VLM-SlideEval: Evaluating VLMs on Structured Comprehension and Perturbation Sensitivity in PPT

Anonymous Author(s)
Affiliation
Address
email

Abstract

Vision-language models (VLMs) are increasingly used to evaluate multimodal content, including presentation slides, yet their slide-specific understanding remains underexplored *despite their growing role as critics in agentic, model-forward pipelines*. We introduce **VLM-SlideEval**, an evaluation framework that probes VLMs along three axes: (1) element-level extraction from slide images aligned to ground truth; (2) robustness to controlled perturbations in geometry, style, and text; and (3) higher-level comprehension, such as recovering a deck's narrative order from shuffled slides. Using publicly available decks from Zenodo[1], we standardize ground-truth element metadata from PowerPoint XML and live renderings into a unified, verifiable schema. Empirically, VLMs underperform on pixel-accurate extraction and show non-trivial agreement, fidelity and consistency under controlled perturbations, while performing better on single-slide content understanding; however, they do not reliably capture narrative structure across slides. These results highlight the limits of current VLMs for slide evaluation and motivate calibrated, critic-in-the-loop evaluators that drive iterative refinement and selection in agentic pipelines.

1 Introduction

Presentation slides are a primary vehicle for conveying structured ideas across domains ranging from education to scientific communication to corporate decision-making. Automatic evaluation of slide quality and content understanding is an emerging and pronounced need, particularly in light of advances in *agentic, model-forward* slide generation Ge et al. [2025a], Fu et al. [2022]. While prior work on document analysis has focused on optical character recognition (OCR) [Xu et al., 2021, Wang et al., 2024, Smith, 2007] and XML-based parsing [Canny, 2025], these approaches are brittle when slides are only available as rendered images, and are limited to low-level layout information without reasoning about higher-level semantics. In contrast, vision-language models (VLMs) promise a unified mechanism for parsing slide content directly from images while also supporting tasks that require semantic or narrative comprehension.

Despite the promise, it remains unclear to what extent current VLMs truly comprehend presentation slides. On one hand, VLMs may struggle with precise pixel-level tasks such as identifying bounding boxes, font attributes, or alignment, since they may not have been directly trained on raw presentation rendering pipelines or large scale OCR data of slide presentations. On the other hand, VLMs may excel at higher-level understanding, such as identifying the role of slide elements (*e.g.,* title, subtitle, body text), inferring content hierarchy, or reasoning over narrative flow across a deck. Understanding these trade-offs is crucial for designing reliable and scalable evaluation pipelines that utilize VLMs.

We introduce **VLM-SlideEval** as a first-class *critic* in agentic, model-forward pipelines and systematically probe VLM slide comprehension. Our contributions are threefold. First, we curate a diverse dataset of PowerPoint decks and extract ground-truth geometry, style, and text via a pipeline combining PowerPoint XML with rasterized renders. Second, we design protocols for low-level

[1] `https://zenodo.org`; HF viewer: `https://huggingface.co/datasets/Forceless/Zenodo10K/viewer/default/pptx`

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

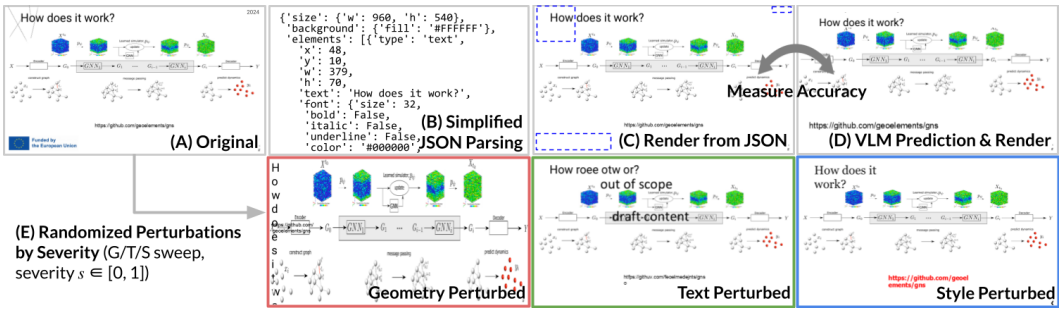


Figure 1: **Evaluation Task Examples:** Top: From an original slide (A), we parse a simplified schema JSON (Table 1) (B), reconstruct a normalized slide (C; blue dashed boxes show theme-embedded content omitted by the schema). A VLM predicts the schema from the re-rendered slide (D), and we score accuracy. Bottom: We subsample 100 decks, retain slides with $\geq$3 visible elements (234 slides total), and apply perturbations to geometry, text, and style with severity $s \in [0, 1]$ (larger s means stronger changes; details in §3). Perturbed slides are then used for VLM quality evaluation and sensitivity analyses (§4).

fidelity and structured comprehension, including element-wise Hungarian alignment and refinement-relevant probes of judge reliability (variance, sensitivity) and robustness via controlled perturbations to geometry, style, and text. Third, we extend evaluation to deck-level narrative by asking VLMs to reorder shuffled slides, assessing coherence.

Applying VLM-SlideEval, we surface clear limits and strengths. VLMs struggle with pixel-accurate extraction and show behavioral divergence under controlled perturbations, yet they competently extract structured content on single slides while remaining unreliable for deck-level narrative. These findings caution against over-reliance on current VLMs for fine-grained slide evaluation and motivate more calibrated critic-in-the-loop refinement and selection gates for agentic pipelines.

2 Related Work

Calibrated VLM evaluators are increasingly *critical* in agentic, model-forward pipelines: they guide candidate selection, drive iterative refinement at inference time, and even supply reward/preference signals for training. Recent work shows verifier-guided decoding that improves performance without weight updates Chakraborty et al. [2025], generalist multimodal judges used both as LMM-as-a-Judge and as reward models Xiong et al. [2025], actor-critic loops that critique and correct reasoning Liu et al. [2025], and refinement-centric benchmarks plus standardization frameworks that emphasize granular, non-saturated measurement Paik et al. [2025], Balachandran et al. [2024]. This motivates a slide-native, *verifiable* evaluator that produces actionable signals at pixel, element, and deck levels.

Yet VLM evaluation remains challenging. Open-ended judging often relies on incomplete visual context and fuzzy rubrics, yielding inconsistent scores Prabhu et al. [2024], while models hallucinate and make perceptual errors in visually grounded reasoning Ma et al. [2024]. Under *controlled manipulations* and counterfactuals, VLMs may inject priors unsupported by pixels and show limited sensitivity to fine-grained changes Guan et al. [2024], Vo et al. [2025]. Robustness studies further find text corruptions especially damaging, lightweight adapters sometimes rivaling full fine-tuning, and broader axes (fairness, toxicity, multilinguality) underexplored Chen et al. [2023], Lee et al. [2024].

Slide presentations sit within multimodal document understanding, where *structured parsing* underpins both comprehension and authoring. Prior work has explored language-driven manipulation of slide *objects* (not pixels) for faster, faithful editing Jung et al. [2025], OCR-free pretraining for screenshots and UI/text layouts that improves element-level parsing Lee et al. [2023], and automatic extraction of deck structure for role identification and accessibility Peng et al. [2023]. In parallel, systems that generate slides from long-form documents highlight the need for *scalable, slide-specific evaluation* under diverse styles and limited metadata Fu et al. [2022], Ge et al. [2025b].

Unlike work that omits a slide-native evaluator, relies on QA proxies, or focuses robustness on charts/UIs, *VLM-SlideEval* provides a slide-specific framework that couples pixel-accurate alignment to PPT-native ground truth with slide-relevant perturbations and deck ordering, *positioning the evaluator as a critic for agentic pipelines*.

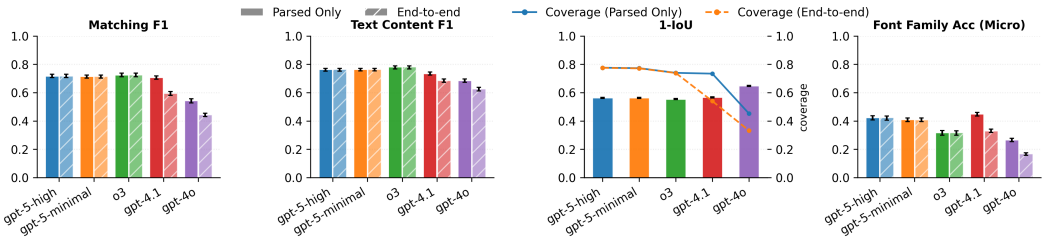


Figure 2: Parsed-only (solid) vs. e2e (hatched) with coverage (*i.e.,* fraction of ground truth instances evaluated for the metric; lines). `o3`/`gpt-5` lead on Matching F1 (0.71–0.72) and Text Content F1 (0.76–0.78); `o3` best in geometry (1–IoU 0.55). Font Family Accuracy is low overall (max 0.42). More results in Fig. 7, Appendix F.

3 Method

Data Source. We sample 100 English-dominant ($\geq 70\%$ by `langid`) `.pptx` decks from Zenodo10K (legacy `.ppt` excluded), totaling 1,948 slides, with CC-BY 4.0 license. Summary statistics are in Appendix A, Table 2.

Ground Truth Element *geometry*, *content*, and *style* are extracted from PowerPoint XML and post-layout rendering. We parse static XML and then query the COM API after a layout pass to recover effective font metrics and tight text bounds (mitigating AutoFit and container/tight-box discrepancies). Elements are stored in a standardized schema with explicit units (Appendix A, Table 1).

VLM Parsing & GT Matching. Slides are rasterized to PNG and sent with a fixed 960×540 coordinate frame; we test five VLMs (via Azure) to return JSON validated against our schema (invalid JSON counts as a parse failure). Each slide is run $N{=}3$ times (low temperature), and metrics are reported per-run and pooled. Predictions are aligned to GT via Hungarian matching (*cf.* [Kuhn, 1955, Stewart et al., 2016, Carion et al., 2020, Dong et al., 2025, Wang et al., 2025]) with a blended cost (IoU, center/size; text adds content similarity) and an acceptance gate; details in Appendix C.

Perturbation Synthesis. *Seeds.* From the same 100 decks we manually select slides well-preserved by the schema and with at least a minimal complexity, ≥ 3 visible text elements, yielding 234 seeds; the reconstructed slide serves as the clean baseline. *Severity knobs.* We generate controlled degradations along *geometry*, *text*, and *style*, parameterized by a single severity $s \in \{0, 0.1, \ldots, 1.0\}$. Magnitudes (*e.g.,* pixel offsets, font-size factors) and event probabilities (*e.g.,* drop/insert text boxes) increase monotonically with s; randomness is seeded per (slide, axis, s). Exports use a Node.js-based PPTX builder and headless rendering. From the 7,722 original+perturbed slides in total (hyperparameters in App. D), we subsample up to 50 slides per severity per axis for evaluation.

Manipulation Check. We assess whether increasing severity $s \in [0, 1]$ yields orderly and proportional degradation using (i) *adjacent POA* - the fraction of consecutive severity steps where y^* does not decrease - and (ii) the *mean absolute calibration error* (MACE) to the identity $y^* = s$, on the normalized $[0, 1]$ scale. Empirically, POA is high (5-pt ≈ 0.95; 100-pt ≈ 0.80) with moderate calibration (overall MACE ≈ 0.34).

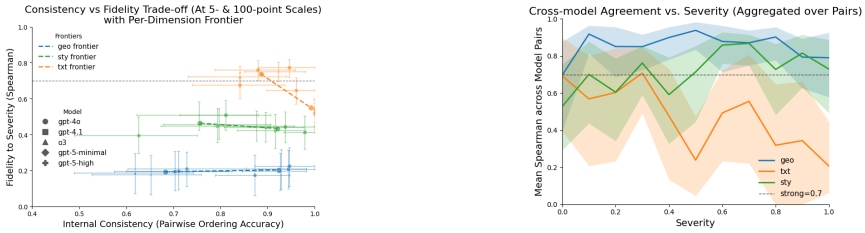
Analysis & Measures We evaluate: (i) *parseability* (schema-valid JSON rate); (ii) *extraction quality* on matched elements (geometry, content, style); (iii) *narrative ordering* (deck reordering; Kendall's τ, Spearman's ρ); and (iv) *perturbation sensitivity* - R^2, POA and $\mathrm{Spearman}(\text{severity}, y^*)$ - comparing different evaluator scales and models. We report bootstrap 95% CIs and isotonic summaries where appropriate. Full metric definitions and evaluator prompts appear in Appendix E.

4 Results

We benchmark five VLMs (Azure API) three main tasks: 1) element-level extraction from single slides, 2) robustness to controlled perturbations, and 3) narrative understanding via slide re-ordering.

Slide Parseability. Parse success declines with slide complexity for `gpt-4.1` (about 93% for simple slides with ≤ 8 elements, 72.1% for (8-16], 32.8% for (16-32], and 18.2% for ≥ 32 elements). `gpt-4o` follows a similar trend but with an earlier decline: about 88.0% for ≤ 8, 57.6% for (8-16], 45.8% for (16-32], with a small (noisy) uptick to 66.7% at ≥ 32 ($N{=}66$). In contrast, `o3` and the `gpt-5` variants remain effectively at ceiling across all bins ($99.5\%+$). See Fig. 6.

Element Prediction Accuracy. Across headline metrics (Fig. 2), `o3` and the `gpt-5` variants lead under e2e. *Matching F1*: Parsed→e2e performance drops ($\Delta \approx 0.12$ for `gpt-4.1` and `gpt-4o`), with `o3` achieving the highest e2e F1 score (0.72), followed by `gpt-5` 0.71–0.72, vs. `gpt-4.1`



(a) **Consistency-fidelity frontier** per dimension. Consistency := POA_{adj} and fidelity := Spearman (severity, y^*) (higher better). Geometry/style show no fidelity gain but lower consistency when moving from 5 to 100-pt scale; text trades fidelity with consistency.

(b) **Cross-model agreement vs. severity**. Spearman agreement across model pairs by severity buckets. Geometry/style pairs often exceed 0.80-0.90; text is lowest (best text pair $\overline{\rho} \approx 0.55$), indicating limited interchangeability on text.

Figure 3: **Evaluation results of model behavior under controlled perturbations.**

0.59 and `gpt-4o` 0.44. *Text Content F1*: `o3` 0.78 (best), `gpt-5` 0.76, `gpt-4.1`/`gpt-4o` 0.69/0.63. *Geometry (1–IoU; lower better)*: `o3` 0.55 (best), `gpt-5` 0.56, `gpt-4.1` 0.57, `gpt-4o` 0.65 (worst). E2e coverage is limited, especially for `gpt-4o` (0.33) and `gpt-4.1` (0.54) vs the rest (0.74-0.78) *Styling (Font Family Acc.)*: overall low (0.17–0.42), with `gpt-5-high` highest (0.42) and `gpt-4o` lowest (0.17). Detailed metrics and parsed-only comparisons appear in Table 4 and App. Fig. 7.

Behavior Under Controlled Perturbations - Scale correspondence. Within each model, an *isotonic link* maps 5-point scores to 100-point scores with high fidelity: $R^2 \in [0.85, 0.89]$ across models ($p = 0.001$), with GPT-4.1 the tightest ($\text{RMSE} = 0.075$) and others close (*e.g.,* GPT-5-high 0.083), on the normalized degradation scale $y^* \in [0, 1]$. This establishes that the two *scales* are largely monotone reparameterizations. *However*, a monotone mapping does not imply identical behavior under controlled severity shifts: coarse 5-point scores reduce quantization jitter and often improve within-slide ordering, whereas 100-point scores expose finer variation that can either reflect genuine sensitivity or add noise. We therefore examine explicit *scale*×*dimension* trade-offs below.

Scale×**dimension trade-offs.** We quantify *internal consistency* as POA_{adj} and *fidelity* as $\text{Spearman}(\text{severity}, y^*)$. We find that for **geometry** and **style**, moving from 5-pt to 100-pt yields *no material fidelity gain* (bootstrap CIs overlap across models) but *reduces POA*, as implied by the flat frontiers (*e.g.,* geometry POA drops from 0.87–0.95 to 0.62–0.73; style from 0.88–0.98 to 0.63–0.81) (Fig. 3a). Thus a coarser scale is preferable for stability in these dimensions. In contrast, for **text**, 100-pt *increases fidelity substantially* (*e.g.,* GPT-5-high $0.51 \rightarrow 0.75$; GPT-5-minimal $0.52 \rightarrow 0.76$) while *lowering POA* (often $1.00 \rightarrow 0.88$–0.92), revealing a consistency–fidelity trade-off.

Model interchangeability. Models *diverge most on text* (Fig. 3b). Even the most convergent text pair (GPT-5-high vs. GPT-5-minimal) attains only $\overline{\rho} \approx 0.55$ (mean of per-severity Spearman), whereas geometry/style pairs frequently exceed 0.80–0.90. Notably, the most divergent geometry pair (*e.g.,* GPT-4o vs. o3) still shows higher agreement ($\overline{\rho} \approx 0.78$) than the most convergent text pair, underscoring that *text quality* is the hardest axis for cross-model agreement.

Narrative in Slide Deck. Overall (Figure 8), the models exhibit difficulty in accurately predicting slide order, with Kendall's τ (0.04-0.12), Spearman's ρ (0.05-0.13), and Exact Match scores (0.10-0.17) only marginally outperforming random guessing, yet remaining below the theoretical upper bound of 1.0. This suggests that the models may struggle to comprehend and reason through the narrative flow of a presentation. Among them, *gpt-4.1* delivered the strongest performance (0.04-0.07 point of improvement) over *gpt-5* with minimal reasoning (Details in Appendix F.3).

5 Conclusion

We present **VLM-SlideEval**, a framework for evaluating slide element extraction, robustness to controlled perturbations, and narrative reordering on a curated PPTX corpus with ground truth. Newer VLMs (*o3*, *gpt-5*) outperform *gpt-4.1*/*gpt-4o*, yet all struggle with pixel-accurate style (*e.g.,* fonts) and cross-slide narrative coherence, and under perturbations exhibit a fidelity-consistency trade-off: geometry/style are comparatively stable, while finer text scales raise sensitivity but reduce internal score consistency. These findings argue for calibrated, slide-native evaluators as first-class critics in agentic/model-forward pipelines, using verifiable signals to gate selection and steer iterative refinement. Limitations include public PPTX, seeded perturbations, and our pipeline; future work spans broader corpora, richer narrative probes, stronger verifiable checks, and judge calibration.

A Ground Truth Extraction Details

Ground truth elements are obtained by parsing the PowerPoint XML specification and cross-checking against a PNG export of the same slides. Each element type (text, rect, line, image, table) is represented in a unified schema with pixel-based geometry and absolute units for fonts and strokes (the full extraction schema is shown in Table 1 below).

Field(s)	Applies to	Unit / Notes
`w`, `h`	slide	px; fixed at 960×540
`x`, `y`, `w`, `h`	rect, text, image, table	px; top-left anchor
`x1`, `y1`, `x2`, `y2`	line	px; line endpoints
`rx`	rect	px; corner radius
`strokeWidth`	rect, line	points (pt); absolute width
`font.size`	text	pt; absolute font size
`font.style`	text	categorical; bold, italic, underscore
`color fields`	text, slide, line, rect	normalized hex (#RRGGBB)
`align`	text	categorical; left/center/right/justify/distributed

Table 1: Schema of extracted ground truth fields (excerpt). See Appendix A for full details.

We normalized the coordinates to the fixed slide size 960×540px, with its origin at the top-left corner. For styling information, font sizes are reported in points, while color values are normalized into #RRGGBB format. This enables precise cross-comparison between extracted ground truth and predictions returned by vision-language models (see Sec. 3). The summary statistics of ground truth element extraction can be found in Table 2

B Predicted Extraction Prompt

```
[System Message]
Analyze the location, size, and styling information of elements in the slide.
The size of the slide is: {TARGET_W} (w) x {TARGET_H} (h) pixels. The screenshot of the slide
      was taken at DPI = 72.
Top-left of the slide is (0,0), +x rightward, +y downward.
All geometry fields are integers in pixels, unless noted otherwise.

Return a JSON object with the following top-level fields for the single slide:
{ size, background, texts:[], rects:[], lines:[], images:[], tables:[] }.
Include every required field exactly as specified.
```

{*Extraction Specification Information: Table 1 Content Here*}

```
[User Message]
{"type": "image_url", "image_url": {"url": "<base64_thumbnail>", "detail": "auto"}}
```

Figure 4: Prompt used for structured extractions from VLMs for a single slide image.

We use a single-slide prompt that (i) fixes the slide coordinate frame at 960×540 px with origin at the top-left; (ii) specifies units per field (pixels for geometry, points for fonts and strokes, hex for colors); and (iii) enumerates the required output schema (size, background, texts, rects, lines, images, tables) with field-level guidance (*e.g.,* x,y are the top-left of the element bbox; lines use x1,y1,x2,y2; rectangle corner radius is rx). The system message instructs the VLM to return a strict JSON object for the single image provided. A compact reference table in the prompt reiterates allowed values (*e.g.,* text align $\in$ left, center, right, justify, distributed) and clarifies that font and stroke widths are in points (absolute), while all positions and sizes are in pixels. The slide image is passed inline as a

	Per deck					Per slide					Total
Category	**Mean**	**SD**	**Min**	**Med**	**Max**	**Mean**	**SD**	**Min**	**Med**	**Max**	**Sum**
Num. of slides	19.48	11.54	1	18.0	46	—	—	—	—	—	1948
All elements	119.01	142.07	1	93.0	1183	6.11	9.03	0	4.0	153	11901
By type											
Text	63.40	58.40	0	49.0	314	3.25	3.34	0	3.0	69	6340
Rect	15.44	63.66	0	2.5	622	0.79	5.28	0	0.0	93	1544
Line	5.64	18.74	0	0.0	167	0.29	2.12	0	0.0	49	564
Image	33.71	33.50	0	28.0	172	1.73	2.54	0	1.0	44	3371
Table	0.82	4.09	0	0.0	40	0.04	0.35	0	0.0	11	82

Table 2: Ground-truth extraction summary across 100 decks and 1,948 slides. Per-deck statistics are computed across decks; per-slide statistics across slides.

base64 PNG. We enforce structured output via the API's JSON schema mode and validate responses with Pydantic; invalid JSON or schema mismatches are marked as parse failures.

Algorithm 1 Hungarian Matching with Blended Geometry+Content Cost and Threshold Gate

Input: $G = \{g_i\}_{i=1}^{m}$, $P = \{p_j\}_{j=1}^{n}$
Params: slide size (W, H); weights $(\alpha, \beta, \gamma, \delta)$; blended acceptance threshold $\tau \in [0, 1]$
Accessors: $\mathrm{box}(e) \to (x, y, w, h)$; $\mathrm{sim}(g, p) \in [0, 1]$ if available (else set δ=0)
Defs:
$\mathrm{IoU}(a, b) = \frac{\mathrm{area}(a \cap b)}{\mathrm{area}(a) + \mathrm{area}(b) - \mathrm{area}(a \cap b)}$
$d_{\mathrm{center}}(a, b) = \frac{\|c(a) - c(b)\|_2}{\sqrt{W^2 + H^2}}$ where $c(\cdot)$ is box center
$\mathrm{size\_rel}(a, b) = \frac{1}{2}\left(\frac{|w_a - w_b|}{\max(\varepsilon, w_a)} + \frac{|h_a - h_b|}{\max(\varepsilon, h_a)}\right)$
Construct $C \in \mathbb{R}^{m \times n}$
for $i = 1$ to m **do**
    for $j = 1$ to n **do**
        $a \leftarrow \mathrm{box}(g_i)$, $b \leftarrow \mathrm{box}(p_j)$
        $c_{\mathrm{iou}} \leftarrow 1 - \mathrm{IoU}(a, b)$; $c_{\mathrm{center}} \leftarrow d_{\mathrm{center}}(a, b)$; $c_{\mathrm{size}} \leftarrow \mathrm{size\_rel}(a, b)$
        $c_{\mathrm{cont}} \leftarrow 1 - \mathrm{sim}(g_i, p_j)$ **if** content available **else** 0
        $C_{ij} \leftarrow \alpha c_{\mathrm{iou}} + \beta c_{\mathrm{center}} + \gamma c_{\mathrm{size}} + \delta c_{\mathrm{cont}}$
    end for
end for
Compute optimal assignment $\mathcal{A} \subseteq \{1..m\} \times \{1..n\}$ by Hungarian on C
Threshold gate and bookkeeping
$\mathcal{M} \leftarrow \varnothing$; matchedG $\leftarrow \varnothing$; matchedP $\leftarrow \varnothing$
for each $(i, j) \in \mathcal{A}$ **do**
    if $C_{ij} \le \tau$ **then**
        $\mathcal{M} \leftarrow \mathcal{M} \cup \{(i, j)\}$; matchedG $\leftarrow$ matchedG $\cup \{i\}$; matchedP $\leftarrow$ matchedP $\cup \{j\}$
    end if
end for
Output: matches '$\mathcal{M}$', false positives '$P \setminus$ matchedP', false negatives '$G \setminus$ matchedG'

C Prediction-to-Ground Truth Matching Algorithm

Let $G = \{g_i\}$ denote the set of ground truth elements and $P = \{p_j\}$ the predicted elements. Each candidate match (g_i, p_j) ($c_{ij} \in C \in \mathbb{R}^{|G| \times |P|}$) we define a blended cost $c_{ij} = \alpha\big(1 - \mathrm{IoU}(g_i, p_j)\big) + \beta\, d_{\mathrm{center}}(g_i, p_j) + \gamma\, \mathrm{size\_rel}(g_i, p_j) + \delta\,\big(1 - \mathrm{sim}(g_i, p_j)\big)$, where IoU is the box overlap, d_{center} is normalized Euclidean center distance, $\mathrm{size\_rel}$ is relative size drift, and sim is a content similarity score (*e.g.,* normalized text similarity). We solve a minimum-cost bipartite matching with the

Hungarian algorithm Kuhn [1955], Carion et al. [2020] on $C = [c_{ij}]$. Finally, we apply a lightweight sanity check: a matched pair (i, j) is accepted iff its blended cost is below a threshold τ (*i.e.,* $c_{ij} \leq \tau$); otherwise it is discarded, yielding an unmatched ground-truth (FN) and prediction (FP). Pseudo code of this procedure can be found in Algorithm 1.

This formulation generalizes naturally to other modalities; only the similarity term $\mathrm{sim}(\cdot)$ is type-dependent. For example, table elements may use cell-value overlap, and images may use caption, color histogram, and object-scene similarity.

D Perturbation Operators and Hyperparameters

Notation. We perturb a slide's element list $\mathcal{E}$ with a single strength knob $s \in [0, 1]$. When $s = 0$ the transform is a no-op (we return a deep copy). All probabilities and noise scales below are monotone in s, and all randomness is seeded for reproducibility.

Geometry (layout/alignment). We act on "box-like" elements with geometry (x, y, w, h) (text, image, table, rect, chart). For each eligible element (sampled with per-element probability π_{geo}; default $= 1.0$):

- **Translation:** $(x', y') = (x + \Delta_x,\ y + \Delta_y)$ with $\Delta_x \sim \mathcal{N}(0, \sigma_x^2)$, $\Delta_y \sim \mathcal{N}(0, \sigma_y^2)$,
$$\sigma_x(s) = (0.04 + 0.16\, s) \cdot W, \quad \sigma_y(s) = (0.04 + 0.16\, s) \cdot H,$$
where (W, H) is slide size (960×540px).
- **Scaling:** $(w', h') = (w \cdot \eta_w,\ h \cdot \eta_h)$, with $\eta_{\{\cdot\}} \sim \exp(\mathcal{N}(0, \sigma_{\log}^2))$ and $\sigma_{\log}(s) = 0.12 + 0.55\, s$.
- **Extreme size (optional):** with probability $p_{\mathrm{ext}}(s) = 0.20\, s$, additionally multiply (w', h') by
$$r \sim \mathrm{Uniform}(0.15, 0.50) \ \text{ or } \ \mathrm{Uniform}(1.5, 10).$$
- **Reposition (optional):** with probability $p_{\mathrm{rep}}(s) = 0.10\, s$, sample a fresh (x', y') uniformly over valid canvas positions (respecting current size).
- **Collapse (optional):** with probability $p_{\mathrm{col}}(s) = 0.08\, s$, set one dimension to $\mathrm{Uniform}(1, 3)$ px (skinny or flat).
- **Bounds:** clamp to $[0, W - w'] \times [0, H - h']$ unless `allow_clipping`.

Text Content. We operate on text elements; non-text are passed through. For each text box (sampled with per-element probability π_{txt}; default $= 1.0$):

- **Character-level noise** with per-character rate $p_{\mathrm{char}}(s) = p_{\min} + (p_{\max} - p_{\min})\, s$, where $p_{\min} = 0.02$, $p_{\max} = 0.25$. For each affected character, apply one of {substitute, delete, insert, adjacent-swap} with weights $(0.50, 0.20, 0.15, 0.15)$. Substitutions/insertions prefer keyboard-neighbor letters; case preserved.
- **Numeric preservation (optional):** after noise, restore the original numeric runs (`\d+(\.\d+)?`) in textual order to limit semantic drift on quantities.
- **Drop boxes (optional):** with probability $p_{\mathrm{drop}}(s) = 0.18\, s$, remove the entire text box.
- **Insert boxes (optional):** with probability $p_{\mathrm{ins}}(s) = 0.35\, s$, insert $n \in \{1, \ldots, \min(\texttt{max\_inserts},\ 1 + \lfloor 3s \rfloor)\}$ irrelevant text boxes. Each insertion samples geometry fractions $w/W \sim \mathrm{U}(0.15,\ 0.35 + 0.35s)$, $h/H \sim \mathrm{U}(0.08,\ 0.22 + 0.28s)$, with uniform valid (x, y). Text is drawn from a small pool (*e.g.,* "lorem ipsum", "TODO: revise"), and default font attributes are assigned (size scales with s; emphasis toggles with small s-scaled probabilities).

Style (typography & color). We act on text elements (per-element probability π_{sty}; default $= 1.0$). Let f denote a font object with fields {`name`, `size`, `bold`, `italic`, `underline`, `color`}.

- **Family switch:** with probability $p_{\mathrm{fam}}(s) = 0.20 + 0.60\, s$, replace `name` by a random choice from a fixed pool excluding the current family.
- **Size jitter:** $\texttt{size}' = \mathrm{clip}_{[6,120]}\big(\texttt{size} \cdot \exp(\mathcal{N}(0, \sigma_{\mathrm{sz}}^2))\big)$ with $\sigma_{\mathrm{sz}}(s) = 0.45\, s$. With probability $p_{\mathrm{szext}}(s) = 0.25\, s$, additionally multiply by $\mathrm{U}(0.12, 3.8)$ to produce tiny/huge outliers.
- **Emphasis toggles:** independently flip {`bold`, `italic`, `underline`} with probability $p_{\mathrm{tog}}(s) = 0.20\, s$.
- **Color:** with probability $p_{\mathrm{inj}}(s) = 0.30\, s$, inject an incongruent palette color (e.g., #FF0000, #FFFF00, #00FFFF, ...). Otherwise jitter the current color in HLS: $\Delta h \sim \mathrm{U}(-30°, 30°)\, s$,

$\Delta\ell \sim \mathrm{U}(-0.25, 0.25)\, s$, $\Delta s \sim \mathrm{U}(-0.20, 0.20)\, s$. With probability $p_{\mathrm{lowc}}(s) = 0.25\, s$, move toward the background color by $c' = (1-\alpha)c + \alpha\, c_{\mathrm{bg}}$ with $\alpha = 0.25 + 0.65\, s$.

- **Background:** with probability $p_{\mathrm{bg}}(s) = 0.20\, s$, jitter the slide background color as above.

E Additional Details for Analysis & Measures

E.1 Slide Parseability

Definition. A slide is counted as *parsed* if the model returns a JSON object that validates against our strict schema (fields, types, units) using Pydantic. Responses that are not valid JSON or violate the schema are marked as failures. Parseability is independent of matching quality (later we report on both the end-to-end - including parse failure cases where they would count towards the denominators of the downstream performance metrics - as well as the parsed-only - excluding parse failure cases from analysis; see Fig. 2 and Fig. 7 for the relevant results).

Complexity. We use GT scene complexity c as the total number of ground truth elements on a slide (sum over text, image, table, line, rect, table).

Reliability curve by complexity. Let $\{B_k\}$ be K quantile bins of c. For each bin B_k we report

$$\widehat{\Pr}(\text{success} \mid c \in B_k) \;=\; \frac{1}{|B_k|} \sum_{i \in B_k} \mathbb{1}\{\text{parsed}_i\},$$

with a 95% bootstrap confidence interval via percentile or BCa intervals.

E.2 Metric Definitions

To investigate the VLM slide comprehension accuracy, we measure a suite of metrics encompassing a diverse set of elements for the three dimensions of quality, as detailed below.

Matching counts & PRF1. For each family and overall (micro), precision $\mathrm{P} = \frac{TP}{TP+FP}$, recall $\mathrm{R} = \frac{TP}{TP+FN}$, and $\mathrm{F1} = \frac{2\mathrm{PR}}{\mathrm{P}+\mathrm{R}}$.

Geometry terms (interpretable). For boxes we report: $1 - \mathrm{IoU}$; center distance d^{center}; relative size r^{size}; for images, aspect-ratio error r^{ar}; for rectangles, radius error r^{rx}; for lines, relative length error r^{len} and angular error r^{ang}. All terms are in $[0, 1]$ after normalization. Lower is better.

Content similarity. Text strings are normalized by lowercasing, replacing "&→and", stripping punctuation, and collapsing whitespace. We compute $s^{\mathrm{content}} = \texttt{SequenceMatcher}(\tilde{t}_{\mathrm{pred}}, \tilde{t}_{\mathrm{gt}}) \in [0, 1]$ and also report $1 - s^{\mathrm{content}}$ where an error term is desired. (Embedding-based similarity is possible but not used in our primary results.)

Style. We measure color differences using CIEDE2000 (ΔE_{00}) computed in CIE $L^*a^*b^*$ space after sRGB→Lab conversion (D65; $k_L = k_C = k_H = 1$). Lower is better. *Rule-of-thumb*: $\Delta E_{00} \lesssim 0.5$ imperceptible, $0.5-1$ barely perceptible, $1-2$ small but visible, $2-3.5$ clearly noticeable under typical viewing. We evaluate: (i) slide background vs. GT; (ii) per element type—font color (`text`), fill and stroke (`rect`), and stroke (`line`). For numeric style fields we report absolute errors in native units: font size (pt) and stroke width (pt). For booleans we report mismatch rates (0/1): bold, italic, underline (for `text`). All statistics are summarized overall and per type using means, standard deviations, and counts; micro-averaged PRF1 is computed from summed TP/FP/FN.

Aggregation. We micro-average PRF1 by summing TP/FP/FN over all slides and runs. For scalar errors we report {mean, stdev, n} over all matched pairs (overall and by type). Where noted, we compute bootstrap 95% CIs (2,000 resamples). Deck-level summaries aggregate per slide, then pool across decks (pooled mean/stdev with sample-size weights).

Units & coordinate frame. All geometry is in pixels in a fixed 960×540 slide frame; stroke width and font size are in points. The rasterization is for screenshots only and does not alter the target coordinate system.

```
[System Message]
[Role]
You score the {DIMENSION} of a PowerPoint slide.

[Scale]
Return ONE integer on the scale {SCALE_MIN}..{SCALE_MAX} (inclusive).
Anchors:
- Min ({SCALE_MIN}): "{LOW_LABEL}".
{OPTIONAL_MID}- Mid ({SCALE_MID}): "{MID_LABEL}".
- Max ({SCALE_MAX}): "{HIGH_LABEL}".

[How to judge]
Consider only:
{CRITERIA_BULLETS}
```

```
[User Message]
{"type": "image_url", "image_url": {"url": "<base64_thumbnail>", "detail": "auto"}}
```

Figure 5: Prompt template used by VLM evaluators on perturbed slides.

E.3 Evaluator Prompts

The prompts used by VLM evaluators for assessing the quality of perturbed slides along text, geometry, and style dimensions are instantiated using a common prompt template (Fig. 5) and supplying the information. For *dimension*, we use {"text quality", "layout geometry", "style"}; we provide two scale set points {(1,5), (1,100)} and corresponding *mid-point* as the mean of the end-points, and labels as {"very poor", "acceptable", "excellent"}. The *how to judge* constraints are shown in Table 3 below:

Text quality	**Layout geometry**	**Style**
• Clarity and plain language • Grammar/spelling • Bullet length (prefer one line) • Concision (avoid fluff)	• Alignment to grid/edges/baselines • Consistent spacing and margins • Balance and visual hierarchy • Element sizing matches importance	• Font family consistency and readability • Font size appropriate for viewing distance • Contrast and color harmony • Consistent emphasis (bold/italic/underline sparingly)

Table 3: *How to judge* constraints used by evaluators.

F Detailed Results

F.1 Slide Parsing Success Rate Conditioned on Scene Complexity

Parseability vs. complexity. Figure 6 visualizes these trends across complexity bins; the per-bin summaries are:

- **gpt-5-high** is essentially at ceiling across all complexity bins: five bins are at $\mathbf{100}\%$ and the remaining two are $\mathbf{99.8}\%$–$\mathbf{99.9}\%$.
- **gpt-5-minimal** is likewise near-ceiling: $\mathbf{99.7}\%$–$\mathbf{100}\%$ in all but one bin; the lowest bin is $\mathbf{99.5}\%$ (16–32).
- **o3** remains at or near ceiling throughout, with $\mathbf{99.7}\%$–$\mathbf{100}\%$ across all bins.
- **gpt-4.1** shows clear sensitivity to complexity: $\mathbf{95.5}\%$ (0–1), $\mathbf{93.7}\%$ (1–2), $\mathbf{92.8}\%$ (2–4), $\mathbf{91.6}\%$ (4–8), then drops to $\mathbf{72.1}\%$ (8–16), $\mathbf{32.8}\%$ (16–32), and $\mathbf{18.2}\%$ (32–∞).

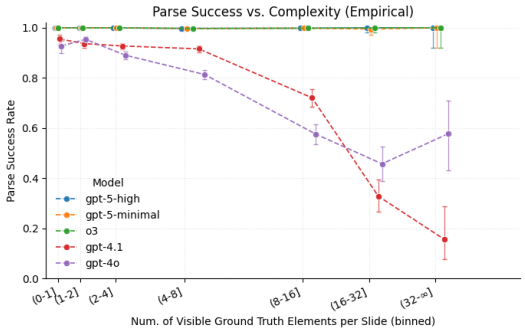


Figure 6: Parse success versus scene complexity (elements per slide) across VLMs. Complexity bins: (0–1], (1–2], (2–4], (4–8], (8–16], (16–32], (32–∞]. GPT-5 and `o3` remain near ceiling across bins, while GPT-4 series degrades with complexity. Estimates in the rightmost bin use small samples (n=66 per model).

- **gpt-4o** underperforms `gpt-4.1` in most bins as complexity grows: $\mathbf{92.7}\%$ (0–1), $\mathbf{95.4}\%$ (1–2), $\mathbf{89.1}\%$ (2–4), $\mathbf{81.4}\%$ (4–8), $\mathbf{57.6}\%$ (8–16), $\mathbf{45.8}\%$ (16–32); the uptick to $\mathbf{66.7}\%$ in 32–∞ reflects small-sample volatility (n=66).

Small sample sizes in the extreme tail (32–∞, n=66 per model) limit certainty there; the overall pattern is near-perfect parseability for the GPT-5 and `o3` models, with sharp degradation for the GPT-4 series as complexity increases.

F.2 Extraction Performance

Fig. 7 summarizes extraction accuracy and geometry error with *Parsed Only* vs. *End-to-end* bars and coverage lines; Table 4 lists per-model metrics, showing e2e (parsed-only) in each cell with best e2e bolded. Overall, `o3` and `gpt-5-{minimal,high}` lead across F1/accuracy and geometry, while `gpt-4.1`/`gpt-4o` degrade more under e2e, consistent with lower coverage.

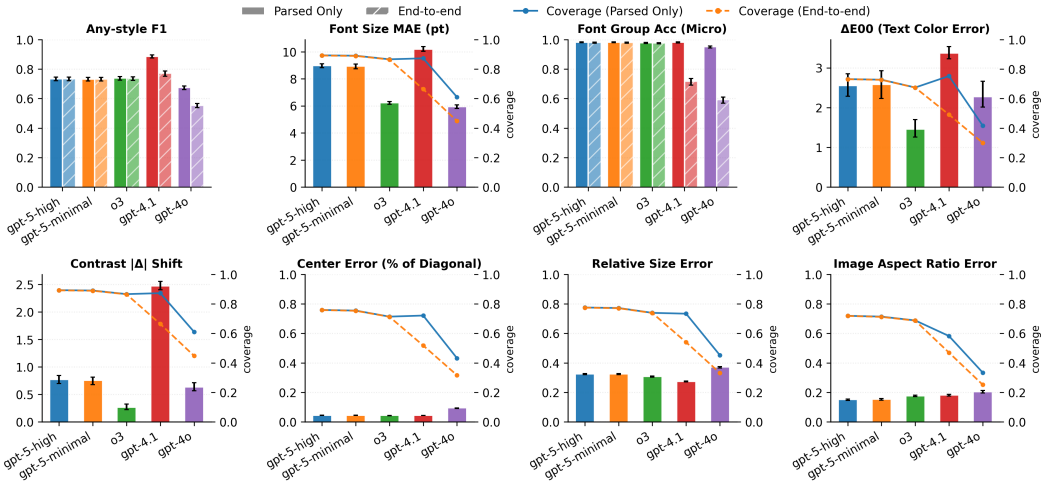


Figure 7: Bars show *Parsed Only* (solid) vs. *End-to-end* (hatched); lines (right axis) show **coverage** (fraction of ground-truth instances evaluated per metric). **Styling** (higher is better): Any-style F1 is moderate overall, with `gpt-4.1` at 0.77 (best) and `gpt-4o` at 0.55 (worst); parsed-only boosts are pronounced for the 4-series (*e.g.*, 0.89 for `gpt-4.1`, 0.67 for `gpt-4o`). **Fonts**: font *group* accuracy is near-perfect for `gpt-5-{minimal,high}` and `o3` (≥ 0.98) but lower for `gpt-4.1`/`gpt-4o` ($\approx 0.72/0.59$); font *family* accuracy is substantially lower across models (0.17–0.42). **Font size**: MAE (pt; lower is better) ranges 5.93–10.18 with `gpt-4o` best. **Color** (lower is better): text ΔE_{00} spans 1.46–3.37 (`o3` best, `gpt-4.1` worst) and contrast $|\Delta|$ shift spans 0.26–2.47 (`o3` best, `gpt-4.1` worst). **Geometry** (lower is better): $1-\text{IoU}$ is best for `o3` (0.55) and worst for `gpt-4o` (0.65); center error is 0.04–0.09, size error 0.27–0.37, and image aspect-ratio error 0.15–0.20. End-to-end coverage is substantially lower for the 4-series than for `o3`/`gpt-5`.

Metric	gpt-4o	gpt-4.1	o3	gpt-5-minimal	gpt-5-high
Element Matching F1	0.44 (0.54)	0.59 (0.71)	**0.72** (0.72)	0.71 (0.71)	0.72 (0.72)
Geometry (micro; lower is better)					
$1 - \text{IoU}$	0.65	0.57	**0.55**	0.56	0.56
Center error (% diag)	0.09	0.04	**0.04**	0.04	0.04
Size error (relative)	0.37	**0.27**	0.31	0.32	0.32
Image AR error	0.20	0.18	0.18	0.15	**0.15**
Content (micro; higher is better)					
Text Content F1	0.63 (0.69)	0.69 (0.73)	**0.78** (0.78)	0.76 (0.76)	0.76 (0.76)
Style (micro; higher is better for style F1 and font accuracies; lower is better for color shifts)					
Any-style F1	0.55 (0.67)	**0.77** (0.89)	0.74 (0.74)	0.73 (0.73)	0.73 (0.73)
Font Family Acc (micro)	0.17 (0.27)	0.33 (0.45)	0.32 (0.32)	0.41 (0.41)	**0.42** (0.42)
Font Group Acc (micro)	0.59 (0.95)	0.72 (0.98)	0.98 (0.98)	0.98 (0.98)	**0.98** (0.98)
Font size MAE (pt)	**5.93**	10.18	6.22	8.92	8.97
Text color ΔE_{00}	2.27	3.37	**1.46**	2.57	2.55
Contrast $\lvert\Delta\rvert$ shift	0.63	2.47	**0.26**	0.75	0.77

Table 4: Extraction accuracy and geometry quality by model. Each cell shows *end-to-end* and (parsed-only) values, when applicable. Higher is better for F1/accuracy; lower is better for error metrics. Best model metric is boldfaced.

F.3 Slide Deck Narrative Order Performance

To assess narrative comprehension, we examine how effectively the VLM reconstructs the original sequence of slides from a randomly shuffled deck (Figure 8). Each deck is segmented into individual slide representations, which are then randomly reordered and input into the model along with a prompt instructing it to restore the correct order. The model's predicted sequence is evaluated against the ground truth using Kendall's τ, Spearman's ρ, and normalized exact match metrics. We report the mean and standard deviation across all decks.

As a preliminary step, we verify whether the models can generate output sequences that match the full length of the original presentations. For instance, if a presentation contains 23 slides, the model should produce an ordered list of 23 elements. According to Figure 6 (left), `GPT-5 high` and `o3` successfully generate nearly complete sequences, whereas other models struggle to even identify the correct number of slides present in the input.

Focusing on presentations with correctly predicted lengths, `GPT-5 minimal` and `GPT-4.1` demonstrate relatively strong performance in ordering accuracy, as measured by Kendall's τ and Spearman's ρ, particularly outperforming `o3`. However, across the board, all models exhibit limited capability in narrative ordering, with scores below 0.15. This indicates substantial room for improvement before approaching the theoretical upper bound of 1.0 across all metrics. While the models appear capable of interpreting slide content and multimodal layout, they still face significant challenges in reasoning through the narrative structure.

G Fonts and Font Groups Used in the Analysis

G.1 Canonicalized Font Names and Counts in the Dataset

Table 5 shows the count statistics of different fonts in text elements present in the ground truth slides.

G.2 Font → Font Group Mapping

```
# Sans
"arial":"sans","calibri":"sans","helvetica":"sans","helvetica neue":"sans","segoe ui":"sans","verdana":"sans",
"tahoma":"sans","gill sans":"sans","inter":"sans","roboto":"sans","open sans":"sans","lato":"sans",
```

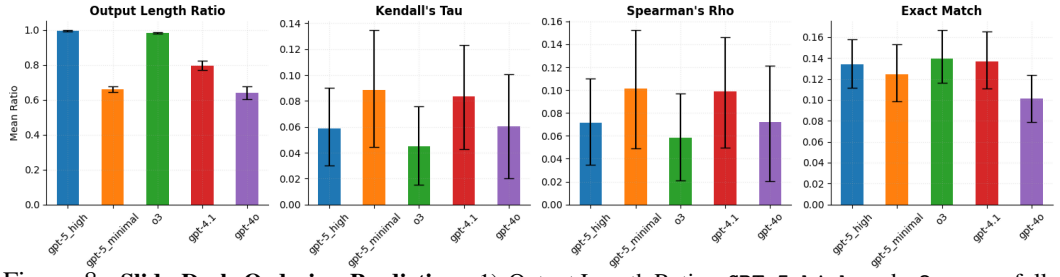


Figure 8: **Slide Deck Ordering Prediction**: 1) Output Length Ratio: `GPT-5 high` and `o3` successfully generate nearly complete sequences 2) Kendall's τ and 3) Spearman's ρ: despite overlapping confidence integrals, `GPT-5 minimal` and `GPT-4.1` show a consistent upward trend among these two measure, indicating potential robustness that warrants further investigation 4) Exact Match: models exhibit similar performance around 0.14 with `GPT-4o` being the lowest.

Font	Count	Font	Count	Font	Count
calibri	2183	arial	1692	unknown	460
lato	260	montserrat	203	roboto	159
open sans	132	century gothic	105	oswald	105
helvetica neue	98	avenir next	97	garamond	70
verdana	66	ibm plex sans	65	corbel	64
georgia	61	source sans pro	53	libre franklin	43
tahoma	41	patrick hand	33	raleway	32
soehne	31	dosis	30	inter	22
times new roman	22	quattrocento sans	20	titillium web	20
bahnschrift	16	barlow	16	cambria	16
elephant	15	franklin gothic	14	nunito	14
gill sans	12	amatic sc	10	american typewriter	10
source code pro	10	ubuntu	9	ibm plex mono	5
palatino linotype	4	aptos	3	handwriting	3
segoe script	3	bookman old style	2	menlo	2
playfair display	2	tenorite	2	bodoni	1
inconsolata	1	pacifico	1	proxima nova	1
segoe ui	1	Total	6340		

Table 5: Frequency of different font families in the ground truth data (sorted descending, row-major)

```
"montserrat":"sans","source sans pro":"sans","libre franklin":"sans","quattrocento sans":"sans",
"ubuntu":"sans","barlow":"sans","bahnschrift":"sans","ibm plex sans":"sans","soehne":"sans","dosis":"sans",
"poppins":"sans","raleway":"sans","titillium web":"sans","nunito":"sans","corbel":"sans","candara":"sans",
"century gothic":"sans","avenir":"sans","avenir next":"sans","franklin gothic":"sans","arial rounded mt":"sans",
# Serif
"times new roman":"serif","georgia":"serif","garamond":"serif","cambria":"serif","palatino linotype":"serif",
"bookman old style":"serif","elephant":"serif","merriweather":"serif","playfair display":"serif",
"bodoni":"serif","bodoni mt":"serif","didot":"serif","tinos":"serif","cmr10":"serif","american typewriter":"serif",
# Mono
"courier new":"mono","courier":"mono","consolas":"mono","menlo":"mono","monaco":"mono","inconsolata":"mono",
"fira mono":"mono","source code pro":"mono","roboto mono":"mono","ibm plex mono":"mono",
# Script / Hand / Display
"comic sans ms":"script","brush script mt":"script","brush script":"script","amatic sc":"script",
"patrick hand":"script","architects daughter":"script","caveat":"script","pacifico":"script","lobster":"script",
"impact":"display","bebas":"display",
# Others
"roboto slab":"serif","carlito":"sans","asana":"serif","tenorite":"sans","aptos":"sans",
"segoe ui emoji":"sans","segoe ui symbol":"sans",
```

G.3 Font Group Frequencies

Table 6 shows the count statistics of different fonts in text elements present in the ground truth slides.

Font	Count	Font	Count	Font	Count
sans	5503	other	569	serif	203
script	47	mono	18	Total	6340

Table 6: Frequency of different font groups in the ground truth data (sorted descending, row-major)

H Reproducibility and Safety Checks for Slide Perturbation

- **Seeding:** All RNG draws use a fixed base seed; per-slide streams can be derived via a deterministic hash of the slide ID.
- **Validity:** Geometry is clamped to the canvas (unless explicitly allowed); sizes are lower-bounded by 1 px. Colors are validated to normalized hex (#RRGGBB) before export.
- **No-op at $s = 0$:** We return an unchanged copy when $s \leq 10^{-12}$.
- **On Monotonicity:** Because operations are stochastic, a single draw at $s{=}1.0$ need not strictly dominate a draw at $s{<}1$, but it does so at expectation (all scales/probabilities are monotone in s).

I Declaration of LLM Usage

We used large language model (LLM) assistants solely for *writing and tooling support*, including (i) manuscript/LaTeX editing, phrasing, and formatting, and (ii) non-substantive code assistance in VS Code (*e.g.,* refactoring, bug fixing, style cleanups, and commenting). All algorithms, evaluation designs, datasets, metrics, and reported results were specified by the authors; LLM-suggested text/code was reviewed, verified, and tested by the authors before inclusion. This usage does not impact the core methodology or conclusions.

References

Vidhisha Balachandran, Jingya Chen, Neel Joshi, Besmira Nushi, Hamid Palangi, Eduardo Salinas, Vibhav Vineet, James Woffinden-Luey, and Safoora Yousefi. Eureka: Evaluating and understanding large foundation models. *arXiv preprint arXiv:2409.10566*, 2024.

Steve Canny. python-pptx: Create open xml powerpoint documents in python. `https://github.com/scanny/python-pptx`, 2025. Accessed: 2025-08-17.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

Souradip Chakraborty, Mohammadreza Pourreza, Ruoxi Sun, Yiwen Song, Nino Scherrer, Furong Huang, Amrit Singh Bedi, Ahmad Beirami, Jindong Gu, Hamid Palangi, et al. Review, refine, repeat: Understanding iterative decoding of ai agents with dynamic evaluation and selection. *arXiv preprint arXiv:2504.01931*, 2025.

Shuo Chen, Jindong Gu, Zhen Han, Yunpu Ma, Philip Torr, and Volker Tresp. Benchmarking robustness of adaptation methods on pre-trained vision-language models. *Advances in Neural Information Processing Systems*, 36:51758–51777, 2023.

Yuyang Dong, Nobuhiro Ueda, KrisztiĂĄn Boros, Daiki Ito, Takuya Sera, and Masafumi Oyamada. Scan: Semantic document layout analysis for textual and visual retrieval-augmented generation. *arXiv preprint arXiv:2505.14381*, 2025.

Tsu-Jui Fu, William Yang Wang, Daniel McDuff, and Yale Song. Doc2ppt: Automatic presentation slides generation from scientific documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 634–642, 2022.

Jiaxin Ge, Zora Zhiruo Wang, Xuhui Zhou, Yi-Hao Peng, Sanjay Subramanian, Qinyue Tan, Maarten Sap, Alane Suhr, Daniel Fried, Graham Neubig, and Trevor Darrell. Autopresent: Designing structured visuals from scratch. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2902–2911, June 2025a.

Jiaxin Ge, Zora Zhiruo Wang, Xuhui Zhou, Yi-Hao Peng, Sanjay Subramanian, Qinyue Tan, Maarten Sap, Alane Suhr, Daniel Fried, Graham Neubig, et al. Autopresent: Designing structured visuals from scratch. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2902–2911, 2025b.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14375–14385, 2024.

Kyudan Jung, Hojun Cho, Jooyeol Yun, Soyoung Yang, Jaehyeok Jang, and Jaegul Choo. Talk to your slides: Language-driven agents for efficient slide editing. *arXiv preprint arXiv:2505.11604*, 2025.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*, pages 18893–18912. PMLR, 2023.

Tony Lee, Haoqin Tu, Chi H Wong, Wenhao Zheng, Yiyang Zhou, Yifan Mai, Josselin S Roberts, Michihiro Yasunaga, Huaxiu Yao, Cihang Xie, et al. Vhelm: A holistic evaluation of vision language models. *Advances in Neural Information Processing Systems*, 37:140632–140666, 2024.

Shuhang Liu, Zhenrong Zhang, Pengfei Hu, Jiefeng Ma, Jun Du, Qing Wang, Jianshu Zhang, Quan Liu, Jianqing Gao, and Feng Ma. Mmc: Iterative refinement of vlm reasoning via mcts-based multimodal critique. *arXiv preprint arXiv:2504.11009*, 2025.

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, et al. Mmlongbench-doc: Benchmarking long-context document understanding with visualizations. *Advances in Neural Information Processing Systems*, 37:95963–96010, 2024.

Gio Paik, Geewook Kim, and Jinbae Im. Mmrefine: Unveiling the obstacles to robust refinement in multimodal large language models. *arXiv preprint arXiv:2506.04688*, 2025.

Yi-Hao Peng, Peggy Chi, Anjuli Kannan, Meredith Ringel Morris, and Irfan Essa. Slide gestalt: Automatic structure extraction in slide decks for non-visual access. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2023.

Viraj Prabhu, Senthil Purushwalkam, An Yan, Caiming Xiong, and Ran Xu. Trust but verify: Programmatic vlm evaluation in the wild. *arXiv preprint arXiv:2410.13121*, 2024.

Ray Smith. An overview of the tesseract ocr engine. In *ICDAR '07: Proceedings of the Ninth International Conference on Document Analysis and Recognition*, pages 629–633, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-2822-8. URL `https://storage.googleapis.com/pub-tools-public-publication-data/pdf/33418.pdf`.

Russell Stewart, Mykhaylo Andriluka, and Andrew Y. Ng. End-to-end people detection in crowded scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

An Vo, Khai-Nguyen Nguyen, Mohammad Reza Taesiri, Vy Tuong Dang, Anh Totti Nguyen, and Daeyoung Kim. Vision language models are biased, 2025. URL `https://arxiv.org/abs/2505.23941`.

Baode Wang, Biao Wu, Weizhen Li, Meng Fang, Yanjie Liang, Zuming Huang, Haozhe Wang, Jun Huang, Ling Chen, Wei Chu, et al. Infinity parser: Layout aware reinforcement learning for scanned document parsing. *arXiv preprint arXiv:2506.03197*, 2025.

Dongsheng Wang, Natraj Raman, Mathieu Sibue, Zhiqiang Ma, Petr Babkin, Simerjot Kaur, Yulong Pei, Armineh Nourbakhsh, and Xiaomo Liu. Docllm: A layout-aware generative language model for multimodal document understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024. URL https://aclanthology.org/2024.acl-long.463/.

Tianyi Xiong, Xiyao Wang, Dong Guo, Qinghao Ye, Haoqi Fan, Quanquan Gu, Heng Huang, and Chunyuan Li. Llava-critic: Learning to evaluate multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13618–13628, 2025.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florêncio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021. URL https://arxiv.org/abs/2012.14740.

J Technical Appendices and Supplementary Material

Technical appendices with additional results, figures, graphs and proofs may be submitted with the paper submission before the full submission deadline (see above), or as a separate PDF in the ZIP file below before the supplementary material deadline. There is no page limit for the technical appendices.

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 clearly state the three evaluation axes, the standardized dataset/ground-truth pipeline, and the principal empirical findings, and they explicitly scope claims to the slide domain (public PPTX corpus, seeded perturbations, defined metrics) with limitations noted; these statements match the methods and results without overgeneralizing.

   Guidelines:

   - The answer NA 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 NA 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: The conclusion explicitly states scope limits (public PPTX corpus, seeded perturbations, rendering/metrics pipeline) and constrains claims to those settings, while summarizing empirical strengths and weaknesses without overgeneralization.

   Guidelines:

   - The answer NA 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: [NA]

Justification: [NA]

Guidelines:

- The answer NA 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: The draft specifies the evaluation pipeline at a reproducible level of detail: fixed slide coordinate frame and units, strict JSON schema validation and Hungarian GT-matching with a threshold gate, fully enumerated perturbation operators with seeded randomness plus explicit reproducibility checks, and metric definitions with bootstrap CIs. Together we believe the provided information is sufficient to replicate the main results within the stated scope.

Guidelines:

- The answer NA 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: [No]

   Justification: Due to ongoing employer data-governance and IP review, we cannot release the curated dataset (or code) at submission time. Nevertheless, the experiments rely on publicly available PPTX sources and accessible model APIs, and we specify preprocessing, seeded perturbations, and metric definitions in sufficient detail to enable faithful reimplementation; pending approval, we plan to release scripts post-review.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the dataset and preprocessing (100 Zenodo decks, 1,948 slides; fixed 960×540 frame; JSON-schema validation; $N=3$ low-temperature runs), the matching/evaluation procedure (Hungarian with a blended cost and acceptance gate), seeded perturbation operators with a severity schedule, and metric definitions with bootstrap CIs and the exact extraction prompt, etc., together sufficient to understand the results.

   Guidelines:

   - The answer NA 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: Key results include bootstrap 95% confidence intervals, reported in the core text and appendix (*e.g.,* parseability by complexity with percentile/BCa intervals), and we state what variability they capture (resampling over slides/bins/severities).

Guidelines:

- The answer NA 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: All inference is via Azure-hosted VLM APIs (no local training/GPUs), and we detail the CPU-only preprocessing/evaluation pipeline—rasterization to 960×540, seeded perturbations, $N{=}3$ runs per slide, and Hungarian matching. The required compute is dominated by API latency and token usage and is reproducible from the described setup.

   Guidelines:

   - The answer NA 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 study uses publicly available PPTX material, collects no personal/sensitive data, involves no human subjects, and preserves anonymity; models are accessed via Azure under organizational policies. We disclose scope and limitations, aligning with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA 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: [Yes]

Justification: Positive impacts include safer deployment of VLM-based slide evaluation and improved accessibility checks (*e.g.,* color contrast overall visibility of elements in a slide). Potential negatives include overreliance on automated judgments and misuse for rigid template policing; we bound claims to evaluation-only.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We release neither models nor datasets; experiments use publicly available PPTX sources with CC-By 4.0 license and hosted APIs, and the curated corpus/scripts remain internal pending employer review, so no high-risk assets are being released. If a future release is approved, it will follow organizational governance with licensing checks, de-identification, and access controls.

Guidelines:

- The answer NA 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: All third-party assets are PPTX decks sourced from Zenodo10K under **CC-BY 4.0**; we cite the collection and include the URL while not redistributing the files. Hosted VLMs are accessed via Azure under provider ToS; no third-party datasets or code are re-packaged or released.

Guidelines:

- The answer NA 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: [NA]

Justification: We introduce an evaluation framework but do not release new assets (datasets/code/models) at submission time; materials remain internal pending employer review. The paper documents schemas, perturbations, and metrics to enable reimplementation, but asset-side documentation is not applicable without a release.

Guidelines:

- The answer NA 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: [NA]

Justification: [NA]

Guidelines:

- The answer NA 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: [NA]

Justification: [NA]

Guidelines:

- The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: LLMs were used only for manuscript/LaTeX editing and non-substantive code assistance in VS Code (refactoring/bug fixes), with all methods, datasets, metrics, and results authored and verified by the authors; see Appendix, *Declaration of LLM Usage*, for details.

Guidelines:

- The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.