Benchmarking and Optimizing Multimodal Structured Generation: The OracleGraph Dataset and PRISM Framework

Zhihong Sun*, **Qi Fan***, **Pan Liu**, **Yang Yi**, **Chen Ye**†
College of Computer Science and Technology
Tongji University
Shanghai, China
{2534122, 2350262, 2353328, 2352475, yechen}@tongji.edu.cn

Abstract

Multimodal Large Language Models (MLLMs) are increasingly asked to emit long, schema-constrained outputs, such as JSON graphs, tables, and code, directly from images. We study why post-training stalls on this task using ancient Chinese document pages, which must be parsed into typed graphs of text regions, annotations, seals, and reading-order relations. Supervised Fine-Tuning (SFT) plateaus, and scalar-reward Reinforcement Learning (DPO, GRPO, SimPO) stays within ± 1.5pp of SFT. We trace this to three coupled pathologies: a single sequence-level reward cannot localize errors in outputs of thousands of tokens (a *policy-level signal gap*); one malformed token zeroes the reward of an entire parse (a *zero-reward parsing cliff*); and policies therefore learn to shorten their outputs (a *short-completion bias*). Vanilla self-distillation (SDPO) even regresses. To measure these failures we introduce **OracleGraph**, a procedurally generated 10,000-page 2.5D benchmark with exact typed-graph ground truth, a 12-dimensional diagnostic VQA suite, and a structured-hallucination taxonomy. To address them we propose **PRISM**, a Policy-Reward Integrated Self-distillation framework in which a hindsight teacher, conditioned on a Diagnostic Graph Report (DGR), supervises a blind student through Graph-F1 reward-weighted distillation and a length-normalized preference loss that counteracts truncation. PRISM improves Graph Reward by $+3.81$pp over vanilla SDPO ($p<0.001$), with gains concentrated on the longest and most structurally complex pages; ablations show these gains come from better node and edge recovery and fewer truncations rather than longer outputs, and full PRISM also improves Graph F1 by $+2.95$pp on the external GraphDoc benchmark. Its $+1.02$pp mean gain over SFT is not significant ($p=0.12$), so we position PRISM primarily as a stabilizer: it attains $2.4\times$ lower cross-seed standard deviation than SFT in-distribution and $14\times$ lower on a cross-book OOD probe, where no method improves the mean. Code and data are available at `https://github.com/209-Tongji/PRISM_2026/`.

1 Introduction

Ancient Chinese documents – such as block-printed editions of *Hongloumeng*, archival ledgers, and hand-copied manuscripts – pose a distinctive challenge for contemporary Multimodal Large Language Models (MLLMs). Unlike modern PDFs, each page is a richly annotated, non-linear surface: vertical main columns interleave with double-line interlinear commentary, top/bottom/side marginalia, fishtail

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

marks, and cross-column fold lines, while vermilion seals and edit marks routinely overlap the text they modify. Parsing such a page into a strict, schema-compliant JSON graph of typed nodes and typed edges (e.g., `READS_AFTER`, `ANNOTATES`, `OVERLAPS`) is therefore not a cosmetic layout-analysis task, but a joint visual-structural-relational reasoning challenge. Reliable open-weight parsers for such corpora are in demand among digital-humanities researchers, yet off-the-shelf 8B-class MLLMs fail almost completely in our measurements.

The difficulty is not specific to Chinese. Ancient pages are an extreme instance of a broader problem: producing *long, schema-valid* outputs that preserve nodes, relations, hierarchy, and reading order. The same requirement arises in scientific and financial document parsing, table and layout-graph extraction, chart-to-code, GUI-hierarchy generation, and diagram or schematic parsing. In each case, the bottleneck is maintaining structural completeness over thousands of constrained tokens.

Supervised Fine-Tuning (SFT) is typically employed to instill a schema prior, but behavioral cloning offers no mechanism to explore and rank candidate parses, and performance plateaus. The contemporary remedy, scalar-reward Reinforcement Learning (e.g., DPO, GRPO, SimPO), exposes three coupled vulnerabilities in this setting. Compressing a multi-thousand-token output into one scalar strips away localized topological signal (a *policy-level signal gap*); a single malformed token renders the whole parse unreadable and its reward zero (a *zero-reward parsing cliff*); and the policy learns to evade the cliff by truncating outputs and emitting structurally thinner graphs (a *short-completion bias*).

To quantify and mitigate these pathologies, we introduce two coupled pieces: **OracleGraph** and **PRISM**. OracleGraph is a procedurally generated 10,000-page 2.5D historical-document benchmark whose typed-graph ground truth is exact by construction, paired with a fine-grained hallucination taxonomy and a diagnostic Visual Question Answering (VQA) suite. PRISM (Policy-Reward Integrated Self-distillation for Multimodal structured extraction) is a self-distillation framework for long structured JSON generation. It engineers multimodal information asymmetry via a Hindsight Teacher guided by a Diagnostic Graph Report (DGR) and combines softmax reward-weighted aggregation (which preserves Graph-F1 rankings) with a length-normalized auxiliary preference loss, explicitly counteracting the short-completion bias without an external reward model. Our contributions are:

- **Typed-Graph Ontology and Taxonomy:** To our knowledge, the first 2.5D typed-graph ontology for ancient Chinese documents (13 node classes, 8 edge classes, 5-level occlusion z-index), enabling a 30+ category structured-hallucination taxonomy that is finer-grained than single-metric evaluations such as POPE [Li et al., 2023] or CHAIR [Rohrbach et al., 2018].
- **The OracleGraph Benchmark:** A 10,000-page corpus ($\sim$275K typed annotations) rendered from programmatic markup, so that node classes, polygons, z-indices, and edges are exact and free of annotator noise, coupled with a 12-dimensional diagnostic VQA suite. Our 8B domain-specific stack is the strongest sub-10B open-weight model we evaluate on its structural subset.
- **Empirical RL Regime Diagnosis:** Under a matched-rank (r=32), 3-seed protocol with n=100 paired bootstrap, standard scalar-reward RL (DPO, GRPO, SimPO) clusters within $\pm$1.5pp of SFT ($p>0.05$) and shortens its outputs during training, consistent with the short-completion bias.
- **The PRISM Framework and Where It Helps:** PRISM yields $+3.81$pp Graph Reward over vanilla self-distillation ($p<0.001$) and $2.4\times$ lower cross-seed variability than SFT. Difficulty-stratified and leave-one-out analyses localize its gains to long, complex pages and to node/edge recall and truncation, not to longer outputs. We do *not* claim a significant in-distribution gain over SFT (p=0.12).
- **Scope Beyond OracleGraph:** Full PRISM, retrained on the human-annotated GraphDoc benchmark [Chen et al., 2025], improves node, edge, and graph F1 over a matched SFT baseline, and OracleGraph-trained PRISM retains a zero-shot advantage on complex GraphDoc pages. PRISM requires gold-derived feedback or a programmatic verifier during post-training.

2 Related Work

Document structure benchmarks and multimodal structured generation. Layout benchmarks such as PubLayNet [Zhong et al., 2019], DocBank [Li et al., 2020], and DocLayNet [Pfitzmann et al., 2022] evaluate flat, largely non-overlapping 2D regions on modern documents. OCR-free document

Table 1: OracleGraph versus representative document-layout and document-graph benchmarks. OracleGraph is smaller than modern-document corpora but is the only one that combines typed relations, explicit occlusion layers, and exact (program-emitted) ground truth with a diagnostic VQA suite.

Benchmark	Domain	Pages	Ground-truth source	Typed relations	Reading order	Occlusion (z)	Diagnostic VQA
PubLayNet [Zhong et al., 2019]	modern scientific articles	358K	automatic XML matching	×	×	×	×
DocBank [Li et al., 2020]	modern arXiv papers	500K	weak supervision (LaTeX)	×	×	×	×
DocLayNet [Pfitzmann et al., 2022]	diverse modern documents	81K	human annotation	×	×	×	×
GraphDoc [Chen et al., 2025]	diverse modern documents	80K	human-verified graphs	✓	✓	×	×
OracleGraph (ours)	ancient Chinese documents	10K	procedural, exact by construction	✓ (8 classes)	✓	✓ (5 levels)	✓ (12 dims)

MLLMs learn structure from large corpora [Kim et al., 2022, Blecher et al., 2023, Hu et al., 2024], and GraphDoc [Chen et al., 2025] already formulates document structure analysis as graph prediction (gDSA), with 80K human-verified modern-document images and relation annotations for reading order and hierarchy. The graph framing itself is therefore prior art. OracleGraph differs in three respects: (i) a 2.5D representation with an explicit 5-level occlusion z-index for physically overlapping elements such as seals stamped over text; (ii) non-linear historical reading orders (e.g., interlinear commentary) in an ancient-document domain; and (iii) procedurally exact ground truth coupled with a diagnostic VQA suite (Table 1). Other lines of work constrain decoding to enforce schemas [Cesista, 2025], or use RL with execution checks [Chen et al., 2026a] or perceptual verification [Wang et al., 2025] to move beyond SFT, whose behavioral cloning struggles with robust spatial reasoning [Rizzoli et al., 2025]. These approaches rely on sparse scalar rewards or ad-hoc verifiers.

Preference optimization and credit assignment. Scalar-reward algorithms such as DPO [Rafailov et al., 2023] and GRPO [Shao et al., 2024, Ju et al., 2025, Rudner et al., 2021] are prone to reward hacking via verbosity in open-domain settings [Wang et al., 2026, MacDiarmid et al., 2025], whereas strict JSON or graph generation presents the opposite failure, a zero-reward cliff [Beigi et al., 2026, Zhou et al., 2025]. Outcome-supervised rewards compress routing decisions into one scalar [Lyu et al., 2025, Ding et al., 2026]; process rewards assign denser credit [Fei et al., 2025, Zhang et al., 2025a] but annotating intermediate 2.5D steps is expensive. Graph-matching rewards have been used for Text-to-SQL RL [Weng et al., 2025], and reference-free length-normalized objectives, most prominently SimPO [Meng et al., 2024], counteract the autoregressive length penalty [Qiyuan et al., 2026, Zou et al., 2026, Chen et al., 2026b, Liao et al., 2026, Gu et al., 2025, Wu et al., 2025]. These sequence-level objectives remain blind to spatial dependencies [Zhang et al., 2025b, Li et al., 2026a].

Self-distillation with privileged information. Self-Distillation Policy Optimization (SDPO) converts rich feedback into dense token-level distillation targets without an external reward model [Hübotter et al., 2026, Li et al., 2026c]; related work exploits privileged teacher information [Li et al., 2026b, Yang et al., 2026], advantage weighting [Peters and Schaal, 2007, Xue et al., 2026], and token- or sequence-level preference signals [Rashid et al., 2025, Zhou et al., 2024, Feng et al., 2025].

Positioning. Each PRISM ingredient has precedent: privileged self-distillation (SDPO), length-normalized preference (SimPO, which PRISM uses *without* a target margin), graph-based rewards (Graph-Reward-SQL), and graph-based layout prediction (GraphDoc). Our contribution is not any single component. It is (a) a diagnosis of why these ingredients fail *in isolation* on long multimodal structured outputs (vanilla SDPO regresses; scalar RL stalls), and (b) an integration, together with matched-compute evidence of which pieces are necessary (§4.3) and how far the recipe transfers (§4.5).

3 Methodology

Problem Formulation. Let $\mathcal{D} = \{I_i, x_i, y_i^*\}_{i=1}^N$ denote a dataset of ancient-document images I, textual prompts x, and ground-truth `OracleGraph` JSON structures y^*. MLLMs frame this as autoregressive generation, outputting tokens $y = (y_1, \ldots, y_T)$ conditioned on a multimodal context $\mathbf{h}^B = [E_{\text{vis}}(I); E_{\text{txt}}(x)]$. SFT establishes a strong structural prior but offers no mechanism to rank parse candidates, while scalar-reward RL suffers from the short-completion bias. **PRISM** (Figure 1) is an asymmetric self-distillation framework that couples dense token-level distillation with a sequence-level length-normalized preference loss, governed by a deterministic Graph-F1 reward.

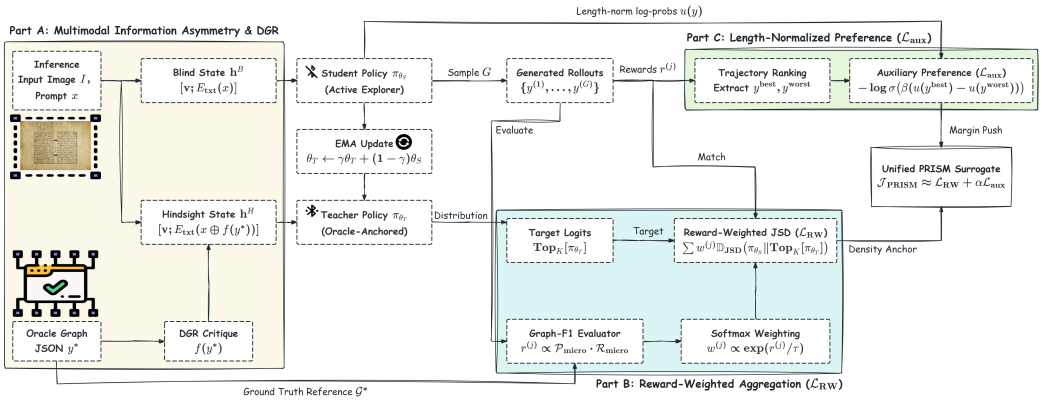


Figure 1: **The Architecture of the PRISM Framework. (Part A) Information Asymmetry & DGR:** The *Teacher Policy* evaluates the multimodal context augmented with a Diagnostic Graph Report (DGR) derived from the ground-truth JSON y^*, establishing a strong structural prior. The *Student Policy* relies solely on the visual input I and prompt to generate G rollouts. **(Part B) Reward-Weighted Aggregation ($\mathcal{L}_{\text{RW}}$):** Rollouts are scored by a programmatic macro *Graph-F1* over node and edge classes; a softmax over rewards weights the renormalized top-K JSD distillation objective, so hallucinated sequences receive little gradient. **(Part C) Auxiliary Preference Injection ($\mathcal{L}_{\text{aux}}$):** A length-normalized preference loss (no target margin) over the best and worst rollouts, computed with the student's log-probabilities, penalizes truncation. **(Part D) EMA Dynamics:** The Teacher's parameters track the Student via an Exponential Moving Average (γ=0.999), keeping a separately parameterized, slowly moving teacher.

3.1 Information Asymmetry and the Hindsight Teacher

Standard SDPO assumes information symmetry, trapping the policy in blind exploration. PRISM introduces a deliberate *information asymmetry* (Figure 1, Part A). During optimization, the **Blind Student** π_{θ_S} samples G=2 graph rollouts $\{y^{(1)}, \ldots, y^{(G)}\}$ conditioned only on $\mathbf{h}^B$. The **Hindsight Teacher** π_{θ_T} is granted privileged access via a **Diagnostic Graph Report (DGR)**, a deterministic natural-language critique $f(y^*)$ enumerating structural constraints and error types, so its context is $\mathbf{h}^H = [E_{\text{vis}}(I); E_{\text{txt}}(x \oplus f(y^*))]$. To avoid distilling long-tail noise, we keep the set $\mathcal{K}_t$ of the teacher's top-K (K=20) tokens at step t and *renormalize* it into a valid distribution, $\tilde{\pi}_T^K(v \mid \cdot) = \pi_{\theta_T}(v \mid \cdot)\mathbb{1}[v \in \mathcal{K}_t]/Z_t$ with $Z_t = \sum_{v' \in \mathcal{K}_t} \pi_{\theta_T}(v' \mid \cdot)$. Without gradients into the teacher, we minimize

$$\mathcal{L}_{\text{JSD}}^{(j)}(\theta_S) = \sum_{t=1}^{|y^{(j)}|} \mathbb{D}_{\text{JSD}}\Big(\pi_{\theta_S}(\cdot \mid \mathbf{h}^B, y_{<t}^{(j)}) \,\Big\|\, \tilde{\pi}_T^K(\cdot \mid \mathbf{h}^H, y_{<t}^{(j)})\Big). \tag{1}$$

3.2 Deterministic Topological Evaluation via Graph-F1

Vanilla SDPO aggregates rollout losses uniformly, treating hallucinated topologies like flawless ones. A parser maps each rollout to a graph $\mathcal{G}^{(j)} = (\mathcal{V}^{(j)}, \mathcal{E}^{(j)})$; unparseable sequences receive $r^{(j)}$=0. A predicted node $\hat{v}$ is a true positive iff it matches a gold node v^* in class, bounding-box IoU $> \delta$=0.5, *and* occlusion layer ($z_{\hat{v}}$=z_{v^*}), under one-to-one greedy assignment; a wrong z-index therefore voids the match. An edge $\hat{e}(\hat{u}, \hat{v}, c_{\hat{e}})$ is a true positive iff both endpoints are matched to the corresponding gold nodes and $c_{\hat{e}} = c_{e^*}$. The **Graph Reward** (used interchangeably with Graph-F1) is the *macro*-F1 over node and edge classes,

$$r^{(j)} = \frac{1}{|\mathcal{C}^{(j)}|} \sum_{c \in \mathcal{C}^{(j)}} \frac{2\,\text{TP}_c}{2\,\text{TP}_c + \text{FP}_c + \text{FN}_c}, \tag{2}$$

where $\mathcal{C}^{(j)}$ is the set of node and edge classes present in the gold or predicted graph. Macro-averaging prevents abundant classes (e.g., text blocks) from swamping rare but topologically critical edges (e.g., `READS_AFTER`).

3.3 Reward-Aware Aggregation and Preference Injection

Reward-Weighted Aggregation. We weight rollouts by a softmax over Graph-F1 with temperature τ=0.5, $w^{(j)} = \exp(r^{(j)}/\tau)/\sum_{k=1}^{G}\exp(r^{(k)}/\tau)$ (Figure 1, Part B), biasing distillation toward structurally stronger trajectories without collapsing onto a single rollout: $\mathcal{L}_{\text{RW}}(\theta_S) = \sum_{j=1}^{G} w^{(j)}\mathcal{L}_{\text{JSD}}^{(j)}(\theta_S)$.

Length-Normalized Auxiliary Preference Loss. Token-level JSD has no sequence-level mechanism to penalize truncation. Let y^{best} and y^{worst} be the highest- and lowest-reward rollouts and $u(y) = \frac{1}{|y|}\sum_{t=1}^{|y|}\log\pi_{\theta_S}(y_t \mid \mathbf{h}^B, y_{<t})$ the length-normalized log-likelihood (Figure 1, Part C):

$$\mathcal{L}_{\text{aux}}(\theta_S) = -\log\sigma\Big(\beta\big(u(y^{best}) - u(y^{worst})\big)\Big), \qquad \beta = 2.5. \tag{3}$$

Unlike SimPO [Meng et al., 2024], we use no target margin: the sigmoid's gradient pushes the likelihood gap continuously, and its magnitude adapts to the current gap rather than to a fixed threshold. The $1/|y|$ factor removes the advantage that shorter sequences otherwise enjoy in summed log-likelihood.

Unified Surrogate Objective. Let $\pi_{\theta_{\text{old}}}$ be the student policy immediately before the current PPO epoch (the policy that generated the rollouts), $\rho_t^{(j)} = \pi_{\theta_S}(y_t^{(j)} \mid \mathbf{h}^B, y_{<t}^{(j)})/\pi_{\theta_{\text{old}}}(y_t^{(j)} \mid \mathbf{h}^B, y_{<t}^{(j)})$, and $\hat{A}^{(j)} = r^{(j)} - \frac{1}{G-1}\sum_{k\neq j} r^{(k)}$ the leave-one-out sequence-level advantage, broadcast to every token of rollout j. With clip ϵ=0.1 and α=0.2, PRISM maximizes

$$\mathcal{J}_{\text{PRISM}}(\theta_S) = \sum_{j=1}^{G}\sum_{t=1}^{|y^{(j)}|}\min\Big(\rho_t^{(j)}\hat{A}^{(j)},\ \text{clip}\big(\rho_t^{(j)}, 1-\epsilon, 1+\epsilon\big)\hat{A}^{(j)}\Big) - \Big(\mathcal{L}_{\text{RW}}(\theta_S) + \alpha\,\mathcal{L}_{\text{aux}}(\theta_S)\Big). \tag{4}$$

3.4 Temporal Stabilization via EMA Dynamics

An instantaneous online update ($\theta_T \equiv \theta_S$) collapses teacher and student into one parameterization, leaving only the prompt-level asymmetry. PRISM instead keeps a separately parameterized teacher updated by an Exponential Moving Average (Figure 1, Part D), $\theta_T \leftarrow \gamma\theta_T + (1-\gamma)\theta_S$, with γ=0.999. In matched experiments (§4.3), a fully frozen teacher performs on par with EMA and both outperform the online variant, so the essential ingredient is a separate teacher; EMA is our default because it tracks the student's distribution while preserving this asymmetry.

4 Experiments

Our evaluation proceeds in five layers: (1) a **diagnostic VQA audit** of zero-shot MLLMs; (2) a **matched-rank RL comparison** isolating algorithmic gains from capacity; (3) an analysis of **where PRISM's gains come from** (difficulty stratification and a unified leave-one-out ablation); (4) **learning dynamics** and DGR verbosity; and (5) **generality** beyond OracleGraph (GraphDoc retraining, zero-shot transfer, and OOD shifts).

4.1 Experimental Setup and Baseline Robustness

Benchmark Composition. OracleGraph is a *procedurally generated* 10,000-page 2.5D benchmark. Each page is rendered by a historical-typesetting engine from a programmatic layout specification (public-domain source text, column grid, interlinear commentary, marginalia, seals, edit marks, fold lines); the oracle graph (node classes, polygons, z-indices, typed edges) is emitted by the same program and is thus exact by construction, with no manual graph annotation. Physical rendering effects (paper texture, ink bleed, seal stamping, optical distortion) add visual realism (App. A.2). Each node carries an occlusion layer $z \in \{0, \ldots, 4\}$, e.g., vermilion seals (z=4) over ink text (z=1); the ontology has 13 node and 8 directed edge classes. The corpus comprises (i) 5,000 pages for SFT warmup and RL alignment; (ii) a 500-page held-out VQA audit split; and (iii) a 4,500-page stress-test split generated via a layout-diversity engine. Rendered example pages are shown in App. A.3.

Metrics. We optimize and report **Graph Reward** (macro Graph-F1, §3.2), averaged over pages. Our leave-one-out ablations additionally report a stricter instance-level metric, **pooled strict F1**,

Table 2: Diagnostic VQA Evaluation on OracleGraph (Accuracy, % ↑). **Avg** is the unweighted 12-metric mean; **Win/12** counts metrics strictly outperforming Qwen2.5-VL-7B. PRISM preserves the VQA capabilities of the SFT model while being optimized for graph generation.

Model	Size	Basic Logic		Global Elements				Advanced 2D Spatial						Avg.	Win/12
		BVR	HCE	NIR	GQR	EXR	SCR	SVC	CCB	RVR	AGC	MBCV	EFD		
Closed-Source API (for reference; thinking-augmented / undisclosed size)															
GPT-4o-mini	-	3.72	1.85	28.38	7.41	21.13	50.00	1.04	2.76	12.24	0.00	3.86	4.07	11.37	2/12
Gemini-2.5-Flash-Thinking†	-	40.61	28.50	53.74	22.83	75.90	32.97	7.80	39.94	75.41	13.80	10.73	19.22	*35.12*	*10/12*
Open-Weight 8B–12B, Non-Reasoning (apples-to-apples comparison set)															
Llama-3.2-11B-Vision-Ins	11B	0.87	0.00	1.70	5.35	5.55	13.64	1.00	1.80	1.48	0.00	3.51	0.00	2.91	0/12
Gemma-3-12B-IT	12B	2.00	1.03	2.31	8.00	26.53	9.45	1.46	3.52	0.83	0.99	1.70	2.47	5.02	0/12
InternVL3-8B	8B	3.37	4.79	0.00	2.07	6.64	3.90	3.42	3.09	2.22	1.37	6.91	1.54	3.28	0/12
InternVL2.5-8B	8B	3.06	7.59	0.81	2.77	8.39	5.82	1.41	1.62	2.15	2.24	6.48	0.27	3.55	0/12
Qwen2.5-VL-7B-Ins	7B	21.91	4.16	55.05	5.99	53.88	40.45	3.38	14.25	15.97	1.98	5.09	4.70	18.90	*ref*
Ovis2-8B	8B	8.25	2.10	21.87	4.03	36.97	55.20	6.69	3.10	15.29	1.08	4.71	2.12	13.45	2/12
Open-weight subset mean	–	6.58	3.28	13.62	4.70	22.99	21.41	2.89	4.56	6.32	1.28	4.73	1.85	*7.85*	–
Open-weight subset max	–	21.91	7.59	55.05	8.00	53.88	55.20	6.69	14.25	15.97	2.24	6.91	4.70	*21.03*	–
Domain-Specific Fine-Tuned MLLMs (Ours, Open-Weight, Non-Reasoning, 8B)															
Qwen3-VL-8B (Base, no adapter)	8B	38.00	20.00	61.00	7.00	25.94	40.00	3.00	21.00	33.00	5.00	10.00	17.00	23.41	9/12
Qwen3-VL-8B (SFT Baseline)	8B	52.00	20.00	60.00	9.00	27.99	44.00	1.00	19.00	35.00	5.00	17.00	24.00	**26.17**	**10/12**
PRISM (seed=42, step 100)	8B	52.00	20.00	60.00	9.00	27.63	44.00	1.00	20.00	34.00	3.00	17.00	25.00	26.05	**10/12**
PRISM (seed=2024, step 180)	8B	51.00	20.00	60.00	8.00	27.64	44.00	1.00	20.00	35.00	3.00	17.00	24.00	25.89	**10/12**

†Gemini-2.5-Flash-Thinking: closed API with inference-time chain-of-thought and undisclosed size; listed for context and excluded from the open-weight 8B–12B non-reasoning comparison. Its EXR score of 75.90 leaves a large gap to all 8B open-weight models, consistent with its thinking / larger-size advantage.

which pools node and edge true/false positives and false negatives over all evaluation pages before computing a single F1, together with node/edge precision and recall and the **truncation rate** (outputs that hit the 3,072-token limit before the JSON is closed). VQA uses task-adaptive scoring (App. A.5); its aggregate is always the unweighted mean of the 12 task scores.

Zero-Shot Baseline Robustness. We first audit existing MLLMs on the 12-dimensional VQA suite (Table 2). On the 8-metric structural subset (*Basic Logic* + *Advanced 2D Spatial*), our SFT and PRISM checkpoints reach 21.6% and 21.5%, versus 8.9% for the strongest same-size open-weight generalist (Qwen2.5-VL-7B), a $2.4\times$ lead. Much of this lead is already present in the Qwen3-VL-8B base (18.4%); domain SFT adds the rest, and PRISM preserves it without catastrophic forgetting (12-metric average 26.17 for SFT vs. 26.05/25.89 for PRISM). Generalists remain strong on *Global Elements* such as long-form extraction (EXR), but fail on *Advanced Spatial* tasks.

4.2 Main RL Results: The Matched-Rank Regime

To assess RL alignment without confounding from parameter scaling or unequal compute, we evaluate PRISM and RL baselines under a matched LoRA rank (r=32) and matched step budget (100 steps). Table 3 reveals three regimes.

Regime I: The SFT Attractor (DPO, GRPO, SimPO). Standard preference-based methods cluster within ± 1.5pp of the SFT baseline (72.94%) without statistical significance. We interpret this as a *policy-level signal gap*: scalar rewards collapse structured graph errors into one dimension, trapping the policy near the SFT attractor. These methods also shorten their outputs during training by 7–8% (App. B.6), the signature of the **short-completion bias**.

Regime II: Schema Collapse (Vanilla-SDPO). Without DGR, reward weighting, and the auxiliary loss, self-distillation regresses by -2.79pp and its schema-parsed rate drops to 94.0%.

Regime III: Stabilized Optimization (PRISM). PRISM attains the highest mean (73.96%) and the lowest cross-seed standard deviation (0.27pp, $2.4\times$ below SFT) while keeping long completions (1431 tokens). It improves on Vanilla-SDPO by $+3.81$pp ($p<0.001$). Its $+1.02$pp gain over SFT is *not* significant ($p=0.12$): the evidence supports PRISM as a method that makes self-distillation work and stabilizes it, not as one that reliably breaks the SFT ceiling.

Interpreting the SFT comparison. Two observations explain why the mean gap over SFT is small. First, SFT already saturates much of the benchmark: in the pre-registered tertile analysis (App. B.2), PRISM is slightly *below* SFT on easy pages (SFT reward ≥ 0.80), above it on medium pages, and level on hard pages, so gains and losses partly cancel in the mean. Second, within a 100-step, r=32 budget, the reward discriminates rollouts well but aggregates weakly into a policy-level advantage (App. B.5). What PRISM changes reliably is the *behavior* of self-distillation: the same family that regresses by -2.79pp without its components becomes the best and most stable method with them, and it does so precisely on the long, complex outputs where the short-completion bias bites (§4.3).

Table 3: RL comparison under a matched-rank (r=32) and matched-compute (100 steps) protocol. PRISM has the highest mean and the lowest cross-seed variability; its gain over SFT is not significant, whereas its gain over Vanilla-SDPO is.

Method (matched r=32, 100 steps)	**Seeds**	**n=100 held-out OracleGraph eval**					
		Reward	**Cross-seed std**	**Δ vs. SFT**	**p vs SFT**	**Parsed %**	**Resp. Len**
Base (Qwen3-VL-8B, no adapter)	1	13.00%	—	−59.94pp	—	—	—
SFT baseline (r=32, 1000 steps)	**3**	72.94%	0.65**pp**	**0**	—	**98.0%**	**1337**
DPO (matched r=32)	3	73.64%	0.39pp	+0.70pp	0.33	98.3%	1312
GRPO (matched r=32)	3	73.74%	1.77pp	+0.80pp	0.41	98.0%	1338
SimPO (matched r=32)	3	72.65%	1.05pp	−0.29pp	0.68	96.7%	1390
Vanilla-SDPO (r=32, no DGR, no RW, no aux pref.)	3	70.15%	1.73pp	−2.79pp	<0.01 (worse)	94.0%	1555
Naive-hp SDPO (r=32, text-LLM hyperparameters)	1	63.3%	—	−9.64pp	—	90.0%	1787
DGR-long (single seed)	1	73.33%	—	+0.39pp	—	97.0%	1447
PRISM (ours, full: SDPO + DGR + RW + aux pref, r=32)	3	**73.96%**	**0.27**pp	**+1.02**pp	0.12	**96.3%**	**1431**
Controlled intra-family ablation:		PRISM vs. Vanilla-SDPO Δ = +3.81pp Graph Reward, 95% CI [+1.95, +5.76]pp, p<0.001.					

p-values and CIs from a 10,000-resample paired bootstrap over the n=100 pages.

Table 4: Per-type Graph-F1 (%) on the n=100 held-out pages (matched protocol). Node types: center-fold (CF), red seal (RS), top marginalia (TM), main text (MT), interlinear annotation (IA). Edge types: `READS_AFTER` (R), `OVERLAPS` (O), `ANNOTATES` (A).

	Node types					**Edge types**		
Method	**CF**	**RS**	**TM**	**MT**	**IA**	**R**	**O**	**A**
Vanilla-SDPO	100.0	98.4	92.4	86.1	78.4	56.2	46.1	40.6
SFT	100.0	98.5	92.8	86.1	**81.7**	57.2	46.8	**41.8**
PRISM	100.0	**98.8**	**92.9**	**87.9**	81.4	**61.0**	**49.6**	40.6
Δ PRISM−SFT (pp)	+0.0	+0.3	+0.1	+1.7	−0.3	**+3.8**	**+2.9**	−1.2

4.3 Where Do PRISM's Gains Come From?

The aggregate gain over Vanilla-SDPO could in principle reflect a uniform shift or simply longer outputs. We test both possibilities with two new analyses (Figure 2, Table 5).

Difficulty decomposition. We stratify the held-out pages into quartiles by the Vanilla-SDPO failure regime and compare PRISM against SDPO over 5 seeds (Figure 2a). The gains are not uniform: they are largest on the lowest-reward quartile (+7.93pp, 95% CI $[+4.10, +12.52]$, p=0.0001), the longest-output quartile (>1,500 tokens; +5.43pp, p=0.010), and the largest gold-graph quartile (+4.12pp, p=0.041), with 5/5 seeds positive in every subset. The largest-gold-graph split depends only on the ground truth, so it is free of the regression-to-the-mean effect that can inflate gains on subsets selected by the baseline's own outcomes (App. B.3). PRISM therefore helps most in the regime it was designed for: long outputs, complex graphs, and rollouts near the parsing cliff.

Unified leave-one-out ablation. All variants share a seed-specific SFT-1000 initialization, a fixed split, LoRA r=32, 100 post-SFT steps, and the same n=100 evaluation pages, averaged over 3 seeds (Table 5). Removing the auxiliary loss is the most damaging single change: pooled strict F1 drops from 0.518 to 0.493, and truncation nearly doubles (3.0% → 5.7%), even though mean output length *increases* (1371 → 1441 tokens). Full PRISM thus produces *shorter* yet more complete graphs (Figure 2b), which rules out length inflation as the explanation. The losses appear in both node and edge precision/recall, with the largest drops in recall (Figure 2c), consistent with truncated outputs that omit objects and relations. Removing reward weighting costs less in-distribution (0.514), and switching everything off (Uniform SDPO) yields the worst result (0.490).

Per-type decomposition. Relative to SFT, PRISM's Graph-F1 gains concentrate on relational structure (Table 4): `READS_AFTER` (+3.8pp) and `OVERLAPS` (+2.9pp) edges, plus main-text nodes (+1.7pp). Easy, high-contrast nodes (center folds, seals, top marginalia) are already saturated, and `ANNOTATES` edges decrease slightly (−1.2pp). Vanilla-SDPO's regression is concentrated on interlinear annotations and reading-order edges, the same categories that truncated outputs lose first.

Teacher parameterization. A frozen teacher (0.5187) and the EMA teacher (0.5181) are statistically indistinguishable, and both keep the full margin over Uniform SDPO (p=0.0041 and p=0.0096). Online self-distillation, where the student is its own teacher and only the prompt is asymmetric, is

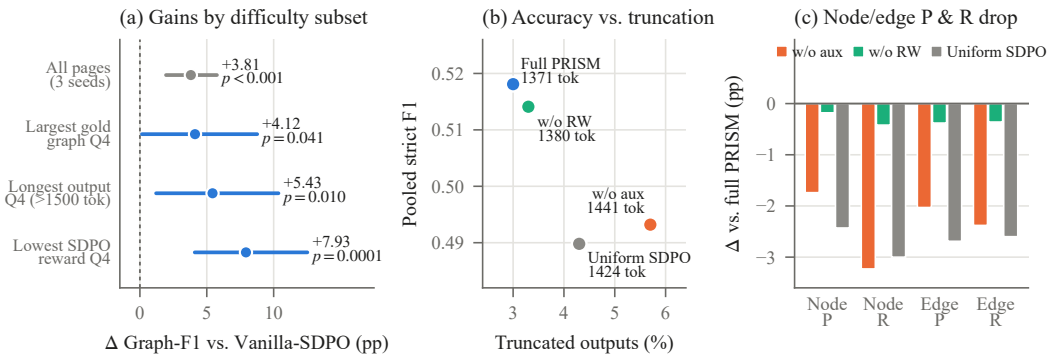


Figure 2: **Where PRISM's gains come from. (a)** PRISM − Vanilla-SDPO Graph-F1 (pp, 95% paired-bootstrap CI) on the full set (3 seeds, Table 3) and on quartile subsets defined by SDPO's failure regimes (5 seeds; 5/5 seeds positive in each). **(b)** Pooled strict F1 versus truncation rate in the unified leave-one-out ablation (3 seeds); labels give mean output length. Full PRISM is the most accurate and the least truncated despite producing the *shortest* outputs. **(c)** Change in node/edge precision and recall relative to full PRISM when a component is removed.

Table 5: Unified component and teacher ablation (matched r=32, 100 steps, identical n=100 pages, 3 seeds). Δ is relative to Uniform SDPO; p from a paired bootstrap. "–" marks quantities not logged for the teacher variants.

Variant	Pooled strict F1	Node P / R	Edge P / R	Trunc. %	Len	Δ vs. Uniform	p
Full PRISM (EMA teacher)	**0.5181**	**0.7067 / 0.6412**	0.3812 / **0.3715**	**3.0**	1371	+0.0284	0.0096
w/o aux loss (RW only)	0.4932	0.6893 / 0.6089	0.3609 / 0.3477	5.7	1441	+0.0034	–
w/o RW (aux only)	0.5141	0.7049 / 0.6370	0.3774 / 0.3679	3.3	1380	+0.0243	–
Uniform SDPO (no DGR, RW, aux)	0.4898	0.6824 / 0.6112	0.3543 / 0.3455	4.3	1424	0	–
Frozen teacher (fixed at init.)	**0.5187**	–	**0.3824** / 0.3682	–	–	+0.0289	0.0041
Online teacher ($\theta_T \equiv \theta_S$)	0.5126	–	0.3747 / 0.3651	–	–	+0.0228	–

directionally weaker (0.5126). The robust ingredient is therefore a *separately parameterized* teacher; EMA is one convenient implementation.

4.4 Learning Dynamics and DGR Verbosity Analysis

Learning Dynamics. The Top-K JSD training loss (Figure 3a) decreases and stabilizes, indicating that reward-weighted distillation aligns the student with the teacher despite the information asymmetry. In the rollout-reward trajectories (Figure 3b), DPO and GRPO plateau with large oscillations, consistent with the short-completion bias producing unstable reward signals, whereas PRISM ends highest among the four families within 100 steps.

DGR Verbosity and Teacher Critique. We sweep DGR verbosity under the matched-r=32 protocol (Table 6). Every DGR variant outperforms the no-DGR baseline. The structural-statistics-only `L2` report has the highest mean, while the `auto` configuration, which routes verbosity by the student's error type, has the lowest cross-seed standard deviation (0.27pp) and is our default.

4.5 Generality: Beyond OracleGraph and Under Distribution Shift

What is OracleGraph-specific? Table 7 separates the parts of PRISM that encode OracleGraph's ontology from those that are task-agnostic. Only the *content* of the DGR and the structural matcher inside the reward depend on the task; reward weighting, the length-normalized loss, and the separate teacher need nothing beyond a scalar score per rollout. The GraphDoc experiments below test exactly this split: we swap in GraphDoc's ontology and evaluator and keep everything else unchanged.

Full PRISM on GraphDoc (retraining). GraphDoc [Chen et al., 2025] differs from OracleGraph in language, domain, layout regime, annotation source (human-verified), and ontology. We keep its official images, splits, node/relation definitions, and evaluator, and serialize its annotations into our graph format without changing their semantics. We instantiate all four PRISM components (DGR-conditioned teacher, Graph-F1 reward weighting, length-normalized loss, EMA teacher); for each of 3 seeds, PRISM starts from the matching GraphDoc SFT-1000 checkpoint and receives 20

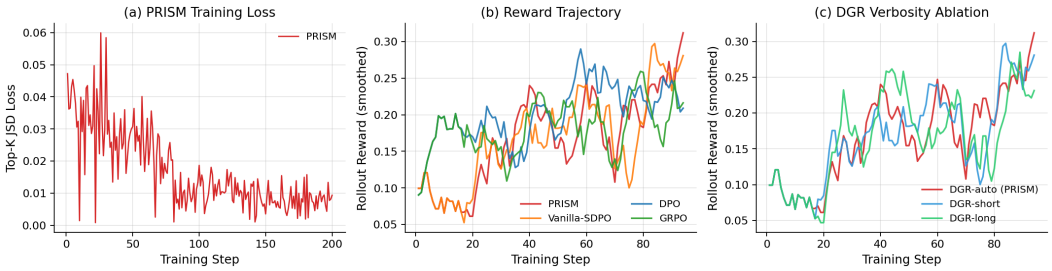


Figure 3: **Learning Dynamics of the PRISM recipe. (a)** Top-K JSD training loss of the PRISM main run (seed 42, 200 steps); it decreases and flattens after step 150. **(b)** Smoothed rollout-reward trajectory (centred moving average, k=7) across four RL families over the first 100 steps. SDPO is visualised by the short-DGR proxy run from the ablation suite; per-seed validation numbers in Table 3 are authoritative. **(c)** DGR-verbosity ablation on the rollout-reward trajectory: DGR-auto (PRISM's default length-adaptive report) versus fixed short and long variants.

Table 6: DGR verbosity ablation (matched-r=32, mean $\pm$ cross-seed std over 3 seeds, same convention as Table 3). The `auto` configuration adjusts feedback to the error type and yields the most stable optimization.

DGR variant	Feedback axes						Graph Reward (3-seed)	Seeds
	P/F	Macro	PerType	Diffs	Gold	Examples		
None (Vanilla SDPO, DGR off)	×	×	×	×	×	×	70.15% ± 1.73pp	3
`L0` (pass/fail only)	✓	×	×	×	×	×	73.49% ± 1.04pp	3
`L2` (+ macro/per-type)	✓	✓	✓	×	×	×	**74.59%** ± 1.14pp	3
`L3` (+ diff items)	✓	✓	✓	✓	×	×	73.77% ± 0.67pp	3
`short` (gold + diff)	×	×	✓	✓	✓	×	73.09% ± 0.93pp	3
`long` (gold + diff + ex.)	×	×	✓	✓	✓	✓	73.12% ± 0.21pp	3
`auto` **(PRISM, Ours)**	*Dynamic*	*Dyn.*	✓	✓	✓	*Dyn.*	73.96% ± **0.27**pp	3

post-training steps. PRISM improves Node F1 by +3.76pp, Edge F1 by +2.67pp, Graph F1 by +2.95pp, and mR_g by +1.89pp (Figure 4a). These simultaneous node and relation gains indicate that the recipe is not tied to OracleGraph's procedural construction or ontology.

Zero-shot transfer. We also evaluate OracleGraph-trained checkpoints directly on GraphDoc, with no weight updates and no GraphDoc-based tuning or checkpoint selection, scoring only the shared text-region ontology. On high-complexity pages (more than the median 12 gold regions, n=23), PRISM improves text precision, recall, and F1 (+2.88pp F1), raises the parse rate from 26.1% to 39.1%, and lowers truncation from 47.8% to 39.1% (Figure 4b), while predicting *fewer* regions per page (34.13 vs. 36.70). Absolute scores stay low under this large language, domain, and ontology shift, so we treat this as a stress test rather than evidence of general zero-shot ability.

Cross-book OOD on OracleGraph. On a 100-page probe from two books outside the 64 training books (*Xiyouji*, *Zizhi Tongjian*), mean Graph Reward does *not* favor PRISM: SFT 20.02%, PRISM 18.29%, DPO 17.41% (no pairwise difference significant, $p \geq 0.19$). PRISM's cross-seed standard deviation is, however, 14× smaller than SFT's (0.16 vs. 2.39pp; Figure 4c). To separate the book (source) shift from the layout-density shift, we split the probe into four matched 25-page cells (two books × low/high density; Table 17). Under the stricter pooled metric, every method collapses (strict F1 ≤ 0.007), and for no method is a source, density, or interaction effect statistically resolved; the overall PRISM−SFT gap is -0.0006 (95% CI $[-0.0030, +0.0011]$). This is a *shared* transfer failure rather than a PRISM-specific one, and the OOD evidence therefore supports stability, not superior mean generalization.

Table 7: Scope of PRISM's components. "Task-specific part" is what must be re-instantiated for a new structured-generation task; all other parts are reused unchanged in the GraphDoc experiments.

Component	Role	Task-specific part	Requirement on a new task
Hindsight teacher + DGR	privileged, dense token-level targets	report content (ontology, error types)	gold structure *or* a programmatic verifier
Graph-F1 reward	rank rollouts by structure	node/edge matcher (IoU, z, classes)	a structural matcher (GraphDoc: official evaluator)
Reward-weighted aggregation	down-weight hallucinated rollouts	none	any scalar score per rollout
Length-normalized aux loss	counteract truncation	none	≥ 2 rollouts per prompt
Separate (EMA) teacher	preserve teacher–student asymmetry	none	none

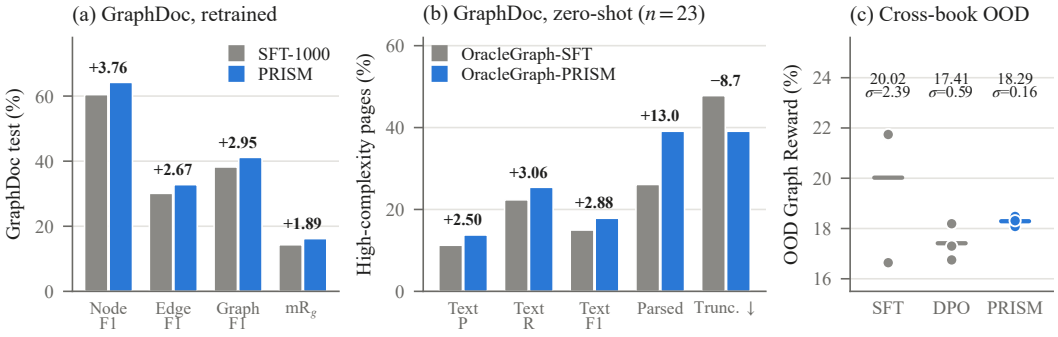


Figure 4: **Beyond the training distribution.** **(a)** Full PRISM retrained on GraphDoc versus the matched GraphDoc SFT-1000 baseline (3-seed means, official evaluator). **(b)** Zero-shot OracleGraph→GraphDoc on high-complexity pages (n=23, shared text-region ontology); truncation is lower-is-better. **(c)** Cross-book OOD Graph Reward per seed (dots), with the mean (bar) and population standard deviation σ over 3 seeds: PRISM's mean is not higher than SFT's, but its spread is $14\times$ tighter.

5 Conclusion, Limitations, and Future Work

We introduce **OracleGraph**, a procedurally generated 10,000-page 2.5D benchmark with exact typed-graph ground truth, a fine-grained hallucination taxonomy, and a 12-dimensional VQA suite for ancient Chinese documents. Our diagnosis shows that scalar-reward RL stalls near SFT on long JSON generation because of a policy-level signal gap, zero-reward parsing cliffs, and the resulting short-completion bias. **PRISM** combines a DGR-conditioned hindsight teacher with reward-weighted distillation and a length-normalized preference loss. It improves on vanilla self-distillation by $+3.81$pp ($p<0.001$), concentrates its gains on long and complex pages, reduces truncation without inflating length, stabilizes training across seeds, and transfers positively to GraphDoc under retraining.

Limitations. (L1) PRISM's $+1.02$pp in-distribution gain over SFT is not significant ($p=0.12$); its supported contribution is stabilizing self-distillation, not reliably breaking the SFT ceiling. (L2) Under cross-book shift, no method improves the mean and all collapse under the strict pooled metric; PRISM reduces variance but does not solve structural OOD transfer. (L3) PRISM needs gold-derived DGRs or an equivalent verifier during post-training. Replacing the DGR with a non-gold programmatic verifier (e.g., renderers or linters for tables and chart-to-code) is plausible but not yet evaluated. (L4) External evidence is limited to one benchmark (GraphDoc, 3 seeds), and OracleGraph pages are synthetic, so transfer to real archival scans remains to be measured. (L5) Reward weighting can amplify noise on pages where the reward is too sparse to rank rollouts (App. B.2).

Future Work. Each limitation suggests a concrete next step: replacing the gold-derived DGR with verifier-generated critiques (HTML renderers for tables, executors or linters for chart-to-code) to test PRISM without gold graphs; a held-out set of real archival scans with expert-verified graphs to measure the synthetic-to-real gap; OOD-aware layout curricula aimed at the shared cross-book failure; adaptive confidence gating of reward weights on reward-sparse pages; and extension of the ontology to multilingual manuscripts.

Broader Impact. OracleGraph and the released 8B checkpoints give digital-humanities researchers an open, locally deployable baseline for structured parsing of historical pages, and the diagnosis of short-completion bias applies to any MLLM that must emit long schema-constrained outputs. Because training pages are synthetic and OOD behavior is weak, outputs should assist, not replace, expert transcription: a dropped annotation or a wrong reading order can change a text's meaning.

References

Mohammad Beigi, Ming Jin, Junshan Zhang, Qifan Wang, and Lifu Huang. Adversarial reward auditing for active detection and mitigation of reward hacking. *CoRR*, abs/2602.01750, 2026. doi: 10.48550/ARXIV.2602.01750. URL https://doi.org/10.48550/arXiv.2602.01750.

Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. Nougat: Neural optical understanding for academic documents, 2023.

Franz Louis Cesista. Multimodal structured generation: Cvpr's 2nd mmfm challenge technical report, 2025. URL https://arxiv.org/abs/2406.11403.

Lei Chen, Xuanle Zhao, Zhixiong Zeng, Jing Huang, Liming Zheng, Yufeng Zhong, and Lin Ma. Breaking the sft plateau: Multimodal structured reinforcement learning for chart-to-code generation, 2026a. URL https://arxiv.org/abs/2508.13587.

Wei Chen, Yubing Wu, Junmei Yang, Delu Zeng, Qibin Zhao, John Paisley, Min Chen, and Zhou Wang. Towards disentangled preference optimization dynamics beyond likelihood displacement, 2026b. URL https://arxiv.org/abs/2604.18239.

Yufan Chen, Ruiping Liu, Junwei Zheng, Di Wen, Kunyu Peng, Jiaming Zhang, and Rainer Stiefelhagen. Graph-based document structure analysis. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Fu0aggezN9.

Ruiyi Ding, Yongxuan Lv, Xianhui Meng, Jiahe Song, Chao Wang, Chen Jiang, and Yuan Cheng. Prpo: Aligning process reward with outcome reward in policy optimization, 2026. URL https://arxiv.org/abs/2601.07182.

Wu Fei, Hao Kong, Shuxian Liang, Yang Lin, Yibo Yang, Jing Tang, Lei Chen, and Xiansheng Hua. Self-guided process reward optimization with redefined step-wise advantage for process reinforcement learning, 2025. URL https://arxiv.org/abs/2507.01551.

Zhili Feng, Dhananjay Ram, Cole Hawkins, Aditya Rawal, Jinman Zhao, and Sheng Zha. Sequence-level large language model training with contrastive preference optimization, 2025. URL https://arxiv.org/abs/2502.16433.

Yanggan Gu, Junzhuo Li, Sirui Huang, Xin Zou, Zhenghua Li, and Xuming Hu. Capturing nuanced preferences: Preference-aligned distillation for small language models, 2025. URL https://arxiv.org/abs/2502.14272.

Anwen Hu, Haiyang Xu, Jiabo Ye, Ming Yan, Liang Zhang, Bo Zhang, Chen Li, Ji Zhang, Qin Jin, Fei Huang, and Jingren Zhou. mplug-docowl 1.5: Unified structure learning for ocr-free document understanding, 2024. URL https://arxiv.org/abs/2403.12895.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation, 2026. URL https://arxiv.org/abs/2601.20802.

Qi Ju, Falin Hei, Zhemei Fang, and Yunfeng Luo. Elo-rated sequence rewards: Advancing reinforcement learning models, 2025. URL https://arxiv.org/abs/2409.03301.

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 (ECCV)*, 2022.

Chen Li, Han Zhang, Zhantao Yang, Fangyi Chen, Zihan Wang, Anudeepsekhar Bolimera, and Marios Savvides. Stelar-vision: Self-topology-aware efficient learning for aligned reasoning in vision, 2026a. URL https://arxiv.org/abs/2508.08688.

Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. Unifying group-relative and self-distillation policy optimization via sample routing, 2026b. URL https://arxiv.org/abs/2604.02288.

Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. Docbank: A benchmark dataset for document layout analysis. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, 2020.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe, 2026c. URL https://arxiv.org/abs/2604.13016.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Peng Liao, Peijia Zheng, Lingbo Li, Shangsong Liang, and Lin Chen. Intrinsic mutual information as a modulator for preference optimization, 2026. URL https://arxiv.org/abs/2604.24804.

Chengqi Lyu, Songyang Gao, Yuzhe Gu, Wenwei Zhang, Jianfei Gao, Kuikun Liu, Ziyi Wang, Shuaibin Li, Qian Zhao, Haian Huang, Weihan Cao, Jiangning Liu, Hongwei Liu, Junnan Liu, Songyang Zhang, Dahua Lin, and Kai Chen. Exploring the limit of outcome reward for learning mathematical reasoning, 2025. URL https://arxiv.org/abs/2502.06781.

Monte MacDiarmid, Benjamin Wright, Jonathan Uesato, Joe Benton, Jon Kutasov, Sara Price, Naia Bouscal, Sam Bowman, Trenton Bricken, Alex Cloud, Carson Denison, Johannes Gasteiger, Ryan Greenblatt, Jan Leike, Jack Lindsey, Vlad Mikulik, Ethan Perez, Alex Rodrigues, Drake Thomas, Albert Webson, Daniel Ziegler, and Evan Hubinger. Natural emergent misalignment from reward hacking in production rl, 2025. URL https://arxiv.org/abs/2511.18397.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward, 2024. URL https://arxiv.org/abs/2405.14734.

Jan Peters and Stefan Schaal. Using reward-weighted regression for reinforcement learning of task space control. *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, pages 262–267, 2007. URL https://api.semanticscholar.org/CorpusID:14180705.

Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S. Nassar, and Peter W. J. Staar. Doclaynet: A large human-annotated dataset for document-layout segmentation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2022.

Deng Qiyuan, Kehai Chen, Min Zhang, and Zhongwen Xu. HiPO: Self-hint policy optimization for RLVR. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=rcb20pHmT1.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Ahmad Rashid, Ruotian Wu, Julia Grosse, Agustinus Kristiadi, and Pascal Poupart. A critical look at tokenwise reward-guided text generation, 2025. URL https://arxiv.org/abs/2406.07780.

Massimo Rizzoli, Simone Alghisi, Olha Khomyn, Gabriel Roccabruna, Seyed Mahed Mousavi, and Giuseppe Riccardi. Civet: Systematic evaluation of understanding in vlms, 2025. URL https://arxiv.org/abs/2506.05146.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Tim G. J. Rudner, Vitchyr H. Pong, Rowan Thomas McAllister, Yarin Gal, and Sergey Levine. Outcome-driven reinforcement learning via variational inference. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=4bzanicqvy8.

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

Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning, 2025. URL https://arxiv.org/abs/2504.08837.

Xiaohua Wang, Muzhao Tian, Yuqi Zeng, Zisu Huang, Jiakang Yuan, Bowen Chen, Jingwen Xu, Mingbo Zhou, Wenhao Liu, Muling Wu, Zhengkang Guo, Qi Qian, Yifei Wang, Feiran Zhang, Ruicheng Yin, Shihan Dou, Changze Lv, Tao Chen, Kaitao Song, Xu Tan, Tao Gui, Xiaoqing Zheng, and Xuanjing Huang. Reward hacking in the era of large models: Mechanisms, emergent misalignment, challenges, 2026. URL https://arxiv.org/abs/2604.13602.

Han Weng, Puzhen Wu, Cui Longjie, Yi Zhan, Boyi Liu, Yuanfeng Song, Dun Zeng, Yingxiang Yang, Qianru Zhang, Dong Huang, Xiaoming Yin, Yang Sun, and Xing Chen. Graph-reward-SQL: Execution-free reinforcement learning for text-to-SQL via graph matching and stepwise reward. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 12917–12943, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.694. URL https://aclanthology.org/2025.findings-emnlp.694/.

Junkang Wu, Xue Wang, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. Alphadpo: Adaptive reward margin for direct preference optimization, 2025. URL https://arxiv.org/abs/2410.10148.

Shuchen Xue, Chongjian GE, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning RL with pretraining in diffusion models, 2026. URL https://openreview.net/forum?id=wekWtzTbUP.

Ning Yang, Hai Lin, Yibo Liu, Baoliang Tian, Guoqing Liu, and Haijun Zhang. Token-importance guided direct preference optimization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=cMEnMVvMw9.

Qi Zhang, Shouqing Yang, Lirong Gao, Hao Chen, Xiaomeng Hu, Jinglei Chen, Jiexiang Wang, Sheng Guo, Bo Zheng, Haobo Wang, and Junbo Zhao. Lets: Learning to think-and-search via process-and-outcome reward hybridization, 2025a. URL https://arxiv.org/abs/2505.17447.

Yizhuo Zhang, Heng Wang, Shangbin Feng, Zhaoxuan Tan, Xinyun Liu, and Yulia Tsvetkov. Generalizable LLM learning of graph synthetic data with reinforcement learning. *CoRR*, abs/2506.00845, 2025b. doi: 10.48550/ARXIV.2506.00845. URL https://doi.org/10.48550/arXiv.2506.00845.

Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. Publaynet: Largest dataset ever for document layout analysis. In *Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR)*, 2019.

Guanghao Zhou, Panjia Qiu, Cen Chen, Jie Wang, Zheming Yang, Jian Xu, and Minghui Qiu. Reinforced mllm: A survey on rl-based reasoning in multimodal large language models, 2025. URL https://arxiv.org/abs/2504.21277.

Wenxuan Zhou, Shujian Zhang, Lingxiao Zhao, and Tao Meng. T-reg: Preference optimization with token-level reward regularization, 2024. URL https://arxiv.org/abs/2412.02685.

Deyu Zou, Yongqiang Chen, Jianxiang Wang, Garry YANG, Mufei Li, Qing Da, James Cheng, Pan Li, and Yu Gong. Reducing belief deviation in reinforcement learning for active reasoning of LLM agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=r8hzDA3pUY.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction separate what PRISM achieves (a $+3.81$pp gain over vanilla self-distillation, $p < 0.001$; $2.4\times$ lower cross-seed std than SFT; gains concentrated on complex pages; positive transfer to GraphDoc under retraining) from what it does not (a significant in-distribution gain over SFT, $p = 0.12$; superior mean OOD generalization). The *policy-level signal gap* for scalar RL is stated as an empirical characterization, not a proven causal mechanism. Each claim is backed by $n = 100$, multi-seed paired-bootstrap results in Section 4, and the scope of the method is stated in Section 4.5 and the Limitations.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We dedicate a specific sub-section to explicit limitations in Section 5 (Limitations and Future Work). We document five constraints: (L1) the lack of statistical significance at $p < 0.05$ for PRISM's in-distribution gain over SFT; (L2) a shared OOD transfer failure under cross-book shift, where PRISM reduces variance but does not improve the mean (separated into source and layout-density factors in Appendix C.3); (L3) the need for gold-derived DGRs or a programmatic verifier, with non-gold verifiers not yet evaluated; (L4) external evidence limited to one benchmark (GraphDoc) and synthetic OracleGraph pages, so transfer to real archival scans is unmeasured; and (L5) the potential for reward weighting to hurt on reward-sparse pages.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.

- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The contributions of this paper are primarily empirical and data-centric (the OracleGraph benchmark and the empirical diagnosis of RL regimes in structured generation). While we formalize the PRISM surrogate objective and EMA dynamics mathematically in Section 3, these are algorithm designs and empirical heuristics rather than formal theoretical theorems requiring mathematical proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We release the full 10,000-page OracleGraph dataset, the 12-dimensional VQA diagnostic suite, the deterministic Graph-F1 evaluation script, and the training code for PRISM via our public GitHub repository (`https://github.com/209-Tongji/PRISM_2026/`). We explicitly document all critical hyperparameters (e.g., LoRA $r = 32$, EMA $\gamma = 0.999$, importance clip $\epsilon = 0.1$) in Section 3, and give a step-by-step description of one PRISM update in Appendix B.1 and exact metric definitions in Appendix A.4. Our RL comparison protocol is executed across 3 random seeds under identical compute budgets and evaluated via 10,000-resample paired bootstrap; the GraphDoc protocols are given in Appendix C.6.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same

dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide a public GitHub repository (`https://github.com/209-Tongji/PRISM_2026/`) containing the source code for PRISM, the deterministic Graph-F1 evaluation script, and the task-adaptive VQA scoring framework. We also fully release the **OracleGraph** dataset, including all 10,000 procedurally generated pages with their oracle graphs and the 12-dimensional VQA suite, under a CC-BY-NC-4.0 license. Detailed instructions to reproduce the SFT and RL environments are provided in the README.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: The core experimental setting, including the data splits (in-distribution vs. cross-book OOD), the matched-compute protocol, and critical hyperparameters (LoRA $r = 32$, EMA $\gamma = 0.999$, importance clip $\epsilon = 0.1$, $g = 2$, and $\max_new_tokens = 3072$) are explicitly detailed in Section 3 and Section 4.1. The graph metrics are specified in Appendix A.4, the task-adaptive VQA metric in Appendix A.5, and the GraphDoc retraining and zero-shot protocols in Appendix C.6.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report the cross-seed standard deviation across 3 random seeds for all matched-rank RL comparisons in Table 3 and the OOD probe in Table 16. Furthermore, we establish the statistical significance of our primary intra-family ablation (PRISM vs. Vanilla-SDPO) and other RL baselines using a rigorous **10,000-resample paired bootstrap** protocol on the $n = 100$ held-out set, explicitly reporting both the 95% Confidence Intervals (CIs) and the exact p-values. We also perform a pre-registered sub-group analysis with Holm-Bonferroni family-wise error correction (Appendix B.2), report bootstrap CIs and seed consistency for the difficulty decomposition (Appendix B.3, including a regression-to-the-mean caveat), and state that the OOD standard deviations are population standard deviations over seeds.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Appendix D.1 (Table 22) reports hardware (NVIDIA H20, 96 GB), GPUs per run, wall-clock per run, and total GPU-hours per experiment group; Table 15 reports inference footprints. The training budget for matched-compute comparisons (100 optimization steps) is stated in Section 4.2.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research involves structural parsing of procedurally rendered pages whose text comes from public-domain classical Chinese works, plus the publicly released GraphDoc benchmark. It does not involve human subjects, living individuals' personally identifiable information (PII), or technologies that could inflict societal harm, fully conforming to the NeurIPS Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: We explicitly note the positive impact of our work on the digital humanities community in Section 1 (providing reliable open-weight parsers for ancient corpora). We do not include a dedicated discussion on negative societal impacts because the structural parsing of ancient, public-domain historical manuscripts does not present foreseeable malicious applications (e.g., no privacy, surveillance, or deepfake risks).

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our released assets (the OracleGraph dataset and the PRISM layout-parsing models) focus strictly on the structural layout analysis of historical literature. They do not fall under the category of high-risk generative models (such as deepfake generators or open-ended LLMs) or privacy-sensitive datasets that would require specific safety filters or gated release safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: OracleGraph pages are rendered by our own pipeline (Appendix A.2) from public-domain classical texts for which no contemporary rights apply; no third-party scans are redistributed. GraphDoc [Chen et al., 2025] and all evaluated models are cited and used under the terms of their official releases. The OracleGraph benchmark is released under the CC-BY-NC-4.0 license for research use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We introduce the **OracleGraph** benchmark and the **PRISM** framework. The dataset composition, typed-graph ontology, and procedural synthesis pipeline are documented in Appendices A.1 and A.2, and the metrics in Appendix A.4. Both the assets and their documentation are released via our public GitHub repository.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The dataset is procedurally generated, and its ground truth is emitted by the rendering program rather than by annotators. The work does not involve crowdsourcing platforms (e.g., MTurk) or human subjects research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This research focuses entirely on the structural parsing of procedurally rendered pages of public-domain Chinese literature and of a public document benchmark, and does not involve human subjects. Thus, IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

A Technical Appendices and Supplementary Material

A.1 The OracleGraph Corpus: Design and Generation

OracleGraph consists of 10,000 high-resolution page images in the style of Chinese ancient documents (principally block-printed novels, historical ledgers, and hand-copied manuscripts). Every page is *procedurally generated*: it is rendered from a programmatic layout specification, and its JSON oracle graph is emitted by the same program (App. A.2). No node, polygon, z-index, or edge is drawn or corrected by hand, so the ground truth is exact with respect to the rendering program. Earlier drafts of this work used the terms "hand-annotated" and "hand-verified"; these referred to the manually authored layout programs and to visual inspection of rendered pages, not to manual graph annotation.

Beyond raw layout, the generator deliberately injects the structural ambiguities that we find to be the dominant failure modes of generalist MLLMs on ancient Chinese content. Each page records its injected scenario in the `adversarial_scenario` field (Listing 1):

- **Type I: Layout Hierarchy & Reading Order.** *Double-line annotation mixing* (small-font paired commentary interleaving main text and being hierarchically conflated), *marginalia misalignment* (wrong vertical anchoring of top/bottom/side notes), *cross-column boundary breakage* (mispredicting which column a sentence continues into), *implicit grid collapse* (columns defined only by absence of a rule, not by an explicit line).
- **Type II: Editing Instructions & Symbols.** *Alteration ignorance* (failing to process historical deletions or insertions encoded by edit marks), *inversion-mark ignorance* (ignoring traditional transposition marks), *punctuation misinterpretation* (reading vermilion grammatical dots as lexical tokens).
- **Type III: Multi-modal Interference & Overlay.** *Dense text-image mismatch*, *sub-image relation loss*, *seal intrusion* (red stamps partially occluding black text), and *multi-color hierarchy confusion* (mixing vermilion commentary with standard main text).

Listing 1: Truncated JSON representation of an OracleGraph page. The schema defines bounding-box polygon, semantic node type, and per-node occlusion-layer index.

```
{
  "image_id": "hongloumeng_page_025",
  "adversarial_scenario": "NOTE_MISANCHORING",
  "metadata": { "width": 1885, "height": 1333 },
  "oracle_graph": {
    "nodes": [
      {
        "node_id": "n_frame1",
        "polygon": [167, 413, 1641, 413, 1641, 1194, 167, 1194],
        "type": "OUTER_PAGE_FRAME",
        "attributes": { "z_index": 0, "color": "black" }
      },
      {
        "node_id": "n_separator1",
        "polygon": [1641, 413, 1641, 1194, 1640, 1194, 1640, 413],
        "type": "COLUMN_SEPARATOR",
        "attributes": { "z_index": 0, "color": "black" }
      }
      // ... (Text nodes, Seals, and Edges omitted for brevity)
    ]
  }
}
```

A.2 Synthesis Pipeline Architecture

Figure 5 summarizes the generator. Its key property is that *the image and the graph are two outputs of one program*: the graph is not inferred from the image, so it inherits none of the bounding-box jitter, annotator disagreement, or subjective topology decisions of manual labeling. This makes the Graph-F1 reward used for RL noise-free with respect to the specification.

1. **Source text.** Text is drawn from public-domain classical works (64 books in the training pool). Two further books, *Xiyouji* and *Zizhi Tongjian*, are held out entirely for the cross-book OOD probe (App. C.2).

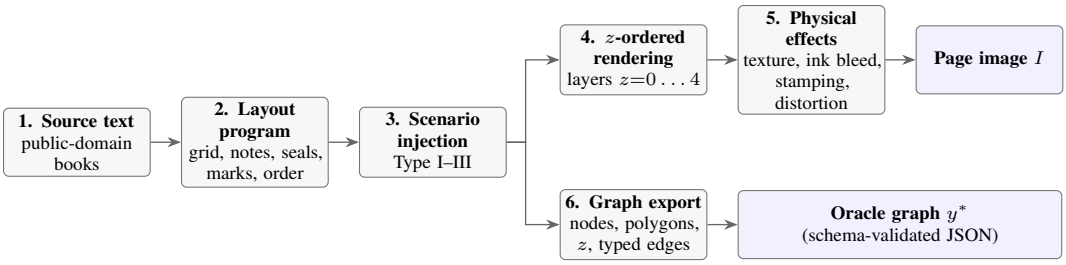


Figure 5: **OracleGraph synthesis pipeline.** One layout program drives both rendering (upper lane) and graph export (lower lane), so the oracle graph is exact by construction.

2. **Layout program.** A historical-typesetting engine instantiates the page geometry (outer frame, column grid, center fold and fishtail marks) and places main-text columns, double-line interlinear commentary, top/bottom/side marginalia, seals, edit and inversion marks, and punctuation dots. Every placed element is assigned one of the 13 node classes, a polygon, a color, and an occlusion layer $z \in \{0, \ldots, 4\}$, and the engine records the directed relations among elements (8 edge classes, e.g., `READS_AFTER`, `ANNOTATES`, `OVERLAPS`).
3. **Scenario injection.** The layout is perturbed to realize one of the Type I–III ambiguities above (e.g., `NOTE_MISANCHORING`), and the perturbation is recorded in the page metadata.
4. **z-ordered rendering.** Elements are rasterized in increasing z, so physical occlusion (e.g., a vermilion seal at z=4 stamped over ink text at z=1) appears in the pixels while both nodes, and the `OVERLAPS` relation between them, remain in the graph. This is what makes the 2.5D evaluation possible: a model must recover the occluded element *and* its layer.
5. **Physical effects.** Historical paper texture, ink bleed, seal-stamping artifacts, and optical distortion are applied to improve visual realism and domain transferability.
6. **Graph export and validation.** The recorded elements and relations are serialized into the JSON schema of Listing 1 and validated against the schema. The 4,500-page stress-test split is produced by the layout-diversity engine of step 2 with broader sampling of layout configurations.

A.3 Examples of Rendered Pages

The following figures show representative rendered pages; each is released with its oracle graph.

A.4 Metric Specification

Graph Reward / Graph-F1 (macro). For each page, predicted and gold nodes are matched one-to-one by greedy assignment in decreasing IoU order. A pair counts as a node TP only if (i) the node classes are equal, (ii) bounding-box IoU > 0.5, and (iii) the occlusion layers are equal ($z_{\hat{v}} = z_{v^*}$). Thus, a correctly localized seal predicted on the wrong layer is simultaneously a false positive and a false negative. An edge is a TP only if both endpoints are matched to the corresponding gold endpoints and the edge class agrees. Per-class F1 is computed for every node and edge class present in the gold or predicted graph, and the page reward is their unweighted (macro) mean, Eq. (2). Unparseable outputs score 0. Reported Graph Reward is the mean over pages and then over seeds. All numbers in the paper use this macro definition; the phrase "micro-averaged" in an earlier draft was an error in the text, not in the implementation.

Pooled strict F1. Used in the unified ablation (Table 5) and the factorial OOD probe (Table 17). Node and edge TP/FP/FN counts, under the same matching rules, are summed over all evaluation pages, and a single F1 is computed from these totals. Unparseable outputs contribute all of their gold objects as FNs. Unlike the page-averaged macro reward, this instance-level metric gives no credit to easy classes on otherwise failed pages, so it is substantially lower in absolute terms. Node/edge precision and recall in Table 5 are computed in the same pooled way.

Truncation and parse rates. An output is *truncated* if generation stops at the 3,072-token limit before the JSON object is closed, and *parsed* if the schema-aware parser recovers a graph.

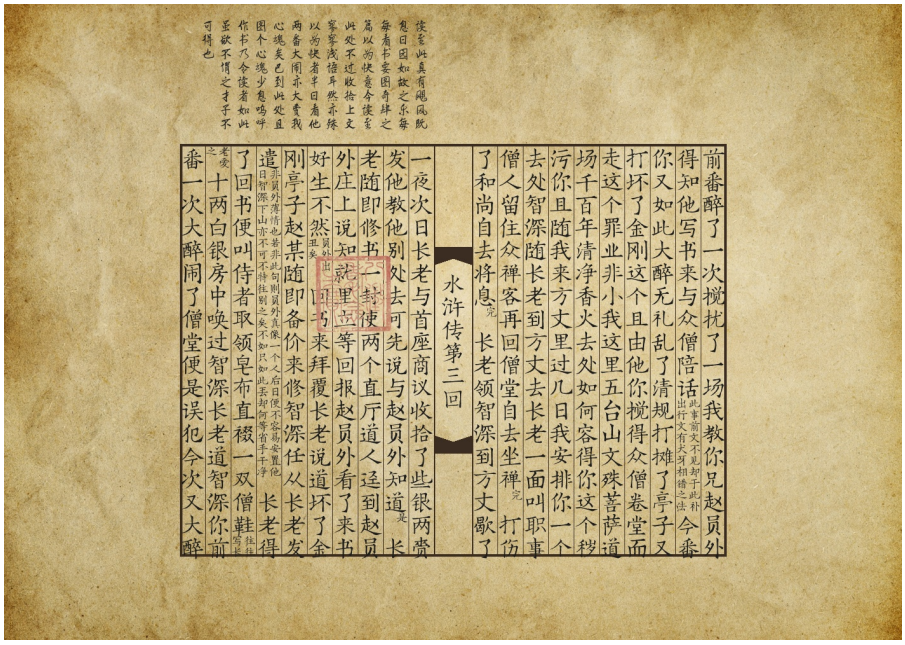


Figure 6: Procedurally rendered page in the style of a block-printed *Water Margin* edition. The page exhibits Type I (double-line interlinear annotations, top marginalia) and Type III (seal intrusion) ambiguities. The paired oracle graph (not shown) encodes the occlusion layer of the seal and the reading-order edges of the overlapping elements.



Figure 7: Procedurally rendered page in the style of a block-printed *The Story of the Stone (Hongloumeng)* edition, featuring main text with cross-column side marginalia, second-layer marginal comments, and cinnabar seals interspersed with the ink text.

A.5 A Task-Adaptive Evaluation Framework for MLLMs in Complex Layout Understanding

Traditional exact match paradigms struggle to objectively reflect the true representational capacity of Multimodal Large Language Models (MLLMs) due to their generative heterogeneity—such as redundant responses, inconsistent numerical expressions, and long-text dislocations—when processing visual question answering tasks on high-resolution ancient document layouts. Inspired by recent studies on alignment and model robustness, this research proposes a task-adaptive quantitative evaluation framework (see Table 8). Tailored to the distinct characteristics of various task dimensions, this framework constructs three customized evaluation models to systematically mitigate evaluation bias through the specific mechanisms detailed below.

Model I: Spatial Localization-Oriented Evaluation (Discrete, 0/1). This model is applied to tasks requiring precise coordinate-to-text anchoring, including **BVR, MBCV, SVC, CCB, RVR, and AGC**. Given the tendency of MLLMs to generate verbose explanations, we implement a *Multi-level Fault Tolerance* mechanism:

- **Exact Match:** Standardized comparison after stripping punctuation and whitespace.
- **Constrained Forward Inclusion:** We perform a reverse search for the Ground Truth (GT) anchor within the prediction. To prevent "cheating" via exhaustive generation, the match is only valid if the total response length is within $5\times$ the length of the GT.
- **Reverse Prefix Matching:** If the model's output is a valid prefix substring of the GT, it is counted as a successful physical localization, accounting for premature sequence truncation.

Scope of the lenient rules. Forward inclusion and reverse prefix matching apply *only* to the six short-answer localization tasks of Model I, where the object of evaluation is whether the model anchored the correct location, not the exact wording of the answer. They are never used for Graph-F1, pooled strict F1, parse/truncation rates, or any JSON-generation metric. Therefore, they cannot mask or inflate the short-completion bias analyzed in the main text, which is measured exclusively on graph outputs.

Model II: Semantic Equivalence-Based Numerical Evaluation (Discrete, 0/1). This model targets quantitative tasks such as **GQR (Grid Query)** and **SCR (Seal Counting)**. It resolves "modality aliasing" through a three-step pipeline:

1. **Heuristic Filtering:** Removal of colloquial prefixes to eliminate false-positive interference from characters like "one".
2. **Unified Numerical Mapping:** A positional weight-based decoding graph converts Chinese numeral strings (0–999) into standardized Arabic formats.
3. **Set Verification:** We extract all continuous numerical values using regular expressions. A score of 1.0 is granted if the GT exists within the predicted numerical set.

Model III: Order-Agnostic Continuous Evaluation (Continuous, 0–1). For long-form extraction tasks like **EXR (Full Extraction)**, where OCR perturbations and layout dislocations are common, we move beyond binary judgments to a *character-set F1* metric (P/R/F1 computed over the multiset of characters treated as unordered, not a position-aware token-F1). We make this design choice *explicit* because character-F1 is ambiguous in the document-parsing literature: "character-F1" can refer to position-aware alignment (e.g., Levenshtein-based), multiset-overlap, or unordered-set-overlap, and the three are not equivalent. We use the **unordered-multiset variant** on EXR specifically so that line-swapping or sequence dislocations caused by complex non-linear ancient layouts do not penalize a correct extraction:

- **Recall (R)** measures the completeness of information extraction: $R = \frac{|\text{GT} \cap \text{Pred}|}{|\text{GT}|}$
- **Precision (P)** constrains model hallucinations and penalizes meaningless redundant generation: $P = \frac{|\text{GT} \cap \text{Pred}|}{|\text{Pred}|}$
- **F1 Score** serves as the final continuous metric for this question type: $F1 = \frac{2 \times P \times R}{P + R}$

EXR therefore measures *character recognition only*. It deliberately ignores order and should not be read as a measure of layout or reading-order understanding. Reading order is evaluated separately and strictly through the `READS_AFTER` edges of Graph-F1, which keeps character-recognition errors apart from topological errors.

Table 8: Illustrative Examples of the Task-Adaptive Evaluation Framework Addressing Generative Heterogeneity.

Response Feature	Ground Truth	Model Response	Scoring Logic	Score
Model I: Spatial Localization-Oriented Short Text (Binary)				
Ideal Match	Incredible	Incredible	Exact Match	1.0
Truncation	Incredible	Incre	Prefix Match	1.0
Severe Hallucination	Incre	Incredible and very...	Length Constraint*	0.0
Model II: Semantic Equivalence-Based Numerical Statistics (Binary)				
Modal Aliasing	15	Fifteen	Semantic Normalization	1.0
Ideal Match	15	15	Set Verification	1.0
Linguistic Noise	1	The 1st choice is 2	Heuristic Filtering	0.0
Model III: Order-Agnostic Continuous Long Text ($F1 \in [0, 1]$)				
Sequence Dislocation	The river flows east	The flows river east	Order-Agnostic F1	1.000
Redundant Generation	The river flows east	The river flows east forever	Hallucination Penalty	0.889

* Penalized due to excessive hallucinations exceeding the $5 \times |\text{GT}|$ length threshold.

A.6 Unweighted vs. QA-Count-Weighted (Micro) VQA Averages

All main-text VQA aggregates (Table 2 and the structural-subset numbers in §4.1) are *unweighted means* of the displayed per-metric accuracies, so that each diagnostic dimension contributes equally and every aggregate can be re-derived from the displayed cells. The 12 metrics do not contain the same number of QA pairs (SEH is empty and omitted; the 12 reported metrics total 1,150 QA). A second natural summary is the *QA-count-weighted micro average*: a single accuracy over all 1,150 QA pairs. Table 9 reports both. The unweighted column is identical to Table 2; an earlier draft listed stale unweighted values for our checkpoints in this table, which caused the reported Table 1/Table 9 discrepancy.

Table 9: Unweighted vs. QA-count-weighted (micro) Avg on the 12-dimensional VQA suite. *Unweighted* equals the **Avg** column of Table 2. *Micro* is the single accuracy over all 1,150 QA pairs.

Model	**Unweighted Avg (main)**	**Micro Avg (per-QA)**	Δ (Micro − Unw.)
GPT-4o-mini	11.37	6.47	−4.90
Gemini-2.5-Flash-Thinking†	35.12	31.46	−3.66
Llama-3.2-11B-Vision-Ins	2.91	2.29	−0.62
Gemma-3-12B-IT	5.02	3.16	−1.86
InternVL3-8B	3.28	3.48	+0.20
InternVL2.5-8B	3.55	3.21	−0.34
Qwen2.5-VL-7B-Ins	18.90	11.91	−6.99
Ovis2-8B	13.45	7.75	−5.70
Qwen3-VL-8B (SFT Baseline, ours)	26.17	20.43	−5.74
PRISM (seed=42, step 100, ours)	26.05	20.61	−5.44
PRISM (seed=2024, step 180, ours)	25.89	20.35	−5.54

Structural subset. We define the structural subset as the 8 metrics of *Basic Logic* (BVR, HCE) and *Advanced 2D Spatial* (SVC, CCB, RVR, AGC, MBCV, EFD), and report its unweighted mean. Using Table 2, it is 21.63 (SFT), 21.50 / 21.38 (PRISM seeds 42 / 2024), 18.38 (Qwen3-VL-8B base), 8.93 (Qwen2.5-VL-7B), 5.42 (Ovis2-8B), 1.08–3.34 (other open-weight models), 29.50 (Gemini-2.5-Flash-Thinking), and 3.69 (GPT-4o-mini). These values supersede the structural-subset numbers of the submitted draft, which were computed from an earlier evaluation run.

B Extended Reinforcement Learning Analysis

B.1 One PRISM Optimization Step

For reproducibility, a single PRISM update on a page (I, x, y^*) proceeds as follows (hyperparameters as in the main text: G=2, K=20, τ=0.5, β=2.5, α=0.2, ϵ=0.1, γ=0.999, LoRA r=32, max_new_tokens=3072).

1. **Rollout.** Set $\pi_{\theta_{\text{old}}} \leftarrow \pi_{\theta_S}$ and sample $y^{(1)}, \ldots, y^{(G)} \sim \pi_{\theta_{\text{old}}}(\cdot \mid \mathbf{h}^B)$; store the per-token log-probabilities under $\pi_{\theta_{\text{old}}}$.
2. **Reward.** Parse each rollout and compute $r^{(j)}$ with the macro Graph-F1 of App. A.4 ($r^{(j)}$=0 if unparseable). Compute $w^{(j)} = \text{softmax}_j(r^{(j)}/\tau)$, the leave-one-out advantage $\hat{A}^{(j)} = r^{(j)} - \frac{1}{G-1}\sum_{k \neq j} r^{(k)}$ (for G=2: $\hat{A}^{(1)} = r^{(1)} - r^{(2)}$), and y^{best}, y^{worst}.
3. **Teacher targets.** Build the DGR from the gold graph (for diff-based levels, also from the rollout's discrepancies with it; Table 6). Run the EMA teacher on $\mathbf{h}^H$, teacher-forced on each student rollout, keep the top-K tokens per position, and renormalize them (Eq. 1). No gradient flows into the teacher.
4. **Student update.** For each PPO epoch, recompute student log-probabilities, the ratios $\rho_t^{(j)}$ (relative to the stored $\pi_{\theta_{\text{old}}}$), $\mathcal{L}_{\text{RW}}$, and $\mathcal{L}_{\text{aux}}$, and take a gradient step on $-\mathcal{J}_{\text{PRISM}}$ (Eq. 4) with respect to the student LoRA parameters.
5. **Teacher update.** $\theta_T \leftarrow \gamma\,\theta_T + (1-\gamma)\,\theta_S$.

B.2 Pre-registered Sub-group Analysis with Family-Wise Correction

We pre-registered a 33/66 tertile split of the n=100 held-out pages on the SFT-r32-1000 per-page reward: cuts at t_{33}=0.7083 and t_{66}=0.7951, giving Easy (n=34), Medium (n=33), and Hard (n=33) buckets. The family of tests is 3 buckets $\times$ 6 baselines = 18 paired-bootstrap comparisons, so we control family-wise error at α=0.05 via Holm-Bonferroni step-down. Table 10 reports the raw and Holm-adjusted p-values together with paired Cohen's d_z.

Table 10: Pre-registered 33/66 tertile sub-group analysis of n=100 held-out pages, with Holm-Bonferroni correction over all 18 (bucket $\times$ baseline) comparisons. Effect size is paired Cohen's d_z. Only Medium $\times$ {SimPO, Vanilla-SDPO} survive at FWE α=0.05 (marked ✓).

Bucket	vs	Δ	Raw p	Holm-adj p	d_z	Sig (adj)
Easy	SFT-r32-1000	−0.019	0.049	0.488	−0.31	
Easy	DPO	+0.013	0.235	1.000	+0.20	
Easy	GRPO	+0.025	0.034	0.374	+0.33	
Easy	SimPO	+0.015	0.121	1.000	+0.26	
Easy	Vanilla-SDPO	+0.025	0.012	0.183	+0.41	
Easy	DGR-long	+0.014	0.276	1.000	+0.18	
Medium	SFT-r32-1000	+0.024	0.015	0.213	+0.40	
Medium	DPO	+0.031	0.005	0.074	+0.42	(borderline)
Medium	GRPO	+0.024	0.023	0.278	+0.36	
Medium	SimPO	+0.038	0.001	**0.014**	+0.46	✓
Medium	Vanilla-SDPO	+0.039	0.001	**0.017**	+0.49	✓
Medium	DGR-long	+0.008	0.328	1.000	+0.16	
Hard	SFT-r32-1000	+0.001	0.884	1.000	+0.01	
Hard	DPO	−0.035	0.264	1.000	−0.19	
Hard	GRPO	−0.043	0.177	1.000	−0.23	
Hard	SimPO	−0.014	0.499	1.000	−0.12	
Hard	Vanilla-SDPO	+0.050	0.021	0.276	+0.36	
Hard	DGR-long	−0.003	0.883	1.000	−0.03	

Mechanism interpretation. *Easy* (SFT $\geq$ 0.80): SFT has near-saturated the graph reward; PRISM *slightly hurts* on raw Δ=−0.019 (loses FWE significance). *Medium* (informative-reward regime): PRISM beats SimPO and Vanilla-SDPO after Holm correction. *Hard* (SFT $<$ 0.71): DPO (+3.5pp)

and GRPO ($+4.3$pp) numerically outperform PRISM on raw Δ (FWE not significant); only Vanilla-SDPO is reliably beaten. This is consistent with the "reward-weighting hurts when reward is too sparse to rank rollouts" interpretation discussed in §5 (L5). Note that these buckets are defined on the *SFT* reward and compare PRISM against every baseline, whereas the difficulty decomposition below is defined on Vanilla-SDPO's failure regimes and compares PRISM only against SDPO; the two analyses answer different questions.

B.3 Difficulty Decomposition: Protocol and Caveats

Protocol. PRISM and Vanilla-SDPO are each trained with 5 seeds under the matched protocol of Table 3. For every held-out page we compute the seed-averaged Graph-F1 difference (PRISM $-$ SDPO), then summarize three quartile subsets: the 25% of pages with the *lowest SDPO reward*, the 25% with the *longest SDPO outputs* (all above 1,500 tokens), and the 25% with the *largest gold graphs* ($|\mathcal{V}^*|+|\mathcal{E}^*|$). Confidence intervals and p-values come from a 10,000-resample paired bootstrap over pages. "Seed consistency" counts the seeds whose subset-mean difference is positive.

Table 11: Difficulty decomposition: PRISM vs. Vanilla-SDPO Graph-F1 gain across quartile subsets (5 seeds). The full-set row is the 3-seed comparison of Table 3.

Subset	Defined on	Mean gain	95% CI	p	Seeds positive
All pages	–	$+3.81$pp	$[+1.95, +5.76]$	< 0.001	3 seeds
Lowest-reward quartile	SDPO outcome	$+7.93$pp	$[+4.10, +12.52]$	0.0001	5/5
Longest-output quartile (>1,500 tok)	SDPO outcome	$+5.43$pp	$[+1.21, +10.36]$	0.0104	5/5
Largest gold-graph quartile	ground truth only	$+4.12$pp	$[+0.15, +8.77]$	0.0405	5/5

Caveat: selection on the baseline's outcome. The first two subsets are selected using SDPO's own per-page outcomes. Pages on which SDPO happened to do poorly will, on average, look better under *any* re-evaluation (regression to the mean), so part of the $+7.93$pp and $+5.43$pp gains may reflect this selection effect rather than PRISM. The largest-gold-graph subset depends only on the ground truth and is free of this bias; its $+4.12$pp gain (p=0.04, 5/5 seeds) is therefore the cleanest evidence that PRISM's advantage grows with structural complexity. We report all three for completeness and base the main-text conclusion on their agreement rather than on the magnitude of the outcome-selected subsets.

B.4 Unified Leave-One-Out Ablation: Details

Protocol. Every variant in Table 5 starts from the seed-specific SFT-1000 checkpoint, uses the same fixed split, LoRA r=32, 100 post-SFT steps, and decoding settings, and is evaluated on the identical n=100 held-out pages. The 3-seed results are pooled as described in App. A.4. "w/o aux" sets α=0; "w/o RW" replaces the softmax weights by uniform weights $1/G$; Uniform SDPO additionally removes the DGR from the teacher context. Frozen and online teachers keep every other PRISM component and change only the teacher update rule (γ=1 and $\theta_T \equiv \theta_S$, respectively).

Table view of Figure 2(c). Removing the auxiliary loss changes node P/R by -1.74/-3.23pp and edge P/R by -2.03/-2.38pp. Removing reward weighting changes them by -0.18/-0.42 and -0.38/-0.36pp. Uniform SDPO changes them by -2.43/-3.00 and -2.69/-2.60pp. Recall drops more than precision when the auxiliary loss is removed, which is the expected footprint of truncation: a truncated graph omits objects and relations rather than hallucinating wrong ones.

Teacher statistics. Relative to Uniform SDPO, the frozen teacher gains $+0.0289$ (p=0.0041), the EMA teacher $+0.0284$ (p=0.0096), and the online teacher $+0.0228$. The frozen-vs.-EMA difference (0.0006) is not statistically resolved, and EMA exceeds online by 0.0055, which is directional only. The conclusion is that a separately parameterized teacher matters more than the specific update rule. This refines the Stage-5 exploratory result in Table 23, where a frozen teacher was much weaker than EMA under a 512-token cap and unmatched rank (App. D).

Relation to the factorial ablation. The single-seed $\pm$RW$\times\pm$aux factorial (Table 19) gives the same ordering in-distribution: removing the auxiliary loss is far more harmful than removing reward weighting. The unified ablation supersedes it as the main evidence because it uses 3 seeds and reports precision, recall, and truncation.

B.5 Reward-Signal Analysis: Rollout-level vs. Policy-level Discriminability

To empirically ground the *policy-level signal gap* characterized in §1, we examine the per-page reward distribution at n=100.

Table 12: Per-page Graph-Reward distribution over n=100 held-out pages. Both methods place $> 73\%$ of pages in the informative regime (0.55–0.85), yet the policy-level mean gap remains marginal: the reward discriminates rollouts well within a page, but standard scalar RL struggles to aggregate this into a policy-level advantage.

Reward Bin	SFT-r32	PRISM (sd=42)	Reward Bin	SFT-r32	PRISM (sd=42)
0.55–0.60	4	5	0.75–0.80	20	11
0.60–0.65	11	5	0.80–0.85	12	19
0.65–0.70	12	15	0.85–0.90	11	17
0.70–0.75	20	18	0.90–0.95	6	4
Overall SFT Mean: 0.738			**Overall PRISM Mean:** 0.736		

Reading. Most pages fall in the informative regime, which argues against a naive "reward is absent" diagnosis. Within each informative page the cross-rollout reward std is ≥ 0.15, while the between-method mean gap among the non-regressing methods (SFT, DPO, GRPO, SimPO, PRISM) is $\leq$ 1pp at matched rank. We therefore *empirically characterize* the regime as one of limited *policy-level signal granularity*: at this compute and rank budget, the scalar graph reward is a strong rollout-ranker but a weak policy-ranker. See §4.3 for which components matter. Vanilla-SDPO is a separate, regressing outlier: it loses pages from both the saturated and informative bins into the sparse bin, which is descriptively consistent with the short-completion / schema-collapse pathology.

B.6 Training-Dynamics Analysis: Train-Eval Dis-alignment and Short-Completion Bias

The n=100 snapshot describes the post-hoc eval distribution but does not speak to what each method does *during* training. Here we complement the snapshot with per-run reward and response-length trajectories over all 100 optimization steps (3 seeds per method, 2 seeds for Vanilla-SDPO, parsed from the training logs). This turns the "policy-level signal gap" characterization from a single-snapshot observation into a trajectory-level observation.

Table 13: Training-time reward trajectory (matched r=32, 100 steps, per-method seed average).

Method	Train-rew [1-50]	Train-rew [51-100]	$\Delta_{\text{early}\to\text{late}}$	Within-run rew std
DPO	0.7145	0.7413	$+0.027$	0.106
GRPO	0.7239	0.7469	$+0.023$	0.094
SimPO	0.7247	0.7458	$+0.021$	0.092
Vanilla-SDPO	0.7280	0.6884	-0.040	0.143
PRISM (Ours)	0.7290	0.7246	-0.004	0.141

Table 14: Response-length trajectory over training steps (evidence of short-completion bias). DPO, GRPO, and SimPO shrink sequences to avoid zero-reward parsing cliffs, whereas PRISM remains nearly flat.

Method	Len [1-50]	Len [51-100]	Δ_{len}	Relative Δ
DPO	1486.5	1381.0	-105	-7.1%
GRPO	1498.7	1385.4	-113	-7.6%
SimPO	1479.1	1382.4	-97	-6.5%
Vanilla-SDPO	1497.7	1578.8	$+81$	$+5.4\%$
PRISM (Ours)	1469.2	1456.5	-13	-0.9%

Four trajectory-level observations.

(i) Train-eval dis-alignment for non-regressing RL methods. DPO, GRPO, and SimPO all exhibit a $+0.02$–0.03 mean training-reward increase from the first-50 to second-50 window, yet on the

n=100 held-out eval their mean-reward gap to SFT is ≤ 0.015 in absolute terms. The reward *signal* is therefore present at training time, but the gain fails to transfer to held-out pages, which is the training-time signature we would expect of the policy-level signal gap.

(ii) PRISM is flat in training reward by design. PRISM's train-time reward is approximately flat (Δ=-0.004), which is mechanistically consistent with reward-weighted top-K JSD aggregation: the gradient is concentrated on the already-best rollouts, so the policy cannot "chase" additional training reward once the top-K is already high; instead the signal is absorbed into stabilization.

(iii) Short-completion bias is directly visible in length trajectories. DPO, GRPO, and SimPO all shrink response length by 7–8% over 100 steps, consistent with the prediction that scalar-reward RL on long structured JSON trades completeness for safer partial parses. Vanilla-SDPO shows the *opposite* pathology ($+5.4\%$), consistent with schema collapse producing repetitive low-reward text. PRISM is nearly flat (-0.9%): the length-normalized auxiliary preference loss on best-vs-worst rollouts suppresses both pathologies. The unified ablation (Table 5) confirms this causally: removing the auxiliary loss nearly doubles truncation.

(iv) Within-run rollout std ≥ 0.09 across methods. Within a single training run, rollout-to-rollout reward std is 0.09–0.14, mirroring the ≥ 0.15 within-page cross-rollout std observed at eval. The reward thus produces a usable per-rollout gradient during training; the limiting factor at our compute / rank budget is not gradient *availability* but the aggregation from rollout-level to policy-level.

Scope. We flag two caveats. First, we cannot causally distinguish between compute budget, LoRA rank, and exploration as co-contributors: a larger-scale run (e.g., r=64 or ≥ 1000 steps) could in principle close the eval gap. Second, these are single-run trajectories per (method, seed); cross-seed confidence intervals on the $\Delta_{\text{early}\to\text{late}}$ differences are wide. We report these values as trajectory-level evidence of the patterns visible in the n=100 snapshot, not as independent statistical tests.

C Deferred Experimental Results

This section provides full tables for deployment efficiency, cross-book OOD evaluation (including the controlled source $\times$ density probe), the per-type Graph-F1 decomposition, the single-seed factorial attribution, and the GraphDoc experiments.

C.1 Deployment Efficiency Analysis

Table 15 compares deployment efficiency across three tiers. Our 8B stack is deployable on a single 24GB GPU for single-batch inference, produces full parses in a single forward pass without test-time reasoning, has zero per-token API cost, and can be reproduced from released weights and training scripts. On the structural subset, only Gemini-2.5-Flash-Thinking scores higher, at the cost of API dependence, undisclosed size, and thinking-mode latency.

Table 15: **Deployment-efficiency comparison.** *Open-weight* means public weights allow local deployment; *TTR* = test-time reasoning / chain-of-thought is required at inference; *1-GPU* indicates the minimal GPU footprint for a single-batch inference at our context length (max_new_tokens=3072, bf16). *Structural subset* is the unweighted mean of the 8 Basic-Logic and Advanced-Spatial metrics of Table 2.

Model	Open-weight?	Size	TTR?	1-GPU inference	Per-token API cost	Structural subset (%)
Gemini-2.5-Flash-Thinking	No	undisclosed	Yes	—	paid	29.50
GPT-4o-mini	No	undisclosed	No	—	paid	3.69
Qwen2.5-VL-7B-Ins (best open)	Yes	7B	No	1× 24GB	free	8.93
Ovis2-8B	Yes	8B	No	1× 24GB	free	5.42
Gemma-3-12B / InternVL2.5/3-8B	Yes	8–12B	No	1× 24–48GB	free	1.75–3.34
Qwen3-VL-8B SFT (ours)	**Yes**	**8B**	**No**	**1× 24GB**	**free**	**21.63**
PRISM (ours, seed 42, step 100)	**Yes**	**8B**	**No**	**1× 24GB**	**free**	**21.50**

C.2 Cross-Book Out-Of-Distribution (OOD) Evaluation

We evaluate on a dedicated 100-page OOD set synthesized from *Xiyouji* (*Journey to the West*) and *Zizhi Tongjian*, two books outside the 64 training books. Table 16 reports the corrected 3-seed means. An earlier draft listed the seed-42 value in place of the mean for SFT and DPO and drew an incorrect

conclusion about OOD ordering. With correct averaging, **SFT has the highest mean** and no pairwise difference is significant. PRISM's distinguishing property under shift is stability: its cross-seed standard deviation is $14\times$ smaller than SFT's and $3.6\times$ smaller than DPO's.

Table 16: Cross-book OOD results on the dedicated 100-page set (combined book $\times$ layout-density shift). Each method is evaluated with 3 seeds $\{42, 7, 2024\}$; per-seed rewards: SFT $16.64/21.69/21.74$, DPO $16.75/17.30/18.19$, PRISM $18.48/18.08/18.31$. $\pm$ is the population standard deviation over seeds. No pairwise mean difference is significant ($p \geq 0.19$, paired bootstrap with 10,000 resamples on aligned 3-seed per-image means).

Method	**n**	**mean Graph Reward (3 seeds)**	**parse-rate**[‡]
SFT-r32-1000	100	$\mathbf{20.02}\% \pm 2.39$pp	55%
DPO (step 100)	100	$17.41\% \pm 0.59$pp	52%
PRISM (step 100)	100	$18.29\% \pm \mathbf{0.16}$pp	58%

[‡]Parse rates as logged in the original OOD evaluation (seed 42 for SFT and DPO).

C.3 Controlled Source $\times$ Layout-Density Factorial Probe

Because the OOD set above mixes a book (source) shift with a layout-density shift, we separate the two with a controlled synthetic factorial probe. The 100 OOD pages are divided into four matched cells of 25 pages: *Xiyouji* or *Zizhi Tongjian*, each at low or high layout density. Table 17 reports pooled strict F1 (App. A.4) per cell.

Table 17: Pooled strict F1 by factorial cell (25 pages per cell). All methods collapse to near zero under this unseen-source reading-order shift.

	Xiyouji		**Zizhi Tongjian**	
Method	**Low density**	**High density**	**Low density**	**High density**
SFT	0.0024	0.0055	0.0054	0.0045
Vanilla-SDPO	0.0047	0.0042	0.0033	0.0050
PRISM	0.0035	0.0053	0.0067	0.0023

Reading. Every method falls to a pooled strict F1 of at most 0.0067, and for no method is a source main effect, a density main effect, or a source $\times$ density interaction statistically resolved. The overall PRISM$-$SFT paired gap is -0.0006 (95% CI $[-0.0030, +0.0011]$, crossing zero). The defensible conclusion is a *shared* zero-shot topological transfer failure across the methods we evaluate, not a PRISM-specific failure, and not evidence that PRISM improves OOD generalization. In particular, the page-averaged Graph Reward of $\sim$18–20% in Table 16 comes mostly from easy node classes, which the pooled strict metric does not reward on otherwise failed pages. Closing this gap, for example with OOD-aware layout curricula, is left to future work.

C.4 Per-Type Graph-F1 Breakdown

The aggregate number hides where the gain appears. Table 18 decomposes the evaluation by 5 node types and 3 primary edge types. PRISM's largest gains over SFT concentrate on edge categories (`READS_AFTER` $+3.8$pp, `OVERLAPS` $+2.9$pp) rather than content extraction, i.e., on topological and relational structure.

C.5 Component-Level Factorial Attribution (Single Seed)

We run a $\pm$RW $\times$ $\pm$aux factorial ablation at matched $r{=}32$ compute with seed 42 (Table 19). In-distribution, **aux** is the principal lever, preventing the policy from collapsing into short outputs, which agrees with the 3-seed unified ablation (Table 5). On a 50-page OOD subset, adding **RW** to aux-only raises the score by 6.2pp. Given a single seed and the shared OOD transfer failure documented above, we treat this OOD effect as suggestive only.

Table 18: **Per-type Graph-F1 breakdown** on n=100 held-out pages. PRISM's gains concentrate on the edge categories (R, O), which are the structural / topological relations rather than content-text extraction. Vanilla-SDPO's regression is largest on IA and R.

Type	SFT F1	PRISM F1	Δ(PRISM−SFT)	Vanilla-SDPO F1
node:CF (center-fold)	100.0%	100.0%	+0.0pp	100.0%
node:RS (red-seal)	98.5%	98.8%	+0.3pp	98.4%
node:TM (top-marginalia)	92.8%	92.9%	+0.1pp	92.4%
node:MT (main-text)	86.1%	**87.9%**	**+1.7pp**	86.1%
node:IA (interlinear-annotation)	81.7%	81.4%	−0.3pp	78.4%
edge:R (READS_AFTER)	57.2%	**61.0%**	**+3.8pp**	56.2%
edge:O (OVERLAPS)	46.8%	**49.6%**	**+2.9pp**	46.1%
edge:A (ANNOTATES)	41.8%	40.6%	−1.2pp	40.6%

Table 19: **Factorial ablation ±RW × ±aux** at matched r=32 and matched compute (single seed, sd=42). Adding *aux* to rw-only gains +3.6pp in-distribution; adding *RW* to aux-only gains +6.2pp on the OOD subset.

Variant	RW	aux	DGR	In-distr n=100	OOD n=50	Δ OOD vs. full PRISM
Vanilla-SDPO	×	×	×	70.15%	—	—
aux-only	×	✓	✓	73.09%	14.23%	−6.2pp
rw-only	✓	×	✓	69.98%	13.56%	−6.8pp
PRISM (full)	✓	✓	✓	**73.59%**†	**20.40%**	**0**

†Single-seed value for sd=42; the 3-seed PRISM mean reported elsewhere is 73.96% ± 0.27pp. Vanilla-SDPO is the 3-seed mean of Table 3.

C.6 GraphDoc Experiments: Retraining and Zero-Shot Protocols

Why GraphDoc. GraphDoc [Chen et al., 2025] is an independently developed, human-verified document-graph benchmark that differs from OracleGraph in language, document domain (modern documents), layout regime, annotation source, and graph ontology. It therefore tests whether PRISM depends on OracleGraph's procedural construction or ontology.

Retraining protocol. We keep GraphDoc's official images, splits, node and relation definitions, semantic ontology, and evaluator, and convert its annotations into our serialized graph format without changing their semantics. The DGR is generated from GraphDoc gold graphs by the same report generator, and the Graph-F1 reward is computed over GraphDoc's node and relation classes. GraphDoc SFT-1000 and PRISM share the base model (Qwen3-VL-8B), training data, serialization, decoding, evaluation protocol, and three seeds; for each seed, PRISM starts from the corresponding SFT checkpoint and receives 20 post-training steps with all four components enabled. This is task-specific retraining, not zero-shot transfer.

Table 20: Full PRISM on GraphDoc (retraining, 3-seed means, official evaluator).

Metric	GraphDoc SFT-1000	GraphDoc PRISM (20 steps)	Gain
Node F1	60.44%	**64.20%**	+3.76pp
Edge F1	30.11%	**32.78%**	+2.67pp
Graph F1	38.21%	**41.16%**	+2.95pp
mR_g	14.31%	**16.20%**	+1.89pp

Zero-shot protocol. We evaluate the OracleGraph-trained SFT and PRISM checkpoints directly on GraphDoc images, with no weight updates, no tuning on GraphDoc labels, and no GraphDoc-based checkpoint selection. Because the full ontologies differ, we evaluate only the shared text-region ontology: OracleGraph's text-bearing node types and GraphDoc's textual region categories are both mapped to a single text-region class, with the mapping fixed before inspecting any prediction. Regions are matched with the one-to-one IoU rule of App. A.4. Complexity is defined solely by the number of gold text regions; pages above the median (12) form the high-complexity subset (n=23).

Table 21: Zero-shot OracleGraph→GraphDoc on high-complexity pages (n=23). "Parsed" = successful structured decoding; "Truncated" = termination at the generation limit before the structure is closed.

Metric	OracleGraph-SFT	OracleGraph-PRISM	Gain
Text precision	11.26%	**13.76%**	+2.50pp
Text recall	22.35%	**25.41%**	+3.06pp
Text F1	14.97%	**17.85%**	+2.88pp
Parsed outputs	26.1%	**39.1%**	+13.0pp
Truncated outputs (↓)	47.8%	**39.1%**	−8.7pp
Predicted regions / page	36.70	34.13	−2.57

Reading. PRISM improves precision and recall while predicting *fewer* regions, so its gain is not explained by indiscriminately generating more output; its higher parse rate and lower truncation rate indicate better structural completeness on complex pages. Absolute performance remains low under the joint language, visual-domain, serialization, and ontology shift. We therefore interpret Table 21 as an ontology-aligned stress test, not as evidence of universal or full-ontology generalization. Taken together with the retraining result, the evidence supports a scoped claim: PRISM is a stabilization recipe for structured generation that applies wherever gold-derived feedback or a programmatic verifier is available, and its full configuration is not limited to OracleGraph's ontology. We do not claim universal applicability or superiority over SFT.

D Additional Ablations and Training Dynamics (Stage-5 Protocol)

D.1 Compute Footprint and Reproducibility Card

All experiments use NVIDIA H20 (96 GB HBM) GPUs. A single RL training run occupies *two* H20s: GPU 0 hosts the student-teacher adapter pair under bf16 and performs the on-policy rollout (g=2, $\max$ _new_tokens=3072); GPU 1 runs the training forward/backward pass on the same adapter in parallel, using NCCL point-to-point communication to shard activations between the two stages. SFT runs are lighter (no rollout stage, no teacher replay): a single H20 reaches ∼2–3 s/step on the 5k-page OracleGraph corpus, so the r=32 SFT-1000 checkpoint (∼3,000 steps including warmup) completes in $\lesssim$ 2 GPU-hours.

Table 22: **Compute Footprint and Hardware Utilization.** Evaluated on NVIDIA H20 (96 GB) GPUs. RL training uses 2× H20 (rollout + train), taking ∼3h wall-clock per 100-step run (≈ 6 GPU-hours). SFT warmup takes ∼1-2 GPU-hours. Evaluation is sharded across 8 GPUs per run for n=100 (wall-clock ≈ 20 min, compute ≈ 3 GPU-hours). OOD re-evaluation batches 4 runs concurrently across 4 nodes.

Task / Bucket	Runs	GPU/run	Wall-clock/run	Total GPU-hours
Matched-rank RL (Table 3): 6 methods × 3 seeds	18	2	3h	∼108
Factorial ±RW×±aux (sd 42, Table 19)	2	2	3h	∼12
DGR verbosity ablation (6 variants × 3 seeds)	18	2	3h	∼108
SFT warmup (r=32, 3 seeds, 1,000 steps)	3	1	$\lesssim$2h	∼6
Evaluation: n=100 val, 8-shard, matched-rank set	10	8	20min	∼27
Evaluation: n=100 book-disjoint OOD, 8-shard, 4 runs	4	8	20min	∼11
Total compute footprint of the submitted experiments				**∼270 GPU-hours**

Concretely, the matched-rank headline set in Table 3 plus its factorial and DGR-verbosity attributions costs ∼230 H20-hours of training; re-evaluating all checkpoints on in-distribution and book-disjoint splits adds ∼40 H20-hours. The experiments added for the camera-ready version (unified ablation, difficulty decomposition, OOD factorial probe, GraphDoc) are not included in this total. Wall-clock throughput is bottlenecked by the student rollout phase ($\max$ _new_tokens=3072 with bf16), not by the student training update; optimization of the rollout engine (e.g., a dedicated SGLang inference server) is a natural scaling lever we leave to reproducers.

Disclaimer: The following ablations were conducted under an earlier *Stage-5 preliminary protocol* (n=30 pages, unmatched LoRA rank, 500-step SFT init, more aggressive learning rates, and a

512-token generation bottleneck). They isolate qualitative design choices (teacher update rule, group size, clip presence/absence, capacity) and are independent of the n=100 / 3-seed / matched-rank protocol used for the headline results in §4.2. Absolute reward numbers in these tables are therefore not directly comparable to the headline 73.96% PRISM number. The *final* PRISM configuration used throughout §4.2 is EMA γ=0.999, importance clip ϵ=0.1, max_new_tokens=3072, LoRA r=32, and group size g=2.

D.2 Stage-5 Mechanistic Ablations

(1) Teacher update rule (Table 23). Under Stage-5, the `online` control (student teaching itself) collapses below the untrained base model (0.061), and EMA (γ=0.995 in this regime) scores far above `frozen` and `trust-region`. Under the matched protocol (Table 5), the frozen and EMA teachers are statistically indistinguishable; the large Stage-5 gap between them is therefore a property of this unmatched, 512-token-capped regime and should not be read as evidence that a frozen teacher is inherently worse. The robust conclusion shared by both protocols is that removing the separate teacher (online) hurts.

Table 23: Stage-5 teacher-update mechanisms (exploratory protocol; untrained base model scores 0.130). Superseded quantitatively by the matched teacher ablation in Table 5.

Teacher Mechanism	Graph Reward ↑	Δ vs. Base
`online` (*teacher* ≡ *student*)	0.061	*-0.069*
`frozen` (Fixed at step 0)	0.228	*+0.098*
`trust-region`	0.265	*+0.135*
`EMA` **(Ours, $\gamma = 0.995$ in Stage-5)**	**0.429**	***+0.299***

(2) Topological recovery (Table 24). Decomposing Stage-5 Graph-F1 by class shows that, in this regime, DPO and GRPO fail on dense relational topologies (Main Text `MT` boxes and `READS_AFTER` edges), while SDPO-based training recovers them partially, trailing the SFT reference. The matched-protocol per-type breakdown is Table 18.

Table 24: Stage-5 fine-grained per-class F1 (exploratory protocol). Standard scalar RL (DPO/GRPO) collapses on Main Text (`MT`) and Reads-After (`R`) in this regime.

Method	Node Types (F1)					Edge Types (F1)		
	MT	**IA**	**TM**	**CF**	**RS**	**R**	**A**	**O**
Untrained Base	0%	10%	40%	85%	24%	0%	0%	0%
DPO (Baseline)	0%	0%	0%	0%	0%	0%	0%	0%
GRPO (Baseline)	0%	3%	36%	100%	13%	0%	0%	0%
SDPO (Ours, Stage-5)	**72%**	**12%**	**75%**	**100%**	**33%**	**33%**	**2%**	**3%**
SFT (Upper Ref.)	*84%*	*74%*	*81%*	*100%*	*93%*	*52%*	*10%*	*22%*

(3) Optimization stability (Table 25). In the Stage-5 scan, a rollout group size of g=2 and an importance clip of ϵ=0.2 performed best; removing within-group contrast (g=1) or clipping (ϵ=0) degraded reward. The final matched configuration uses ϵ=0.1.

Table 25: Stage-5 RL optimization stability ablation. Within-group advantages (g=2) and importance clipping are both important in this regime.

Hyperparameter	Configuration Variant	Graph Reward ↑
Group Size (g)	$g = 1$ (No in-group contrast)	0.256
	$g = 3$ (High optimization noise)	0.306
	$g = 2$ **(Ours, best in scan)**	**0.429**
Importance Clip (ϵ)	$\epsilon = 0$ (No clipping)	0.249
	$\epsilon = 0.2$ **(Stage-5 setting)**	**0.429**

(4) Capacity vs. algorithm (Table 26). Increasing LoRA rank (r=32, 64) for SDPO on a merged SFT base did not help in this regime (Graph Reward 19.4% and 14.9%), while PRISM at r=16 reached 71.8%, indicating that the gain does not come from adapter capacity.

Table 26: Stage-5 algorithm vs. parameter-capacity attribution on a merged SFT base.

Configuration (merged base, Stage-5)	Graph Reward ↑	CER ↓	BLEU ↑
PRISM ($r = 16$ on merged base)	**71.8%**	**0.084**	**0.892**
SDPO ($r = 32$ on merged base)	19.4%	0.353	0.643
SDPO ($r = 64$ on merged base)	14.9%	0.319	0.663

D.3 Convergence and Structural Alignment Stability

As illustrated in Figure 8(a) and (b), the **top-K JSD loss** during the SDPO main run exhibits a smooth, monotonic decrease, dropping by approximately 77% over 200 steps. This indicates that the Student policy consistently aligns its token-level distribution with the gold-conditioned EMA-Teacher.

Interestingly, the **Graph-similarity Reward** peaks around step 125 before exhibiting a "reward collapse." Analysis of the *Average Response Length* (Figure 8(d)) reveals that this is not due to policy divergence, but rather a **Length-Constraint Bottleneck**. As the model internalizes the Teacher's highly detailed graph schema, its outputs systematically grow in length, eventually saturating at the 512-token truncation limit set during this stage-5 regime. This truncation causes the JSON structure to break, resulting in a zero reward. This observation motivated the 3,072-token limit of the matched protocol.

D.4 Stage-5 Exploratory Stability: SDPO vs. DPO and GRPO (Unmatched Protocol)

Under the early stage-5 regime (lighter LoRA rank, 500-step SFT init, more aggressive lr schedule) and initialized from the same SFT checkpoint, Figure 8(c) shows that DPO and GRPO exhibit larger reward fluctuations than SDPO and sometimes drop to lower structural fidelity, while SDPO maintains a smoother reward curve under this regime. **We do *not* interpret this as a general claim that scalar-reward RL is fundamentally unstable**: the matched-r=32, 3-seed comparison in the main text (§4.2, Table 3) instead shows that DPO, GRPO, and SimPO all cluster within $\pm$1.5pp of the SFT baseline and that Vanilla-SDPO actually *regresses* by -2.8pp. The dense asymmetric-distillation signal helps only after adding the DGR + reward-weighted + length-normalized components (i.e., full PRISM). We retain the stage-5 plot here for training-curve completeness and defer to the main-text matched-rank tables for any quantitative claim.

D.5 Ablation on DGR Feedback Verbosity (Training Dynamics)

Figures 8(e) and (f) explore the impact of **Diagnostic Graph Report (DGR)** verbosity on the Teacher's guidance. We find that:

- **DGR-short (0 items):** Provides a stable but slow learning signal due to the lack of specific error-correction hints.
- **DGR-medium (10 items, Default):** Offers the best balance, providing enough structural context to accelerate convergence without overwhelming the Teacher's context window.
- **DGR-long (30 items):** While providing the richest feedback, it introduces slight noise and variance in the early stages of training, suggesting a diminishing return on feedback length.

D.6 Exploratory DGR Verbosity Scan under the Stage-5 Regime

An earlier exploratory scan of DGR verbosity was run under the lighter Stage-5 regime prior to the matched-r=32 protocol used in the main text. The rewards observed under this regime are therefore on a different scale and *not directly comparable* to Table 6; we include Table 27 here only as a qualitative consistency check.

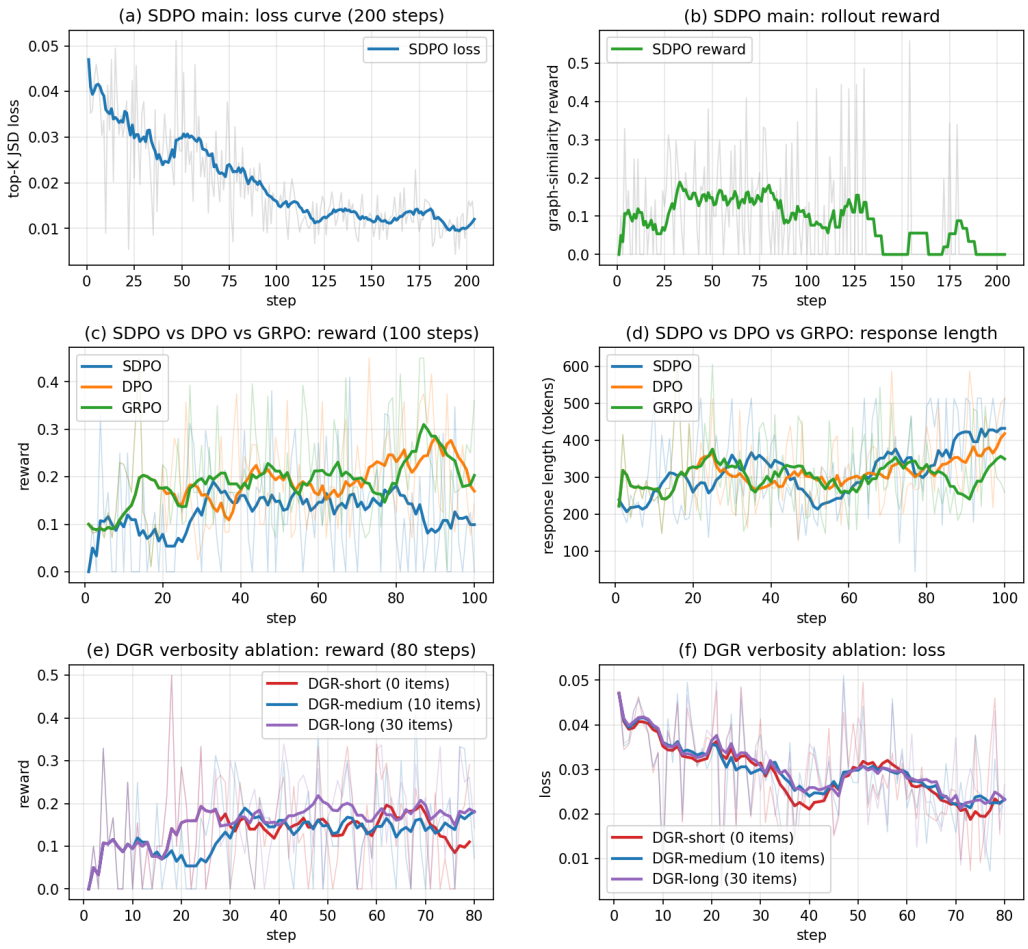


Figure 8: **Stage-5 Exploratory Training and Ablation Analysis.** *All panels are from the stage-5 exploratory regime (unmatched LoRA rank, 500-step SFT init, 512-token gen cap); absolute values are not directly comparable to the matched-r=32 main-text tables. Retained here for training-dynamics intuition only.* **(a-b)** Main-run stability showing JSD loss reduction and reward trends. **(c-d)** Benchmarking SDPO against DPO/GRPO baselines under the stage-5 protocol; in the *matched* 3-seed comparison in §4.2 the three methods are much closer, and Vanilla-SDPO regresses rather than leads. **(e-f)** Impact of DGR feedback granularity on alignment efficiency.

Table 27: *Exploratory / stage-5 regime only.* Configuration and Performance of Diagnostic Graph Report (DGR) Levels prior to the matched-r=32 protocol. In this regime, non-gold variants (`L0`-`L3`) score well below gold-conditioned variants and `auto` scores above fixed `short`/`long`. Under the matched protocol (Table 6) all DGR variants are close in mean and `auto` is distinguished mainly by its lower variance, so only the qualitative benefit of DGR over no-DGR carries over.

Level	Feedback Components Included						Graph Reward[§]
	Pass/Fail	Macro Stats	Per-Type Stats	Diff Items	Gold Ref.	Examples	
`L0`	✓	×	×	×	×	×	0.189
`L1`	✓	✓	×	×	×	×	0.173
`L2`	✓	✓	✓	×	×	×	0.147
`L3`	✓	✓	✓	✓	×	×	0.172
`short`	×	×	✓	✓	✓	×	0.304
`long`	×	×	✓	✓	✓	✓	0.230
`auto` **(Ours)**	×	×	✓	✓	✓	*Dynamic*	**0.429**

[§]Single-seed, stage-5 regime; absolute values not comparable to Table 6.