c, \tau_j) + \mu_2 \kappa_{\mathrm{geo}}(c, \Pi_j) \\ &\quad + \mu_3 \kappa_{\mathrm{alias}}(c, A_j) \Big\}. \end{aligned} \tag{10}$$

The candidate function f_θ can use graph retrieval, sequence labeling, a layout-aware LLM, an MLLM, or a heuristic extractor. $\kappa_{\mathrm{type}}, \kappa_{\mathrm{geo}}, \kappa_{\mathrm{alias}}$ measures type consistency, agreement with spatial priors, and alias-trigger consistency. The candidate generator proposes hypotheses; grounding and verification filter them.

Table fields also require row and column consistency. When field s_j is a line-item attribute, the candidate is the tuple $c = (v, \mathrm{row}, \mathrm{col}, B)$ rather than one value span. Its score combines cell text, column-header semantics, and same-row entity relations to prevent values from adjacent or summary rows from entering the target field.

$$\begin{aligned} S_{\mathrm{tab}}(c, s_j) &= \eta_1 \,\mathrm{sim}(\mathrm{col_header}(c), A_j) + \eta_2 \,\mathrm{coh_row}(c) + \eta_3 \,\mathrm{sim_cell}(v_c, \tau_j) \\ &\quad + \eta_4 \,\mathrm{rel_table}(c, \Psi_j), \quad \eta_l \geq 0, \ \sum_{l=1}^{4} \eta_l = 1. \end{aligned} \tag{11}$$

4.5 Evidence-grounded field states link values to page regions

Evidence grounding separates VGA-KIE from ordinary generative KIE. For candidate c_{jl}, the framework must find at least one supporting region $E_{jl} \subseteq E(G)$ in evidence space $E(G)$. Evidence can be a token span, text line, text block, table cell, key-value path, or image crop. VGA-KIE defines grounding as joint alignment among a candidate value, field schema, and document region, not as a bounding box attached after extraction.

$$\begin{aligned} E_j^{(l^*)} &= \operatorname*{arg\,max}_{E \subseteq E(G),\, |E| \leq M} S_g\big(c_{jl}, E, s_j \mid G\big), \\ S_g(c, E, s_j) &= \log \sum_{e \in E} \exp\big(S_e(c, e, s_j)\big) - \zeta \Omega_{\mathrm{red}}(E). \end{aligned} \tag{12}$$

The term $\Omega_{\mathrm{red}}(E)$ penalizes redundant or conflicting evidence regions. Each evidence score combines textual, spatial, and semantic agreement with schema compatibility and structural-path consistency:

$$\begin{aligned} S_e(c, e, s_j) &= \omega_1 M_{\mathrm{txt}}(v_c, T_e) + \omega_2 M_{\mathrm{box}}(B_c, B_e) + \omega_3 M_{\mathrm{sem}}(c, e) + \omega_4 M_{\mathrm{sch}}(e, s_j) \\ &\quad + \omega_5 M_{\mathrm{path}}(c, e, G), \quad \omega_i \geq 0, \ \sum_{i=1}^{5} \omega_i = 1. \end{aligned} \tag{13}$$

M_{txt} measures normalized agreement between candidate and evidence text, and M_{box} measures geometric overlap or proximity. M_{sem} measures semantic similarity, M_{sch}

tests field-type and alias conditions, and M_{path} measures path support between the candidate and a key, table header, or contextual field. Evidence for an invoice total, for example, should contain the amount and have a plausible layout path to a key such as total, amount due, or grand total.

Candidate-evidence grounding support is $g_{jl} = \sigma\big(S_g(c_{jl}, E_{jl}, s_j)\big)$. When $g_{jl} < \tau_g$, VGA-KIE assigns $u_{jl} =$ unsupported, and the candidate cannot transition directly to pass. When text matching is weak but visual regions, table paths, or cross-field constraints provide partial support, the verifier can return uncertain. Internal label u_{jl} controls candidate filtering and repair only. Unsupported Output Rate is independently audited over released outputs by a frozen external evidence matcher.

When a dataset provides entity boxes or field-region annotations, the grounding module can use supervised training. Without region labels, it can use weak supervision or heuristic alignment. A supervised grounding loss is

$$\begin{aligned} L_{\text{ground}} &= L_{\text{pos}} + \lambda_{\text{neg}} L_{\text{rank}}, \\ L_{\text{pos}} &= -\sum_{c^\star, s^\star, e^+} \log p(e^+ \mid c, s), \quad p(e \mid c, s) = \operatorname*{softmax}_{e \in E(G)} S_e(c, e, s), \\ L_{\text{rank}} &= \sum_{e^-} \max\big(0,\, m_{\text{rank}} + S_e(c, e^-, s) - S_e(c, e^+, s)\big). \end{aligned} \tag{14}$$

4.6 Typed failure diagnosis tests multiple constraints

The verifier maps candidate c, evidence E, and field schema s_j to $\boldsymbol{V}(c, E, s_j; \Omega) = \big(z(c), r(c), \rho(c)\big)$. Here, z is the discrete verification state, r is the dominant failure reason, and ρ is the final reliability score. VGA-KIE formulates verification as multi-constraint consistency. Every rule $\omega_m \in \Omega$ produces a normalized satisfaction value $R_m(c, E, s_j) \in [0, 1]$ and violation magnitude $\Delta_m(c, E, s_j) = 1 - R_m(c, E, s_j)$.

$$\begin{aligned} R(c, E, s_j) &= \big[R_{\text{fmt}}, R_{\text{spa}}, R_{\text{evi}}, R_{\text{sem}}, R_{\text{cross}}, R_{\text{tab}}\big], \quad \Delta_m(c, E, s_j) = 1 - R_m(c, E, s_j), \\ S_v(c, E, s_j; \Omega) &= \sum_{m=1}^{6} \beta_m R_m(c, E, s_j) - P_{\text{pair}}(c, E, s_j), \\ P_{\text{pair}} &= \sum_{1 \le m < n \le 6} \beta_{m,n} \max\{0, \Delta_m + \Delta_n - \epsilon\}, \\ \beta_m &\ge 0,\ \beta_{m,n} \ge 0, \quad \sum_{m=1}^{6} \beta_m = 1. \end{aligned} \tag{15}$$

The first term aggregates rule satisfaction, and the second penalizes compound errors. A spatial mismatch combined with missing evidence, for example, often identifies the wrong field rather than simple formatting noise. Format verification checks field types and normalization rules for dates, amounts, identifiers, and addresses. Spatial verification tests whether a value occurs in a schema-permitted key neighborhood, line, lower region, or table cell. Evidence verification requires support from an OCR span, cell, or visual region. Semantic verification tests field aliases against local context. Cross-field verification checks arithmetic, temporal, or entity constraints. Table verification checks rows, columns, header-value relations, and line-item structure. Table 1 summarizes the six rule types.

Table 1 Verification types, representative constraints, and detectable failures in VGA-KIE.

Verification type	Representative constraint	Detectable failure
Format verification	The value must satisfy the field type and its normalization pattern.	Format or normalization error
Spatial verification	The value must occur in a schema-permitted spatial relation to its key.	Key-value spatial mismatch
Evidence verification	The candidate must be supported by an aligned OCR span, table cell, or image region.	Unsupported value
Semantic verification	The local context must match a permitted semantic alias of the target field.	Semantic conflict
Cross-field verification	Related fields must satisfy declared arithmetic, temporal, or entity constraints.	Cross-field inconsistency
Table-structure verification	The value must occur in the correct table row, column, or line item.	Table-alignment error

The verification state follows a dual-threshold rule with grounding gating:

$$z(c) = \begin{cases} \text{pass}, & S_v(c, E, s_j; \Omega) \geq \tau_{\text{pass}} \wedge g(c) \geq \tau_g \wedge \max_m \Delta_m \leq \delta_{\text{safe}}, \\ \text{fail}, & S_v(c, E, s_j; \Omega) < \tau_{\text{fail}} \vee g(c) < \tau_{\min} \vee \max_m \Delta_m > \delta_{\text{crit}}, \\ \text{uncertain}, & \text{otherwise.} \end{cases} \tag{16}$$

The dominant failure reason is assigned only to fail or uncertain states and equals the largest weighted violation:

$$r(c) = \begin{cases} \text{none}, & z(c) = \text{pass}, \\ \underset{m \in \{1,\ldots,6\}}{\arg\max}\ \beta_m \Delta_m(c, E, s_j), & z(c) \neq \text{pass}. \end{cases} \tag{17}$$

The verification output conditions the repair policy. When $r(c) =$ spatial mismatch, repair expands the graph neighborhood or relocates the key-value path. When $r(c) =$ unsupported value, it repeats grounding or inspects an OCR-free visual region. When $r(c) =$ table alignment error, it re-estimates row and column structure.

The final field reliability score ρ combines generator score, evidence support, rule satisfaction, and repair cost:

$$\begin{aligned} \rho(c) = \sigma\Big(&\lambda_1 \log \pi(c) + \lambda_2 S_g(c, E^* \mid G, s_j) + \lambda_3 S_v(c, E^*, s_j; \Omega) \\ &- \lambda_4 |H(c)| - \lambda_5 U_{\text{parse}}(c) - \lambda_6 \mathbb{1}\big[z(c) \neq \text{pass}\big]\Big). \end{aligned} \tag{18}$$

$|H(c)|$ is repair-trajectory length, $U_{\text{parse}}(c)$ measures parsing uncertainty such as low OCR score, unstable region detection, or conflicting table boundaries, and $\mathbb{1}\big[z(c) \neq$

pass] penalizes a candidate that has not passed verification. This score distinguishes a field supported by clear initial evidence from one that passes only after repair.

4.7 Failure-conditioned repair with re-verification

The repair component applies targeted correction only to fail or uncertain fields. It treats repair as a field-level state update rather than rerunning extraction over the whole document. The dominant failure reason selects the action, and every updated candidate must undergo renewed evidence linking and verification.

Available actions are Relocate, ReOCR, Rerank, Reextract, and Normalize. Relocate searches nearby or schema-related regions. ReOCR handles incomplete text or weak OCR scores within an evidence region. Rerank orders existing candidates by verification and evidence scores. Reextract carries the failure reason and current evidence into extraction for the failed field only. Normalize standardizes dates, amounts, identifiers, telephone numbers, and addresses.

Let the state of field s_j before repair step t be

$$\chi_j^{(t)} = \big(C_j^{(t)}, c_j^{(t)}, E_j^{(t)}, z_j^{(t)}, r_j^{(t)}, H_j^{(t)}\big). \tag{19}$$

$C_j^{(t)}$ is the current candidate set, $c_j^{(t)}$ is the selected candidate, and $E_j^{(t)}$ is its linked evidence. $z_j^{(t)} \in \{\text{pass}, \text{fail}, \text{uncertain}\}$ is the verification state, $r_j^{(t)}$ is the dominant failure reason, and $H_j^{(t)}$ is the field-level repair history. The repair-action set is

$$A_{\text{rep}} = \big\{\text{Relocate}, \text{ReOCR}, \text{Rerank}, \text{Reextract}, \text{Normalize}\big\}. \tag{20}$$

Action selection depends on the failure reason, verification state, and recent gain:

$$a_j^{(t)} = \pi_{\text{rep}}\big(r_j^{(t)}, z_j^{(t)}, \Delta_v^{(t)}, H_j^{(t)}\big), \quad a_j^{(t)} \in A_{\text{rep}}. \tag{21}$$

$\Delta_v^{(t)}$ denotes the verification-score change from the latest repair. Given action $a_j^{(t)}$, the candidate set is updated locally:

$$\Delta_v^{(t)} = S_v\big(c_j^{(t)}, E_j^{(t)}, s_j; \Omega\big) - S_v\big(c_j^{(t-1)}, E_j^{(t-1)}, s_j; \Omega\big), \quad \Delta_v^{(0)} = 0. \tag{22}$$

$$C_j^{(t+1)} = R_{a_j^{(t)}}\big(C_j^{(t)}, G, s_j, E_j^{(t)}, r_j^{(t)}\big). \tag{23}$$

The system then repeats evidence linking and verification over the updated candidates:

$$E_c^{(t+1)} = \underset{E \subseteq E(G),\, |E| \le M}{\arg\max} S_g\big(c, E, s_j \mid G\big), \quad c \in C_j^{(t+1)}. \tag{24}$$

$$\big(z_c^{(t+1)}, r_c^{(t+1)}, \rho_c^{(t+1)}\big) = \boldsymbol{V}\big(c, E_c^{(t+1)}, s_j; \Omega\big), \quad c \in C_j^{(t+1)}. \tag{25}$$

The final candidate follows the unified objective rather than generator score alone:

$$\begin{aligned}\Phi_j^{(t+1)}(c) = \lambda_c S_c(c \mid G, s_j) + \lambda_g S_g\big(c, E_c^{(t+1)}, s_j\big) + \lambda_v S_v\big(c, E_c^{(t+1)}, s_j; \Omega\big) \\ - \lambda_h \big|H_c^{(t+1)}\big| - \lambda_u \mathbb{1}\big[z_c^{(t+1)} \neq \text{pass}\big].\end{aligned} \tag{26}$$

$$c_j^{(t+1)} = \arg\max_{c \in C_j^{(t+1)}} \Phi_j^{(t+1)}(c). \tag{27}$$

The weights satisfy $\lambda_c, \lambda_g, \lambda_v, \lambda_h, \lambda_u \geq 0$. S_c, S_g, S_v contains candidate-generation compatibility, evidence support, and verification satisfaction. $\big|H_c^{(t+1)}\big|$ is the repair-history length of candidate c, and $\mathbb{1}\big[z_c^{(t+1)} \neq \text{pass}\big]$ penalizes a candidate that has not passed. The objective evaluates candidate quality, evidence support, rule satisfaction, and repair cost together.

The loop stops when the field passes verification, reaches repair budget K_r, or produces no verification gain for two consecutive repairs:

$$\text{Stop}_j^{(t+1)} = 1 \iff z_j^{(t+1)} = \text{pass} \ \vee \ (t+1 \geq K_r) \ \vee \ \big(\Delta_v^{(t+1)} < \epsilon \wedge \Delta_v^{(t)} < \epsilon\big). \tag{28}$$

Every repair attempt is appended to the field history:

$$H_j^{(t+1)} = H_j^{(t)} \cup \big\{(t, a_j^{(t)}, r_j^{(t)}, z_j^{(t+1)}, \rho_j^{(t+1)})\big\}. \tag{29}$$

Algorithm 1 implements atomic-field inference. The parser produces observations and the document graph. The algorithm then iterates over $Q_D = (q_1, \ldots, q_m)$, reads the adapter's budgeted candidate pool, links evidence, and runs multi-type verification. It applies a local repair only when the best candidate is fail or uncertain, then updates the state after renewed linking and verification. The loop performs budgeted candidate search for one atomic request; it does not train an internal candidate generator or unconditionally re-extract the document.

4.8 Repair paths connect failure reasons to state transitions

Fig. 3 maps three failure sources to verifier signals and permitted local actions. Each path begins with an initial candidate and evidence state. Typed failure diagnosis selects re-grounding, relocation, reranking, or joint repair, and re-verification moves the candidate to pass, fail, or uncertain.

A semantic conflict triggers schema-aware reranking when the candidate disagrees with field semantics or cross-field constraints. Relocation, neighbor reranking, or row and column relinking narrows the search after a spatial or table-alignment error. An unsupported value requires renewed grounding and remains uncertain when no evidence can be established. Evidence relinking and re-verification gate every update, making the failure reason and repair history part of the field output rather than a post-hoc explanation.

Algorithm 1 Verification-guided atomic-field inference in VGA-KIE.

Require: D, $Q_D = \{q_j = (D, s_j, \kappa_j)\}_{j=1}^{m}$, P, Γ, Ω, K_c, K_r
Ensure: $Y = \{y_j\}_{j=1}^{m}$, $y_j = (v_j, E_j, \rho_j, z_j, r_j, H_j)$

$X \leftarrow P(D)$
$G \leftarrow \mathrm{BuildDocumentGraph}(X)$
for each atomic request $q_j = (D, s_j, \kappa_j) \in Q_D$ **do**
  $C_j^{(0)} \leftarrow \mathrm{TopK}_{K_c}\big(\Gamma(D, X, q_j)\big)$
  for each candidate $c \in C_j^{(0)}$ **do**
    $E_c^{(0)} \leftarrow \arg\max_{E \subseteq E(G),\, |E| \le M} S_g(c, E, s_j \mid G)$
    $\big(z_c^{(0)}, r_c^{(0)}, \rho_c^{(0)}\big) \leftarrow \boldsymbol{V}(c, E_c^{(0)}, s_j; \Omega)$
  end for
  $\big(c_j^{(0)}, E_j^{(0)}, z_j^{(0)}, r_j^{(0)}, \rho_j^{(0)}\big) \leftarrow \mathrm{SelectBest}\big(C_j^{(0)}, \Phi_j^{(0)}\big)$
  $H_j^{(0)} \leftarrow \emptyset$; $t \leftarrow 0$
  while $\mathrm{Stop}_j^{(t)} = 0$ **do**
    $a_j^{(t)} \leftarrow \pi_{\mathrm{rep}}\big(r_j^{(t)}, z_j^{(t)}, \Delta_v^{(t)}, H_j^{(t)}\big)$
    $C_j^{(t+1)} \leftarrow R_{a_j^{(t)}}\big(C_j^{(t)}, G, s_j, E_j^{(t)}, r_j^{(t)}\big)$
    for each candidate $c \in C_j^{(t+1)}$ **do**
      $E_c^{(t+1)} \leftarrow \arg\max_{E \subseteq E(G),\, |E| \le M} S_g(c, E, s_j \mid G)$
      $\big(z_c^{(t+1)}, r_c^{(t+1)}, \rho_c^{(t+1)}\big) \leftarrow \boldsymbol{V}(c, E_c^{(t+1)}, s_j; \Omega)$
    end for
    $\big(c_j^{(t+1)}, E_j^{(t+1)}, z_j^{(t+1)}, r_j^{(t+1)}, \rho_j^{(t+1)}\big) \leftarrow \mathrm{SelectBest}\big(C_j^{(t+1)}, \Phi_j^{(t+1)}\big)$
    $H_j^{(t+1)} \leftarrow H_j^{(t)} \cup \big\{(t, a_j^{(t)}, r_j^{(t)}, z_j^{(t+1)}, \rho_j^{(t+1)})\big\}$
    $t \leftarrow t + 1$
  end while
  $y_j \leftarrow \mathrm{BuildOutput}\big(c_j^{(t)}, E_j^{(t)}, \rho_j^{(t)}, z_j^{(t)}, r_j^{(t)}, H_j^{(t)}\big)$
end for
return Y

5 Experimental Design and Evaluation Protocol

5.1 Experimental setup and evaluation objectives

Under fixed data splits, field schemas, input processing, normalization rules, and evaluation scripts, the experiments addressed four questions: (1) whether the coupled closed loop improves field-level extraction accuracy; (2) whether evidence states improve the locatability of accepted fields; (3) whether failure conditions improve failure detection and provide effective routes for local repair; and (4) whether re-verification limits harmful rewrites and additional latency while preserving repair gains. Publicly reported values are used only to position the work in the literature. The mechanism claims rely primarily on paired controls and component ablations.

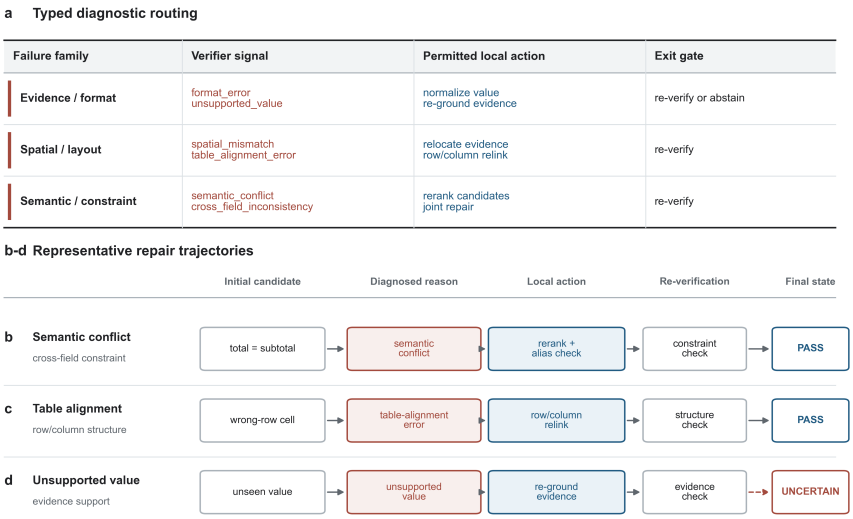


Fig. 3 Typed diagnostic routing and bounded repair in VGA-KIE. a, Verifier signals map each failure family to permitted local actions and an explicit exit gate. b-d, Representative trajectories show repair after semantic conflict, repair after table-alignment error, and an uncertain state for an unsupported value. Every repaired candidate is re-verified, and a candidate is withheld when no supporting evidence region can be established.

5.2 Datasets, task settings, and splits

The core experiments used FUNSD, SROIE, and CORD as a reproducible benchmark combination. These datasets cover noisy scanned forms, fixed-field receipts, and hierarchical receipt semantics, respectively, and therefore represent three characteristic difficulties addressed by VGA-KIE. The key, value, header, and other entities in FUNSD depend strongly on two-dimensional position, reading order, and local adjacency, making the dataset suitable for testing whether the layout graph and schema-conditioned neighborhoods reduce key-value mismatches. The relatively fixed company, address, date, and total fields in SROIE support evaluation of format verification, evidence linking, amount and date normalization, and field-level repair under OCR noise. CORD contains menu items, quantities, prices, and hierarchical structures and is used primarily to evaluate line-item assignment, cross-field consistency, and table-alignment repair.

For all three datasets, each document D_i was treated as an independent resampling unit, and each atomic field instance $q_{ij} = (D_i, s_j, \kappa_{ij})$ defined in Section 3.1 was treated as a scoring unit. Atomic requests within the same document share OCR, layout, and contextual observations and were therefore not treated as independent samples in uncertainty estimation. For every atomic request, the system recorded the field value, evidence, reliability score, verification state, failure reason, and repair history. The primary Field-F1 used the complete target atomic inventory: atoms with a final state of fail or uncertain, or with an empty released value, were treated as missing

predictions and were not removed from the denominator. This rule prevents selective abstention from artificially inflating accuracy.

DocILE, RealKIE, and UNIKIE-BENCH differ from FUNSD, SROIE, and CORD in annotation granularity, input form, field schema, and evaluation protocol and were therefore excluded from the public background comparison in Table 2. DocILE is better suited to evaluating field localization, multi-page structure, and line items in business documents [41]. RealKIE supports analysis of long enterprise documents, sparse fields, and realistic OCR noise [13]. UNIKIE-BENCH supports analysis of open-category fields, long-tail schemas, complex layouts, and grounding and hallucination risks in MLLMs [14].

The core evaluation protocol fixed Original FUNSD (2019), the public KIE processing version of SROIE Task 3, and CORD v2.0.0. The protocol specified the published 149/50 training/test pools for FUNSD, the 626/347 pools for SROIE, and the official 800/100/100 split for CORD. For FUNSD and SROIE, original document identifiers were sorted by SHA-256; the first 15 and 63 documents from the respective training pools formed the validation sets, and the remaining 134 and 563 documents formed the training sets. The official test sets remained unchanged. The experimental records fixed dataset versions, sources, document-file hashes, and split-manifest hashes, and the main text reports results aggregated by a shared evaluation script.

Each split manifest was stored in UTF-8 with LF line endings and one original document identifier per line, together with its SHA-256 hash. The data manifest also recorded the dataset version, source, license, retrieval or snapshot information, the SHA-256 hash of each original file, and a document-level aggregate hash. Before each experiment, the pipeline checked both document-identifier overlap and file-content hashes across the training, validation, and test sets. Dataset-aware field schemas, field mappings, and normalization rules were then frozen. All methods were evaluated on the same field inventory with the same scoring script.

5.3 Comparison methods and technical-route analysis

Table 2 groups representative approaches by candidate-generation and localization strategy. LayoutLMv3 and FormNet are OCR-based layout encoders; Donut and DocParser represent OCR-free generation; DocLLM-7B is a layout-aware generative LLM; LMDX and STNet/See then Tell emphasize localization or grounding; and LayTextLLM, LayTokenLLM-8B, and direct MLLM prompting represent recent layout-token and direct-generation routes.

These public results come from different papers, data-processing pipelines, and experimental protocols and therefore do not constitute a unified ranking. Table 2 is used to characterize the emphasis of each route in layout encoding, generation, and document adaptation. The attribution of VGA-KIE's mechanisms instead relies on Tables 3–5 under the shared protocol, rather than on differences among heterogeneous public values.

Existing routes differ mainly in how they generate and localize field candidates. OCR-based layout encoders use two-dimensional positions and local structure to improve field or entity prediction. OCR-free and generative models reduce dependence

on external OCR but must still address outputs that cannot be localized. Layout-token LLMs strengthen layout-semantic modeling. LMDX and STNet/See then Tell have already introduced localization or visual grounding into KIE [11, 12]; accordingly, this work does not claim novelty merely for providing an evidence region. Based on the mechanisms explicitly described in those publications, their principal objectives are candidate generation or answer localization. VGA-KIE further studies how evidence, failure conditions, and re-verification can be organized as a single field-state transition. This comparison is limited to the mechanisms reported in the cited papers and does not exclude unreported post-processing in their implementations.

The distinction of VGA-KIE is not another candidate encoder, but a change in the field output and decision unit. The evidence-grounded field state binds the value, evidence, and history into one updateable object. Typed failure diagnosis compresses multiple constraint violations into conditions for action routing. Failure-conditioned repair with re-verification requires every repair result to pass through the same evidence and verification gates. The novelty claim concerns the decision coupling among these mechanisms, not priority for each underlying operation in isolation.

To separate candidate quality from the contribution of the post-hoc reliability mechanism, the main experiments fixed LayTokenLLM-8B as the candidate generator and used LayTokenLLM-8B direct extraction as the internal paired control. Both configurations read the same LayTokenLLM-8B checkpoint, input-processing outputs, and frozen top-k candidate cache. Direct extraction returned the highest-ranked candidate and its generator provenance region, whereas LayTokenLLM-8B with VGA-KIE continued with evidence linking, typed verification, targeted repair, and re-verification. The provenance region was used only for external evidence auditing of direct extraction and is not equivalent to VGA-KIE's evidence-linking module. This paired design isolates the contribution of the post-hoc layer at the same candidate interface; the other baselines in Table 3, including LayoutLMv3 and LayTextLLM, do not share this candidate cache.

Table 2 Representative KIE methods and publicly reported F1 scores (%) on FUNSD, SROIE, and CORD. Values are not directly comparable across source protocols.

Method / source	Approach type	FUNSD	SROIE	CORD
LayoutLMv3 [19, 28]	OCR-based layout encoder	92.08	77.70	97.46
FormNet [5, 19]	OCR-based structural encoder	84.69	–	97.28
Donut [9, 28]	OCR-free generation	–	81.70	84.10
DocParser [28]	OCR-free IE	–	87.30	84.50
DocLLM-7B [7]	Layout-aware generative LLM	51.80	91.90	67.40
LMDX [12]	Localization / grounding	–	–	95.57
STNet / See then Tell [11]	Grounded OCR-free KIE	–	88.30	88.80
LayTextLLM [25]	Layout-token LLM	63.30	96.00	97.30
LayTokenLLM-8B [26]	Layout-token LLM	81.62	–	78.30
Direct MLLM prompting [30, 31]	Black-box MLLM	53.20	71.30	78.80

5.4 Main results, reliability, and evaluation metrics

Tables 3–5 use the same test-document inventory, field schemas, OCR and layout parsing products, normalization rules, and evaluation script. Each model received OCR tokens, layout tokens, page images, or their combination according to its interface. The primary paired comparison fixed LayTokenLLM-8B: direct extraction and LayTokenLLM-8B with VGA-KIE shared the same checkpoint, input processing, decoding configuration, and frozen top-k candidate cache. Thresholds and candidate and repair budgets were selected only on training and validation data and frozen before testing.

Evaluation denominator. The frozen protocol first used dataset-specific mappings to convert gold annotations and predictions into the atomic key $q_j = (D, s_j, \kappa_j)$ defined in Section 3.1. It then concatenated the request sequence $Q_D = (q_1, \ldots, q_m)$ for each document to form the global test inventory Q. Single-valued fields used fixed instance keys, while repeated entities, table rows, and hierarchical fields used protocol-defined entity keys, row keys, or canonical structural paths. Unmatched gold atoms were counted as missing predictions, and unmatched predicted atoms were counted as false positives. The frozen inventory contained 497 documents and 4,720 target atoms: 50 FUNSD documents with 2,332 atoms, 347 SROIE documents with 1,388 atoms, and 100 CORD documents with 1,000 atoms. Dataset mappings, match order, and tie-handling rules must remain frozen with the evaluation script.

During training, LayTextLLM and LayTokenLLM-8B used the same LoRA fine-tuning procedure. LoRA rank, scaling factor, dropout, target modules, learning rate, batch size, number of training epochs, and optimization schedule used shared settings. At inference, LayoutLMv3, LayTextLLM, and LayTokenLLM-8B shared task-level parameters fixed by the same configuration file, including inference batches, candidate-output constraints, confidence thresholds, and pre-evaluation normalization. After selection on the validation sets, the paired control froze the same LayTokenLLM-8B checkpoint and candidate cache, and VGA-KIE did not update the candidate-generator parameters. Because the three model types use layout-encoder and layout-token LLM interfaces, model-specific differences were retained only where required by network architecture, input encoding, and output form.

A complete record for each evaluated configuration should include the atomic key, field value, initial-candidate provenance, linked evidence, candidate-level support label u_{jl}, final state z_j, failure reason, and repair trajectory, together with the upstream ranking score π_{jl}, grounding support score g_{jl}, and final reliability score ρ_j. Run metadata should also record the candidate-adapter checkpoint, OCR and layout parser versions, K_c, K_r, verification thresholds, scoring weights, random seeds, hardware, and evaluation-script version. Tables 3–5 currently report aggregate point estimates; aggregate tables do not replace atomic-level outputs and run logs.

Field-value metrics. For each dataset, let Y_{pred} be the set of non-empty predictions whose final state is pass, Y_{gold} the set of target gold fields, and C_{match} the set of matched predictions for which both the field name and normalized field value satisfy

the evaluation script. Field-level precision, recall, and F1 are defined as follows:

$$P_f = \frac{|C_{\text{match}}|}{|Y_{\text{pred}}|}, \tag{30}$$

$$R_f = \frac{|C_{\text{match}}|}{|Y_{\text{gold}}|}, \tag{31}$$

$$F_{1,f} = \frac{2P_f R_f}{P_f + R_f}. \tag{32}$$

Fields with a final state of fail or uncertain and empty outputs were treated as missing predictions. Field-F1 was calculated independently for each dataset, and Avg. in Table 3 is the arithmetic mean of the three dataset-level Field-F1 values.

Evidence matching. For a field instance with gold evidence, the matcher used token overlap, cell overlap, or bounding-box intersection over union according to the evidence type. The matching rule and its threshold were determined only on the validation sets and frozen for testing. Fields whose evidence could not be uniquely localized were labeled `evidence_not_evaluable` and excluded from Evidence Accuracy.

Gold evidence was derived by locating each normalized gold value in the token, table-cell, and bounding-box annotations. A unique exact match could be confirmed directly. Multiple matches and evidence spanning lines or cells were resolved with predefined uniqueness and region-consistency rules. Only fields without uniquely determinable evidence were excluded, and the number excluded must be reported.

Evidence and support metrics. Let A_e be the set of fields with a final state of pass and evaluable gold evidence, and $Q_{\text{pred},e}$ the set of all released, non-empty target field instances with evaluable gold evidence. Evidence Accuracy is the proportion of A_e whose predicted evidence satisfies the frozen matching function against the gold evidence. Unsupported Output Rate is the proportion of $Q_{\text{pred},e}$ that an independent external matcher cannot link to an acceptable token span, table cell, or bounding box. The current run released only pass fields, and all 4,475 released fields had evaluable evidence; the two metrics therefore happened to share the same denominator in this run. Gold provenance was matched for 4,153 of 4,475 fields, giving 92.8% Evidence Accuracy. Valid document evidence could not be linked for 121 of 4,475 fields, giving a 2.7% Unsupported Output Rate. Some predictions were linked to document evidence but not to the gold provenance, so the two percentages sum to less than 100%.

Acceptance Coverage. The system formed the release set A by jointly requiring a non-empty value, non-empty evidence, and a pass state, as specified in Section 3.3. Let N_Q and N_A denote the number of frozen target atoms and the number of released atoms, respectively. Then AcceptanceCoverage $= N_A/N_Q$. VGA-KIE released 4,475 of 4,720 target atoms, giving an Acceptance Coverage of 94.8%. Acceptance Coverage is reported with Unsupported Output Rate to expose the effect of selective abstention on support metrics. Because external methods do not provide atomic-level counts under the same definition, only the point estimate for VGA-KIE is reported.

Failure detection. For every initial top-1 candidate q, the gold failure label was set to 1 if its normalized field value did not match the gold value or, when gold evidence was evaluable, its provenance region did not satisfy the frozen evidence matcher;

otherwise, the label was set to 0. The predicted failure label was set to 1 when the verifier returned fail or uncertain and to 0 when it returned pass. The aggregate Precision, Recall, and F1 were 87.5%, 88.4%, and 87.9%, respectively, corresponding to archived confusion counts of approximately TP = 350, FP = 50, and FN = 46. Because the current archive retains metrics to one decimal place only, these integers are approximate audit values and do not support additional significant digits. This metric evaluates binary failure versus no-failure detection; macro-F1 over the six dominant failure reasons requires per-class annotations.

Repair metrics. Let R^- be the set of initially incorrect fields that entered repair. Repair Success is the number of fields that changed from incorrect to correct with external evidence divided by $|R^-|$. Of 400 initially incorrect fields entering repair, 282 became correct after renewed evidence linking and re-verification, giving a Repair Success of 70.5%. To measure the net damage caused by the complete post-hoc process to initially correct candidates, let R^+ be all initially correct fields exposed to the process, including both unchanged and rewritten fields. False Repair Rate is the number of fields that changed from correct to incorrect divided by $|R^+|$. Of 4,100 initially correct fields, 86 were ultimately changed to incorrect values, giving 2.1%. This definition avoids using the number of fields merely flagged incorrectly by the verifier as the risk denominator for all correct candidates. A zero denominator is reported as N/A.

Reliability score. The field reliability score $\rho_q \in [0, 1]$ is used only for candidate ranking and verification-state decisions. In this study, ρ_q is described as a reliability score and is not interpreted as a calibrated probability.

Statistical boundary. Tables 3–5 report single aggregate point estimates on the fixed test sets. The current archive does not contain reproducible document-level bootstrap outputs, variance across random seeds, or confidence intervals. The manuscript therefore does not use the term statistically significant or interpret small numerical differences as stable rankings. A future reproduction should use documents as the resampling unit, report 95% bootstrap confidence intervals, and estimate intervals for paired differences on the shared test documents.

Efficiency boundary. Latency (rel.) in Table 4 uses LayTokenLLM-8B direct extraction as 1.00×. When candidates are loaded from the frozen cache, 1.32× measures only the relative overhead of the current post-hoc reliability layer. It is not the total end-to-end latency of OCR, candidate generation, and post-hoc processing and cannot be compared across hardware. Absolute latency per document, p50 and p95, batch size, hardware, and the repair-trigger rate require reproducible run logs.

Paired primary-task comparison. LayTokenLLM-8B direct extraction achieved Field-F1 scores of 90.86, 94.72, and 94.88 on FUNSD, SROIE, and CORD, respectively. Adding VGA-KIE increased the scores to 94.91, 99.17, and 98.86, corresponding to gains of 4.05, 4.45, and 3.98 percentage points. The two configurations achieved mean Field-F1 values of 93.49 and 97.65, an average gain of 4.16 percentage points under the protocol. They shared the same frozen LayTokenLLM-8B checkpoint, input processing, decoding configuration, and candidate cache. Because the archive does not contain document-level uncertainty outputs, these differences are reported as paired point estimates on the fixed test sets and are not described as statistically significant. Table 3 summarizes Field-F1 under the controlled training and inference

settings. All four methods shared the test documents, data splits, OCR and layout products, and evaluation script. Avg. is the arithmetic mean of Field-F1 across the three datasets.

Table 3 Field-F1 (%) under the shared evaluation protocol.

Method	Model type	FUNSD	SROIE	CORD	Avg.
LayoutLMv3	OCR-based layout encoder	91.42	93.06	94.28	92.92
LayTextLLM	Layout-token LLM	90.35	95.78	95.41	93.85
LayTokenLLM-8B direct extraction	Layout-token LLM	90.86	94.72	94.88	93.49
LayTokenLLM-8B + VGA-KIE (ours)	Layout-token LLM + Ground-Verify-Repair	94.91	99.17	98.86	97.65

Note. All methods use the shared evaluation protocol. The two LayTokenLLM-8B rows use the same frozen checkpoint, preprocessing, decoding configuration, and candidate cache. Values are not derived from the heterogeneous public results in Table 2.

The paired differences in Table 3 were positive for all three datasets, although the strongest external comparator differed by dataset. The table establishes VGA-KIE's performance position under the shared protocol. Because LayTokenLLM-8B direct extraction and LayTokenLLM-8B with VGA-KIE shared the same candidate cache, their difference can be used for paired attribution of the gain. Other baselines, including LayoutLMv3 and LayTextLLM, have different candidate spaces and are used only for performance positioning.

Under the shared protocol, VGA-KIE achieved 94.91, 99.17, and 98.86 on FUNSD, SROIE, and CORD, with an average of 97.65. Relative to the strongest external baseline on each dataset, with scores of 91.42, 95.78, and 95.41, these values represent improvements of 3.49, 3.39, and 3.45 percentage points. Relative to LayTextLLM, the strongest external baseline by average score at 93.85, the mean improvement was 3.80 percentage points.

Table 3 positions VGA-KIE against representative methods, but mechanism attribution relies primarily on two stronger forms of internal evidence. First, Table 4 compares direct extraction with LayTokenLLM-8B plus VGA-KIE using the same frozen LayTokenLLM-8B candidate cache, thereby isolating the joint increment of the post-hoc reliability layer. Second, Table 5 removes evidence linking, typed verification, repair, and verify-again under the same candidate set and evaluation script, distinguishing the contributions of evidence state, failure conditions, and re-verification gating. The novelty argument therefore does not rest on post-hoc interpretation of heterogeneous baselines.

Table 4 reports the paired results under the shared frozen LayTokenLLM-8B candidate cache and presents the VGA-KIE point estimates together with their audit counts. Direct extraction and LayTokenLLM-8B plus VGA-KIE achieved average Field-F1 values of 93.49 and 97.65, respectively, a difference of 4.16 percentage points under the aggregate protocol. VGA-KIE achieved 94.8% Acceptance Coverage (4,475/4,720). In the external evidence audit, Evidence Accuracy increased from

76.8% to 92.8% (4,153/4,475), and Unsupported Output Rate decreased from 12.4% to 2.7% (121/4,475).

For VGA-KIE, Verification Failure Precision, Recall, and F1 were 87.5%, 88.4%, and 87.9%, with approximate confusion counts of TP = 350, FP = 50, and FN = 46. Repair Success was 70.5% (282/400), and False Repair Rate was 2.1% (86/4,100). LayTokenLLM-8B direct extraction performs neither post-hoc verification nor repair, so the corresponding metrics are N/A. These gains were accompanied by a 1.32× relative latency for the post-hoc layer under the frozen candidate-cache condition. Because document-level bootstrap intervals and across-seed variance are unavailable, all values remain point estimates on the fixed test sets.

Table 4 Paired reliability results for LayTokenLLM-8B direct extraction and LayTokenLLM-8B + VGA-KIE under the same frozen candidate cache. Counts are reported as numerator/denominator; the symbol ≈ denotes approximate verification counts. Unsupported Output Rate is computed by an independent external matcher.

Metric	LayTokenLLM-8B direct	LayTokenLLM-8B + VGA-KIE	VGA-KIE audit count / scope
Avg. Field-F1 ↑	93.49	97.65	Arithmetic mean across FUNSD, SROIE, and CORD
Acceptance Coverage ↑	Unavailable	94.8%	4,475 / 4,720 target field instances
Evidence Accuracy ↑	76.8%	92.8%	4,153 / 4,475 released, evidence-evaluable fields
Unsupported Output Rate ↓	12.4%	2.7%	121 / 4,475 released non-empty fields
Verification Failure	N/A	P 87.5% / R 88.4% / F1 87.9%	TP/FP/FN ≈ 350/50/46
Repair Success ↑	N/A	70.5%	282 / 400 initially wrong fields entering repair
False Repair Rate ↓	N/A	2.1%	86 / 4,100 initially correct fields exposed to the post-hoc pipeline
Latency (rel.)	1.00×	1.32×	Frozen-candidate post-hoc latency; excludes OCR and candidate generation

5.5 Main results, reliability, and evaluation metrics

Table 5 uses the full VGA-KIE system as the reference and removes the layout graph, evidence linking, typed verification, repair, and verify-again under the same candidate set and evaluation script. These ablations test how the five components affect field accuracy, evidence support, and the repair boundary.

Table 5 Component ablation for LayTokenLLM-8B + VGA-KIE under the same frozen candidate cache and evaluation script. Repair metrics are reported as Repair Success (RS) / False Repair Rate (FR), in %; zero-denominator values are N/A.

Variant	Field-F1 (%)	Evidence Accuracy (%)	Unsupported Output Rate(%)	Repair metrics (%)
Full VGA-KIE	97.65	92.8	2.7	RS 70.5 / FR 2.1
w/o layout graph	95.42	89.6	4.3	RS 64.8 / FR 2.5
w/o evidence linking	94.18	81.3	9.2	RS 54.8 / FR 3.9
w/o typed verification	95.06	87.6	6.1	RS 56.2 / FR 4.3
w/o repair	94.73	90.1	4.7	RS N/A / FR N/A
w/o verify-again	96.62	91.5	3.4	RS 71.8 / FR 5.8

Evidence linking produced the largest change in the complete result. Removing it reduced Field-F1 from 97.65 to 94.18 (−3.47), reduced Evidence Accuracy from 92.8% to 81.3% (−11.5 percentage points), increased Unsupported Output Rate to 9.2%, and reduced Repair Success to 54.8%. These coupled changes indicate that locatable evidence both constrains field acceptance and provides the state needed for subsequent diagnosis and repair.

Without typed verification, Field-F1, Evidence Accuracy, and Repair Success were 95.06, 87.6%, and 56.2%, respectively. Without the layout graph, the same three metrics were 95.42, 89.6%, and 64.8%. These results show that typed diagnosis and layout structure both contribute to repair routing, although their effects are smaller than that of evidence linking across the complete state chain.

Repair and verify-again had distinct functions. Removing repair produced 94.73 Field-F1 and 90.1% Evidence Accuracy, while Repair Success and False Repair Rate were not applicable. Removing verify-again retained 96.62 Field-F1, but increased False Repair Rate from 2.1% to 5.8% and produced 71.8% Repair Success. Repair Success for the complete system was corrected to 70.5% (282/400) based on the audit count. Repair therefore primarily supplies state transitions from incorrect to correct, whereas verify-again primarily limits harmful rewrites. Their joint action defines the closed-loop boundary of failure-conditioned repair.

Boundary of the novelty claim. Evidence Accuracy, Unsupported Output Rate, and the evidence-linking ablation support the unified field state. Binary Verification Failure F1 and the typed-verification ablation support the role of failure conditions. Repair Success, False Repair Rate, and the repair and verify-again ablations support conditional repair and the acceptance gate. This evidence supports the effectiveness of the combined reliability layer, but it does not show that all six failure types have

been diagnosed accurately at the class level or that replacing the candidate generator would preserve the same gain.

5.6 Mechanism trajectories and repair boundaries

Fig. 3 presents the general action routing, whereas Fig. 3 focuses on three terminal boundaries: an OCR error that passes verification after local repair, a field rejected because document evidence is absent, and a harmful rewrite caused by repair under an ambiguous layout. Each trajectory follows the sequence evidence, initial candidate, failure condition, local action, and final state, allowing direct comparison among successful repair, unsupported rejection, and false repair.

In Fig. 4a, format constraints and weak evidence trigger ReOCR and text-span adjustment. In Fig. 4b, expanding the search still fails to establish an evidence link, so the system returns unsupported/fail. Fig. 4c shows that re-verification can still accept an incorrect rewrite under ambiguous context. These panels are mechanism illustrations rather than empirical failure cases with dataset names, document identifiers, and real crops. They therefore do not establish incidence, diagnostic accuracy, or repair performance and are excluded from the denominators of Tables 4 and 5.

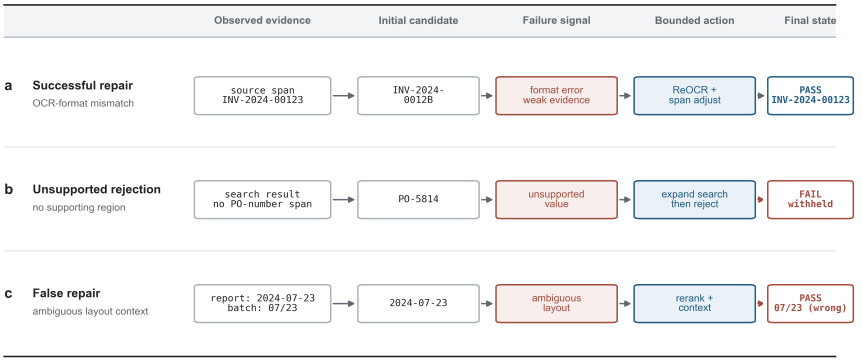


Fig. 4 Illustrative mechanism-level state transitions at the repair boundary. a, ReOCR and span adjustment correct an OCR-format failure before the candidate passes re-verification. b, Evidence expansion still finds no supporting region, so the field is withheld. c, Ambiguous layout context causes an initially correct value to be rewritten and incorrectly accepted. These scenarios explain decision logic only; they are not dataset-grounded cases and are excluded from all metrics.

6 Discussion

The scientific value of VGA-KIE lies in making reliability decisions that are usually implicit in KIE explicit. A larger candidate generator can improve field values but does not automatically provide auditable answers to whether the evidence is sufficient, which type of constraint has failed, or whether a repaired value should be accepted. VGA-KIE organizes these questions as a single field-state transition, which is relevant

to business, financial, and administrative documents that require review, auditing, or downstream automated processing.

The unified candidate interface provides interface-level modularity, not demonstrated model independence. OCR-based systems, layout-aware LLMs, and OCR-free MLLMs can be mapped to the same candidate structure, but differences in score calibration, provenance granularity, and error distributions can change the benefits of verification and repair. The supported design conclusion is therefore limited to interface compatibility; the present results do not establish stable gains across candidate generators.

The novelty argument has three empirical boundaries. First, Verification Failure F1 is a binary metric: it supports effective failure detection but does not establish accurate discrimination among all six dominant failure reasons. Second, the current results use a fixed LayTokenLLM-8B candidate interface and do not establish external validity across candidate generators. Third, Fig. 4 is a mechanism illustration, not a qualitative experiment with document identifiers and real evidence crops.

The experiments that would most directly strengthen the argument correspond to these three boundaries. A confusion matrix and per-type precision, recall, and F1 should be reported for the six failure reasons. Candidate-generator swap experiments should be performed with at least one OCR-based generator and one LLM or MLLM generator. Dataset-grounded qualitative cases should be sampled from field-level outputs on FUNSD, SROIE, and CORD and should include the document identifier, real crop, initial candidate, gold value, failure reason, and repair result. These experiments would separately test typed diagnosis, external validity of the interface, and auditability of actual errors.

7 Conclusion

This work introduced VGA-KIE, a post-hoc reliability method that organizes evidence states, failure conditions, and re-verification gating as a unified field-state transition. The evidence-grounded field state binds each field value to document evidence, typed failure diagnosis converts constraint violations into action conditions, and failure-conditioned repair with re-verification requires every repaired candidate to pass the evidence and verification gates again. The contribution lies in the decision coupling among these mechanisms, not in claiming priority for grounding, rule checking, or retrying in isolation.

Under the shared evaluation reported in Tables 3–5, LayTokenLLM-8B with VGA-KIE achieved an average Field-F1 of 97.65. In the paired comparison using the same frozen candidate cache, it improved on LayTokenLLM-8B direct extraction at 93.49 by 4.16 percentage points. VGA-KIE released 4 475 of 4 720 target fields, giving 94.8% Acceptance Coverage; it achieved 92.8% Evidence Accuracy (4 153/4 475), a 2.7% Unsupported Output Rate (121/4 475), 70.5% Repair Success (282/400), and a 2.1% False Repair Rate (86/4 100). The paired comparison in Table 4 supports the incremental value of the combined post-hoc layer, while the ablations in Table 5 associate that increment with evidence linking, typed verification, repair, and re-verification. The conclusion is limited to the present shared protocol, the

LayTokenLLM-8B candidate interface, and binary failure detection; class-level diagnosis of six failure types, gains across candidate generators, and qualitative cases from real documents require separate experiments.

These conclusions apply to the shared evaluation protocol, the three core datasets, and the fixed LayTokenLLM-8B candidate condition. Publicly reported values are used only as literature context and do not enter local gain attribution. The evidence supports a combined post-hoc layer with evidence state, failure conditions, and a re-verification gate. Compatibility across candidate generators, per-class failure diagnosis, generalization to long multi-page documents, and real failure cases remain to be established independently.

Acknowledgements. Thanks to all the authors for their contributions to this paper. The authors would like to express their gratitude to the reviewers, whose helpful suggestions greatly improved the presentation of this paper.

Declarations

- Funding: This work is supported by Henan Provincial Science and Technology Research Project (252102110358).
- Conflict of interest: The authors declare no conflicts of interest to report regarding the present study.
- Ethics approval and consent to participate:Not applicable.
- Consent for publication:All authors have approved the publication of this manuscript and agree to submit the work for publication. No participant-related privacy issues or ethical risks are involved in this study.
- Data availability : This study used the publicly available FUNSD, SROIE, and CORD datasets. The datasets can be accessed through their original publications and official project pages. This study did not create a new dataset.
- Materials availability: All experimental materials and auxiliary data that support the findings of this study are available from the corresponding author upon reasonable request.
- Code availability :The relevant source code and experimental programs generated during this study are available from the corresponding author upon reasonable request.
- Author contribution: Conceptualization, YS and NK; methodology, YS; software, YS and BP; validation, BP and MW; data curation, YS and MW; writing—original draft preparation, YS; writing—review and editing, YS and NK; visualization, BP. All authors have read and agreed to the published version of the manuscript.

References

[1] Jaume, G., Ekenel, H.K., Thiran, J.-P.: Funsd: A dataset for form understanding in noisy scanned documents. In: 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW), pp. 1–6 (2019). https://doi.org/10.1109/icdarw.2019.10029

[2] Huang, Z., Chen, K., He, J., Bai, X., Karatzas, D., Lu, S., Jawahar, C.V.: Icdar2019 competition on scanned receipt ocr and information extraction. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1516–1520 (2019). https://doi.org/10.1109/icdar.2019.00244

[3] Liu, X., Gao, F., Zhang, Q., Zhao, H.: Graph convolution for multimodal information extraction from visually rich documents. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers), pp. 32–39 (2019). https://doi.org/10.18653/v1/N19-2005

[4] Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.: Layoutlm: Pre-training of text and layout for document image understanding. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1192–1200 (2020). https://doi.org/10.1145/3394486.3403172

[5] Lee, C.-Y., Li, C.-L., Dozat, T., Perot, V., Su, G., al.: Formnet: Structural encoding beyond sequential modeling in form document information extraction. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 3735–3754 (2022). https://doi.org/10.18653/v1/2022.acl-long.260

[6] Park, S., Shin, S., Lee, B., Lee, J., Surh, J., Seo, M., Lee, H.: Cord: A consolidated receipt dataset for post-ocr parsing. In: NeurIPS Workshop (2019)

[7] Wang, D., Raman, N., Sibue, M., Ma, Z., Babkin, P., al.: Docllm: A layout-aware generative language model for multimodal document understanding. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 8529–8548 (2024). https://doi.org/10.18653/v1/2024.acl-long.463

[8] Fujitake, M.: Layoutllm: Large language model instruction tuning for visually rich document understanding. In: Proceedings of the Language Resources and Evaluation Conference, pp. 10219–10224 (2024). https://doi.org/10.63317/428wjtmc3xvh

[9] Kim, G., Hong, T., Yim, M., Nam, J., Park, J., al.: Ocr-free document understanding transformer. In: Lecture Notes in Computer Science, pp. 498–517 (2022). https://doi.org/10.1007/978-3-031-19815-1_29

[10] Hu, A., Xu, H., Ye, J., Yan, M., Zhang, L., al.: mplug-docowl 1.5: Unified structure learning for ocr-free document understanding. In: Findings of the Association for Computational Linguistics: EMNLP 2024, pp. 3096–3120 (2024). https://doi.org/10.18653/v1/2024.findings-emnlp.175

[11] Liu, S., Zhang, Z., Hu, P., Ma, J., Du, J.: See then Tell: Enhancing Key Information Extraction with Vision Grounding (2024)

[12] Perot, V., Kang, K., Luisier, F., Su, G., Sun, X.: Lmdx: Language model-based document information extraction and localization. In: Findings of the Association for Computational Linguistics: ACL 2024, pp. 15140–15168 (2024). https://doi.org/10.18653/v1/2024.findings-acl.899

[13] Townsend, B., May, M., Mackowiak, K., Wells, C.: RealKIE: Five Novel Datasets for Enterprise Key Information Extraction (2024)

[14] Ji, Y., Xu, Z., Liu, Z., Chen, Z., Zhang, Q.: Unikie-bench: Benchmarking large multimodal models for key information extraction in visual documents. In: Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 6331–6352 (2026). https://doi.org/10.18653/v1/2026.acl-long.287

[15] Zhang, C., Zhao, Y., Xie, Y., Yuan, C., Tu, Y.: Unveiling the deficiencies of pre-trained text-and-layout models in real-world visually-rich document information extraction. In: Findings of the Association for Computational Linguistics: EACL 2026, pp. 1–16 (2026). https://doi.org/10.18653/v1/2026.findings-eacl.1

[16] Zhong, X., Tang, J., Yepes, A.J.: Publaynet: Largest dataset ever for document layout analysis. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1015–1022 (2019). https://doi.org/10.1109/icdar.2019.00166

[17] Hong, T., Kim, D., Ji, M., Hwang, W., Nam, D., Park, S.: Bros: A pre-trained language model focusing on text and layout for better key information extraction from documents. Proceedings of the AAAI Conference on Artificial Intelligence **36**(10), 10767–10775 (2022) https://doi.org/10.1609/aaai.v36i10.21322

[18] Xu, Y., Xu, Y., Lv, T., Cui, L., Wang, F., al.: Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 2579–2591 (2021). https://doi.org/10.18653/v1/2021.acl-long.201

[19] Huang, Y., Lv, T., Cui, L., Lu, Y., Wei, F.: Layoutlmv3: Pre-training for document ai with unified text and image masking. In: Proceedings of the 30th ACM International Conference on Multimedia, pp. 4083–4091 (2022). https://doi.org/10.1145/3503161.3548112

[20] Xu, Y., Lv, T., Cui, L., Wang, G., Lu, Y., al.: LayoutXLM: Multimodal Pre-training for Multilingual Visually-rich Document Understanding (2021)

[21] Powalski, R., Borchmann, L., Jurkiewicz, D., Dwojak, T., Pietruszka, M., Palka, G.: Going full-tilt boogie on document understanding with text-image-layout transformer. In: Lecture Notes in Computer Science, pp. 732–747 (2021). https://doi.org/10.1007/978-3-030-86331-9_47

[22] Appalaraju, S., Jasani, B., Kota, B.U., Xie, Y., Manmatha, R.: Docformer: End-to-end transformer for document understanding. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 973–983 (2021). https://doi.org/10.1109/iccv48922.2021.00103

[23] Li, Y., Qian, Y., Yu, Y., Qin, X., Zhang, C., al.: Structext: Structured text understanding with multi-modal transformers. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 1912–1920 (2021). https://doi.org/10.1145/3474085.3475345

[24] Tang, Z., Yang, Z., Wang, G., Fang, Y., Liu, Y., al.: Unifying vision, text, and layout for universal document processing. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 19254–19264 (2023). https://doi.org/10.1109/cvpr52729.2023.01845

[25] Lu, J., Yu, H., Wang, Y., Ye, Y., Tang, J.: A bounding box is worth one token - interleaving layout and text in a large language model for document understanding. In: Findings of the Association for Computational Linguistics: ACL 2025, pp. 7252–7273 (2025). https://doi.org/10.18653/v1/2025.findings-acl.379

[26] Zhu, Z., Luo, C., Shao, Z., Gao, F., Xing, H.: A simple yet effective layout token in large language models for document understanding. In: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 14472–14482 (2025). https://doi.org/10.1109/cvpr52734.2025.01349

[27] Zhang, J., You, Z., Wang, J., Le, X.: Sail: Sample-centric in-context learning for document information extraction. Proceedings of the AAAI Conference on Artificial Intelligence **39**(24), 25868–25876 (2025) https://doi.org/10.1609/aaai.v39i24.34780

[28] Dhouib, M., Bettaieb, G., Shabou, A.: Docparser: End-to-end ocr-free information extraction from visually rich documents. In: Lecture Notes in Computer Science, pp. 155–172 (2023). https://doi.org/10.1007/978-3-031-41734-4_10

[29] Hu, A., Xu, H., Zhang, L., Ye, J., Yan, M., al.: mplug-docowl2: High-resolution compressing for ocr-free multi-page document understanding. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 5817–5834 (2025). https://doi.org/10.18653/v1/2025.acl-long.291

[30] Shi, Y., Peng, D., Liao, W., Lin, Z., Chen, X., al.: Exploring OCR Capabilities of GPT-4V(ision): A Quantitative and In-depth Evaluation (2023)

[31] Liu, Y., Li, Z., Huang, M., Yang, B., Yu, W., Li, C., al.: Ocrbench: on the hidden mystery of ocr in large multimodal models. Science China Information Sciences **67**(12) (2024) https://doi.org/10.1007/s11432-024-4235-6

[32] Blecher, L., Cucurull, G., Scialom, T., Stojnic, R.: Nougat: Neural Optical Understanding for Academic Documents (2023)

[33] Lee, K., Joshi, M., Turc, I., Hu, H., Liu, F., al.: Pix2Struct: Screenshot Parsing as Pretraining for Visual Language Understanding (2022)

[34] Mathew, M., Karatzas, D., Jawahar, C.V.: Docvqa: A dataset for vqa on document images. In: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 2199–2208 (2021). https://doi.org/10.1109/wacv48630.2021.00225

[35] Gralinski, F., Stanislawek, T., Wroblewska, A., al.: Kleister: Key information extraction datasets involving long documents with complex layouts. In: Lecture Notes in Computer Science, pp. 564–579 (2021). https://doi.org/10.1007/978-3-030-86549-8_36

[36] Khang, M., Jung, S.C., Park, S., Hong, T.: KIEval: Evaluation Metric for Document Key Information Extraction (2025)

[37] Li, M., Xu, Y., Cui, L., Huang, S., Wei, F., Li, Z., Zhou, M.: Docbank: A benchmark dataset for document layout analysis. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 949–960 (2020). https://doi.org/10.18653/v1/2020.coling-main.82

[38] Li, M., Cui, L., Huang, S., Wei, F., Zhou, M., Li, Z.: TableBank: Table Benchmark for Image-based Table Detection and Recognition (2020)

[39] Smock, B., Pesala, R., Abraham, R.: Pubtables-1m: Towards comprehensive table extraction from unstructured documents. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4624–4632 (2022). https://doi.org/10.1109/cvpr52688.2022.00459

[40] Li, M., Lv, T., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., Wei, F.: Trocr: Transformer-based optical character recognition with pre-trained models. Proceedings of the AAAI Conference on Artificial Intelligence **37**(11), 13094–13102 (2023) https://doi.org/10.1609/aaai.v37i11.26538

[41] Simsa, J., Siencnik, S.K., Uresova, Z., al.: Docile benchmark for document information localization and extraction. In: Lecture Notes in Computer Science, pp. 147–166 (2023). https://doi.org/10.1007/978-3-031-41679-8_9