BEYOND VISUAL BOUNDS: GENERATION-AWARE LAYOUT ANALYSIS FOR EDITABLE VISUAL DOCUMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models (VLMs) have advanced text recognition and document parsing. However, when a parsed text element is edited, bounding boxes that capture only its visible extent do not specify the space available for replacement text or how that text should be aligned within that space. We introduce Generation-aware Layout Analysis (GLA), a task that recovers text, allocated regions, semantic categories, and alignment to provide spatial constraints for subsequent element-level visual editing. This representation could support editable slide and document reconstruction, translation with layout preservation, and automated design editing. For training and evaluation on this task, we construct a 37K-image training dataset spanning nine domains and introduce GLA-Bench, a 1,000-image benchmark with four tasks. To predict dense coordinate and attribute sequences accurately and efficiently, we introduce joint supervised fine-tuning (SFT), reinforcement learning (RL) with layout-specific rewards, and accelerated decoding. To recover local feedback obscured by image-level aggregation, we introduce EAA-GRPO, which augments group-relative policy optimization (GRPO) with Element-level Advantage Assignment (EAA). EAA compares predictions for the same element across rollouts to provide field-specific supervision. To accelerate decoding, we adapt a block-diffusion drafter to the repetitive structure of parsing outputs. On GLA-Bench, our 4B parser reaches 46.59 Joint@.75 in full parsing, versus 6.26 for the best evaluated zero-shot baseline. Layout-only RL improves full parsing on both backbones. Task-adapted drafting achieves 2.18× autoregressive throughput with the SFT target fixed.

1 INTRODUCTION

Layout analysis aims to recover the structure of visual documents by identifying content elements, their semantic roles, and their spatial organization. Existing approaches detect and classify document regions (Zhong et al., 2019; Pfitzmann et al., 2022), generate text or markup directly from images with OCR-free models (Kim et al., 2022; Lee et al., 2023; Wei et al., 2024), and recover interface elements and their locations with screen parsers (Baechler et al., 2024; Lu et al., 2024). However, their representations do not explicitly distinguish a text element's visible extent from the space available for replacement content or specify its alignment within that space.

Slides, posters, document pages, and interfaces often circulate as raster images even when their content must later be revised. Replacing or translating a text element requires knowing how much space the new text can occupy and how it should be aligned with the surrounding design. For example, Figure 1 shows the website title "O For Paz" being expanded to "O For Paz International Marketplace." Using the visible text bounds to constrain this edit forces the longer title to wrap and smaller font size, whereas recovering the wider allocated title region allows it to remain on a single line with suitable size. The English-to-German translation example illustrates a similar issue: fitting the longer translation into the original text box shrinks the text, while recovering the available region allows it to wrap at a readable size. Recovering these spatial constraints can support editable slide and document reconstruction, image to slide, translation with layout preservation, and automated design editing (Hazimeh et al., 2025; Qian et al., 2024; Tuo et al., 2024).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

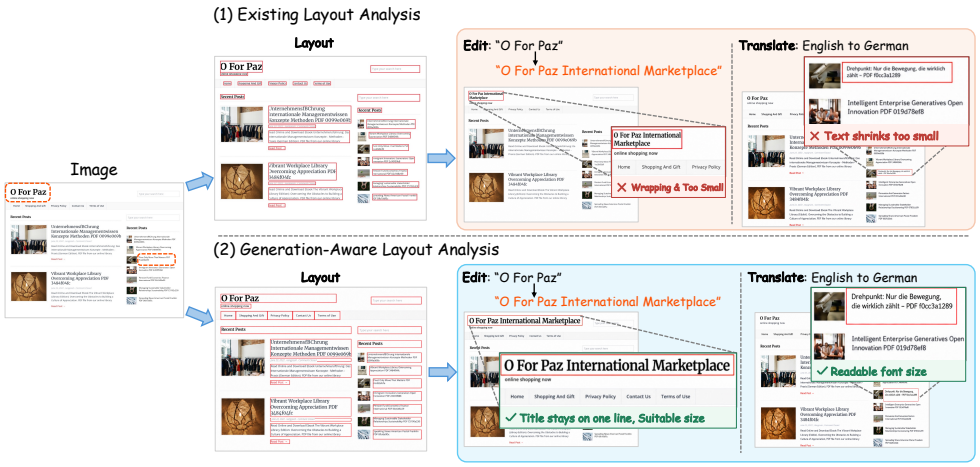


Figure 1: Visible-text localization and generation-aware layout analysis. A box enclosing the rendered text leaves the surrounding space available for replacement content unspecified. GLA recovers allocated regions and alignment to provide placement constraints for editing and translation.

To address this gap, we introduce GLA, a task that extends layout analysis to recover text, allocated regions, semantic categories, and alignment from raster images. As illustrated in Figure 1, an allocated region specifies the space available to a text element given the surrounding layout. This region is inferred from containers, margins, neighboring elements, and whitespace. The semantic category describes the element's role, while alignment specifies its placement within the region. Together, these fields provide spatial constraints for subsequent element-level visual editing. This compact representation can be evaluated independently of complete document reconstruction (Hazimeh et al., 2025), with fonts, appearance, and layer structure left to downstream systems.

To support training and evaluation, we construct a training dataset spanning nine domains with 37K samples, and curate GLA-Bench, a 1,000-image benchmark. The benchmark is carefully annotated by humans for text, allocated regions, semantic categories, and alignment, while the training set consists of automatically annotated HTML-rendered images. We evaluate four complementary tasks: regional text recognition (`bbox2text`), allocated-region localization (`text2bbox`), layout-only parsing (`img2layout`), and full parsing (`img2full`).

To meet the demands for accurate prediction and efficient decoding of dense coordinate and attribute sequences in GLA, we propose a framework combining multi-task SFT, RL, and accelerated decoding. Joint SFT across the four tasks establishes a shared parsing capability, while RL with task-specific rewards refines localization and attribute prediction in layout-only parsing. However, standard group-relative policy optimization (GRPO) (Shao et al., 2024) assigns a shared sequence-level advantage based on aggregated image-level rewards. In dense layouts, this can obscure differences in element-level quality across rollouts, limiting feedback on local improvements. We therefore augment GRPO with Element-level Advantage Assignment (EAA), yielding EAA-GRPO. EAA compares predictions for the same element across rollouts and assigns element-level relative advantages to the corresponding coordinate fields while retaining image-level feedback. To improve decoding efficiency for long parsing outputs containing large numbers of coordinates, we explore two speculative decoding approaches: multi-token prediction (MTP) (Gloeckle et al., 2024) and DFlash with block-diffusion drafting (Chen et al., 2026). We further adapt the DFlash drafter to the repeated field structures and coordinate patterns in parsing outputs.

Experiments on GLA-Bench show that SFT improves allocated-region recovery more than visible-text localization on both backbones (Table 3). Our 4B parser reaches 46.59 Joint@.75 in full parsing, versus 6.26 for the best evaluated zero-shot baseline (Table 2). RL trained only on layout parsing improves full-parsing Joint@.75 by 2.63–4.13 points across both backbones; EAA-GRPO adds 1.05 points over GRPO on Qwen3.5-4B (Table 4). With the SFT target fixed, task adaptation raises DFlash throughput from 0.87× to 2.18× the autoregressive baseline (Table 6). Our contributions are as follows: (1) We formulate GLA to recover text, allocated regions, semantic categories, and alignment from raster images, providing spatial constraints for element-level visual editing. (2) We

construct a 37K-image, automatically annotated HTML-rendered training set spanning nine domains and curate GLA-Bench, a 1,000-image human-annotated benchmark supporting four complementary tasks. (3) We combine multi-task SFT, EAA-GRPO, and speculative decoding for accurate and efficient parsing. EAA compares the same element across rollouts to guide coordinate prediction, while MTP and task-adapted DFlash accelerate decoding.

2 Related Work

Visual parsing and editable text structure. Visual document analysis detects and labels semantic regions or generates text or markup directly from images. PubLayNet (Zhong et al., 2019) and DocLayNet (Pfitzmann et al., 2022) label document regions such as titles, paragraphs, and tables; OCR-free parsers recover content and, in some cases, page structure (Kim et al., 2022; Lee et al., 2023; Li et al., 2025). Screen parsers extend localization to interface elements and their functions (Baechler et al., 2024; Lu et al., 2024). Although these methods may localize semantic blocks rather than only glyphs, their representations do not explicitly separate visible text bounds from regions allocated for replacement content or specify alignment within them. Semantic derendering reconstructs editable artifacts such as SVG from slide images (Hazimeh et al., 2025), jointly accounting for text, appearance, and graphical elements. GLA recovers allocated text regions, semantic roles, and alignment. This independently measurable representation can provide constraints for editing and translation.

Credit assignment for structured visual predictions. GRPO typically assigns all tokens a shared response-level advantage (Shao et al., 2024), including in visual perception and grounding (Shen et al., 2025a). More methods explore finer-grained credit assignment. MetaSpatial's 3D-SPO assigns coordinate-token credit using physical penalties (Pan & Liu, 2026); TokenAdv routes reward-specific advantages to temporal-localization and answer tokens (Shen et al., 2025b). MCR-GRPO estimates each box's marginal contribution through leave-one-out scoring and normalizes these contributions within each response (Han et al., 2026). However, within-response normalization compares boxes with different localization difficulty, so even the best sampled prediction of a difficult element can receive negative local credit. EAA instead compares predictions of the same element across rollouts, providing relative element-level feedback while retaining image-level supervision.

Efficient decoding of structured outputs. Speculative decoding reduces target-model steps by verifying drafted continuations in parallel (Leviathan et al., 2023; Chen et al., 2023). We study two proposal strategies: MTP and block-diffusion drafting with DFlash (Gloeckle et al., 2024; Chen et al., 2026). HunyuanOCR-1.5 already adapts DFlash to long OCR outputs and reports length- and content-dependent speedups (Li et al., 2026). We examine both on element-level and full-image GLA tasks with different output lengths but a shared representation.

3 Dataset and Benchmark

We construct the training set and GLA-Bench using a shared element-level annotation schema. The training set consists of automatically annotated HTML-rendered images, whereas GLA-Bench is carefully annotated by humans. We first specify the annotation target and data curation procedures, then define four complementary tasks and their evaluation criteria.

3.1 Representation and Annotation

Element representation. Given an image $I \in \mathbb{R}^{H\times W\times 3}$, its annotation is a sequence of elements

$$Y = [e_1, \ldots, e_N], \qquad e_i = (t_i, b_i^{\text{alloc}}, c_i, a_i), \tag{1}$$

where t_i is textual content, $b_i^{\text{alloc}} = (x_1, y_1, x_2, y_2)$ is an axis-aligned allocated region, c_i identifies the element's semantic role or structural type, selected from 13 categories (*e.g.*, `Title`, `Text`, `Table`), and $a_i \in \{\texttt{left}, \texttt{center}, \texttt{right}, \texttt{justify}\}$ specifies horizontal alignment within that region. Coordinates are quantized to $[0, 1000]$, and records are serialized in reading order.

Allocated-region annotation criteria. An allocated region defines the space available to an element within the surrounding layout, rather than the tight bounds of its rendered content. Region

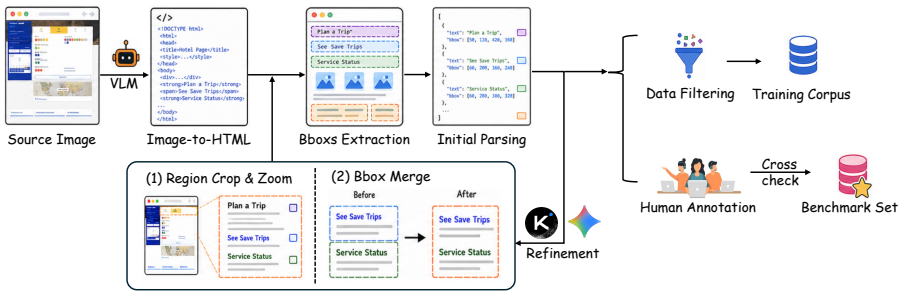


Figure 2: **Dataset construction pipeline.** Image-to-HTML parsing initializes the layout. Iterative refinement combines crop-and-zoom re-parsing with bounding-box merging, feeding results back into bounding-box extraction and parsing. Refined records are converted to the shared annotation schema. Training annotations are filtered for validity, while those for reserved benchmark images undergo manual correction and cross-checking to produce GLA-Bench.

boundaries follow a hierarchy of observable layout cues. Explicit boundaries, such as containers, panels, cells, and separators, take precedence, followed by repeated grid or column structures and then boundaries implied by neighboring elements. Whitespace and alignment patterns provide supporting evidence when stronger cues are absent. When multiple extents are plausible, the target is a conservative region supported by these cues, without extending into unconstrained whitespace. Visible content is typically contained within its allocated region, but containment is not a hard constraint: clipping, rotation, overflow, and decorative rendering can violate it.

3.2 Dataset Construction and Benchmark Curation

The source pool contains 38,418 images across nine domains. Appendix B.1 details their composition. Before annotation, we reserve 1,000 images through domain-stratified sampling for GLA-Bench; the remaining 37,418 images form the training set.

Annotation construction. Figure 2 summarizes the annotation pipeline for the training set and GLA-Bench. We use image-to-HTML parsing to initialize the layout because HTML explicitly represents text, containers, and their spatial organization, providing a structured basis for extracting candidate elements and regions. However, conversion to HTML can omit small or densely arranged text, approximate spacing and container boundaries, or fragment a visually coherent region into multiple boxes. We therefore introduce an iterative refinement loop that revisits the source image. Re-parsing enlarged local crops helps recover details difficult to resolve at full-image scale and refine region boundaries. Bounding-box merging consolidates fragmented predictions where the visual layout supports a single allocated region. Refinement results feed back into bounding-box extraction and parsing to update candidate records. The refined records are then converted to the schema in Eq. 1. For the training set, records are retained if they are parseable, contain nonempty text, and have valid coordinates and supported attribute values. This yields approximately 778K automatically annotated elements across 37,418 training images, averaging 20.8 elements per image.

GLA-Bench curation. For the reserved benchmark images, candidate annotations undergo exhaustive manual correction and cross-checking under the same annotation schema and allocation policy. Annotators correct text and allocated regions, verify categories and alignment, add omitted elements, remove duplicates and spurious detections, and correct serialization order. Remaining ambiguities are referred for adjudication. The resulting GLA-Bench contains 19,858 elements across 1,000 images, averaging 19.9 elements per image and covering all 13 categories. The element-count distribution is long-tailed, with a few images exceeding 100 elements. Appendix B.1 and Figure 6 give more detailed statistics. We plan an independent agreement assessment covering element coverage, transcription, region geometry, category, and alignment.

3.3 Diagnostic Tasks and Evaluation

Diagnostic tasks. Table 1 specifies four tasks. Element-level tasks isolate recognition given a reference region and localization given target text; full-image tasks require element discovery, with or

Table 1: **Diagnostic tasks derived from GLA-Bench.** The tasks assess complementary aspects of element recognition, localization, layout parsing, and full-image parsing.

Group	Task	Input	Output	Instances	Evaluation focus
Element-level	`bbox2text`	I, b_i^{alloc}	t_i	867	Text recognition
	`text2bbox`	I, t_i	b_i^{alloc}	934	Region localization
Full-image	`img2layout`	I	$[(b_i^{\text{alloc}}, c_i, a_i)]_{i=1}^{N}$	1,000	Layout-only parsing
	`img2full`	I	$[(t_i, b_i^{\text{alloc}}, c_i, a_i)]_{i=1}^{N}$	1,000	Full parsing

without transcription. Each full-image task uses all 1,000 benchmark images; the smaller element-level subsets bring the total to 3,801 task instances.

Element-level metrics. For `bbox2text`, we report character error rate (CER) and exact-match accuracy (EM), with the same text normalization used for full parsing. For `text2bbox`, we report mIoU and mean accuracy (mAcc). The latter averages localization accuracy over IoU thresholds $\mathcal{T} = \{0.50, 0.55, \ldots, 0.95\}$:

$$\text{mAcc} = \frac{1}{M|\mathcal{T}|} \sum_{j=1}^{M} \sum_{\tau \in \mathcal{T}} \mathbf{1}\Big[\text{IoU}(\hat{b}_j, b_j) \geq \tau\Big], \tag{2}$$

where M is the number of evaluated query instances. These metrics measure average region overlap and success under increasingly strict boundary requirements, respectively.

Full-image metrics. For `img2layout` and `img2full`, predictions are matched one-to-one with reference elements using geometry only. We report geometric F1 at IoU thresholds 0.50 and 0.75 and mean F1 over $\mathcal{T}$, counting unmatched predictions and references as false positives and false negatives, respectively. Category macro-F1 and alignment accuracy are computed over matched elements. Joint@τ for `img2layout` requires IoU of at least τ together with correct category and alignment. For `img2full`, it additionally requires an exact match between the normalized predicted and reference text. Element-level metrics are averaged over task instances, whereas full-image metrics are computed per image and then averaged. Appendix C.4 analyzes zero category and Joint scores.

4 Model Design

Generation-aware layout parsing must recover spatial constraints for many elements while keeping long structured outputs practical to generate. This creates two requirements: prediction must be sensitive to fine-grained local regions, and decoding must handle dense sequences of coordinates and attributes efficiently. Our framework addresses both with a single instruction-conditioned VLM and three complementary components. Multi-task SFT establishes shared recognition and layout-recovery capabilities; EAA-GRPO provides targeted element-level feedback; and speculative decoding exploits output regularities to reduce sequential generation cost.

4.1 Multi-Task SFT

Generation-aware parsing requires both local element understanding and full-image structure recovery. To combine these capabilities, we jointly train a single VLM on all four tasks, conditioned on an image, a task instruction, and any supplied text or region. Element-level tasks supervise regional text recognition and allocated-region localization; full-image tasks supervise element discovery and joint field prediction. Training minimizes the autoregressive next-token loss with paraphrased instructions and a fixed field order, yielding a shared parser that initializes RL.

4.2 Element-level Advantage Assignment for GRPO

SFT learns to reproduce records but does not directly optimize geometric accuracy or attribute correctness. We therefore apply RL to `img2layout`. However, dense outputs introduce a further challenge for GRPO (Shao et al., 2024): a single image-level reward aggregates many elements, so improvements to one region can be offset by errors in another. We propose the Element-level

Advantage Assignment (EAA) scheme for GRPO that compares the same element across rollouts while retaining image-level supervision. We call the resulting algorithm EAA-GRPO.

Task-specific rewards. For an image with N annotated elements, we sample G outputs and match each to the annotations using partial one-to-one assignment with normalized L1 and generalized-IoU (Rezatofighi et al., 2019) costs. Let i index a reference element and z_{gi} denote the IoU between its allocated region and the matched prediction in output g. The composite element reward is

$$R_{gi}^{\text{elem}} = U_{gi} + 0.1C_{gi} + 0.1H_{gi}, \qquad U_{gi} = u(z_{gi}),$$
$$u(z) = \frac{1}{|\mathcal{T}|}\sum_{\tau\in\mathcal{T}} \mathbf{1}[z \geq \tau] + \eta\,\mathbf{1}[z \geq 0.75], \tag{3}$$

where $\mathcal{T} = \{0.50, 0.55, \ldots, 0.95\}$ and $\eta = 0.5$. The bonus emphasizes the Joint@.75 localization threshold. Category and alignment correctness are $C_{gi} = \mathbf{1}[\hat{c}_{gi} = c_i]$ and $H_{gi} = \mathbf{1}[\hat{a}_{gi} = a_i]$, respectively, with hats denoting matched predictions. Missed elements receive zero for all components.

Averaging element rewards and adding detection and format terms gives

$$R_g^{\text{img}} = \frac{1}{N}\sum_{i=1}^{N} R_{gi}^{\text{elem}} + 0.1F_g^{\text{det}} + 0.05F_g^{\text{fmt}}, \tag{4}$$

where F_g^{det} and F_g^{fmt} score detection and format validity. The denominator N penalizes omissions. GRPO normalizes rewards across the G outputs:

$$A_g^{\text{img}} = \frac{R_g^{\text{img}} - \text{Mean}_{h=1}^{G} R_h^{\text{img}}}{\text{Std}_{h=1}^{G} R_h^{\text{img}} + \epsilon_{\text{img}}}, \tag{5}$$

where $\epsilon_{\text{img}} > 0$ stabilizes normalization. Equal image rewards yield zero shared advantage even when individual-element quality differs.

Element-level relative advantages. For each element i, EAA instead normalizes R_{gi}^{elem} across $\mathcal{V}_i$, the outputs with unambiguous matches or zero-reward misses; ambiguous assignments are excluded:

$$A_{gi}^{\text{box}} = \frac{R_{gi}^{\text{elem}} - \text{Mean}_{h\in\mathcal{V}_i} R_{hi}^{\text{elem}}}{\max\big(\text{Std}_{h\in\mathcal{V}_i} R_{hi}^{\text{elem}}, \epsilon_{\text{box}}\big)}, \tag{6}$$

where $\epsilon_{\text{box}} = 0.05$. This advantage captures relative element quality that image-level aggregation can conceal.

Coordinate-specific credit assignment. To guide region prediction, we assign A_{gi}^{box} to the coordinate-token positions $\mathcal{S}_{gi}^{\text{box}}$ of eligible matched elements, indexed by $\mathcal{I}_g$. This set excludes ambiguous matches and trusted boxes with IoU at least 0.95; all tokens retain image-level supervision. The final advantage for token t is

$$A_{g,t}^{\text{final}} = A_g^{\text{img}} + \sum_{i\in\mathcal{I}_g} \mathbf{1}[t \in \mathcal{S}_{gi}^{\text{box}}] A_{gi}^{\text{box}}. \tag{7}$$

Missed elements have no coordinate spans for local updates. We use $A_{g,t}^{\text{final}}$ in the GRPO clipped token objective, averaging coordinate contributions within each span and normalizing by N. Thus element-level feedback refines coordinates while image-level supervision covers the output.

4.3 Efficient Inference

Long outputs containing many coordinates and attributes make autoregressive parsing costly. Their repeated fields, quantized coordinates, and limited attribute vocabularies motivate speculative decoding: a lightweight predictor proposes multiple tokens, and the target model verifies them in parallel to accept a valid prefix (Leviathan et al., 2023; Chen et al., 2023). MTP (Gloeckle et al., 2024) proposes multiple future tokens through lightweight prediction layers, while DFlash (Chen et al., 2026) uses a block-diffusion drafter conditioned on target-model features. To improve draft–target agreement on field transitions and coordinate patterns, we adapt the DFlash drafter to GLA outputs while keeping the SFT target frozen. Both approaches aim to reduce sequential target-model passes. We evaluate net throughput gains, accounting for drafting and verification overhead in Section 5.4.

Table 2: **Main results on GLA-Bench.** General-purpose VLMs are evaluated zero-shot; the lower block reports task-specific post-training for two 4B backbones. Joint is Joint@.75. All values are percentages; CER can exceed 100%.

	bbox2text		text2bbox		img2layout		img2full			
Model	CER↓	EM↑	mIoU↑	mAcc↑	mF1↑	Joint↑	mF1↑	Cat.F1↑	Align.↑	Joint↑
General-purpose VLMs (zero-shot)										
Qwen2.5-VL-72B	225.95	20.18	42.49	19.40	11.40	4.06	9.85	56.45	75.57	4.05
Qwen3-VL-235B	58.38	68.50	48.41	27.78	15.08	6.74	13.33	57.28	78.00	6.18
InternVL3-78B	135.94	40.95	13.56	3.45	2.46	0.35	1.43	51.25	72.55	0.36
InternVL3.5-241B	93.25	53.86	44.81	24.28	8.05	2.71	7.30	54.21	70.83	2.45
Kimi-K2.6	264.90	54.90	48.90	28.81	15.03	6.55	13.89	66.00	73.70	6.26
Task-specific post-training										
Qwen3-VL-4B	68.71	61.94	47.15	24.24	11.25	0.00	11.16	0.00	57.69	0.00
Qwen3-VL-4B$_{+\text{SFT}}$	10.00	87.77	73.13	59.76	51.82	43.70	50.94	79.97	91.27	43.96
Qwen3-VL-4B$_{+\text{SFT}+\textbf{EAA-GRPO}}$	9.98	88.24	73.41	60.39	55.36	46.95	53.97	80.53	91.81	46.59
Qwen3.5-4B	127.64	48.10	33.21	14.05	7.84	0.00	0.17	0.00	48.86	0.00
Qwen3.5-4B$_{+\text{SFT}}$	13.71	86.39	66.05	47.47	40.41	33.88	38.95	78.64	93.02	32.93
Qwen3.5-4B$_{+\text{SFT}+\textbf{EAA-GRPO}}$	12.01	86.74	67.41	49.52	44.85	38.20	43.25	82.26	93.53	37.06

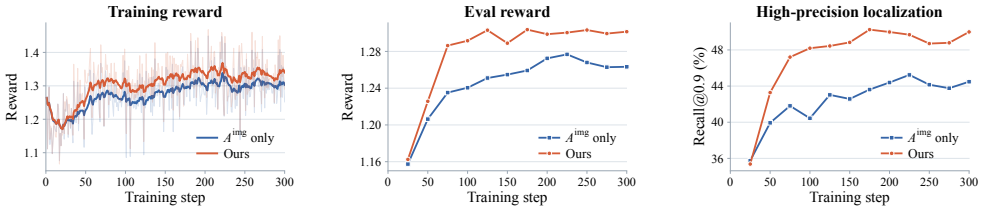


Figure 3: **RL training dynamics on Qwen3-VL-4B.** GRPO (A^{img} only) and EAA-GRPO (Ours) use the same reward. Left: training reward. Middle and right: evaluation reward and Recall@0.9 on 64 `img2layout` validation images held out from the training partition.

5 Experiments

5.1 Experimental Setup

We evaluate on GLA-Bench, which contains 1,000 human-annotated images across nine domains and 3,801 task instances. Training and benchmark images are disjoint. We follow Section 3.3, using Joint@.75 for main full-image comparisons. Scores are percentages; improvements are reported in percentage points. We adopt Qwen3-VL-4B (Bai et al., 2025a) and Qwen3.5-4B as backbones and use representative general-purpose VLMs (Bai et al., 2025b; Zhu et al., 2025; Wang et al., 2025) as zero-shot baselines. For each 4B backbone, Table 2 reports the base model, multi-task SFT, and SFT followed by EAA-GRPO. SFT uses all four tasks, while RL uses `img2layout` training instances.

5.2 Main Results

Full-image parsing. Table 2 compares zero-shot and post-trained models across the four tasks. After SFT and EAA-GRPO, Qwen3-VL-4B reaches 46.95 and 46.59 Joint@.75 on `img2layout` and `img2full`, respectively. The strongest evaluated zero-shot models on these tasks reach only 6.74 and 6.26, respectively. Qwen3.5-4B also exceeds both zero-shot results, reaching 38.20 and 37.06. SFT accounts for most of the improvement: on Qwen3-VL-4B, it raises full-parsing geometric mF1 from 11.16 to 50.94 and Joint@.75 from 0.00 to 43.96. Category macro-F1 and alignment accuracy also increase on matched elements. Together, these results show improvements in both allocated-region recovery and joint prediction of geometry, text, category, and alignment.

Element-level tasks. The element-level results in Table 2 show that SFT improves recognition and localization when the target region or text is supplied. For Qwen3-VL-4B, recognition EM rises from 61.94 to 87.77 and localization mAcc from 24.24 to 59.76; Qwen3.5-4B shows the same trend, gaining 38.29 points in EM and 33.42 in mAcc. These results show that the gains extend beyond element discovery to transcription and allocated-region localization for specified elements.

Table 3: **Localization in conventional LA and GLA.** We report mIoU (%) before and after SFT on `text2bbox`, with visible-text bounds for Conventional LA and allocated regions for GLA. Δ is GLA minus Conventional LA.

Model	Conventional LA	GLA	Δ
Qwen3-VL-4B	58.32	47.15	−11.17
Qwen3-VL-4B + SFT	61.47	**73.13**	+11.66
Qwen3.5-4B	39.34	33.21	−6.13
Qwen3.5-4B + SFT	57.44	**66.05**	+8.61

Table 4: **Ablation of RL components.** Results (%) on `img2full` with Qwen3.5-4B. GRPO w/o Det./Fmt. omits detection and format terms; GRPO and EAA-GRPO share the reward in Eq. 4.

Method	mF1↑	Cat. F1↑	Align.↑	Joint@.75↑
SFT	38.95	78.64	93.02	32.93
GRPO w/o Det./Fmt.	41.52	80.56	93.28	35.27
GRPO	41.81	80.64	93.39	36.01
EAA-GRPO (Ours)	**43.25**	**82.26**	**93.53**	**37.06**

Table 5: **MTP layer sweep on Qwen3.5-4B.** Score is Joint@.5 on `img2full`. TPS denotes tokens per second; speedup uses the MTP-off baseline in this table.

Setting	Score	TPS	Speedup
MTP off	57.18	218.03	1.00×
Layer 1	57.18	246.67	1.13×
Layer 2	57.07	322.81	1.48×
Layer 3	57.11	348.29	1.60×
Layer 4	57.12	**393.13**	**1.80**×
Layer 5	57.08	368.40	1.69×

Table 6: **DFlash decoding on GLA-SFT.** DFlash uses block-diffusion speculative decoding; TA adapts the drafter on task-specific data with the target fixed. Speedup uses the AR baseline in this table. Acc. rate / Acc. len. report the draft-token acceptance rate / mean accepted tokens per verification step.

Method	TPS	Speedup	Acc. rate	Acc. len.
AR	240.33	1.00×	–	–
DFlash	209.27	0.87×	8.63%	1.29
DFlash + TA	**524.64**	**2.18**×	**37.78%**	**5.67**

Effect of RL. The SFT+EAA-GRPO results in Table 2 improve layout-parsing Joint@.75 over SFT by 3.25 points on Qwen3-VL-4B and 4.32 on Qwen3.5-4B. Although RL uses only `img2layout` training instances, full-parsing Joint@.75 also increases by 2.63 and 4.13 points, respectively. Full-parsing geometric mF1 rises by 3.03 and 4.30 points, alongside gains in category macro-F1 and alignment accuracy on matched elements. These improvements show transfer from layout-only RL to full parsing and are consistent with better structural prediction shared by the two tasks.

5.3 Ablation Studies

Generation-aware localization. Table 3 compares visible-text and allocated-region localization on the same `text2bbox` queries, each evaluated against its corresponding reference regions. Before SFT, both backbones achieve higher mIoU against visible-text bounds. SFT raises Qwen3-VL-4B's allocated-region mIoU from 47.15 to 73.13 (+25.98 points), versus +3.15 for visible-text localization. Qwen3.5-4B shows the same pattern, gaining 32.84 points on allocated regions versus 18.10 on visible-text bounds. Allocated-region mIoU exceeds visible-text mIoU after SFT on both backbones. The larger gains for allocated regions support the learnability of spatial allocation.

RL ablation. Table 4 evaluates Qwen3.5-4B on `img2full`, with RL training restricted to `img2layout`. EAA-GRPO raises geometric mF1 from 38.95 to 43.25 over SFT and, with the reward fixed, exceeds GRPO by 1.44 mF1 and 1.05 Joint@.75 points. Complementing these benchmark results, Figure 3 tracks Qwen3-VL-4B over 300 RL steps, with evaluation on 64 `img2layout` validation images held out from the training set. EAA-GRPO achieves higher evaluation reward from step 50 onward and reaches approximately 50% Recall@0.9 at step 300, versus 45% for GRPO. Together, these results show that layout-only RL improves full-parsing localization, while element-level feedback yields additional gains in geometric accuracy and high-IoU recall.

5.4 Inference Efficiency

Task-adapted drafting. Table 6 compares autoregressive decoding, a generic DFlash drafter, and its task-adapted counterpart with the SFT target fixed. The generic drafter achieves 209.27 TPS, below the autoregressive baseline of 240.33 TPS (0.87×). Task adaptation raises throughput to 524.64 TPS (2.18× the baseline), draft-token acceptance from 8.63% to 37.78%, and mean accepted length from 1.29 to 5.67 tokens per verification step. Higher acceptance is consistent with better draft–target agreement. Here, task adaptation turns a net slowdown into a throughput gain.

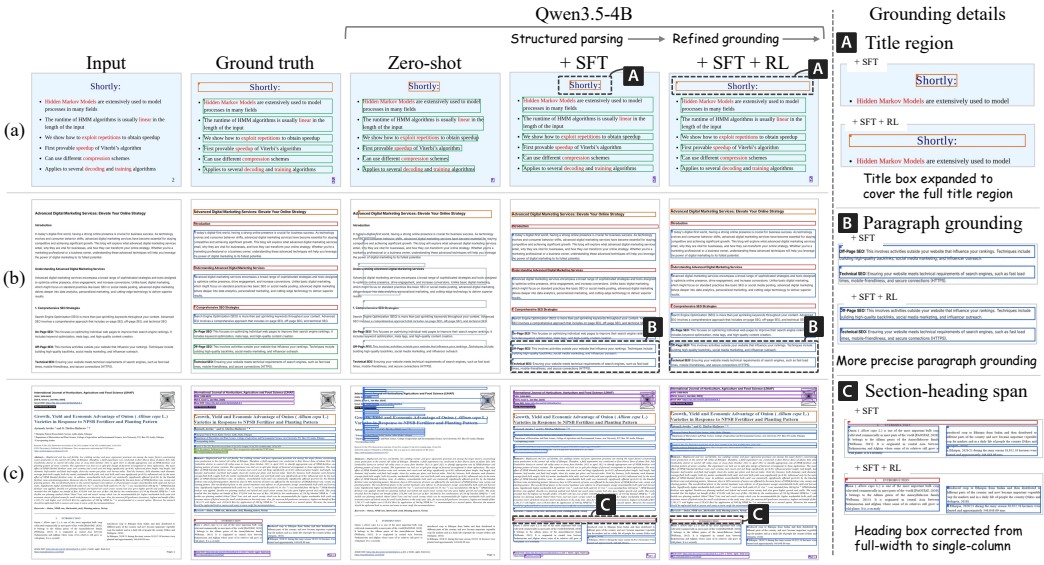


Figure 4: **Qualitative comparison of GLA.** Qwen3.5-4B predictions in the zero-shot, SFT, and SFT+RL settings are shown alongside the inputs and ground truth for (a) a slide, (b) a single-column document, and (c) a scientific article. Insets A–C highlight the recovery of the allocated title region, the refinement of paragraph boundaries, and the correction of a section-heading span to its column.

Multi-token prediction. Table 5 reports the MTP layer sweep on Qwen3.5-4B. Across one to five layers, Joint@.5 ranges from 57.07 to 57.18, within 0.11 points of the MTP-off score of 57.18. Four layers achieve the highest measured throughput: 393.13 TPS versus 218.03 TPS for the baseline (1.80×), with Joint@.5 of 57.12. A fifth layer reduces throughput to 368.40 TPS (1.69×), while the score remains close at 57.08. Thus, increasing prediction depth does not monotonically improve speed; four layers provide the best observed throughput with little change in Joint@.5 in this sweep.

5.5 Qualitative Analysis

Figure 4 compares zero-shot, SFT, and SFT+RL predictions of Qwen3.5-4B against the ground-truth allocated regions. SFT produces more coherent element regions on the dense document pages, organizing the title, abstract, and two-column text. RL further refines these regions according to their layout context. In example (a), the box around "Shortly:" expands from the visible text bounds to the full title region, recovering the horizontal space allocated to the heading. In example (b), RL adjusts the paragraph boundaries to include the wrapped lines highlighted in inset B. In example (c), it corrects the "Introduction" heading from a page-wide span to the left column. Together, these examples show how SFT and RL adapt region extents to layout structure, providing spatial constraints that could support subsequent content generation.

6 Conclusion

We introduce GLA to recover text, allocated regions, semantic categories, and alignment as spatial constraints for editing. Our training set and GLA-Bench support learning and evaluation across nine domains. Multi-task SFT and EAA-GRPO yield a 4B parser reaching 46.59 Joint@.75 in full parsing. Explicit supervision improves allocated-region recovery, while gains from layout-only RL extend to full parsing on both backbones, suggesting shared structural capabilities. EAA-GRPO's gains over GRPO support element-level coordinate feedback. Task-adapted drafting achieves 2.18× autoregressive throughput with the SFT target fixed, highlighting the value of adapting decoding to structured outputs. These findings support spatial allocation as a learnable intermediate representation for future editing and translation systems.

AI USE STATEMENT

In this work, we used generative AI tools to assist with manuscript writing and polishing and to construct HTML-rendered training data and candidate benchmark annotations. For manuscript preparation, these tools helped organize the text, refine the language, and improve clarity, readability, and consistency. For dataset construction, generative AI tools produced HTML reconstructions and candidate structured annotations, as described in Section 3.2. The generated training annotations underwent checks for parseability, nonempty text fields, valid coordinates, and supported attribute values. Candidate annotations for GLA-Bench underwent comprehensive manual correction and cross-checking, including verification of text, regions, categories, and alignment, as well as correction of omissions and removal of duplicates. The authors reviewed all AI-assisted manuscript revisions and carefully verified the data generation and quality control procedures described above. We take full responsibility for the final content of this work, including AI-assisted text and data and all associated research claims.

ETHICS STATEMENT

This work aims to support layout understanding and reconstruction of editable visual documents. The data cover web and mobile interfaces, slides, documents, infographics, and charts. We consider data provenance and usage permissions, potential privacy risks, and errors and biases that may arise from automatically generated annotations. During the research, we reviewed the terms of use of each data source and identified and processed content containing personal information to mitigate privacy risks. Before releasing the data, we plan to further review the applicable redistribution permissions and usage restrictions and conduct additional privacy checks on the proposed release. The content and scope of the release will be determined based on these reviews.

To improve data quality, we apply rule-based filtering to the training annotations and perform comprehensive manual correction and cross-checking of the benchmark annotations to reduce automatic annotation errors. Nevertheless, differences in source distributions, subjectivity in region annotation, and biases in the generative models may still affect data quality and evaluation results.

Document layout analysis and reconstruction technologies could also be used for unauthorized information extraction or misleading content manipulation. We advocate responsible use of our research outputs that respects the intellectual property rights associated with the original content, protects personal privacy, and complies with applicable terms of use. Reconstructed or edited content should undergo appropriate verification in practical applications.

REPRODUCIBILITY STATEMENT

To support reproducibility, Section 3 describes the annotation schema, data construction pipeline, dataset splits, evaluation tasks, and metrics. Section 4 presents multi-task supervised fine-tuning, reinforcement learning with element-level advantage assignment, and efficient inference mechanisms. Appendix C.1 provides the hardware environment and numerical precision, key SFT and RL training configurations, MTP and DFlash training and inference settings, and the timing boundaries, throughput calculations, and corresponding autoregressive baselines for each acceleration experiment. Upon acceptance, we will release the training and inference code, trained model weights, GLA-Bench, and training data, together with annotation guidelines, fixed dataset split identifiers, prompts, configuration files, and evaluation scripts, to facilitate independent evaluation and reproduction of our results. Third-party data will be released in accordance with applicable licenses and redistribution conditions.

REFERENCES

Gilles Baechler, Srinivas Sunkara, Maria Wang, Fedir Zubach, Hassan Mansoor, Vincent Etter, Victor Cărbune, Jason Lin, Jindong Chen, and Abhanshu Sharma. Screenai: A vision-language model for ui and infographics understanding. *arXiv preprint arXiv:2402.04615*, 2024.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

Jian Chen, Yesheng Liang, and Zhijian Liu. Dflash: Block diffusion for flash speculative decoding. *arXiv preprint arXiv:2602.06036*, 2026.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022.

Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th annual ACM symposium on user interface software and technology*, pp. 845–854, 2017.

Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction. *arXiv preprint arXiv:2404.19737*, 2024.

Xinheng Han, Jianfei Wang, Yu Chen, Xiang Wang, Shuai Li, Weixing Li, and Feng Pan. Credit the right box: Marginal contribution assignment for structured visual perception. *arXiv preprint arXiv:2608.01055*, 2026.

Adam Hazimeh, Ke Wang, Mark Collier, Gilles Baechler, Efi Kokiopoulou, and Pascal Frossard. Semantic document derendering: Svg reconstruction via vision-language modeling. *arXiv preprint arXiv:2511.13478*, 2025.

Geewook Kim, Teakgyu Hong, Moonbin Yim, Jeongyeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. OCR-free document understanding transformer. In *European Conference on Computer Vision*, 2022.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*, 2023.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, 2023.

Gengluo Li, Xingyu Wan, Shangpin Peng, Weinong Wang, Hao Feng, Yongkun Du, Binghong Wu, Zheng Ruan, Zhiqiong Lu, Liang Wu, et al. Hunyuanocr-1.5: Making lightweight OCR VLMs faster and better. *arXiv preprint arXiv:2607.04884*, 2026.

Yumeng Li, Guang Yang, Hao Liu, Bowen Wang, and Colin Zhang. dots.ocr: Multilingual document layout parsing in a single vision-language model. *arXiv preprint arXiv:2512.02498*, 2025.

Ilya Loshchilov, Frank Hutter, et al. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5(5):5, 2017.

Yadong Lu, Jianwei Yang, Yelong Shen, and Ahmed Awadallah. Omniparser for pure vision based gui agent. *arXiv preprint arXiv:2408.00203*, 2024.

Zhenyu Pan and Han Liu. Metaspatial: Reinforcing 3d spatial reasoning in vlms for the metaverse. In *International Conference on Learning Representations*, volume 2026, pp. 117974–117993, 2026.

Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S Nassar, and Peter Staar. Doclaynet: A large human-annotated dataset for document-layout segmentation. pp. 3743–3751, 2022.

Zhipeng Qian, Pei Zhang, Baosong Yang, Kai Fan, Yiwei Ma, Derek F. Wong, Xiaoshuai Sun, and Rongrong Ji. Anytrans: Translate anytext in the image with large scale models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 2432–2444, 2024.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

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

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, Ruochen Xu, and Tiancheng Zhao. VLM-R1: A stable and generalizable R1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025a.

Xiaoqian Shen, Min-Hung Chen, Yu-Chiang Frank Wang, Mohamed Elhoseiny, and Ryo Hachiuma. Zoom-zero: Reinforced coarse-to-fine video understanding via temporal zoom-in. *arXiv preprint arXiv:2512.14273*, 2025b.

Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. Slidevqa: A dataset for document visual question answering on multiple images. *arXiv preprint arXiv:2301.04883*, 2023.

Yuxiang Tuo, Yifeng Geng, and Liefeng Bo. Anytext2: Visual text generation and editing with customizable attributes. *arXiv preprint arXiv:2411.15245*, 2024.

Jordy Van Landeghem, Ruben Tito, Lukasz Borchmann, Michal Pietruszka, et al. Document understanding dataset and evaluation (DUDE). *arXiv preprint arXiv:2305.08455*, 2023.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Haoran Wei, Chenglong Liu, Jinyue Chen, Jia Wang, Lingyu Kong, Yanming Xu, Zheng Ge, Liang Zhao, Jianjian Sun, Yuang Peng, Chunrui Han, and Xiangyu Zhang. General OCR theory: Towards OCR-2.0 via a unified end-to-end model. *arXiv preprint arXiv:2409.01704*, 2024.

Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. Publaynet: Largest dataset ever for document layout analysis. *arXiv preprint arXiv:1908.07836*, 2019.

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

Appendix

A METHODS DETAILS

This section supplements the model design in Section 4. We describe the shared training formulation, the element-level feedback in EAA-GRPO, and the two speculative decoding approaches. Hardware, hyperparameters, and measurement settings are given in Appendix C.1.

A.1 SHARED MULTI-TASK FORMULATION

A single instruction-conditioned VLM is trained jointly on `bbox2text`, `text2bbox`, `img2layout`, and `img2full`. The first two tasks supervise regional text recognition and allocated-region localization; the full-image tasks supervise element discovery and structured field prediction. As in Section 4.1, SFT minimizes the autoregressive next-token loss using paraphrased instructions and a fixed field order. The resulting checkpoint initializes RL, which uses only `img2layout` training instances. The annotation schema and task interfaces are detailed in Appendices B.3 and B.5.

A.2 ELEMENT-LEVEL FEEDBACK IN EAA-GRPO

Matching and reward construction. For each image, the policy samples G outputs. Partial one-to-one assignment using normalized L1 and generalized-IoU costs associates each output with reference elements. The composite element reward R_{gi}^{elem} combines thresholded IoU, category correctness, and alignment correctness, as defined in Eq. 3. Missed elements receive zero reward. Averaging over all N reference elements and adding detection and format terms produces R_g^{img} in Eq. 4.

Comparisons across rollouts. Image-level GRPO normalizes R_g^{img} across the sampled outputs. EAA additionally compares the composite rewards for the same reference element across $\mathcal{V}_i$, which includes unambiguous matches and zero-reward misses while excluding ambiguous assignments. Its element-level advantage A_{gi}^{box} follows Eq. 6. This comparison isolates differences in local element quality that image-level aggregation can obscure.

Coordinate-specific updates. The local advantage is added only at the coordinate-token positions of eligible matched elements, while every output token retains the shared image-level advantage, as in Eq. 7. The eligible set excludes ambiguous matches and trusted boxes with IoU at least 0.95. Missed elements contribute zero rewards to the across-rollout comparison but have no coordinate spans for local updates. The combined advantage enters the clipped GRPO token objective; coordinate contributions are averaged within each span and normalized by N. Category and alignment correctness affect the composite element reward, but the additional local advantage is assigned to coordinates.

Illustrative example. Figure 5 shows two elements and two illustrative rollouts. Category and alignment are correct in both rollouts, so $R_{gi}^{\text{elem}} = U_{gi} + 0.2$. IoUs of 0.82 and 0.62 yield composite rewards of 1.4 and 0.5, respectively. The title and URL swap these rewards between rollouts. Their summed rewards are therefore equal, but their element-level advantages have opposite signs: rollout 1 receives positive local feedback for the title and negative feedback for the URL, while rollout 2 receives the reverse. This controlled example illustrates how EAA distinguishes local quality despite equal summed rewards for the illustrated elements.

A.3 SPECULATIVE DECODING

As described in Section 4.3, MTP proposes future tokens through prediction layers, while DFlash uses a block-diffusion drafter conditioned on target-model features. The target model verifies drafted continuations in parallel. Task adaptation trains the DFlash drafter on `img2layout` and `img2full` outputs while keeping the SFT target fixed, aiming to improve agreement on repeated

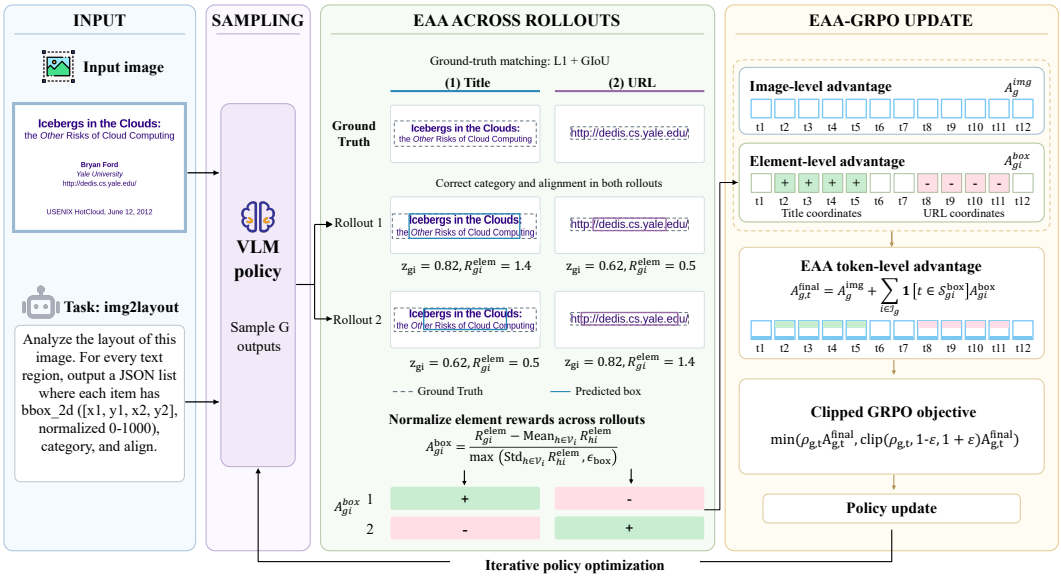


Figure 5: **Element-level Advantage Assignment in EAA-GRPO.** A real image is paired with controlled, illustrative region predictions. After reference matching, EAA compares composite rewards for the same element across rollouts. The update panel shows rollout 1: blue outlines represent shared image-level feedback, while green and pink highlights represent positive and negative local feedback on coordinate tokens. Token strips are schematic positions rather than literal tokenizer boundaries. The combined advantage enters the clipped GRPO objective; $\rho_{g,t}$ is the current-to-old token probability ratio and ϵ is the clipping parameter. Only two elements are shown.

fields and coordinate patterns. Appendix C.1 gives the training and timing settings used for the acceleration experiments.

B GLA-Bench Details

This section expands on Section 3, covering source composition, allocated-region criteria, the shared annotation schema, human verification, and task interfaces. Experimental prompts and human model-comparison results appear in Appendices C.2 and C.3, respectively.

B.1 Source Domains and Data Splits

Table 7 reports the nine sources: Web UI and Rico (Deka et al., 2017) provide web and mobile interfaces; SlideVQA (Tanaka et al., 2023), SlideShare, and SlideGen provide slides; DUDE (Van Landeghem et al., 2023) provides document pages; and the remaining sources cover infographics, charts, and website screenshots.

Before annotation, 1,000 of the 38,418 source images are reserved by domain-stratified sampling for GLA-Bench; 37,418 form the training partition. All rendered images, annotations, and task instances inherit their source-image split, preventing overlap between training and evaluation across the four tasks.

HTML rendering and image provenance. During dataset construction, we convert source images into HTML representations, render the HTML, and save the results as new raster images. These HTML-rendered images and their automatically generated annotations form the training set. The nine domains describe the provenance of the source content; the training images are the rendered HTML reconstructions. All rendered images and derived task instances retain the split assigned to their source image. This process complements the annotation refinement and benchmark verification in Section 3.2.

Table 7: **Source composition of the training set and GLA-Bench.** Counts refer to source images. The 37,418 training images and 1,000 benchmark images are disjoint. All task instances inherit their source-image split.

Source	Training	GLA-Bench	Total
Web UI	11,130	264	11,394
Rico (Mobile UI)	7,960	195	8,155
SlideVQA	4,618	122	4,740
DUDE	3,853	105	3,958
Infographics	3,161	89	3,250
Charts	2,974	85	3,059
SlideShare	1,673	56	1,729
SlideGen	1,197	46	1,243
Website screenshots	852	38	890
Total	37,418	1,000	38,418



(a) (b) (c)

Figure 6: **Overview of the training data and GLA-Bench.** (a) Source-domain composition of the full image pool. (b) Word cloud summarizing content in GLA-Bench. (c) Distribution of annotated bounding boxes per image in GLA-Bench, with counts above 100 grouped into the final bin.

Figure 6 shows source proportions, benchmark text content, and the long-tailed distribution of elements per image in GLA-Bench.

B.2 Allocated Regions Across Domains

Annotation target. An allocated region records the space assigned to an element by the surrounding design. Explicit containers, panels, cells, and separators take precedence over repeated grids or columns, followed by neighboring elements, whitespace, and alignment. When several extents are plausible, annotation selects a conservative region supported by these cues, without extending into unconstrained whitespace. This available space permits replacement content while preserving the composition. Visible content is usually contained within its allocated region, but clipping, rotation, overflow, and decorative rendering can violate containment, as discussed in Section 3.1.

Cross-domain examples. Figures 7 and 8 contrast visible-text bounds with allocated regions for one image from each source. The difference is especially clear in short text placed inside a wider container: the SlideShare title follows the gray panel, Rico labels follow interface controls, and the SlideGen diagram text follows its nodes. Document columns and webpage sidebars constrain the horizontal extent of prose, while chart labels and values occupy regions determined by the surrounding chart structure. These examples show why allocation depends on the local design rather than a fixed padding rule. Both representations describe the same source image; the comparison isolates the spatial information supplied by GLA.

B.3 Semantic Categories and Alignment

Semantic categories identify an element's role in the visual composition, while alignment specifies how its text is placed within the allocated region. Both attributes use a shared vocabulary across all nine source domains and both full-image parsing tasks.

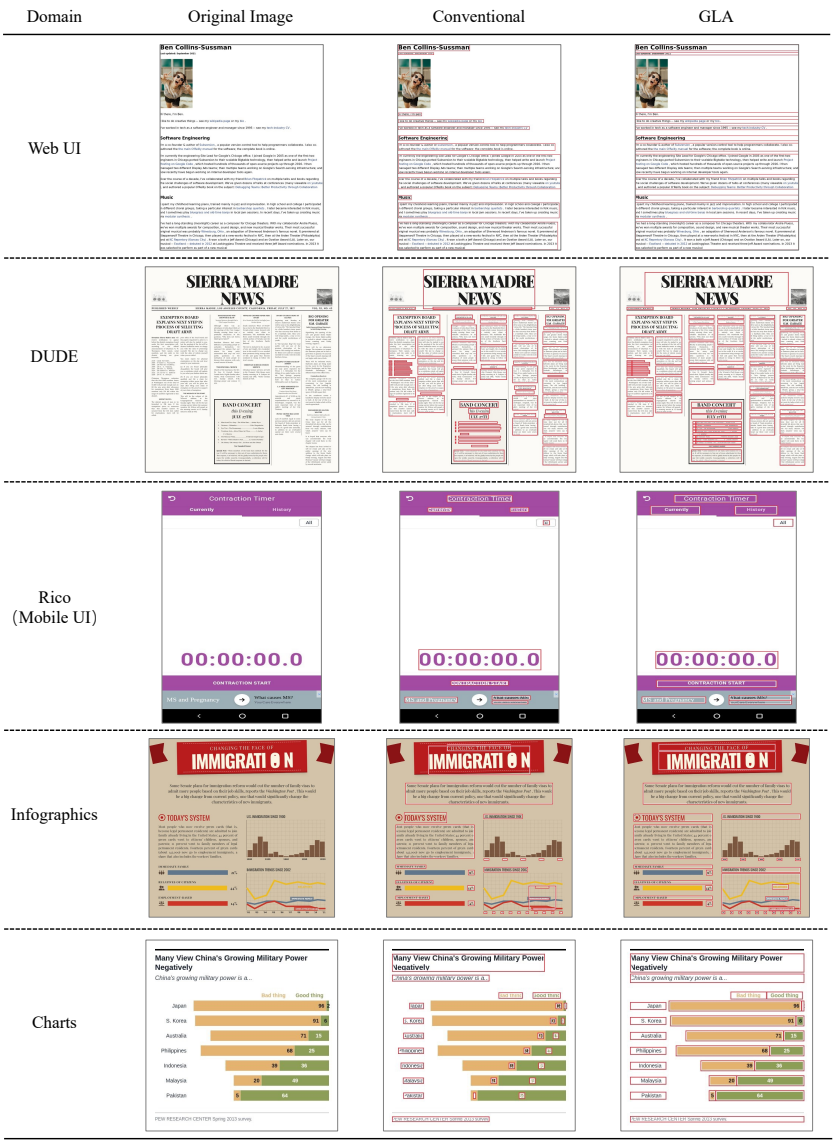


Figure 7: **Visible-text bounds and allocated regions across five domains.** Each row shows the original image, conventional visible-text bounds, and GLA allocated regions. From top to bottom: Web UI, DUDE, Rico, Infographics, and Charts. Columns, interface controls, infographic panels, and chart components determine the space assigned to each text element.

Semantic categories. The 13 categories are defined below in the same order as the panels in Figure 9. Category labels retain the exact capitalization, hyphens, and underscores shown here in serialized outputs.

- **Title.** The main heading of a page or content block, identifying its central topic.
- **Section-header.** A heading or subheading that introduces a section, column, or content group. This category also includes form field names and descriptive interface labels, such as "Email."
- **Text.** Body paragraphs, descriptions, and other content text, including field values associated with form or interface labels.
- **List-item.** An entry in a bulleted or numbered list, or one of a sequence of parallel entries whose arrangement conveys a list.

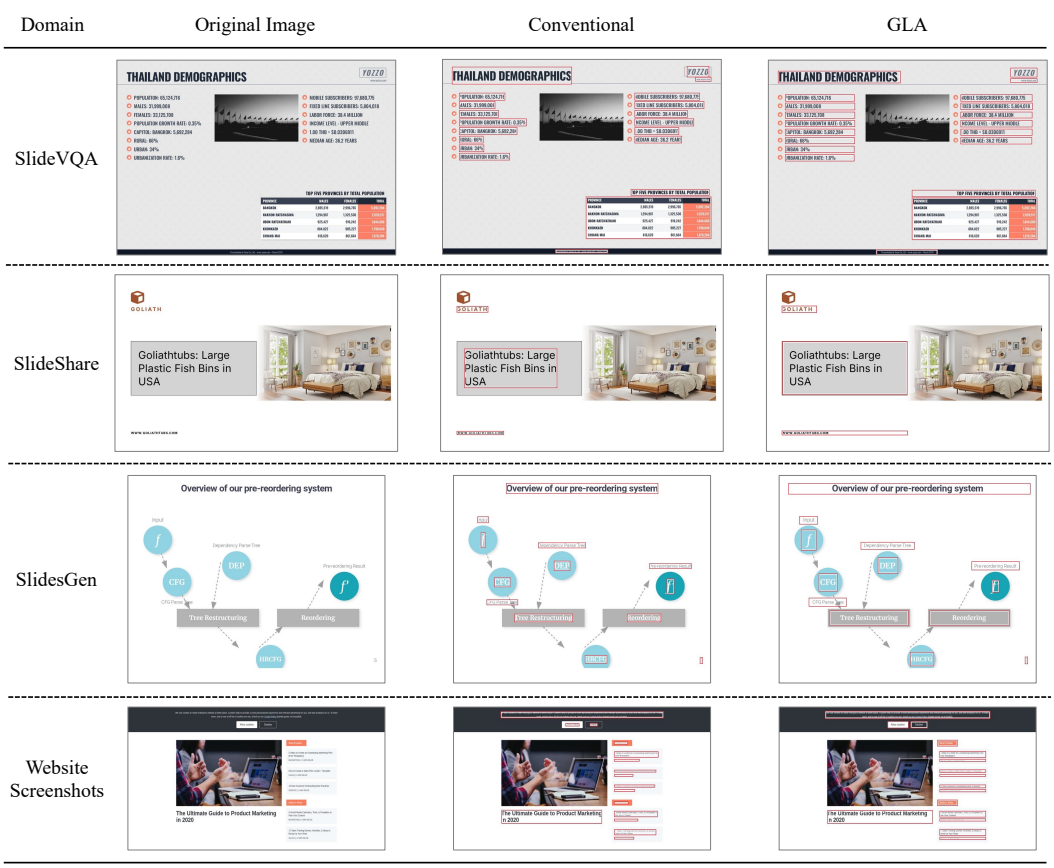


Figure 8: **Visible-text bounds and allocated regions across four further domains.** The columns follow Figure 7. From top to bottom: SlideVQA, SlideShare, SlideGen (labeled SlidesGen in the image), and website screenshots. Slide columns, explicit text containers, diagram nodes, and webpage components provide allocation cues. In the SlideShare example, the gray panel defines the title's available region.

- **Caption.** Text describing an associated figure, table, illustration, or other visual, typically placed adjacent to that visual.
- **Footnote.** A supplementary note, clarification, or source attribution associated with the page or one of its visual elements.
- **Page-header.** Information in a page-level header or top bar, including text in interface headers and mobile status bars.
- **Page-footer.** Information in a page-level footer, including copyright notices, legal statements, and footer URLs.
- **Table.** Text within table headers and cells, organized by the table's row-and-column structure.
- **Formula.** A mathematical or chemical expression or equation presented as a text element.
- **ui_button.** Action text on a clickable button or link, such as "Submit" or "Sign up."
- **chart_text.** Text within a chart or diagram, including internal titles, axis and tick labels, legends, data values, and node labels.
- **Other.** Text outside the preceding semantic roles, or text whose role cannot be determined from the visual context.

Category assignment follows the element's function in context. A title inside a chart is `chart_text`, whereas an external description of that chart is a `Caption`. Within a form, `Section-header` identifies the field name, `Text` its value, and `ui_button` the submit action. These distinctions give the same visual structure a consistent semantic interpretation across domains.

Alignment. The four alignment values describe horizontal text placement relative to the element's own allocated region:

- **left.** Text is aligned to the region's left edge.
- **center.** Text is centered about the region's horizontal midpoint.
- **right.** Text is aligned to the region's right edge.
- **justify.** Text spacing is adjusted so that full lines span the region's width and align with both edges.

A button in a left-hand sidebar can therefore have `center` alignment when its label is centered within the button's region. Region geometry and alignment jointly specify the space available for replacement content and its placement within that space.

Distribution of semantic roles and alignment. Category and alignment distributions characterize two complementary aspects of the benchmark: the roles assigned to text and the placement constraints associated with those roles. Figure 10 presents the distribution visualization. Reporting category frequencies alongside category macro-F1 distinguishes the prevalence of a role from its contribution to category-level evaluation; reporting alignment frequencies provides context for interpreting alignment accuracy.

B.4 Human Annotation and Quality Control

Benchmark verification. Manual correction and cross-checking cover transcription, allocated regions, category, alignment, element coverage, and reading order. Reviewers add omissions and remove duplicate or spurious elements. The verified benchmark contains 19,858 elements across 1,000 images, averaging 19.9 elements per image, and supplies the shared reference for all four tasks.

Resolving reference-annotation conflicts. Disputed reference annotations are reviewed by additional independent annotators under the same guidelines, without access to previous votes. A strict-majority vote selects the annotation. If no candidate obtains a majority, an expanded reviewer pool votes on the leading alternatives; persistent ties proceed to expert adjudication using the allocation criteria. The retained annotation and decision rationale are recorded alongside the text, region, category, and alignment.

Reference-annotation review establishes the benchmark targets. The separate human evaluation of model outputs is described in Appendix C.3.

B.5 Task Interfaces and Serialized Outputs

The four tasks derive their targets from the same element annotation, linking recognition and localization to complete structured parsing. Figure 11 instantiates all four on one webpage. For `bbox2text`, the input is the image and a reference region, and the output is its text: the highlighted field label yields `Name`. For `text2bbox`, the input is the image and a text query: "Take Action Petition" yields $[731, 89, 979, 110]$. The full-image tasks discover all elements. `img2layout` returns an array of records with `bbox_2d`, `category`, and `align`; `img2full` adds `text` to each record.

Coordinates refer to the entire image. A pixel position (u, v) in an image of width W and height H is serialized as

$$x = \mathrm{round}\Big(1000\frac{u}{W}\Big), \qquad y = \mathrm{round}\Big(1000\frac{v}{H}\Big). \tag{8}$$

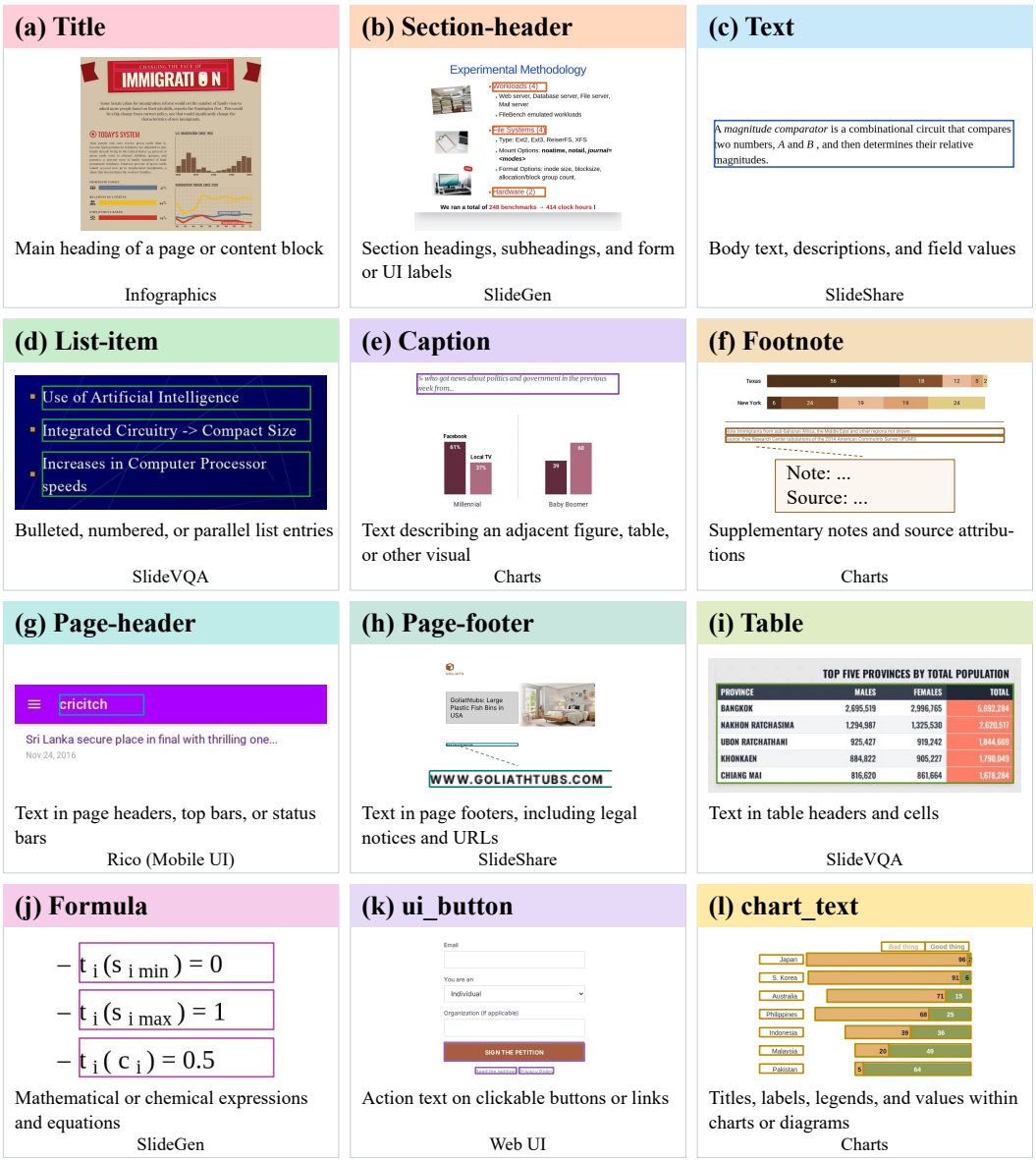


Figure 9: **The 13 semantic categories in GLA-Bench.** Panels (a–l) show representative crops with regions highlighted for the category named in each panel. Panel (f) calls out note and source prefixes; panel (h) enlarges a footer. Panel (m) defines `Other`.

Each box contains four integers $[x_1, y_1, x_2, y_2]$ in $[0, 1000]$, with $x_1 \leq x_2$ and $y_1 \leq y_2$. Annotation records follow reading order. Category and alignment values use the vocabulary in Appendix B.3.

Evaluation protocol. The official task metrics follow Section 3.3. `bbox2text` uses CER and EM, and `text2bbox` uses mIoU and mAcc. Full-image predictions are matched one-to-one using geometry only. Category macro-F1 and alignment accuracy are evaluated on matched elements. Joint@τ requires IoU of at least τ and correct category and alignment; `img2full` additionally requires exact agreement between normalized predicted and reference text. Full-image metrics are computed per image and then averaged. Appendix C.4 illustrates how category and transcription errors can yield zero joint correctness despite high IoU.

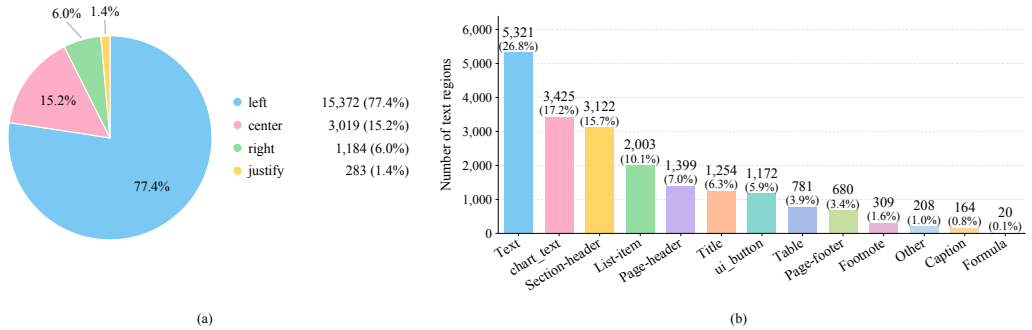


Figure 10: **Alignment and semantic-category distributions in GLA-Bench.** Statistics cover 19,858 annotated text regions across 1,000 images. (a) Frequencies of the four alignment values. (b) Frequencies of the 13 semantic categories, with counts and percentages above the bars.

C Experiment Details

This section supplements Section 5 with training and inference settings, prompt specifications, human evaluation, and additional quantitative and qualitative diagnostics.

C.1 Training and Inference Setup

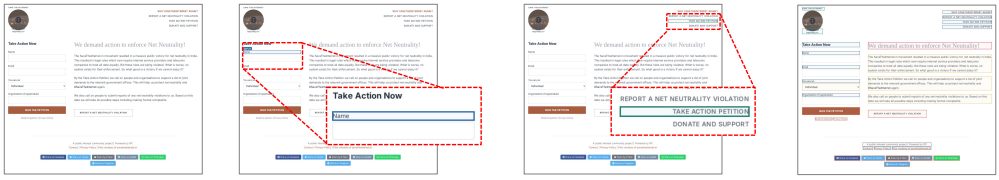
Supervised fine-tuning. We initialize the GLA parser with Qwen3-VL-4B-Instruct and Qwen3.5-4B, respectively. Training uses eight NVIDIA H20 GPUs, each with 96 GB of memory, together with bfloat16 mixed precision, FlashAttention (Dao et al., 2022), and gradient checkpointing. During SFT, we fine-tune all parameters of the language model while keeping the vision encoder frozen. Both models are trained for one epoch on the combined data from all four tasks, with an effective batch size of 64 and a maximum sequence length of 8,192 tokens. We use AdamW (Loshchilov et al., 2017) with an initial learning rate of 1×10^{-5}, a weight decay of 0.1, and a cosine learning-rate schedule. The warmup ratios are 3% for Qwen3-VL-4B and 5% for Qwen3.5-4B.

Reinforcement learning. RL starts from the corresponding SFT checkpoint and uses only `img2layout` training instances, with both the vision encoder and visual projector frozen. We use AdamW with a learning rate of 1×10^{-6}, cosine scheduling, and a warmup ratio of 10%. Each optimization update comprises 32 images, with 16 outputs sampled per image at a temperature of 0.9 and a maximum generation length of 4,096 tokens. The GRPO clipping parameter and KL regularization coefficient are set to 0.2 and 0.04, respectively. EAA uses a standard-deviation floor of 0.05 for element-level advantage normalization. Following Section 4.2, trusted matched boxes with IoU at least 0.95 are excluded from additional coordinate-level updates.

For the Qwen3-VL-4B training-dynamics comparison in Figure 3, we reserve 64 images from the training partition for `img2layout` validation. The evaluation reward and Recall@0.9 curves are measured on this subset, not on GLA-Bench.

Speculative decoding. We conduct both MTP and DFlash acceleration experiments on Qwen3.5-4B. For MTP, auxiliary prediction training is performed jointly during SFT with a loss weight of 0.1; Table 5 reports the one-to-five-layer sweep. DFlash uses a separate Qwen3.5-4B SFT checkpoint trained without MTP as its target model. Its drafter is initialized from a pretrained DFlash model and adapted using `img2layout` and `img2full` data while keeping the target model frozen. The drafter is trained for one epoch using AdamW with a learning rate of 2×10^{-5}.

Throughput measurement. Acceleration experiments are conducted on H20 GPUs with one concurrent request. For each method, the AR baseline uses the same target checkpoint with speculative decoding disabled. MTP uses a single H20 GPU, bfloat16 precision, and vLLM 0.19.1 (Kwon et al., 2023). All inference configurations use greedy decoding with a maximum generation length of 4,096 tokens. Throughput is computed as the total number of output tokens divided by the wall-clock time



(a) Original Image (b) bbox2text (c)text2bbox (d)img2layout / img2full

Bbox2text Prompt

What is the text content inside the region [21,272,312,292]? The coordinates are normalized to 0-1000. Output only the text.

Bbox2text Output

Name

Text2bbox Prompt

Find the region that contains the text "Take Action Petition" and output its bounding box as [x1, y1, x2, y2], normalized to 0-1000.

Text2bbox Output

[731,89,979,110]

Img2layout Prompt

Analyze the layout of this image. For every text region, output a JSON list where each item has bbox_2d ([x1, y1, x2, y2], normalized 0-1000), category, and align..

Img2layout Output

```
[
  {"bbox_2d":[21,13,133,30],"category":"Page-header","align":"center"},
  {"bbox_2d":[731,32,979,50],"category":"Section-header","align":"left"},
  {"bbox_2d":[731,61,979,81],"category":"Section-header","align":"left"},
  {"bbox_2d":[731,89,979,110],"category":"Section-header","align":"left"},
  {"bbox_2d":[731,118,979,140],"category":"Section-header","align":"left"},
   ......

  {"bbox_2d":[436,923,564,951],"category":"ui_button","align":"center"}
]
```

Img2full Prompt

Perform full layout-aware parsing of this image. For every text region output a JSON list where each item has text, bbox_2d ([x1, y1, x2, y2], normalized 0-1000), category and align.

Img2full Output

```
[
  {"text":"Save The Internet","bbox_2d":[21,13,133,30],"category":"Page-header","align":"center"},
  {"text":"Why Savetheinternet Again?","bbox_2d":[731,32,979,50],"category":"Section-header","align":"left"},
  {"text":"Report a Net Neutrality Violation","bbox_2d":[731,61,979,81],"category":"Section-header","align":"left"},
  {"text":"Take Action Petition","bbox_2d":[731,89,979,110],"category":"Section-header","align":"left"},
  {"text":"Donate and Support","bbox_2d":[731,118,979,140],"category":"Section-header","align":"left"},
   ......

  {"text":"Share on Telegram","bbox_2d":[436,923,564,951],"category":"ui_button","align":"center"}
]
```

Figure 11: **Four task interfaces on the same image.** The upper panels show the source image, the region queried by `bbox2text`, the text localized by `text2bbox`, and the full-image annotation. The lower panels pair each instruction with its target output. Regional tasks return text or a coordinate array; full-image tasks return element records. Ellipses abbreviate intermediate records for display and are not part of the serialized outputs.

over the measurement interval. For MTP, timing includes inference-process initialization, data processing, and generation. For DFlash, timing begins after server initialization and warmup and covers request processing through response delivery, including vision encoding, prefill, and decoding. We report speedups relative to the respective AR baseline for each experiment.

C.2 Zero-shot Prompt Specifications

Figures 12–14 present excerpts of task instructions and expanded zero-shot templates, emphasizing category definitions and output conventions. GLA's category vocabulary and output conventions are task-specific. During supervised training, the parser receives annotated examples of these con-

System instruction for zero-shot evaluation

You are a meticulous document layout analysis assistant. Always follow the user's exact output format and emit only the requested JSON or value.

bbox2text | Region text recognition

Canonical user prompt

<image>
What is the text content inside the region {bbox}? The coordinates are normalized to 0-1000. Output only the text.

Zero-shot user prompt (complete) General-purpose models

<image>
What is the text content inside the region {bbox}? The coordinates are normalized to 0-1000. Output only the text.

Coordinates are normalized to [0,1000] relative to the entire image. Please ****output only the original text within the region itself****, without quotation marks, explanations, or Markdown.

text2bbox | Text grounding

Canonical user prompt

<image>
Find the region that contains the text "{text}" and output its bounding box as [x1, y1, x2, y2], normalized to 0-1000.

Zero-shot user prompt (complete) General-purpose models

<image>
Find the region that contains the text "{text}" and output its bounding box as [x1, y1, x2, y2], normalized to 0-1000.

Coordinates are normalized to [0,1000] relative to the entire image (0 = left/top, 1000 = right/bottom). Please ****output only four integers****, strictly in the format [x1, y1, x2, y2], without any additional text or Markdown. ****Coordinates must be integers between 0 and 1000; decimals, 0-1 fractions (e.g., [0.0, 0.1, ...] is incorrect), and percentages are prohibited.****

Figure 12: **Regional task prompts.** Task-level and expanded zero-shot excerpts are shown with the shared system instruction. `<image>` denotes the separately supplied image; `{bbox}` and `{text}` are instance-specific placeholders. Chinese portions are translated into English for presentation.

ventions. Zero-shot evaluation does not assume such familiarity: requesting a category field alone leaves the taxonomy and label names unspecified. The expanded prompts therefore state these requirements at inference time, giving general-purpose models an explicit specification of the expected output.

For the full-image tasks, the templates define all 13 categories, prescribe four alignment values, and specify image-normalized integer coordinates. Definitions distinguish contextual roles, including form field names (`Section-header`) and chart-internal titles (`chart_text`). They also require one object per visual text block and name its output fields. The `img2full text` field contains transcription with line breaks replaced by spaces. Regional templates specify text or a four-integer array. Chinese instructions are translated into English in the figures.

The complete experimental prompts also specify allocated regions, following Appendix B.2: region boundaries reflect available space inferred from containers, columns, neighboring elements, and whitespace. The figure excerpts focus on category and serialization instructions. The experiments use the same category definitions and fallback rule throughout: `Text` denotes ordinary prose or field values with an identifiable role; `Other` is used when no defined category fits or the role cannot be determined from context. Uncertainty alone does not make an element `Text`.

System instruction for zero-shot evaluation

You are a meticulous document layout analysis assistant. Always follow the user's exact output format and emit only the requested JSON or value.

img2layout | Layout parsing

Canonical user prompt

<image>
Analyze the layout of this image. For every text region, output a JSON list where each item has bbox_2d ([x1, y1, x2, y2], normalized 0-1000), category, and align.

Zero-shot user prompt (complete) General-purpose models

<image>
You are a professional layout/document/UI parsing assistant. Given the input image, detect ****all**** text regions (titles, body text, captions, table text, chart text, buttons, page headers and footers, etc.; do not miss small or dense text) and output a JSON array (****output only the JSON array itself****, without Markdown code fences or explanations).

Coordinate convention: Represent every bounding box as [x1, y1, x2, y2] (****integers****, not decimals), normalized to [0, 1000]. 0 corresponds to the left/top image boundary, and 1000 to the right/bottom boundary. Conversion formulas: x = round(pixel_x / image_width * 1000), y = round(pixel_y / image_height * 1000). Ensure x1<=x2 and y1<=y2. ****Coordinates must be integers between 0 and 1000; decimals, 0-1 fractions, and percentages are prohibited.****

Each array element must be an object with the following fields in order: bbox_2d, category, align:
- "bbox_2d": [x1, y1, x2, y2], normalized to [0,1000] (see the coordinate convention).
- "category": See the category definitions below.
- "align": One of ['left', 'right', 'center', 'justify'], indicating the horizontal alignment of text within the region.
category must be exactly one of the following 13 labels (names are case- and hyphen-sensitive and must be returned exactly as written):
- "Title": The most prominent main title of an entire page or block, using the largest font size and heading the whole page or block.
- "Section-header": Headings or subheadings for chapters, columns, or sections; also form field names and descriptive UI labels (e.g., 'First name' / 'Email' / 'Phone').
- "Text": Body paragraphs, descriptive text, field values associated with labels, card or product body text, and other content text (the default category when uncertain).
- "List-item": List entries (marked with -, bullets, or numbers, or arranged as parallel short entries).
- "Caption": Captions or descriptive text for figures, tables, illustrations, or videos, usually adjacent to the corresponding media.
- "Footnote": Footnotes, notes, or source attributions, usually near the bottom and in a smaller font.
- "Page-header": Information strips across the top, such as page headers, top bars, or mobile status bars.
- "Page-footer": Page footers, including copyright notices, legal statements, privacy policies, terms, bottom URLs, or cookie notices.
- "Table": Text within data tables, including headers, cells, and table bodies with a clear row-and-column structure.
- "Formula": Mathematical or chemical equations, including LaTeX notation (e.g., E=mc^2 or \frac{a}{b}).
- "ui_button": Clickable action text on buttons or calls to action (e.g., Submit / Buy Now / Sign up / Book Now).
- "chart_text": Axes, tick labels, legends, data annotations, and titles within charts (bar, line, pie, scatter, radar, heatmap, etc.); also text within diagrams, such as flowchart or structural-diagram labels, infographic nodes, and schematic labels.
- "Other": Text that clearly belongs to none of the above categories or whose category cannot be determined (use sparingly).

Rules:
- Cover the entire page without omissions. Use one object per visual text block. Do not merge multiple independent paragraphs into one object or split a single paragraph into multiple objects.
- Assign each object the single most appropriate category; use "Text" or "Other" when uncertain.
- Output valid JSON only, without any preceding or following text.

Figure 13: **Layout-only parsing prompts.** The excerpts specify coordinates, output fields, 13 category labels, four alignment values, and text-block grouping; the complete experimental prompts also define allocated regions. Chinese portions are translated into English; colored headers are figure annotations.

System instruction for zero-shot evaluation

You are a meticulous document layout analysis assistant. Always follow the user's exact output format and emit only the requested JSON or value.

img2full | Full layout-aware parsing

Canonical user prompt

<image>
Perform full layout-aware parsing of this image. For every text region output a JSON list where each item has text, bbox_2d ([x1, y1, x2, y2], normalized 0-1000), category and align.

Zero-shot user prompt (complete) General-purpose models

<image>
You are a professional layout/document/UI parsing assistant. Given the input image, detect ****all**** text regions (titles, body text, captions, table text, chart text, buttons, page headers and footers, etc.; do not miss small or dense text) and output a JSON array (****output only the JSON array itself****, without Markdown code fences or explanations).

Coordinate convention: Represent every bounding box as [x1, y1, x2, y2] (****integers****, not decimals), normalized to [0, 1000]. 0 corresponds to the left/top image boundary, and 1000 to the right/bottom boundary. Conversion formulas: x = round(pixel_x / image_width * 1000), y = round(pixel_y / image_height * 1000). Ensure x1<=x2 and y1<=y2. ****Coordinates must be integers between 0 and 1000; decimals, 0-1 fractions, and percentages are prohibited.****

Each array element must be an object with the following fields in order: text, bbox_2d, category, align:
- "text": The original text in the region, transcribed verbatim, with line breaks replaced by spaces.
- "bbox_2d": [x1, y1, x2, y2], normalized to [0,1000] (see the coordinate convention).
- "category": See the category definitions below.
- "align": One of ['left', 'right', 'center', 'justify'], indicating the horizontal alignment of text within the region.
category must be exactly one of the following 13 labels (names are case- and hyphen-sensitive and must be returned exactly as written):
- "Title": The most prominent main title of an entire page or block, using the largest font size and heading the whole page or block.
- "Section-header": Headings or subheadings for chapters, columns, or sections; also form field names and descriptive UI labels (e.g., 'First name' / 'Email' / 'Phone').
- "Text": Body paragraphs, descriptive text, field values associated with labels, card or product body text, and other content text (the default category when uncertain).
- "List-item": List entries (marked with -, bullets, or numbers, or arranged as parallel short entries).
- "Caption": Captions or descriptive text for figures, tables, illustrations, or videos, usually adjacent to the corresponding media.
- "Footnote": Footnotes, notes, or source attributions, usually near the bottom and in a smaller font.
- "Page-header": Information strips across the top, such as page headers, top bars, or mobile status bars.
- "Page-footer": Page footers, including copyright notices, legal statements, privacy policies, terms, bottom URLs, or cookie notices.
- "Table": Text within data tables, including headers, cells, and table bodies with a clear row-and-column structure.
- "Formula": Mathematical or chemical equations, including LaTeX notation (e.g., E=mc^2 or \frac{a}{b}).
- "ui_button": Clickable action text on buttons or calls to action (e.g., Submit / Buy Now / Sign up / Book Now).
- "chart_text": Axes, tick labels, legends, data annotations, and titles within charts (bar, line, pie, scatter, radar, heatmap, etc.); also text within diagrams, such as flowchart or structural-diagram labels, infographic nodes, and schematic labels.
- "Other": Text that clearly belongs to none of the above categories or whose category cannot be determined (use sparingly).

Rules:
- Cover the entire page without omissions. Use one object per visual text block. Do not merge multiple independent paragraphs into one object or split a single paragraph into multiple objects.
- Assign each object the single most appropriate category; use "Text" or "Other" when uncertain.
- Output valid JSON only, without any preceding or following text.

Figure 14: **Full-parsing prompts.** The excerpts specify geometry, attributes, and verbatim transcription, replacing line breaks with spaces; the complete experimental prompts also define allocated regions. Chinese portions are translated into English; colored headers are figure annotations.

Table 8: **Human arena validation on full parsing.** A 20-reviewer pool supplies three judgments per comparison and 300 vote participations per model across opponents. The table reports tie-adjusted WR, image-bootstrap 95% confidence intervals, human ranks, and benchmark ranks based on Joint@.75 over the shared 100-image pool.

Model	WR (%)	95% CI	Human rank	Bench rank
Training-stage arena				
Qwen3-VL-4B	7.50	[3.96, 11.75]	3	3
Qwen3-VL-4B$_{+\text{SFT}}$	65.00	[57.87, 72.00]	2	2
Qwen3-VL-4B$_{+\text{SFT+EAA-GRPO}}$	77.50	[70.89, 83.82]	1	1
Zero-shot arena				
Qwen2.5-VL-72B	51.50	[44.25, 58.63]	3	3
Qwen3-VL-235B	68.50	[61.08, 75.82]	2	2
InternVL3-78B	17.17	[11.32, 23.41]	5	5
InternVL3.5-241B	41.17	[33.50, 48.94]	4	4
Kimi-K2.6	71.67	[65.22, 78.25]	1	1

C.3 Human Evaluation of Model Outputs

Two pairwise arenas. Human evaluation assesses `img2full` using a shared pool of 100 images, sampled by source domain from GLA-Bench. The training-stage arena compares Qwen3-VL-4B, its SFT variant, and its SFT+EAA-GRPO variant. The zero-shot arena compares the five general-purpose models in Table 2. A pool of 20 reviewers supplies three independent judgments per image–pair comparison. Each model receives 300 judgments across its opponents. With balanced opponent allocation, each model pair uses 50 images in the training-stage arena and 25 in the zero-shot arena, giving 150 and 250 comparisons and 450 and 750 judgments, respectively. Each judgment evaluates both models in a pair. Images can recur across pairs; per-model coverage counts image–pair participations.

Blinding and judgment criteria. Reviewers inspect the source image and two anonymized outputs rendered with the same visual conventions. Model identities, benchmark scores, and reference annotations are hidden. Presentation order is randomized, with the two orientations differing by at most one comparison per pair. Reviewers select the left output, the right output, or a tie, considering element coverage, allocated-region fidelity, transcription, category, and alignment. Decisions follow a shared rubric and include a brief reason.

Win rates and human rankings. Let $y_{ijxr} \in \{0, \frac{1}{2}, 1\}$ denote a loss, tie, or win for model i against j on image x, as judged by reviewer r. Let $\mathcal{R}_{ijx} \subseteq \{1, \dots, 20\}$ be the three reviewers assigned to that comparison. Let $\mathcal{X}_{ij}$ contain the N_{ij} images assigned to pair (i, j), where $N_{ij} = 50$ or 25 in the two arenas. For M models, pairwise preference and opponent-averaged win rate are

$$\widehat{p}_{ij} = \frac{1}{3N_{ij}} \sum_{x \in \mathcal{X}_{ij}} \sum_{r \in \mathcal{R}_{ijx}} y_{ijxr}, \qquad \mathrm{WR}_i = \frac{100}{M-1} \sum_{j \neq i} \widehat{p}_{ij}. \tag{9}$$

All three initial judgments contribute equally. Images are equally weighted within each pair, and opponents receive equal weight. Human ranks follow descending unrounded WR_i, with equal values assigned midranks. We report 95% confidence intervals from 10,000 bootstrap resamples of the 100-image pool, retaining the recorded pairs and judgments for each sampled image. Each resample uses its actual number of judgments for each pair.

Inter-rater agreement. Agreement is computed from the three initial judgments, before discussion or adjudication. For comparison b, let n_{bk} count votes for outcome $k \in \{\text{left}, \text{tie}, \text{right}\}$, using the same randomized orientation for all three reviewers. With $L = \sum_{i<j} N_{ij}$ comparisons,

$$P_o = \frac{1}{6L} \sum_{b=1}^{L} \sum_{k} n_{bk}(n_{bk} - 1), \quad P_e = \sum_{k} \left(\frac{\sum_b n_{bk}}{3L} \right)^2, \quad \kappa = \frac{P_o - P_e}{1 - P_e}. \tag{10}$$

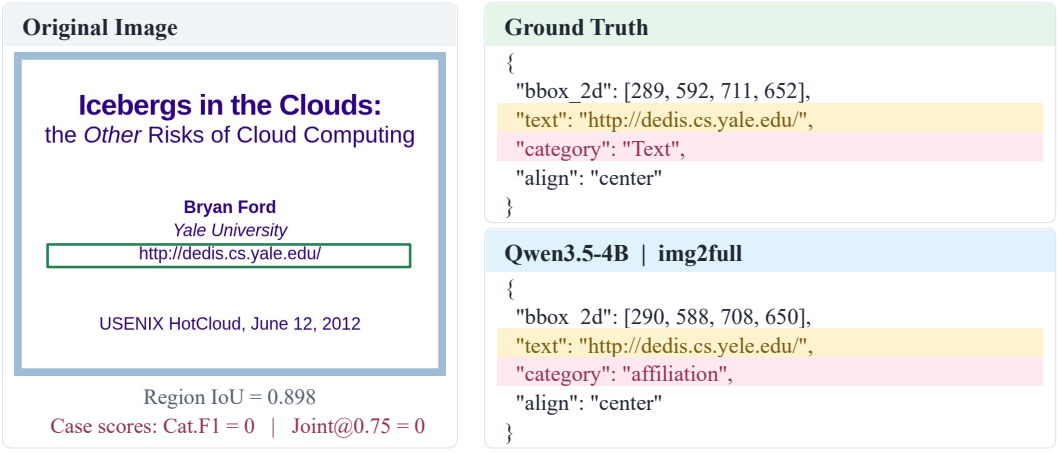


Figure 15: **High geometric overlap with category and transcription errors.** The Qwen3.5-4B prediction localizes the URL at IoU 0.898 with correct center alignment, but assigns `affiliation` instead of `Text` and transcribes `yale` as `yele` in the URL. The category mismatch fails the category-agreement condition, and the transcription error also violates the exact-text condition for joint correctness in `img2full`.

Agreement is quantified by P_o, Fleiss' κ, and the fraction U of comparisons with unanimous judgments. An independent audit checks flagged decisions against the rubric; additional reviews are recorded separately from the initial votes used for these statistics.

Agreement with benchmark rankings. Benchmark ranks use Joint@.75 scores on the shared 100-image pool within each arena. Spearman's ρ_s and Kendall's τ_b compare these ranks with human ranks, separately from inter-rater agreement. Table 8 reports model win rates and compares human ranks with benchmark ranks. The results show the high degree of consistency between human ranks and our evaluated ranks, which demonstrates the reliability of our proposed evaluation metrics.

C.4 Analysis of Zero Category and Joint Scores

Connection to the main results. Table 2 reports 0.00 Joint@.75 for the Qwen3-VL-4B and Qwen3.5-4B baselines on both `img2layout` and `img2full`, and 0.00 Cat.F1 on `img2full`. These reported scores motivate examining category agreement alongside region overlap. Figure 15 illustrates the distinction with a URL region: its predicted box has IoU 0.898 and correct center alignment, but the category `affiliation` differs from the reference `Text`. The prediction also transcribes `yale` as `yele` in the URL. High region overlap and correct alignment therefore coexist with errors in category and transcription.

Category matching. Under Section 3.3, predictions and references are matched one-to-one using geometry. Category agreement is evaluated over matched elements using the labels in Appendix B.3. For the matched pairs of a single image, category c contributes

$$F_{1,c} = \frac{2\mathrm{TP}_c}{2\mathrm{TP}_c + \mathrm{FP}_c + \mathrm{FN}_c}, \qquad F_{1,\mathrm{macro}} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} F_{1,c}, \tag{11}$$

where $\mathcal{C}$ is the set of categories included in the macro-average. If no matched pair has the same category on both sides, then $\mathrm{TP}_c = 0$ for every c, yielding zero category F1. In Figure 15, `affiliation` is outside the benchmark vocabulary and does not match the reference `Text`.

Joint correctness under the benchmark protocol. Under Section 3.3, a joint-correct element must satisfy the IoU threshold and have the correct category and alignment. Full parsing additionally requires an exact match between normalized predicted and reference text. Category disagreement is therefore sufficient to make an element fail the joint criterion, even when its geometry and alignment

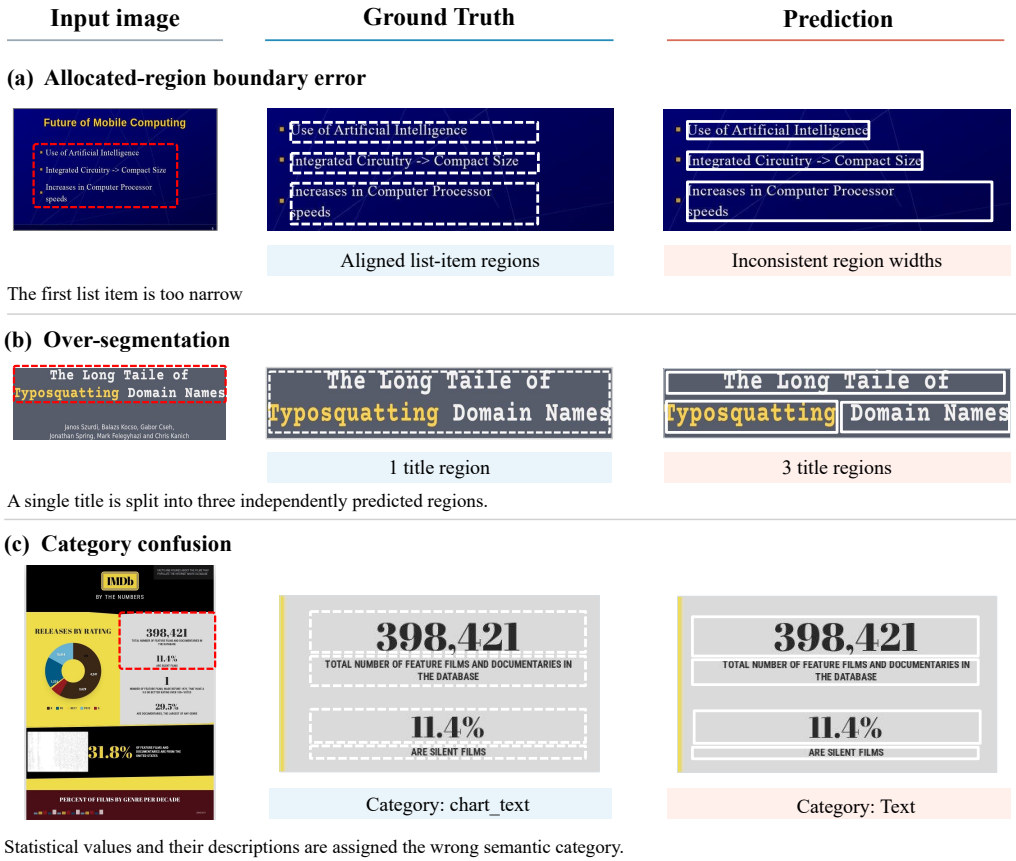


Figure 16: **Structural consistency in layout parsing.** Columns show input images, reference annotations, and parser predictions. (a) List-item reference regions share a common width, while predicted widths vary. (b) One reference title region corresponds to three predicted regions across line and style changes. (c) Statistical values and their descriptions receive `chart_text` in the reference and `Text` in the prediction.

are correct. In Figure 15, both category disagreement and the URL transcription error violate the joint criterion.

Output validity and task correctness. The displayed prediction is syntactically valid JSON and contains the expected fields, yet its category and text values are incorrect. JSON validity therefore does not imply category agreement or joint correctness. The example shows how accurate localization can coexist with zero joint correctness for an individual element. The full-image scores in Table 2 follow the benchmark's per-image evaluation and averaging protocol.

C.5 Structural Consistency in Layout Parsing

The qualitative examples complement Section 5.5 by examining three types of prediction error: inconsistent allocated-region widths, fragmented element grouping, and incorrect semantic categories. Figure 16 compares reference annotations with parser predictions in representative layouts. List items share a reference width that defines their available horizontal space; a title spanning two lines and multiple text styles forms one editable element; and statistical values and descriptions take their semantic role from the surrounding infographic. These cases show why generation-aware parsing represents geometry and categories together at the level of semantic elements.